LEARNING METRICS, GRAPHS, AND RANKINGS: NEW THEORY AND APPLICATIONS

by

Blake Joseph Mason

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

(Electrical and Computer Engineering)

at the

UNIVERSITY OF WISCONSIN–MADISON

2020

Date of final oral examination: December 4, 2020

The dissertation is approved by the following members of the Final Oral Committee:
Robert Nowak, Professor, Electrical and Computer Engineering
Dimitris Papailiopoulos, Assistant Professor, Electrical and Computer Engineering
Timothy T. Rogers, Professor, Psychology
Stephen Wright, Professor, Computer Science
Lalit Jain, Assistant Professor, Business, University of Washington

ACKNOWLEDGMENTS

I was told in February of my first year, in the Corral Room over negronis, that one of the hard things about graduate school is not only doing the work but maintaining the environment to be productive in. As much as this document bears my name, it is the product of the myriad conversations, friendships, and pieces of advice that have seen me to its completion, and I am hopelessly indebted to far more people than I can name here. Insufficient as it may be for their many kindnesses, I would like to thank a few people without whom I should never have made it to this place or anywhere near it.

I am grateful for my advisor, Rob Nowak. His steady hand at the helm of this grand expedition has seen it through all manner of wind and weather and to a destination I could not have envisioned at the outset. I've appreciated his insight for problems and proofs and the simultaneous latitude he's given me to chase new ideas and directions all the while knowing he had my back. I remember in my second year, he conjectured almost exactly what would go on to be a core result in my first machine learning paper. That it took us five more months to prove the result serves to underscore the vast wealth of knowledge and experience that I have had the privilege to tap. I took me too long to see beyond his professional accomplishment to realize just how genuinely good of a guy he is, and I am all the more thankful for that.

Lalit Jain has been a friend, a mentor, a competitor in ECE 830, and all manners of kind to me over these past five years. I am especially in his debt for every time he has let me stay with him through our various conferences and trips. I remember one particular hike we took in the Cascades. As we were leaving the summit in terrible weather, the clouds briefly parted and we could see the expanse of the valley and river beneath us. In our numerous research conversations over the years, I have at times experienced the same clarity and infinite horizon. I want to thank Lalit for all that he has taught me and for pushing me to be a better researcher.

During my degree, I was part of the LUCID program at Wisconsin. I owe a thank you to Tim Rogers and Caitlin Iverson for running such an incredible

interdisciplinary experience. I am especially thankful to Martina Rau for our collaborations through LUCID. Rebecca Willet was a source of insight and guidance. Thank you for the advice and every party you threw. Laurent Lessard provided many wonderful conversations and always went out of his way to be helpful. I'd also like to thank Ardhendu Tripathy, Ayon Sen, Jack Wolf, and Jenna Nobles who have all been great collaborators. Thank you Kwang-Sung Jun for first introducing me to bandits. Thank you Marshall Müller and David Nieman for making me a better mathematician. During my degree, I have appreciated the many insights of Dimitris Papailiopoulos and Steve Wright and want to thank them for serving on my committee. Finally, I owe Justin Haldar at USC a thank you for pushing me to apply to the University of Wisconsin and writing a letter for me.

I moved over 2000 miles to Madison and knew no one when I arrived. I will always appreciate and remember fondly those lab mates I had in my first few years, the meals we shared, the perennial games of Hanabi, and just how welcome they made me feel. Thank you Aniruddha Bhargava, Eric Hall, Lalit Jain, Daniel Pimentel-Alarcón, Urvashi Oswal, Xin Jiang, Ian Kinsella, Ari Biswas, and Sumeet Katariya. Scott Sievert gave me mittens and taught me, who had never seen snowfall, how to survive a winter. For that and for all he has taught me about coding, I am thankful. Cindy Harrison was the angel of the third floor of the WID, and every one of us that worked late into night there is in her debt. I miss the conversations we had and the vibrance, humor, and energy she brought to life. Thank you to Jackson Burgess for every revelrous weekend we shared in Iowa and for the ceaseless empathy and perspective. Chris Magnano, Ross Kleiman, Andrew Ontano, Silvia Di Gregorio, Jay Grenda, AJ Gazin, and Eileen Jennings have all been kind friends. Mike O'Neill has been a great friend and has taught me all manners of things about optimization. Davis Gilton's amiability, humor, and holistic approach to research and life enriched my time here. Thank you especially for letting me stay with you in Chicago and for showing me Sedona. Thank you Katie Hunt for the cocktails, meat and cheese platters, and every rough time we've endured together. Nick Moran has been a faithful friend for many years. I am especially thankful that he came to visit me my first summer here. I am thankful for Em Huang's steady friendship,

welcome ear, and thoughtful advice. Thank you Karen Mason for the homemade cookies and every wonderful conversation we had. Michael McDonald has been a presence and friend throughout my life, and I appreciate the visits he made to Madison during my degree. Lastly, I want to thank Lauren Liedel for her hilarity, generosity, and constant friendship.

I want to thank my parents, Mary Ann and Greg, without whose wisdom, love, and support I would be adrift. My dad has been a sounding board, a mentor, and a source of perspective. I am thankful to have in my mom a confidant and kindred spirit. I owe a special thank you to my sister Dana who has always rooted for me and who has been my council and my friend. Nick Karp has been the brother I've needed throughout my life. He has seen me through both the tempestuous and the placid, and I appreciate his sagacious approach to life on which I may stay my mind. Finally, I need to thank Laura Palarz for her unwavering support, humor, and belief in me.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

The past decade has seen an explosion in the use of machine learning to model, to understand, and to make recommendations to people. Machine learning systems recommend videos and music, control advertising, and even generate content for webpages and educational resources. This has ignited a resurgence of classical questions in psychology, such as embedding from preference judgments, attacked and analyzed with modern machine learning tools. The goal is to use machine learning to improve inference, provide new techniques, and better model people. Learning directly from human generated data presents several challenges that researchers must face. First, as human judgments are frequently noisy and subjective, researchers are often limited in the form of queries that they can reliably ask people such as simple binary 'yes' or 'no' comparisons. Second, collecting crowdsourced data can be time-intensive, and expensive if expertise is required; hence, it is important to develop methods to actively choose which data to query and to collect only the most informative samples when learning from the crowd. Third, people are heterogeneous in their skills and abilities. In settings such as crowdsourced labelling, researchers must identify a large pool of capable workers to answer queries, but it can be difficult to estimate individuals' ability to perform a potentially abstract task when choosing amongst a diverse group.

In this thesis, we approach all three of these challenges. First, motivated by questions in educational psychology, we develop method for low-dimensional metric learning from triplet comparisons of the form "item i and item j are closer to each other than item i and item k." We characterize how difficult metric learning is through geometric and statistical arguments and provide a simple and efficient convex optimization to learn metrics optimally. Informed by these theoretical results for metric learning, we develop a new way of studying people's perception of images and also provide an empirical comparison of different active sampling algorithms for metric learning which attempt to reduce the sample complexity.

We pay special attention to reduce the sample complexity of learning from people as data collection is a bottleneck in many practical problems. This thesis

studies active sampling algorithms to learn nearest neighbor graphs and nearest neighbor data structures. Nearest neighbors is an efficient and flexible way to model people's preferences. Algorithms for both methods rely on multi-armed bandits, a class of active sampling algorithms, to reduce the sample complexity of learning. Finally, we propose and provide an algorithm for a new multi-armed bandit objective tailored to settings where one wishes to find many high-performing alternatives: such as many highly accurate crowd-workers for a crowdsourcing task.

1 INTRODUCTION

This thesis focuses on core challenges of learning from human generated data. The goal throughout is to develop algorithms and theory that helps researchers understand and model people and to develop methods that more efficiently gather data from people for machine learning tasks.

Learning directly from human generated data presents several challenges that researchers must face. First, as human judgments are frequently noisy and subjective, researchers are often limited in the form of queries that they can reliably ask people such as simple binary 'yes' or 'no' comparisons. Second, collecting crowdsourced data can be time-intensive, and expensive if expertise is required; hence, it is important to develop methods to actively choose which data to query and to collect only the most informative samples when learning from the crowd. Third, people are heterogeneous in their skills and abilities. In settings such as crowdsourced labelling, researchers must identify a large pool of capable workers to answer queries, but it can be difficult to estimate individuals' ability to perform a potentially abstract task when choosing amongst a diverse group. In this thesis, we have approached each of these challenges focusing on the following areas:

- **Low-dimensional metric learning** We show that distance metrics can be efficiently learned from noisy, binary comparison data and give some of the first learning theory guarantees for metric learning.

- **Actively learning nearest neighbor graphs** We develop an adaptive sampling method to learn nearest neighbor graphs which can be used to represent peoples' beliefs of similarity. The algorithm is the first to achieve the optimal complexity in the presence of noise.

- **Actively finding every near-optimal alternative** We develop a multi-armed bandit approach to actively detect every alternative that achieves near-optimal performance, such as every worker that performs highly on a given task. This

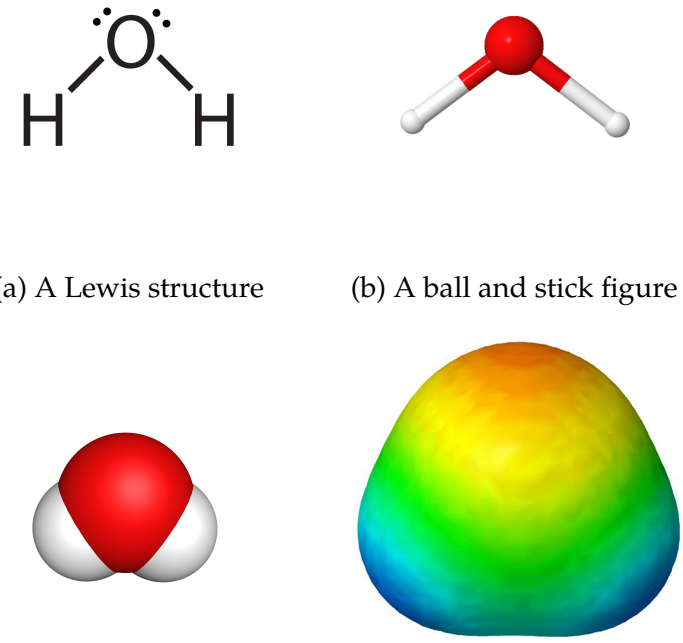
filled a gap in the literature and provided new analytical tools for studying bandits.

1.1 Organization

In Chapter 2, we introduce a problem arising in educational psychology proposed by collaborators wishing to develop personalized, intelligent tutoring systems. This project serves as the overarching motivation for much of my work in machine learning. Specifically, researchers wish to determine which features students pay attention to when looking at images of molecules found in textbooks in settings where methods such as eye-tracking are untenable. We propose a technique based in similarity learning, a relative of metric learning, for this task. Next, in Chapter 3, we develop new theory for triplet metric learning, establishing learning rates and recovery guarantees. Then, in Chapter 4, we use the techniques and guarantees developed in Chapter 3 to develop a novel method for cognitive task analysis and apply it to the question proposed in Chapter 2. This closes some questions about metric learning, but raises others about data efficiency. To address the data efficiency question, in Chapter 5, we develop an active technique to efficiently learn nearest neighbor graphs from noisy samples and apply it to human preference data, showing that it is possible to apply active methods to reduce the sample complexity of learning from people. In Chapter 6, motivated by the task of selecting crowd-workers to label datasets, we propose and analyze a method to actively find every near-optimal alternative. Finally, in Chapter 7 we apply the algorithm from Chapter 6 to develop a method to learn data structures to answer nearest-neighbor queries from noisy data, generalizing the results of Chapter 5 while achieving the same optimal rate.

1.2 Similarity Learning Methods for Intelligent Tutoring Systems

To succeed in Science, Technology, Engineering, and Mathematics (STEM), students need to learn to use visual representations that depict important domain material, such as graphs, models, and figures. As an example, in Figure 1.1, we show four different visual representations of the water molecule. Each highlights different properties of the molecule. Most prior research has focused on conceptual knowledge about visual representations (ie, what concepts are depicted in the visual) that is acquired via verbally mediated forms of learning. However, students also need perceptual fluency: the ability to rapidly and effortlessly translate among representations. Perceptual fluency is acquired via non-verbal, implicit learning processes. Non-verbal mediation is common to perception tasks, but presents a challenge for learning. For skills that are not verbally mediated, people are unable to accurately say *how* they form the judgments they do or what it is they are focusing on when they make their judgments. A classic example from the psychology literature comes from asking people to guess gender based on images of faces. Humans are extremely accurate at this task. However, when asked what *features* of the image led them to their conclusion, their answers are often at odds with the judgments they provide. A challenge for instructional interventions in classrooms that focus on implicit learning is to model students' knowledge acquisition. Because implicit learning is non-verbal, we cannot rely on traditional methods, such as expert interviews or student think-alouds. In this chapter, we propose a similarity learning (a relative of metric learning) technique to assess how people perceive similarity between visual representations. We used this approach to model how undergraduate students perceive similarity between visual representations of chemical molecules. The approach achieved good accuracy in predicting students' similarity judgments and expands expert predictions of how students might perceive visual representations of molecules. This chapter is adapted from (Rau et al., 2016), and the work was completed in collaboration with the authors therein.



(a) A Lewis structure

(b) A ball and stick figure

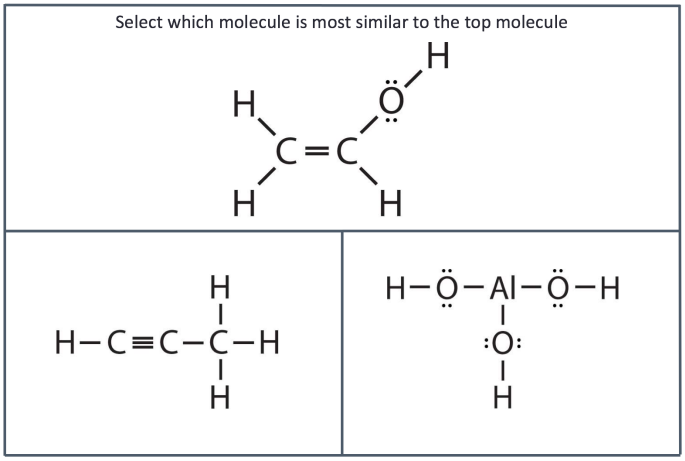
(c) A space-filling model

(d) An electrostatic potential map

Figure 1.1: Different visual representations of the water molecule

1.3 Learning Low-Dimensional Metrics from Triplets

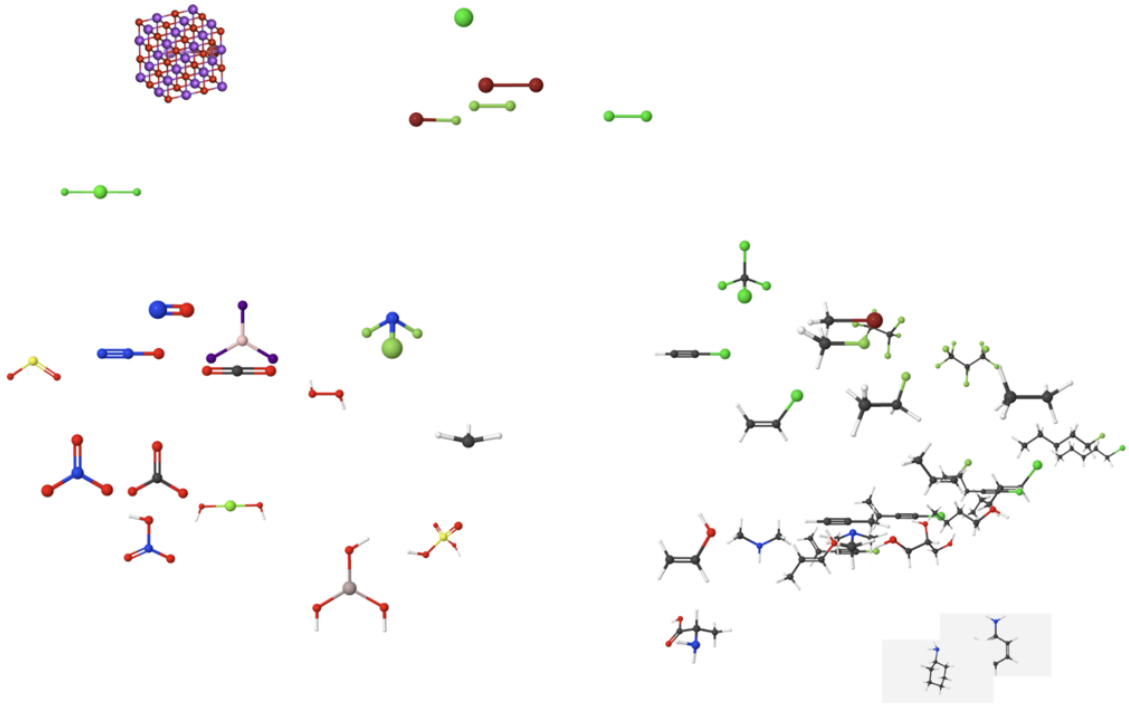
Metric learning is a promising tool for learning from human judgments. In this problem, one is given n items with feature vectors $\mathbf{x}_1, \cdots, \mathbf{x}_n \in \mathbb{R}^d$ and collects data in the form of triplets "item $\mathbf{x}_i$ is closer to item $\mathbf{x}_j$ than it is to item $\mathbf{x}_k$." One seeks to learn semidefinite matrix $\mathbf{K}$ such that the distance metric $\|\mathbf{x}_i - \mathbf{x}_j\|_{\mathbf{K}}^2 := (\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{K} (\mathbf{x}_i - \mathbf{x}_j)$ agrees with the triplet constraints as well as possible. Triplets are commonly collected in psychological experiments and used to model test subjects' beliefs Cox and Cox (2008). A fundamental question is *why and how* people form their beliefs. Using metric learning, we developed a method to identify *what features* best explain peoples' judgments. This research is motivated by collaborations



(a) A triplet query between three Lewis structures

with educational psychologists described in Chapter 2. We gathered images from chemistry textbooks and collected triplets as shown in Figure 1.2a, asking students which of the bottom two molecules is most similar to the top. By leveraging metric, we were able to compute the importance of any molecular feature, such as the presence of a bond or atom, to students' similarity judgments and generated the visualization shown in Figure 1.3a accordingly. The method relies on learning a metric given by a matrix $\mathbf{K}$ that predicts students' judgments and then computing which features explain these judgements as a function of $\mathbf{K}$. Despite the promise of metric learning for psychological experiments and its application to modern tasks such as facial recognition Schroff et al. (2015), there was little theoretical analysis in this area prior to this thesis.

This thesis provides some of the first learning theory results for metric learning and the first recovery result. All results hold for noisy data as might be collected from people. We focus on *low-dimensional* metric learning where $\mathbf{K}$ has low-rank corresponding to the common assumption in psychology that comparisons are well modeled coming as from a latent low-dimensional subspace Cox and Cox (2008). We additionally analyze the case of $\mathbf{K}$ being sparse and low-rank, corresponding to the assumption that only k of the d features correlate to people's judgements. In this chapter, we provide a simple convex optimization to learn a k-dimensional metric



(a) Triplet embedding

on points in $\mathbb{R}^d$ that predicts unseen triplets in $O(kd\log(n))$ samples, and we show that this rate is optimal. Surprisingly, we also show that in the more restricted case that one wishes to learn a sparse and low-rank $\mathbf{K}$, the same number of samples is needed, and the additional structure is not helpful. Finally, we provide the first recovery result for metric learning. If there exists a true $\mathbf{K}^*$ which generates the data with noise, we give the first result that shows it can be estimated from data. This chapter is adapted from (Mason et al., 2017) and is collaborative with its authors.

1.4 Application of Metric Learning to Cognitive Science and Intelligent Tutoring Systems

Using the new theoretical results about metric learning developed in Chapter 3, we develop a new technique for cognitive task analysis and apply it to the problem first

introduced in Chapter 2. To benefit from visuals in STEM instruction, students need representational competencies that enable them to see meaningful information in the visuals. Most research has focused on explicit conceptual representational competencies. In addition, implicit perceptual competencies allow students to efficiently see meaningful information in visuals. Most common methods to assess students' representational competencies rely on verbal explanations or assume that students explicitly attend to the visuals. However, because perceptual competencies are implicit and not necessarily verbally accessible, these methods are ill-equipped to assess perceptual competencies. We address these shortcomings with a method that draws on similarity learning methods, a type of machine learning method that learns visual features that account for participants' responses to triplet comparisons of visuals. In Experiment 1, 614 chemistry students judged the similarity of Lewis structure representations of chemical molecules. In Experiment 2, 489 chemistry students judged the similarity of ball-and-stick models. Our results showed that our method can detect visual features that drive students' perception of visual representations of chemical molecules. Our inspection of the features suggests that students' conceptual knowledge about molecules informed perceptual competencies through top-down processes. Further, Experiment 2 tested whether we can improve the efficiency of the method with active sampling. Results showed that regular random sampling methods yield higher accuracy than active sampling for small sample sizes. Together, the experiments provide the first method to assess students' perceptual competencies implicitly, without asking them to verbalize their knowledge or assuming explicit visual attention. These findings have implications for the design of instructional interventions that help students acquire perceptual representational competencies. This chapter originally appeared in the Journal of Cognitive Science (Mason et al., 2019a) and is collaborative with its authors.

1.5 Actively Learning Nearest Neighbor graphs

Though it is powerful, metric learning is not ideal for some tasks where one wishes to learn from human data. When either the number of features, d, or the number

of salient features, k, is large, collecting sufficient data from participants to learn a full metric can be expensive. Furthermore, metric learning traditionally restricts practitioners to distances between items in a Euclidean space. In some learning tasks that use human generated data, such as computing word embeddings for natural language processing, Euclidean structure has been shown to be overly restrictive as compared to other metric spaces Dhingra et al. (2018). To address these two challenges, we developed the first method to actively learn nearest neighbor graphs from noisy data.

Actively learning nearest neighbor graphs gives practitioners a fast and flexible method to learn similarity and preference. Precisely, consider n points $\mathbf{x}_1, \cdots, \mathbf{x}_n$, in a metric space $(\mathcal{M}, d)$ where the distance function $d(\cdot, \cdot)$ is unknown. One wishes to learn the graph where each $\mathbf{x}_i$ is connected via an edge to its nearest neighbor $\mathbf{x}_{i^*} = \arg\min_{j \neq i} d(\mathbf{x}_i, \mathbf{x}_j)$. For instance, in the chemistry example, each molecule image is a point and edges connect pairs that students judge to be similar. Given noiseless access to the distance measure d, these graphs can be learned efficiently in as few as $O(n \log(n))$ distance measurements Vaidya (1989). While modelling human judgements as being from a latent distance function is common, people's judgments can be noisy Coombs (1964), and this makes existing techniques ill-suited. My thesis provides the first algorithm to achieve the optimal $O(n \log(n))$ rate for learning nearest neighbor graphs while only assuming *noisy* estimates of distance. The method is fully agnostic to the underlying metric space and does not require Euclidean assumptions. Furthermore, we demonstrate efficiency of our method empirically and theoretically, needing only $O(n \log(n)\Delta^{-2})$ queries in favorable settings, where Δ^{-2} accounts for the effect of noise. Using crowd-sourced data collected for a subset of the UT Zappos50K dataset, we apply our algorithm to learn which shoes people believe are most similar and show that it beats both an active baseline and ordinal embedding. This chapter originally appeared in (Mason et al., 2019b) and is collaborative with the authors of that work.

1.5.1 Actively Finding Every Near-Optimal Alternative

Crowdsourcing is commonly used to label datasets, but efficiently finding crowd-workers capable of performing specialized tasks, such as identifying dog breeds or molecules, can be challenging Doroudi et al. (2016). To quickly label a dataset, it is desirable to have as many workers as possible, but these workers must be highly accurate to minimize label noise Kazai et al. (2013). This problem may be modelled as one of *pure-exploration in multi-armed bandits*. Each worker has an unknown average accuracy that may be estimated by having them label individual points where the truth is known. The pool of workers may be modeled as n distributions, referred to as 'arms', with unknown means. Existing bandit objectives are ill-suited for settings where practitioners wish to find many arms with high means. The popular objective of finding the k arms with the largest means is unsuitable as it does not provide a guarantee for the arms it returns. For instance, when finding workers, actively searching for and hiring the 100 best workers if only 10 truly good workers exist in the pool is wasteful. Setting a fixed threshold and finding workers that exceed that threshold corrects this issue, but this raises the question of how high a threshold should be set. If it is too high, algorithms will not find any workers. Instead, to efficiently find skilled crowd-workers in as few samples as possible, we seek an active algorithm to find every worker that performs almost as well as the best worker.

We address this by proposing the all-ϵ identification objective. Given a set of n distributions with means $\mu_1, \cdots, \mu_n$ and a value of $\epsilon > 0$, the objective is to identify the subset of arms $\{i : \mu_i > (1-\epsilon)\max_j \mu_j\}$ with a fixed probability $1-\delta$. Semantically, this allows one to find every worker within a factor of $1-\epsilon$ of the best. For instance, $\epsilon = .05$ finds every worker with an average performance within 95% of the best. This objective is naturally robust to the underlying distribution of means. It guarantees that every near optimal arm will be detected and that no significantly suboptimal arm will be returned. In this chapter we present several algorithms that are tailored to different practical settings. To establish the optimality of these methods, we develop a new analytical tool to study finite-time lower bounds. Finite

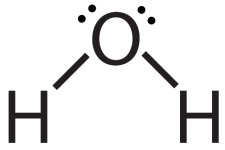
time guarantees are not captured by traditional bounding techniques but can have large practical impacts Simchowitz et al. (2017). Finally, in addition to strong theoretical results, we show that these algorithms perform well in practice. In particular, we show that one exceeds or matches the performance of several oracle baselines on a real-world dataset. This chapter will appear at NeurIPS 2020 Mason et al. (2020), and is collaborative with its authors.

2.1 Introduction

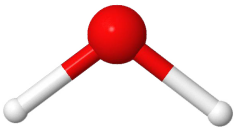
Visual representations are ubiquitous instructional tools in science, technology, engineering, and math (STEM) domains (Ainsworth, 2008; (US), 2006). For example, instructors use the visual representations shown in Figure 2.1 to help students learn about chemical bonding. Yet, to a novice student, these visual representations may not be helpful because the student may not know how to interpret the representations. For instance, does the red color in the ball-and-stick figure (Figure 2.1-b) mean the same thing as in the electrostatic potential map (EPM; Figure 2.1-d)? (It does not.)

Instructors often ask students to use visual representations that they have never seen before to make sense of concepts that they have not yet learned about (Wertsch and Kazak, 2011; Airey and Linder, 2009), an issue known as the *representation dilemma* (Dreher and Kuntze, 2015). Hence, to succeed in STEM, students need *representational competencies* that enable them to use visual representations to make sense of and solve domain-relevant problems (Ainsworth, 2006; Gilbert, 2005). One crucial representational competency is the ability to interpret visual representations; that is, to map visual representations to the abstract concepts they depict (Ainsworth, 2006; Schnotz, 2005). For example, students need to understand how the representations in Figure 2.1 show information about the molecule. For the Lewis structure (Figure 2.1-a), the student may map the unbonded electrons shown as dots to conceptual knowledge about how polarity in chemical molecules and infer that the water molecule has a local negative charge by the Oxygen atom.

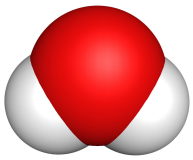
Educational technologies are particularly suitable to support representational competencies because they can provide adaptive support while students solve domain-relevant problems (Koedinger et al., 2006; VanLehn, 2011). Such adaptive support relies on a cognitive model that infers whether the student has learned target skills based on her/his interactions with the technology. Research shows that adapting instruction to students' representational competencies can enhance those



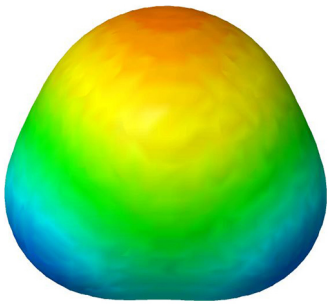
(a) A Lewis structure



(b) A ball and stick figure



(c) A space-filling model



(d) An electrostatic potential map

Figure 2.1: Visual representations of the water molecule.

competencies (Tuckey et al., 1991) and learning of domain knowledge (Davidowitz and Chittleborough, 2009).

However, educational technologies for representational competencies have two critical limitations. First, they typically focus on one set of representational competencies: students' conceptual understanding of representations (e.g., the ability to explain how visual features depict concepts). This focus mimics education psychology research's focus on conceptual learning (Ainsworth, 2006; Seufert, 2003). Conceptual knowledge is invariably intertwined with a second type of representational competency: *perceptual knowledge* (Kellman and Massey, 2013; Massey et al., 2013); the ability to rapidly and effortlessly recognize conceptual information based

on visual features of the representations. This ability results from textitimplicit forms of learning. For example, expert chemists simply "see" that the molecules depicted in Figure 2.1 have a local negative charge by the Oxygen atom, without having to make a an effortful conceptual inference.

Second, of the few educational technologies that enhance perceptual fluency, their adaptive capabilities are limited and their perceptual supports rely solely on performance measures (e.g., accuracy, response times) to adapt to students' representational competencies (Massey et al., 2013; Kellman et al., 2010). They do not use a cognitive model of the latent skills that students acquire through perceptual learning. As a result, they cannot provide specific feedback when students make mistakes. Decades of research showing that cognitive models can dramatically increase the effectiveness of educational technologies (VanLehn, 2011; Anderson et al., 1990) suggest that we must address this limitation and create adaptive instruction for perceptual knowledge.

These limitations likely result from cognitive modeling's traditional focus on explicit, verbally accessible knowledge. To develop cognitive models, researchers analyze how students think about target skills (Koedinger et al., 2006; Rau et al., 2013). We typically ask students to verbalize their problem-solving steps (Clark, 2014; Schraagen et al., 2000). Yet, verbalization is not suitable for assessing perceptual learning processes, which are implicit and not verbally accessible (Kellman and Massey, 2013; Koedinger et al., 2012). Therefore, instructional designers have to rely on "educated guesses" as to which visual features students may pay attention to. These educated guesses are based on the novice-expert literature, which documents the fact that novices tend to rely on surface features; that is, easily perceivable visual cues such as color and shape, to judge the similarity between stimuli items. By contrast, experts rely on visual features that are conceptually relevant and hence make more refined distinctions between visual features. Thus, to create adaptive perceptual supports, we need to develop cognitive models for perceptual learning.

Our research takes a first step towards developing a cognitive model for perceptual learning by assessing students' perceptual knowledge of a common visual representation in chemistry. In particular, we investigate *research question 1*: Which

visual features do students focus on when presented with visual representations? To address this question, we asked hundreds of students to judge the similarity between visual representations of molecules. We then used similarity learning–a method that provides a formal approach to investigating how people perceive similarity among visual stimuli. This method allowed us to estimate latent factors that account for the perceived similarity relationships between representations. Because we can map these latent factors to the visual features in the representations, this approach allows us to investigate which visual features are most salient to students' perceptions of similarity. Comparing these visual features to "educated guesses" allowed us to test *research question 2*: Do the visual features we identified as salient via metric learning correspond to visual features that students are expected to attend to based on the expert-novice literature on perceptual learning? In addition, we investigated a methodological *research question 3*: How many similarity judgments we need to assess students' perceptual knowledge?

Although we address these questions in the context of a particular domain with a particular visual representation, this chapter makes two important broader contributions. First, it provides an empirical validation of the "educated guesses" that developers of perceptual learning technologies typically rely on. Second, it establishes a methodology to assess perceptual knowledge that can serve as a basis for a cognitive model of perceptual learning. These contributions build the foundation for the development of adaptive instruction for perceptual knowledge and other implicit knowledge.

2.2 Experiment

2.2.1 Visual Representations of Molecules

For our experiment, we selected visual representations of chemical molecules common in undergraduate instruction. Lewis structure representations are the most commonly used visual representations in undergraduate chemistry textbooks. We reviewed textbooks and online instructional materials and listed the frequency

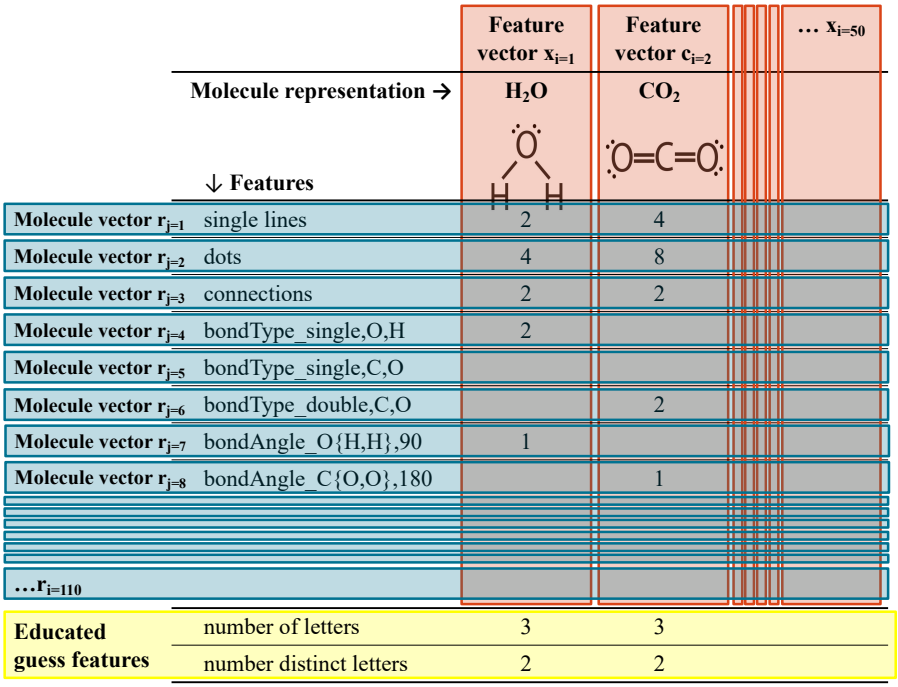


Figure 2.2: Example features for H_2O and CO_2 molecule representations with educated guess features in yellow, feature vectors in red, and molecule vectors in blue.

of all occurring molecules using their chemical names (e.g., H_2O) and common names (e.g., water). For our experiment, we chose the 50 most common molecules.

First, we created *educated guess features* (Figure 2.2, yellow) that correspond to expert assessments of which visual features students may attend to when making similarity judgments. To obtain these educated guesses, we reviewed the literature on chemistry expertise (Rappoport and Ashkenazi, 2008; Talanquer, 2009) and on perceptual learning (Kellman and Massey, 2013; Goldstone et al., 2010), and conducted learner-centered interviews with undergraduate and PhD students in chemistry (Rau and Evenstone, 2014). We identified 6 educated guess features: number of total letters, number of distinct letters, number of total bonds, number of single bonds, number of unbonded electrons, and molecule geometry (linear, planar, tetrahedral).

To investigate which visual features drive students' similarity judgments, we quantitatively described the visual features of the molecule representations. To this end, we created *feature vectors* for each of the molecules (see Figure 2.2, red) that describe which visual features the representation contains (e.g., bond angles, the numbers of specific atoms, or the numbers of different atoms present). The feature vectors of our corpus of molecule representations contained a total of 110 features. The 50 feature vectors collectively form matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{50}]$ where $\mathbf{x}_i \in \mathbb{R}^{110}$ is the feature vector for the ith molecule.

We aggregated each element of the feature vectors into *molecule vectors* for individual features (Figure 2.2, blue). Each molecule vector consisted of 50 values describing how many times the feature occurred in each representation. As molecule vectors make up the rows of our matrix of 110 features by 50 molecules shown in Figure 2.2, we will refer to the molecule vector for the jth feature as $\mathbf{r}_j$. Thus, feature vectors provide a numeric description of the visual information present in each representation, whereas molecule vectors provide a numeric description of overall patterns of visual features in the dataset for all representations.

2.2.2 Similarity Judgment Tasks

Students completed similarity judgment tasks that were presented as triplet comparisons (see Figure 2.3). Given a representation of a molecule (the "target-molecule"), students were asked to choose which molecule was most similar to the given one. For each task, the student chose between one of the two choice-molecules that they perceived to be more similar to the target-molecule. After each task, another triplet was generated uniformly at random from our corpus of molecule representations.

We delivered the similarity judgment tasks via NEXT; a cloud- based machine learning platform (Jamieson et al., 2015). NEXT allows users to upload their own content and query participants to perform judgment tasks. It uses machine learning algorithms to automate data collection and analyze results. More information about the plat- form can be found at `http://nextml.org`. In NEXT, students first received a brief description of the study and then worked through a sequence of 50 similarity

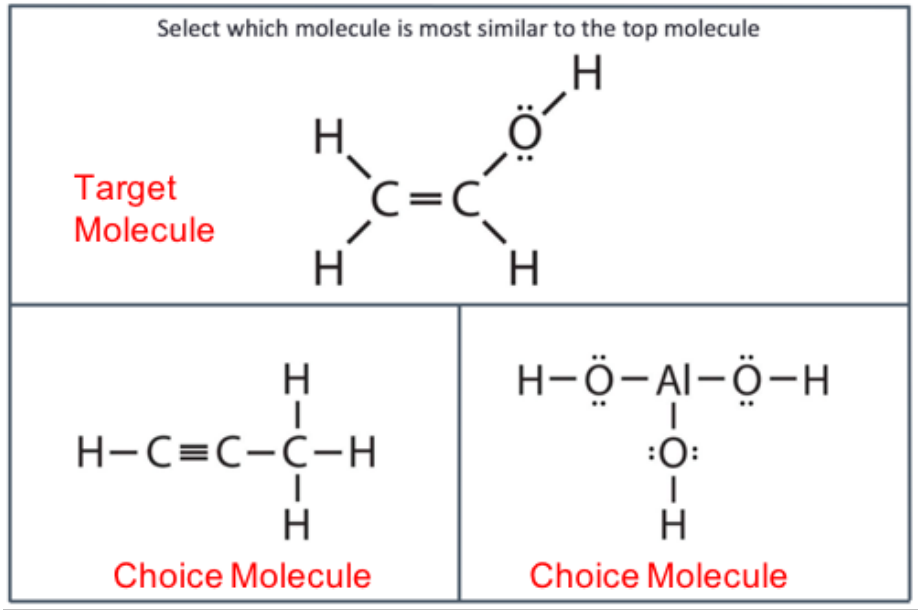


Figure 2.3: Example of a similarity judgment task: given the molecule on the top, students were asked which of the two molecules at the bottom is most similar.

judgment tasks. Students were instructed that these tasks are not a test and that there is right or wrong answer, but that we they are simply asked about their personal perceptions of similarities among molecule representations.

2.2.3 Dataset

Undergraduate students enrolled in an introductory chemistry course at a large U.S. university were invited to participate in a survey on learning with visual representations. The course had an enrollment of 781 students. Participation was voluntary. Altogether, we collected 26,180 responses from 563 (possibly non-unique) students. 61.6% of the students completed all 50 similarity judgment tasks. On average, students completed 46.5 tasks. Each similarity judgment in response to a triplet comparison task was associated with the feature vectors ($\mathbf{x}_i$) and molecule vectors ($\mathbf{r}_j$) of the three molecule representations, as described in section 2.2.1.

2.3 Analysis

In the following, we describe how we used similarity learning to investigate which visual features drive students' similarity judgments. We first provide a brief introduction into the similarity learning method in general. Then, we describe how we applied this method to our dataset in particular.

2.3.1 Introduction to Similarity Learning

In general, the goal of similarity learning is to learn a similarity function f that agrees with students' similarity judgments in the following sense: if item i is judged to be more similar to j than to k, then $f(i,j) < f(i,k)$. The function f can be thought as quantifying the perceived distance or dissimilarity between pairs. Alternatively, the function could quantify the perceptual similarity (inverse distance) between pairs, in which case $f(i,j) > f(i,k)$.

People are better at providing ordinal (i.e., comparative) responses than at providing fine-grained quantitative judgments or ratings Stewart et al. (2005). For example, when asked to compare the visual representations in Figure 2.3, people find it easier to judge whether the target molecule is more similar to the left or the right choice molecule than to judge their similarity on a rating scale. However, it is challenging to learn embeddings from comparisons due to the sheer number of possible triplet comparisons that could be made; the number of distinct triplets is proportional to n^3. For example, in our case of $n = 50$ molecule representations, there exist nearly $125,000$ distinct triplets. Researchers have observed that while triplet comparisons are easy to answer, they can become tedious and boring after extended sessions (Bijmolt and Wedel, 1995). Since we hypothesize that perceived dissimilarities can be accurately represented in d-dimensional space, it is reasonable to conjecture that if the embedding dimension is low (i.e., $d \ll n$), then there will be a high degree of redundancy among the triplet comparisons. In fact, researchers have observed that a small subset of these triplet comparisons often suffice to learn a reasonably accurate embedding, lending support to this conjecture (Agarwal et al., 2007; Johnson, 1973; Tamuz et al., 2011).

2.3.2 Similarity Learning Approaches

We applied two similarity learning approaches in this chapter: similarity learning by ranking (Chechik et al., 2010) and non-metric multi-dimensional scaling. In both cases, we modeled the perceptual similarity between molecules i and j as

$$S_{ij} = x_i^T A x_j$$

Here A is a symmetric matrix that parameterizes the model. The (k, l)th element of the matrix, A_{kl}, represents the importance of the interaction of feature k and feature l in the model. Since we assume A is symmetric $A_{kl} = A_{lk}$ and $S_{kl} = S_{lk}$. Before introducing these approaches, let us define some notation. There are N triplet comparisons. For the nth triplet, let i_n denote the target-molecule and let j_n and k_n denote the two choice-molecules. Let y_n denote the student's judgment, specifically $y_n = +1$ if the student decided j_n was more similar to i_n and $y_n = -1$ otherwise. Each of the $p = 50$ diagrams also has m associated features (e.g., numbers of different atoms, bonds, etc.). Arrange the features for each molecule representation into an $m \times 1$ molecular feature vector, and the $m \times 1$ feature vectors into a $m \times P$ matrix, X. The ith column of X, denoted x_i, contains the m features for molecule i. The jth row of X, denoted r_j, is a molecule vector for feature j containing the value of feature j for all 50 representations.

2.3.2.1 Approach 1: Similarity Learning by Ranking

This approach learns matrix A in our model of perceptual similarity directly from triplet responses via linear regression.

$$S_{ij} = x_i^T A x_j$$

where x_i and x_j are $m \times 1$ dimensional feature vectors of the m features of molecule representations i and j. The matrix A is $m \times m$, and the similarity learning problem is to estimate A that minimizes the number of disagreements between the ranking predictions for each triple (i.e., either $S_{ij} > S_{ik}$ or vice-versa) and the comparative

judgments collected from the students, as proposed by (Chechik et al., 2010).

The first step in this analysis was to estimate $\mathbf{A}$. Formally, the estimation of $\mathbf{A}$ can be written as the following optimization problem. Let $\mathbb{S}_m$ be the set of all $m \times m$ symmetric matrices. Solve for $\mathbf{A}$ that minimizes:

$$\widehat{A} := \arg\min_{\mathbf{A}\in\mathbb{S}_m} \sum_{n=1}^{N} (y_n - \boldsymbol{x}_{i_n}^T \mathbf{A} \boldsymbol{x}_{j_n} + \boldsymbol{x}_{i_n}^T \mathbf{A} \boldsymbol{x}_{k_n})^2.$$

The matrix $\mathbf{A}$ that minimizes the sum of squared errors weights the similarities between the diagram features so as to predict perceptual similarity judgments. In general, the solution $\mathbf{A}$ will place some weight on all m features. We anticipate that the visual features that are not salient do not strongly affect students' similarity judgments and therefore have lower weights in $\mathbf{A}$.

Taking this thinking a step further, we could consider many different optimizations of the type above, where in each case we use different subsets of the features, in order to determine which are most predictive of student judgments. Indeed, some features may- be totally irrelevant and worsen, rather than help, the prediction of students' similarity judgments. Unfortunately, searching over all possible subsets of features is computationally infeasible, so we instead consider the following optimization that approximates this search problem called `sparse-COMET` (Atzmon et al., 2015).

$$\widehat{A} := \arg\min_{\mathbf{A}\in\mathbb{S}_m} \sum_{n=1}^{N} (y_n - \boldsymbol{x}_{i_n}^T \mathbf{A} \boldsymbol{x}_{j_n} + \boldsymbol{x}_{i_n}^T \mathbf{A} \boldsymbol{x}_{k_n})^2 + \lambda \sum_{k=1}^{m} \|\mathbf{A}(k,:)\|_2$$

This optimization method uses a cost function that consists of two terms. The first term represents least squares data-fitting cost in the previous optimization. The second term is a Group LASSO penalty, which encourages solutions that have many columns equal to 0. If a column in $\mathbf{A}$ is all zero, then the corresponding feature is not used for prediction. The number of zero-valued columns in the solution depends on $\lambda > 0$. Note that we recover the previous optimization when $\lambda = 0$. Larger values of λ produce sparser solutions that effectively use fewer features. Features crucial for prediction are excluded only if λ is exceedingly large.

The second step in this analysis was to tune the parameter λ and then to assess the prediction accuracy of our method. To this end, we used 10-fold cross validation. Specifically, we randomly split the complete dataset into 10 equal sized subsets. We removed 2 random subsets as hold-out data and kept the remaining data as training data. We then solved the optimization above with the training data over a range of different λ values. For each λ, we scored prediction accuracy on one set of hold-out data to select the optimal value. Then, using our chosen λ value, we solved the optimization again to obtain a final $\mathbf{A}$ using $9/10$ of the data, and assessed the prediction accuracy on remaining $1/10$ of the data.

The final step was to rank the features based on the weights in matrix. Due to the Group LASSO penalty in the loss function, many of the columns in the resulting matrix are zero. To get the aggregate weight of each relevant feature, we computed the length (norm) of each non-zero column and ranked accordingly.

2.3.2.2 Approach 2: Ordinal Embedding

In this approach, rather than directly making predictions of similarity based on feature vectors and triplet responses, we first used students' similarity judgments to learn an embedding that spatially represents the similarity of molecule representations as distances in 2-dimensional space. We then identified molecule vectors that account for the distribution of molecule representations in the embedding space.

The first step in this analysis was to learn an embedding. We applied non-metric multidimensional scaling (NMDS) to the 26, 180 triplet comparison responses collected from the experiment to learn an embedding of the 50 molecule representations in a two-dimensional space (Agarwal et al., 2007). Embedding in two dimensions allows visualizing the perceived similarity computed by NMDS. The embedding reflects the consensus among students as to which molecular representations were more or less similar. We created 50 different embeddings, using multiple random initializations per embedding in order to account for the non-convexity of NMDS.

The second step was to validate the embedding. To this end, we computed a

distance matrix for each embedding. To validate the distance matrices, we used the following cross-validation procedure. We selected 6000 triplet comparison responses uniformly at random to serve as a hold-out dataset. From the remaining triplets, we randomly selected training sets of different size, ranging from 1000 to 20,000 triplet comparison responses. We computed embeddings for each training set. We then used these embeddings and the associated distance matrices to predict students' similarity judgments. Next, we used the distances in the embedding as a predictor of judgments in the hold-out set; the prediction errors quantify how well the embedding reflects the judgments. We repeated this procedure for training sets of different size. We performed 50-fold cross validation to calculate average prediction error on the learned embeddings. This procedure allowed assessing how prediction performance relates to the training set size (i.e., how many triplets were used to compute an embedding).

The third step in our analysis, after validating our embedding procedure, was to compute an embedding and corresponding distance matrix from the full set of triplets. Since the distance between points in the embedding corresponds to their perceived dissimilarity, we computed a similarity matrix defined as the element-wise inverse of the distance matrix, scaled from 0 to 1. This becomes matrix $\mathbf{S}$.

The fourth step was to identify which features, represented by the feature vectors, drive students' similarity judgments. Because the embedding was performed in 2 dimensions, we can consider the problem of only choosing 2 feature vectors to combine and compare combinations of pairs of feature vectors to the similarity matrix. For each possible pair, we performed a least squares optimization to find the ideal uniform scaling to match an outer product of our feature vectors to the similarity matrix.

$$\widehat{A} := \arg\min_{\mathbf{A}\in\mathbb{S}_m} \sum_{i,j=1}^{p} (\mathbf{S}_{i,j} - \mathbf{x}_i^T \mathbf{A} \mathbf{x}_j)^2$$

subject to $\mathbf{A}_{s,t} = 0$ for all s, t not equal to k, l or l, k. In other words, only let the k, l elements of $\mathbf{A}$ be non-zero and optimize these. This equates to fitting $\mathbf{S}$ to the molecule vectors for features k and i. Here, $\mathbf{S}_{i,j}$ represents the value of

the perceptual similarity between molecules i and j from the embedding. The magnitude of resulting value of $\mathbf{A}_{kl}$ tells us how important the interaction of features k and i is in representing the similarity. This is basically a correlation coefficient, and it only gauges the marginal value of this interaction (i.e., in isolation of all other interactions). In each case, after learning a matrix $\mathbf{A}$ we computed the corresponding residual value between similarity matrix $\mathbf{S}$ and our combination of 2 features. After performing all possible combinations of pairs of features, we ranked pairs of features in ascending order of residual values, with the smallest residuals being the best approximation of our observed similarity matrix. To evaluate the feature rankings, we used 10-fold cross-validation by performing identical tests on 10 different similarity matrices computed from different embeddings based on equal numbers of triplets to ensure that the original embedding and the non-convexity of NMDS was not a factor in the final ranking of feature pairs.

2.4 Results

2.4.1 Identifying Important Visual Features

To address *research question 1*, we used the two similarity learning approaches just described to identify which visual features account for students similarity judgments.

2.4.1.1 Approach 1: Similarity Learning by Ranking

Recall that the first approach entailed learning a similarity function that describes students' perceived similarity between molecule representations. This approach yielded an average 69% prediction accuracy of students' similarity judgments (assessed via 10-fold cross validation). This finding indicates that there was consensus over which representations were more or less similar, but also that there were some disagreements among students' similarity judgments.

To identify which visual features account for students similarity judgments, we estimated the weights for each feature in the learned matrix $\mathbf{A}$. The stronger a

Feature	Avg. Weight
Distinct letters	4.50%
Single bonds between Oxygen and Hydrogen	3.45%
180-degree angle in Hydrogen-Carbon-Fluorine	3.16%
Double bonds between Oxygen and Nitrogen	3.03%
Number of Nitrogen atoms	2.99%
Double bonds between Carbon and Oxygen	2.78%
120-degree angle in Hydrogen-Carbon-Hydrogen	2.73%
Number of Oxygen atoms	2.64%
180-degree angle in Carbon-Carbon-Oxygen	2.62%
Single bonds between Carbon and Oxygen	2.37%

Table 2.1: Top 10 features from the ranking of features with strong weights obtained by Approach 1.

feature's weight in $\mathbf{A}$, the more this feature affected students' similarity judgments. Hence, the feature's weight corresponds to its saliency in students' perception of molecule representations.

Table 2.1 shows the 10 most important features, as determined by a ranking of features according to their aggregate weight computed from matrix $\mathbf{A}$. These results show that the most highly ranked feature is the number of distinct letters, which corresponds to an aggregate educated guess feature. Specific visual features that are relevant to organic molecules were also ranked highly (e.g., the number of single bonds between Oxygen and Hydrogen atoms, the number of bonds between Carbon and Oxygen, the number of Nitrogen and Oxygen atoms). These specific visual features were present in many of the molecules in our dataset. Several visual features also included geometric aspects, specifically bond angles. These features indicate the presence of chemical functional groups that are relevant to predicting molecule's reactive behaviors.

2.4.1.2 Approach 2: Ordinal Embedding

Recall that approach s learns an embedding that represents the similarity of molecule representations as distances in a d-dimensional space, from which we then extracted

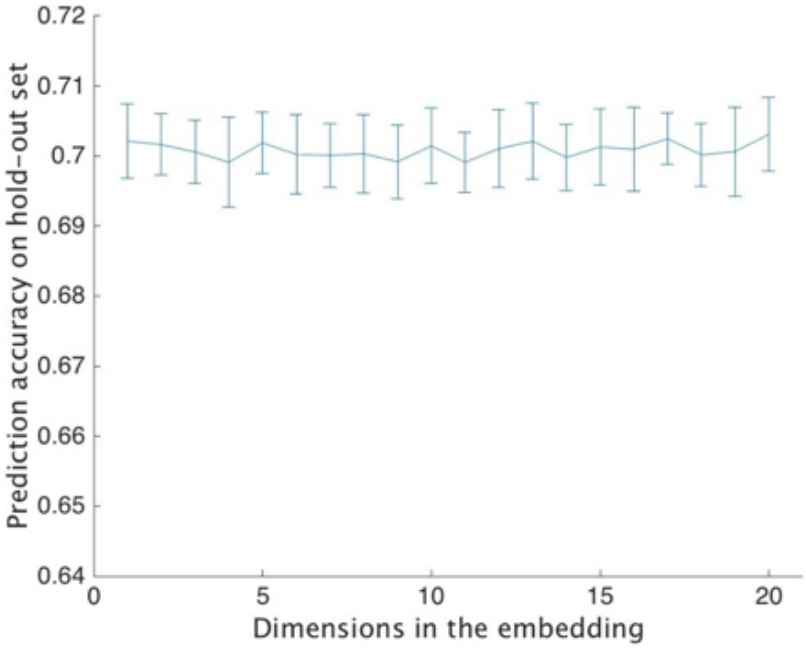


Figure 2.4: Prediction accuracy on hold-out set by number of dimensions in embedding.

the most important features. First, we established how many dimensions we need to consider (i.e., which d to choose in representing similarity of molecule representations in a d-dimensional space). Using the process of 50-fold cross validation described above, we calculated unit through 20 dimensional embeddings of perceptual similarity. We used 20, 000 triplets in this computation to ensure that the number of triplets did not affect the prediction accuracy as the dimension became large. Figure 2.4 shows that there is no drop in prediction accuracy when embedding in low dimensions versus high, suggesting that perceptual similarity can be accurately rep- resented in a low dimensional subspace, and that there is a high degree of redundancy in the data. This result shows that students' responses agreed on the relative similarity about 70% of the time.

Next, we generated a 2-dimensional embedding that describes students' perceived similarity between the molecule representations. Figure 5 shows this embed-

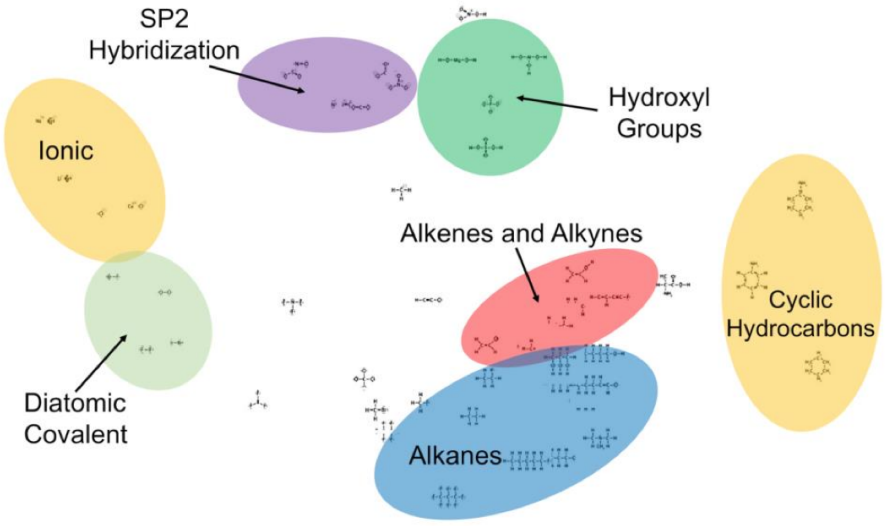


Figure 2.5: 2-dimensional similarity embedding. Distances between molecule representations correspond to students' perceived dissimilarity between them (i.e., molecule representations that are depicted close to one another are perceived to be similar).

ding, illustrating that molecules naturally form clusters based on their perceptual similarity. These clusters correspond to specific chemical properties shared among the molecules, such the presence of a particular type of bond or a functional group. We color-coded and labeled some of these clusters to illustrate these characteristics of students' perceptions. This illustration lends face validity to our embedding approach.

From this embedding, we extracted an ordered list of the feature pairs that best capture students' similarity judgments, shown in Table 2.2. The feature pairs in this table were ranked based on how well they approximate the similarity matrix computed from the embedding in Figure 2.5. The same feature may appear twice in a pair to account for the possibility that a weighted combination of a feature with itself better reflects the observed similarity structure than does a pair of features. In sum, these results show that the most highly ranked features are general visual features, which correspond to the aggregate educated guess features (e.g., number

Feature	Avg. Weight
1	Distinct Letters & Distinct Letters
2	Total letters & Distinct Letters
3	Different Letters & Single Bonds
4	Total Bonds & Distinct Letters
5	Different Letters & Carbons
6	Hydrogens & Distinct Letters
7	Total Letters & Total Letters
8	Total Letters & Single Letters
9	Total Letters & Unbonded Electrons
10	Distinct Letters & Carbon-Hydrogen Bonds

Table 2.2: Top 10 feature pairs from the learned embeddings (approach 2). Each row corresponds to a pair of feature vectors ranked in accordance with how accurately they described the observed similarity structure from the embedding.

of letters, number of lines). Specific visual features that are relevant to hydrocarbon molecules were also ranked highly (e.g., the number of Carbon and Hydrogen atoms). These specific features were present in many of the molecules in our dataset.

2.4.1.3 Comparing the Similarity Learning Approaches

While both methods agreed upon the top ranked feature, the similarity learning by ranking approach ranked structural features of the representations that were relevant to hydrocarbons and organic molecules more highly. As the ranking from this method follow predictive power, this ranking indicates that students' judgments of similarity can best be predicted, and therefore explained, through a combination of the number of different letters and the structural features involving Carbon, Hydrogen, and Oxygen.

2.4.2 Comparison with "Educated Guesses"

To address *research question 2* (do the visual features we identified as salient via metric learning correspond to visual features that students are expected to attend

to?), we compared the results from the similarity learning approaches to the educated guess features that we had determined based on the expert-novice literature on perceptual learning. Overall, the results from both metric learning approaches agree with the educated guesses: aggregate features that describe general visual features were ranked to be most important by both metric learning approaches. The similarity learning by ranking approach also yielded a number of visual features that are specific to the types of molecules in our corpus; in particular, visual representations that are highly relevant for comparing organic molecules.

2.4.3 Number of Similarity Judgments Needed

We addressed our methodological research question 3 (how many similarity judgments we need to assess students' perceptual knowledge) with the ordinal embedding approach. Specifically, we tested how many triplet comparisons are required to compute a representative embedding of the underlying similarity. Figure 6 shows that gains in prediction accuracy of the embedding were no longer statistically significant beyond 7000 triplet comparisons.

2.4.4 Differences Between the Two Approaches

The two methods are different and potentially complementary. There is no definitively correct way to fit the common model $\mathbf{S}_{ij} = \mathbf{x}_i^T \mathbf{A} \mathbf{x}_j$ to data. The main differences in the final rankings they produce stems from how we are learning matrix $\mathbf{A}$ and the restrictions we put on its structure. In approach 1 we are directly working with triplet responses which are perhaps noisy due to disagreements in students' individual judgments of perceptual similarity, but we are placing fewer restrictions on the learned matrix, allowing for more feature interaction. In approach 2, NMDS is useful for capturing perceived similarity in aggregate, but we enforce much stronger restrictions on the structure of $\mathbf{A}$, namely that only two features may interact at once, giving a clearer picture of the importance of a pair of features.

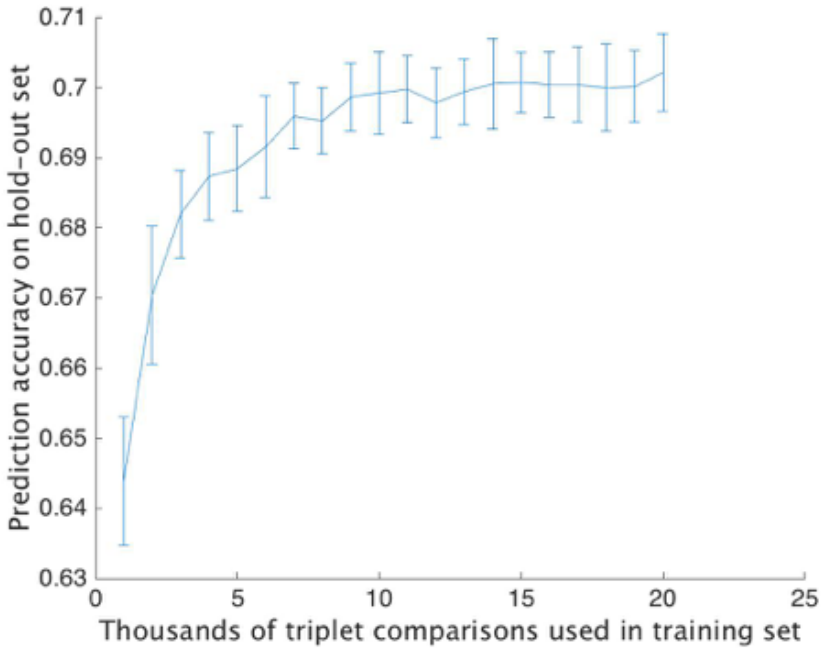


Figure 2.6: Prediction accuracy on hold-out set by number of triplet comparison judgments used in the training set.

2.5 Discussion

We applied similarity learning approaches to assess which visual features students focus on when presented with visual representations. We compared two approaches, one that allows us to assess the predictive power of the identified features, and one that allows representing the perceived similarity in a d-dimensional space. Both approaches yield similar results as to which visual features are salient to students. Hence, both approaches address research question 1: Which visual features do students focus on when presented with visual representations? We found that students' similarity judgments of Lewis structures appear to be driven by general visual features such as the number of total and distinct letters, as well as by visual features specific to the types of molecules in our dataset (e.g., number of Hydrogen / Carbon atoms).

Our results also address research question 2: Do the visual features we identified as salient via similarity learning correspond to visual features that students are expected to attend to based on the expert-novice literature on perceptual learning? We found that the identified general visual features align with educated guesses based on the literatures on expertise and perceptual learning, which validates the common "educated guess" approach that instructional designers have to rely on in the absence of assessments of perceptual knowledge. Our results also suggest that, in addition to these general features, students learn to pay attention to key visual features that are highly domain-specific; such as features that indicate the presence of functional groups that are predictive of chemical behaviors. Furthermore, our results show that a few key features predict students' perceptions of similarity between visual representations with accuracy of about 70%.

Finally, we addressed our methodical research question 3: How many similarity judgments we need to assess students' perceptual knowledge? Our results show that about $7,000$ responses to triplet comparison tasks are sufficient in assessing a population's perceptual knowledge. Using a survey with 50 triplet comparison tasks (as in our experiment), that means an N of 140 participants will yield valid assessments of perceptual knowledge.

2.6 Limitations

Although both similarity learning approaches had rigorous theoretical backing, we made a few assumptions about our triplet comparison data that had inherent limitations of note. In both of these methods, we are not modeling individual students, but rather the population as a whole. Consequently, we assume that the triplets and therefore the judgments of similarity are independent of one another. This assumption allows us to learn the rankings of features and feature pairs for the students' collectively, but it does not provide a ranking for an individual. Further, because judging similarity representations is a subjective task, students' judgments may in certain cases conflict with one another. Even with an extremely large number of similarity judgments, complete consensus is unlikely, and therefore, perfect

prediction of student judgments is similarly difficult to achieve. Hence, future research needs to investigate how to expand the present approach to modeling individual perceptual knowledge.

Another limitation pertains to the ordinal embedding procedure. For visualization purposes, we embedded the molecules into a 2-dimensional space. Higher dimensional embedding may more accurately capture perceptual dissimilarities. Future research should explore this question.

2.7 Future Directions

We will expand our research to other types of visual representations typically used in chemistry instruction (see Figure 2.1). Further, we will gather data from expert chemists and compare them to data from novices and advanced learners. Based on this comparison, we will identify a "perceptual knowledge gap" between students and experts. Specifically, we will identify visual features that experts attend to but students do not.

Further, we will expand similarity learning so that it can assess an individual student's perceptual knowledge in real time. The current approach is limited in that it requires a large number of similarity judgments to assess students' perceptual knowledge, which is only feasible if we are interested in assessing perceptual knowledge of a population of interest (e.g., novices, advanced students, experts), and because we assume independence among similarity judgments. To address this limitation, we will combine our similarity learning approach with cognitive modeling methods (e.g., Bayesian knowledge tracing). For example, a similarity judgment survey may provide a prior for in a cognitive model, and students' performance on perceptual learning tasks may inform the choice of representations for a small number similarity judgment tasks interspersed in the learning activity.

This expansion will provide the basis for the design of adaptive instruction for perceptual knowledge that can provide appropriate sequences of perceptual learning tasks that draw students' attention to visual features they yet have to learn. Further, knowing which visual features students have not yet learned can serve

as a basis for the design of visual feedback that highlights visual features when students make mistakes on perceptual learning tasks.

In sum, we will use the similarity learning approach described in this chapter both to design instruction for perceptual learning and to assess perceptual knowledge as a learning outcome.

2.8 Conclusions

This chapter described a new approach to assess students' perceptual knowledge. We used this approach to validate the "educated guesses" approach. In addition, we offer more formal pathways for instructional designers to create perceptual learning assessments. Because developing adaptive instruction for perceptual knowledge relies on such assessments, this chapter makes an important contribution to cognitive modeling research.

This chapter also makes important contributions to machine learning. We provide a new mathematical approach to quantify the accuracy of perceptual embeddings learned from similarity judgments. Specifically, we derived bounds on the accuracy of embeddings learned from small numbers of comparative judgments by adapting recently developed large-sample analysis methods (Arias-Castro et al., 2017). This approach provided new algorithms for generating embeddings that are provably accurate. We investigated new methods for embedding based on spectral methods inspired by spectral ranking algorithms (Negahban et al., 2012). Our experiment yielded an empirical validation with perceptual data from undergraduates, as well as new machine learning methods to assess how visual features predict or encode perceptual similarity judgments. Specifically, we explored the application of group Lasso algorithms for automatically selecting the most perceptually salient features (Yuan and Lin, 2006). Our experiment empirically evaluated the group Lasso approach.

In sum, our work provides a crucial stepping stone towards adaptive instruction for perceptual knowledge. Perceptual knowledge is by definition implicit and does not lend itself to the kinds of techniques used in traditional cognitive modeling

approaches (e.g., think-alouds, interviews). We presented and evaluated two similarity learning approaches that can determine which visual features students attend to when perceiving visual representations.

Acknowledgements We thank Professor John Moore in the Chemistry Department for his help in recruiting participants for this study, and the Learning Understanding Cognition Intelligence and Data Science group at UW Madison for their suggestions.

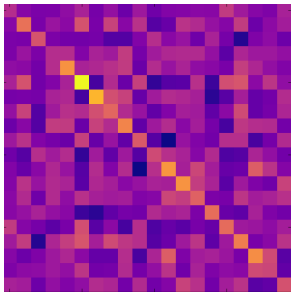
3.1 Low-Dimensional Metric Learning

This chapter studies the problem of learning a low-dimensional Euclidean metric from comparative judgments. Specifically, consider a set of n items with high-dimensional features $\mathbf{x}_i \in \mathbb{R}^p$ and suppose we are given a set of (possibly noisy) distance comparisons of the form

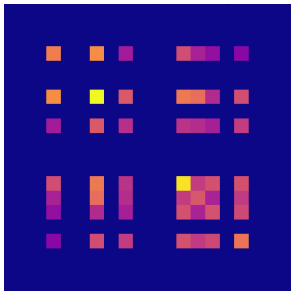
$$\text{sign}(\text{dist}(\mathbf{x}_i, \mathbf{x}_j) - \text{dist}(\mathbf{x}_i, \mathbf{x}_k)),$$

for a subset of all possible triplets of the items. Here we have in mind comparative judgments made by humans and the distance function implicitly defined according to human perceptions of similarities and differences. For example, the items could be images and the $\mathbf{x}_i$ could be visual features automatically extracted by a machine. Accordingly, our goal is to learn a $p \times p$ symmetric positive semi-definite (psd) matrix $\mathbf{K}$ such that the metric $d_{\mathbf{K}}(\mathbf{x}_i, \mathbf{x}_j) := (\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{K}(\mathbf{x}_i - \mathbf{x}_j)$, where $d_{\mathbf{K}}(\mathbf{x}_i, \mathbf{x}_j)$ denotes the squared distance between items i and j with respect to a matrix $\mathbf{K}$, predicts the given distance comparisons as well as possible. Furthermore, it is often desired that the metric is *low-dimensional* relative to the original high-dimensional feature representation (i.e., $\text{rank}(\mathbf{K}) \leqslant d < p$). There are several motivations for this:

- Learning a high-dimensional metric may be infeasible from a limited number of comparative judgments, and encouraging a low-dimensional solution is a natural regularization.
- Cognitive scientists are often interested in visualizing human perceptual judgments (e.g., in a two-dimensional representation) and determining which features most strongly influence human perceptions. For example, educational psychologists in Rau et al. (2016) collected comparisons between visual representations of chemical molecules in order to identify a small set of vi-



(a) A general low rank psd matrix



(b) A sparse and low rank psd matrix

Figure 3.1: Examples of $\mathbf{K}$ for $p = 20$ and $d = 7$. The sparse case depicts a situation in which only some of the features are relevant to the metric.

sual features that most significantly influence the judgments of beginning chemistry students.

- It is sometimes reasonable to hypothesize that a small subset of the high-dimensional features dominate the underlying metric (i.e., many irrelevant features).
- Downstream applications of the learned metric (e.g., for classification purposes) may benefit from robust, low-dimensional metrics.

With this in mind, several authors have proposed nuclear norm and $\ell_{1,2}$ group lasso norm regularization to encourage low-dimensional and sparse metrics as in Fig. 3.1b (see Bellet et al. (2015) for a review). Relative to such prior work, the contributions of this work are three-fold:

1. We develop novel upper bounds on the generalization error and sample complexity of learning low-dimensional metrics from triplet distance comparisons. Notably, unlike previous generalization bounds, our bounds allow one to easily quantify how the feature space dimension p and rank or sparsity $d < p$ of the underlying metric impacts the sample complexity.
2. We establish minimax lower bounds for learning low-rank and sparse metrics that match the upper bounds up to polylogarithmic factors, demonstrating

the optimality of learning algorithms for the first time. Moreover, the upper and lower bounds demonstrate that learning sparse (and low-rank) metrics is essentially as difficult as learning a general low-rank metric. This suggests that nuclear norm regularization may be preferable in practice, since it places less restrictive assumptions on the problem.

3. We use the generalization error bounds to obtain model identification error bounds that quantify the accuracy of the learned $\mathbf{K}$ matrix. This problem has received very little, if any, attention in the past and is crucial for interpreting the learned metrics (e.g., in cognitive science applications). This is a bit surprising, since the term "metric learning" strongly suggests accurately determining a metric, not simply learning a predictor that is parameterized by a metric.

3.1.1 Comparison with Previous Work

There is a fairly large body of work on metric learning which is nicely reviewed and summarized in the monograph Bellet et al. (2015), and we refer the reader to it for a comprehensive summary of the field. Here we discuss a few recent works most closely connected to this work. Several authors have developed generalization error bounds for metric learning, as well as bounds for downstream applications, such as classification, based on learned metrics. To use the terminology of Bellet et al. (2015), most of the focus has been on must-link/cannot-link constraints and less on relative constraints (i.e., triplet constraints as considered in this chapter). Generalization bounds based on algorithmic robustness are studied in Bellet and Habrard (2015), but the generality of this framework makes it difficult to quantify the sample complexity of specific cases, such as low-rank or sparse metric learning. Rademacher complexities are used to establish generalization error bounds in the must-link/cannot-link situation in Guo and Ying (2014); Ying et al. (2009); Bian and Tao (2012), but do not consider the case of relative/triplet constraints. The sparse compositional metric learning framework of Shi et al. (2014) does focus on relative/triplet constraints and provides generalization error bounds in terms of

covering numbers. However, this work does not provide bounds on the covering numbers, making it difficult to quantify the sample complexity. To sum up, prior work does not quantify the sample complexity of metric learning based on relative/triplet constraints in terms of the intrinsic problem dimensions (i.e., dimension p of the high-dimensional feature space and the dimension d of the underlying metric), there is no prior work on lower bounds, and no prior work quantifying the accuracy of learned metrics themselves (i.e., only bounds on prediction errors, not model identification errors). Finally we mention that Fazel et al. Oymak et al. (2015) also consider the recovery of sparse and low rank matrices from linear observations. Our situation is very different, our matrices are low rank because they are sparse - not sparse and simultaneously low rank as in their case.

3.2 The Metric Learning Problem

Consider n known points $\mathbf{X} := [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n] \in \mathbb{R}^{p \times n}$. We are interested in learning a symmetric positive semidefinite matrix $\mathbf{K}$ that specifies a metric on $\mathbb{R}^p$ given ordinal constraints on distances between the known points. Let $\mathcal{S}$ denote a set of triplets, where each $t = (i, j, k) \in \mathcal{S}$ is drawn uniformly at random from the full set of $n\binom{n-1}{2}$ triplets $\mathcal{T} := \{(i, j, k) : 1 \leqslant i \neq j \neq k \leqslant n, j < k\}$. For each triplet, we observe a $y_t \in \{\pm 1\}$ which is a noisy indication of the triplet constraint $d_{\mathbf{K}}(x_i, x_j) < d_{\mathbf{K}}(x_i, x_k)$. Specifically we assume that each t has an associated probability q_t of $y_t = -1$, and all y_t are statistically independent.

Objective 1: Compute an estimate $\widehat{\mathbf{K}}$ from $\mathcal{S}$ that predicts triplets as well as possible.

In many instances, our triplet measurements are noisy observations of triplets from a true positive semi-definite matrix $\mathbf{K}^*$. In particular we assume

$$q_t > 1/2 \iff d_{\mathbf{K}^*}(\mathbf{x}_i, \mathbf{x}_j) < d_{\mathbf{K}^*}(\mathbf{x}_i, \mathbf{x}_k) \, .$$

We can also assume an explicit known *link function*, $f : \mathbb{R} \to [0, 1]$, so that $q_t = f(d_{\mathbf{K}^*}(\mathbf{x}_i, \mathbf{x}_j) - d_{\mathbf{K}^*}(\mathbf{x}_i, \mathbf{x}_k))$.

Objective 2: Assuming an explicit known link function f estimate $\mathbf{K}^*$ from $\mathcal{S}$.

3.2.1 Definitions and Notation

Our triplet observations are nonlinear transformations of a linear function of the Gram matrix $\mathbf{G} := \mathbf{X}^T\mathbf{K}\mathbf{X}$. Indeed for any triple $t = (i, j, k)$, define

$$\begin{aligned} \mathbf{M}_t(\mathbf{K}) &:= d_{\mathbf{K}}(x_i, x_j) - d_{\mathbf{K}}(x_i, x_k) \\ &= x_i^T\mathbf{K}x_k + x_k^T\mathbf{K}x_i - x_i^T\mathbf{K}x_j - x_j^T\mathbf{K}x_i + x_j^T\mathbf{K}x_j - x_k^T\mathbf{K}x_k \,. \end{aligned}$$

So for every $t \in \mathcal{S}$, y_t is a noisy measurement of $\operatorname{sign}(\mathbf{M}_t(\mathbf{K}))$. This linear operator may also be expressed as a matrix

$$\mathbf{M}_t := x_i x_k^T + x_k x_i^T - x_i x_j^T - x_j x_i^T + x_j x_j^T - x_k x_k^T \,,$$

so that $\mathbf{M}_t(\mathbf{K}) = \langle \mathbf{M}_t, \mathbf{K} \rangle = \operatorname{Trace}(\mathbf{M}_t^T\mathbf{K})$. We will use $\mathbf{M}_t$ to denote the operator and associated matrix interchangeably. Ordering the elements of $\mathcal{T}$ lexicographically, we let $\mathcal{M}$ denote the linear map,

$$\mathcal{M}(\mathbf{K}) = (\mathbf{M}_t(\mathbf{K}) | \text{ for } t \in \mathcal{T}) \in \mathbb{R}^{n\binom{n-1}{2}}$$

Given a PSD matrix $\mathbf{K}$ and a sample, $t \in \mathcal{S}$, we let $\ell(y_t\langle \mathbf{M}_t, \mathbf{K} \rangle)$ denote the loss of $\mathbf{K}$ with respect to t; e.g., the 0-1 loss $\mathbb{1}_{\{\operatorname{sign}(y_t\langle \mathbf{M}_t, \mathbf{K} \rangle) \neq 1\}}$, the hinge-loss $\max\{0, 1 - y_t\langle \mathbf{M}_t, \mathbf{K} \rangle\}$, or the logistic loss $\log(1 + \exp(-y_t\langle \mathbf{M}_t, \mathbf{K} \rangle))$. Note that we insist that our losses be functions of our triplet differences $\langle \mathbf{M}_t, \mathbf{K} \rangle$. Further, note that this makes our losses invariant to rigid motions of the points x_i. Other models proposed for metric learning use scale-invariant loss functions Heim et al. (2015).

For a given loss ℓ, we then define the empirical risk with respect to our set of observations $\mathcal{S}$ to be

$$\widehat{R}_{\mathcal{S}}(\mathbf{K}) := \frac{1}{|\mathcal{S}|} \sum_{t \in \mathcal{S}} \ell(y_t\langle \mathbf{M}_t, \mathbf{K} \rangle).$$

This is an unbiased estimator of the true risk $R(\mathbf{K}) := \mathbb{E}[\ell(y_t\langle \mathbf{M}_t, \mathbf{K} \rangle)]$ where the

expectation is taken with respect to a triplet t selected uniformly at random and the random value of y_t.

Finally, we let $\mathbf{I}_n$ denote the identity matrix in $\mathbb{R}^{n\times n}$, $\mathbf{1}_n$ the n-dimensional vector of all ones and $\mathbf{V} := \mathbf{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T$ the *centering matrix*. In particular if $\mathbf{X} \in \mathbb{R}^{p\times n}$ is a set of points, $\mathbf{XV}$ subtracts the mean of the columns of $\mathbf{X}$ from each column. We say that $\mathbf{X}$ is centered if $\mathbf{XV} = 0$, or equivalently $\mathbf{X1}_n = 0$. If $\mathbf{G}$ is the Gram matrix of the set of points $\mathbf{X}$, i.e. $\mathbf{G} = \mathbf{X}^T\mathbf{X}$, then we say that $\mathbf{G}$ is centered if $\mathbf{X}$ is centered or if equivalently, $\mathbf{G1}_n = 0$. Furthermore we use $\|\cdot\|_*$ to denote the nuclear norm, and $\|\cdot\|_{1,2}$ to denote the mixed $\ell_{1,2}$ norm of a matrix, the sum of the ℓ_2 norms of its rows. Unless otherwise specified, we take $\|\cdot\|$ to be the standard operator norm when applied to matrices and the standard Euclidean norm when applied to vectors. Finally we define the $\mathbf{K}$-norm of a vector as $\|\mathbf{x}\|_{\mathbf{K}}^2 := \mathbf{x}^T\mathbf{K}\mathbf{x}$.

3.2.2 Sample Complexity of Learning Metrics.

In most applications, we are interested in learning a matrix $\mathbf{K}$ that is low-rank and positive-semidefinite. Furthermore as we will show in Theorem 3.1, such matrices can be learned using fewer samples than general psd matrices. As is common in machine learning applications, we relax the rank constraint to a nuclear norm constraint. In particular, let our constraint set be

$$\mathcal{K}_{\lambda,\gamma} = \{\mathbf{K} \in \mathbb{R}^{p\times p} | \mathbf{K} \text{ positive-semidefinite}, \|\mathbf{K}\|_* \leqslant \lambda, \max_{t\in\mathcal{T}} \langle \mathbf{M}_t, \mathbf{K}\rangle \leqslant \gamma\}.$$

Up to constants, a bound on $\langle \mathbf{M}_t, \mathbf{K}\rangle$ is a bound on $\mathbf{x}_i^T\mathbf{K}\mathbf{x}_i$. This bound along with assuming our loss function is Lipschitz, will lead to a tighter bound on the deviation of $\widehat{R}_S(\mathbf{K})$ from $R(\mathbf{K})$ crucial in our upper bound theorem.

Let $\mathbf{K}^* := \min_{\mathbf{K}\in\mathcal{K}_{\lambda,\gamma}} R(\mathbf{K})$ be the true risk minimizer in this class, and let $\widehat{\mathbf{K}} := \min_{\mathbf{K}\in\mathcal{K}_{\lambda,\gamma}} \widehat{R}_S(\mathbf{K})$ be the empirical risk minimizer. We achieve the following prediction error bounds for the empirical risk minimzer.

Theorem 3.1. *Fix $\lambda, \gamma, \delta > 0$. In addition assume that $\max_{1\leqslant i\leqslant n} \|\mathbf{x}_i\|^2 = 1$. If the loss*

function ℓ is L*-Lipschitz, then with probability at least* $1-\delta$

$$R(\widehat{\mathbf{K}}) - R(\mathbf{K}^*) \leqslant 4L\left(\sqrt{\frac{140\lambda^2\frac{\|\mathbf{X}\mathbf{X}^T\|}{n}\log p}{|\mathcal{S}|}} + \frac{2\log p}{|\mathcal{S}|}\right) + \sqrt{\frac{2L^2\gamma^2\log 2/\delta}{|\mathcal{S}|}}$$

Note that past generalization error bounds in the metric learning literature have failed to quantify the precise dependence on observation noise, dimension, rank, and our features $\mathbf{X}$. Consider the fact that a $p \times p$ matrix with rank d has $O(dp)$ degrees of freedom. With that in mind, one expects the sample complexity to be also roughly $O(dp)$. We next show that this intuition is correct if the original representation $\mathbf{X}$ is isotropic (i.e., has no preferred direction).

The Isotropic Case. Suppose that $\mathbf{x}_1, \cdots, \mathbf{x}_n$, $n > p$, are drawn independently from the isotropic Gaussian $\mathcal{N}(\mathbf{0}, \frac{1}{p}\mathbf{I})$. Furthermore, suppose that $\mathbf{K}^* = \frac{p}{\sqrt{d}}\mathbf{U}\mathbf{U}^T$ with $\mathbf{U} \in \mathbb{R}^{p\times d}$ is a generic (dense) orthogonal matrix with unit norm columns. The factor $\frac{p}{\sqrt{d}}$ is simply the scaling needed so that the average magnitude of the entries in $\mathbf{K}^*$ is a constant, independent of the dimensions p and d. In this case, $\operatorname{rank}(\mathbf{K}^*) = d$ and $\|\mathbf{K}^*\|_F = \operatorname{trace}(\mathbf{U}^T\mathbf{U}) = p$. These two facts imply that the tightest bound on the nuclear norm of $\mathbf{K}^*$ is $\|\mathbf{K}^*\|_* \leqslant p\sqrt{d}$. Thus, we take $\lambda = p\sqrt{d}$ for the nuclear norm constraint. Now let $z_i = \sqrt{\frac{p}{\sqrt{d}}}\mathbf{U}^T\mathbf{x}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$ and note that $\|\mathbf{x}_i\|_{\mathbf{K}}^2 = \|z_i\|^2 \sim \chi_d^2$. Therefore, $\mathbb{E}\|\mathbf{x}_i\|_{\mathbf{K}}^2 = d$ and it follows from standard concentration bounds that with large probability $\max_i \|\mathbf{x}_i\|_{\mathbf{K}}^2 \leqslant 5d\log n =: \gamma$ see Davidson and Szarek (2001). Also, because the $\mathbf{x}_i \sim \mathcal{N}(\mathbf{0}, \frac{1}{p}\mathbf{I})$ it follows that if $n > p\log p$, say, then with large probability $\|\mathbf{X}\mathbf{X}^T\| \leqslant 5n/p$. We now plug these calculations into Theorem 3.1 to obtain the following corollary.

Corollary 3.2 (Sample complexity for isotropic points)**.** *Fix* $\delta > 0$*, set* $\lambda = p\sqrt{d}$*, and assume that* $\|\mathbf{X}\mathbf{X}^T\| = O(n/p)$ *and* $\gamma := \max_i \|\mathbf{x}_i\|_K^2 = O(d\log n)$*. Then for a generic* $\mathbf{K}^* \in \mathcal{K}_{\lambda,\gamma}$*, as constructed above, with probability at least* $1-\delta$*,*

$$R(\widehat{\mathbf{K}}) - R(\mathbf{K}^*) \;=\; O\left(\sqrt{\frac{dp(\log p + \log^2 n)}{|\mathcal{S}|}}\right)$$

This bound agrees with the intuition that the sample complexity should grow roughly like dp, the degrees of freedom on $\mathbf{K}^*$. Moreover, our minimax lower bound in Theorem 3.4 below shows that, ignoring logarithmic factors, the general upper bound in Theorem 3.1 is unimprovable in general.

Beyond low rank metrics, in many applications it is reasonable to assume that only a few of the features are salient and should be given nonzero weight. Such a metric may be learned by insisting $\mathbf{K}$ to be row sparse in addition to being low rank. Whereas learning a low rank $\mathbf{K}$ assumes that distance is well represented in a low dimensional subspace, a row sparse (and hence low rank) $\mathbf{K}$ defines a metric using only a subset of the features. Figure 3.1 gives a comparison of a low rank versus a low rank and sparse matrix $\mathbf{K}$.

Analogous to the convex relaxation of rank by the nuclear norm, it is common to relax row sparsity by using the mixed $\ell_{1,2}$ norm. In fact, the geometry of the $\ell_{1,2}$ and nuclear norm balls are tightly related as the following lemma shows.

Lemma 3.3. *For a symmetric positive semi-definite matrix* $\mathbf{K} \in \mathbb{R}^{p \times p}$, $\|\mathbf{K}\|_* \leqslant \|\mathbf{K}\|_{1,2}$.

Proof. $$\|\mathbf{K}\|_{1,2} = \sum_{i=1}^{p} \sqrt{\sum_{j=1}^{p} \mathbf{K}_{i,j}^2} \geqslant \sum_{i=1}^{p} \mathbf{K}_{i,i} = \mathrm{Trace}(\mathbf{K}) = \sum_{i=1}^{p} \lambda_i(\mathbf{K}) = \|\mathbf{K}\|_* \qquad \square$$

This implies that the $\ell_{1,2}$ ball of a given radius is contained inside the nuclear norm ball of the same radius. In particular, it is reasonable to assume that it is easier to learn a $\mathbf{K}$ that is sparse in addition to being low rank. Surprisingly, however, the following minimax bound shows that this is not necessarily the case.

To make this more precise, we will consider optimization over the set

$$\mathcal{K}'_{\lambda,\gamma} = \{\mathbf{K} \in \mathbb{R}^{p \times p} | \mathbf{K} \text{ positive-semidefinite}, \|\mathbf{K}\|_{1,2} \leqslant \lambda, \max_{t \in \mathcal{T}} \langle \mathbf{M}_t, \mathbf{K} \rangle \leqslant \gamma\}.$$

Furthermore, we must specify the way in which our data could be generated from noisy triplet observations of a fixed $\mathbf{K}^*$. To this end, assume the existence of a *link function* $f : \mathbb{R} \to [0, 1]$ so that $q_t = \mathbb{P}(y_t = -1) = f(\mathbf{M}_t(\mathbf{K}^*))$ governs the observations. There is a natural associated logarithmic loss function ℓ_f corresponding to

the log-likelihood, where the loss of an arbitrary $\mathbf{K}$ is

$$\ell_f(y_t\langle \mathbf{M}_t, \mathbf{K}\rangle) = \mathbb{1}_{\{y_t=-1\}} \log \frac{1}{f(\langle \mathbf{M}_t, \mathbf{K}\rangle)} + \mathbb{1}_{\{y_t=1\}} \log \frac{1}{1-f(\langle \mathbf{M}_t, \mathbf{K}\rangle)}$$

Theorem 3.4. *Choose a link function* f *and let* ℓ_f *be the associated logarithmic loss. For* p *sufficiently large, then there exists a choice of* γ, λ, $\mathbf{X}$, *and* $|\mathcal{S}|$ *such that*

$$\inf_{\widehat{\mathbf{K}}} \sup_{\mathbf{K}\in\mathcal{K}'_{\lambda,\gamma}} \mathbb{E}[R(\widehat{\mathbf{K}})] - R(\mathbf{K}) \geqslant C\sqrt{\frac{C_1^3 \ln 4}{2} \frac{\lambda^2 \frac{\|\mathbf{X}\mathbf{X}^T\|}{n}}{|\mathcal{S}|}}$$

where $C = \frac{C_f^2}{32}\sqrt{\frac{\inf_{|x|\leqslant\gamma} f(x)(1-f(x))}{\sup_{|v|\leqslant\gamma} f'(v)^2}}$ *with* $C_f = \inf_{|x|\leqslant\gamma} f'(x)$, C_1 *is an absolute constant, and the infimum is taken over all estimators* $\widehat{\mathbf{K}}$ *of* $\mathbf{K}$ *from* $|\mathcal{S}|$ *samples.*

Importantly, up to polylogarithmic factors and constants, our minimax lower bound over the $\ell_{1,2}$ ball matches the upper bound over the nuclear norm ball given in Theorem 3.1. *In particular, in the worst case, learning a sparse and low rank matrix* $\mathbf{K}$ *is no easier than learning a* $\mathbf{K}$ *that is simply low rank.* However in many realistic cases, a slight performance gain is seen from optimizing over the $\ell_{1,2}$ ball when $\mathbf{K}^*$ is row sparse, while optimizing over the nuclear norm ball does better when $\mathbf{K}^*$ is dense. We show examples of this in the Section 3.3. The proof is given in the supplementary materials.

Note that if γ is in a bounded range, then the constant C has little effect. For the case that f is the logistic function, $C_f \geqslant \frac{1}{4}e^{-y_t\langle \mathbf{M}_t, \mathbf{K}\rangle} \geqslant \frac{1}{4}e^{-\gamma}$. Likewise, the term under the root will be also be bounded for γ in a constant range. The terms in the constant C arise when translating from risk and a KL-divergence to squared distance and reflects the noise in the problem.

3.2.3 Sample Complexity Bounds for Identification

Under a general loss function and arbitrary $\mathbf{K}^*$, we can not hope to convert our prediction error bounds into a recovery statement. However in this section we will

show that as long as $\mathbf{K}^*$ is low rank, and if we choose the loss function to be the log loss ℓ_f of a given link function f as defined prior to the statement of Theorem 3.4, recovery is possible.

Firstly, note that under these assumptions we have an explicit formula for the risk,

$$R(\mathbf{K}) = \frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}} f(\langle \mathbf{M}_t, \mathbf{K}^*\rangle)\log\frac{1}{f(\langle \mathbf{M}_t, \mathbf{K}\rangle)} + (1 - f(\langle \mathbf{M}_t, \mathbf{K}^*\rangle))\log\frac{1}{1-f(\langle \mathbf{M}_t, \mathbf{K}\rangle)}$$

and

$$R(\mathbf{K}) - R(\mathbf{K}^*) = \frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}} KL(f(\langle \mathbf{M}_t, \mathbf{K}^*\rangle)\|f(\langle \mathbf{M}_t, \mathbf{K}\rangle)).$$

The following theorem shows that if the excess risk is small, i.e. $R(\widehat{\mathbf{K}})$ approximates $R(\mathbf{K}^*)$ well, then $\mathcal{M}(\widehat{\mathbf{K}})$ approximates $\mathcal{M}(\mathbf{K}^*)$ well. The proof, given in the supplementary materials, uses standard Taylor series arguments to show the KL-divergence is bounded below by squared-distance.

Lemma 3.5. *Let* $C_f = \inf_{|x|\leqslant\gamma} f'(x)$. *Then for any* $\mathbf{K} \in \mathbf{K}_{\lambda,\gamma}$,

$$\frac{2C_f^2}{|\mathcal{T}|}\|\mathcal{M}(\mathbf{K}) - \mathcal{M}(\mathbf{K}^*)\|^2 \leqslant R(\mathbf{K}) - R(\mathbf{K}^*).$$

The following may give us hope that recovering $\mathbf{K}^*$ from $\mathcal{M}(\mathbf{K}^*)$ is trivial, but the linear operator $\mathcal{M}$ is non-invertible in general, as we discuss next. To see why, we must consider a more general class of operators defined on Gram matrices. Given a symmetric matrix $\mathbf{G}$, define the operator $\mathbf{L}_t$ by

$$\mathbf{L}_t(\mathbf{G}) = 2\mathbf{G}_{ik} - 2\mathbf{G}_{ij} + \mathbf{G}_{jj} - \mathbf{G}_{kk}$$

If $\mathbf{G} = \mathbf{X}^T\mathbf{K}\mathbf{X}$ then $\mathbf{L}_t(\mathbf{G}) = \mathbf{M}_t(\mathbf{K})$, and more so $\mathbf{M}_t = \mathbf{X}\mathbf{L}_t\mathbf{X}^T$. Analogous to $\mathcal{M}$, we will combine the $\mathbf{L}_t$ operators into a single operator $\mathcal{L}$,

$$\mathcal{L}(\mathbf{G}) = (\mathbf{L}_t(\mathbf{G})| \text{ for } t \in \mathcal{T}) \in \mathbb{R}^{n\binom{n-1}{2}}.$$

Lemma 3.6. *The null space of $\mathcal{L}$ is one dimensional, spanned by* $\mathbf{V} = \mathbf{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T$.

The proof is contained in the supplementary materials. In particular we see that $\mathcal{M}$ is not invertible in general, adding a serious complication to our argument. However $\mathcal{L}$ is still invertible on the subset of centered symmetric matrices orthogonal to $\mathbf{V}$, a fact that we will now exploit. We can decompose $\mathbf{G}$ into $\mathbf{V}$ and a component orthogonal to $\mathbf{V}$ denoted $\mathbf{H}$,

$$\mathbf{G} = \mathbf{H} + \sigma_{\mathbf{G}}\mathbf{V}$$

where $\sigma_{\mathbf{G}} := \frac{\langle \mathbf{G}, \mathbf{V}\rangle}{\|\mathbf{V}\|_F^2}$, and under the assumption that $\mathbf{G}$ is centered, $\sigma_G = \frac{\|\mathbf{G}\|_*}{n-1}$. Remarkably, the following lemma tells us that a *non-linear* function of $\mathbf{H}$ uniquely determines $\mathbf{G}$.

Lemma 3.7. *If $n > d + 1$, and $\mathbf{G}$ is rank d and centered, then $-\sigma_{\mathbf{G}}$ is an eigenvalue of $\mathbf{H}$ with multiplicity $n - d - 1$. In addition, given another Gram matrix $\mathbf{G}'$ of rank d', $\sigma_{\mathbf{G}'} - \sigma_{\mathbf{G}}$ is an eigenvalue of $\mathbf{H} - \mathbf{H}'$ with multiplicity at least $n - d - d' - 1$.*

Proof. Since $\mathbf{G}$ is centered, $\mathbf{1}_n \in \ker \mathbf{G}$, and in particular $\dim(\mathbf{1}_n^\perp \cap \ker \mathbf{G}) = n-d-1$. If $x \in \mathbf{1}_n^\perp \cap \ker \mathbf{G}$, then

$$\mathbf{G}x = \mathbf{H}x + \sigma_{\mathbf{G}}\mathbf{V}x \Rightarrow \mathbf{H}x = -\sigma_{\mathbf{G}}x.$$

For the second statement, notice that $\dim(\mathbf{1}_n^\perp \cap \ker \mathbf{G} - \mathbf{G}') \geqslant n - d - d' - 1$. A similar argument then applies. □

If $n > 2d$, then the multiplicity of the eigenvalue $-\sigma_G$ is at least $n/2$. So we can trivially identify it from the spectrum of $\mathbf{H}$. This gives us a *non-linear* way to recover $\mathbf{G}$ from $\mathbf{H}$.

Now we can return to the task of recovering $\mathbf{K}^*$ from $\mathcal{M}(\widehat{\mathbf{K}})$. Indeed the above lemma implies that $\mathbf{G}^*$ (and hence $\mathbf{K}^*$ if $\mathbf{X}$ is full rank) can be recovered from $\mathbf{H}^*$ by computing an eigenvalue of $\mathbf{H}^*$. However $\mathbf{H}^*$ is recoverable from $\mathcal{L}(\mathbf{H}^*)$, which is itself well approximated by $\mathcal{L}(\widehat{\mathbf{H}}) = \mathcal{M}(\widehat{\mathbf{K}})$. The proof of the following theorem makes this argument precise.

Theorem 3.8. *Assume that* $\mathbf{K}^*$ *is rank* d, $\widehat{\mathbf{K}}$ *is rank* d', $n > d + d' + 1$, $\mathbf{X}$ *is rank* p *and* $\mathbf{X}^T\mathbf{K}^*\mathbf{X}$ *and* $\mathbf{X}^T\widehat{\mathbf{K}}\mathbf{X}$ *are all centered. Let* $C_{d,d'} = \left(1 + \frac{n-1}{(n-d-d'-1)}\right)$. *Then with probability at least* $1-\delta$,

$$\frac{n\sigma_{min}(\mathbf{X}\mathbf{X}^T)^2}{|\mathcal{T}|}\|\widehat{\mathbf{K}} - \mathbf{K}^*\|_F^2 \leqslant \frac{2LC_{d,d'}}{C_f^2}\left[\left(\sqrt{\frac{140\lambda^2\frac{\|\mathbf{X}\mathbf{X}^T\|}{n}\log p}{|\mathcal{S}|}} + \frac{2\log p}{|\mathcal{S}|}\right) + \sqrt{\frac{2L^2\gamma^2\log\frac{2}{\delta}}{|\mathcal{S}|}}\right]$$

where $\sigma_{min}(\mathbf{X}\mathbf{X}^T)$ *is the smallest eigenvalue of* $\mathbf{X}\mathbf{X}^T$.

The proof, given in the supplementary materials, relies on two key components, Lemma 3.7 and a type of *restricted isometry property* for $\mathcal{M}$ on $\mathbf{V}^\perp$. Our proof technique is a streamlined and more general approach similar to that used in the special case of ordinal embedding. In fact, our new bound improves on the recovery bound given in Jain et al. (2016a) for ordinal embedding.

We have several remarks about the bound in the theorem. If $\mathbf{X}$ is well conditioned, e.g. isotropic, then $\sigma_{\min}(\mathbf{X}\mathbf{X}^T) \approx \frac{n}{p}$. In that case $\frac{n\sigma_{\min}(\mathbf{X}\mathbf{X}^T)^2}{|\mathcal{T}|} \approx \frac{1}{p^2}$, so the left hand side is the average squared error of the recovery. In most applications the rank of the empirical risk minimizer $\widehat{\mathbf{K}}$ is approximately equal to the rank of $\mathbf{K}^*$, i.e. $d \approx d'$. Note that If $d + d' \leqslant \frac{1}{2}(n-1)$ then $C_{d,d'} \leqslant 3$. Finally, the assumption that $\mathbf{X}^T\mathbf{K}^*\mathbf{X}$ are centered can be guaranteed by centering $\mathbf{X}$, which has no impact on the triplet differences $\langle \mathbf{M}_t, \mathbf{K}^*\rangle$, or insisting that $\mathbf{K}^*$ is centered. As mentioned above C_f will be have little effect assuming that our measurements $\langle \mathbf{M}_t, \mathbf{K}\rangle$ are bounded.

3.2.4 Applications to Ordinal Embedding

In the ordinal embedding setting, there are a set of items with unknown locations, $z_1, \cdots, z_n \in \mathbb{R}^d$ and a set of triplet observations $\mathcal{S}$ where as in the metric learning case observing $y_t = -1$, for a triplet $t = (i, j, k)$ is indicative of the $\|z_i - z_j\|^2 \leqslant \|z_i - z_k\|^2$, i.e. item i is closer to j than k. The goal is to recover the z_i's, up to rigid motions, by recovering their Gram matrix $\mathbf{G}^*$ from these comparisons. Ordinal embedding case reduces to metric learning through the following observation. Consider the case when $n = p$ and $\mathbf{X} = \mathbf{I}_p$, i.e. the x_i are standard basis vectors.

Letting $\mathbf{K}^* = \mathbf{G}^*$, we see that $\|x_i - x_j\|_{\mathbf{K}}^2 = \|z_i - z_j\|^2$. So in particular, $\mathbf{L}_t = \mathbf{M}_t$ for each triple t, and observations are exactly comparative distance judgements. Our results then apply, and extend previous work on sample complexity in the ordinal embedding setting given in Jain et al. (2016a). In particular, though Theorem 5 in Jain et al. (2016a) provides a consistency guarantee that the empirical risk minimizer $\widehat{\mathbf{G}}$ will converge to $\mathbf{G}^*$, they do not provide a convergence rate. We resolve this issue now.

In their work, it is assumed that $\|z_i\|^2 \leqslant \gamma$ and $\|\mathbf{G}\|_* \leqslant \sqrt{d}n\gamma$. In particular, sample complexity results of the form $O(dn\gamma \log n)$ are obtained. However, these results are trivial in the following sense, if $\|z_i\|^2 \leqslant \gamma$ then $\|\mathbf{G}\|_* \leqslant \gamma n$, and their results (as well as our upper bound) implies that true sample complexity is significantly smaller, namely $O(\gamma n \log n)$ which is independent of the ambient dimension d. As before, assume an explicit link function f with Lipschitz constant L, so the samples are noisy observations governed by $\mathbf{G}^*$, and take the loss to be the logarithmic loss associated to f.

We obtain the following improved recovery bound in this case. The proof is immediate from Theorem 3.8.

Corollary 3.9. *Let $\mathbf{G}^*$ be the Gram matrix of n centered points in d dimensions with $\|\mathbf{G}^*\|_F^2 = \frac{\gamma^2 n^2}{d}$. Let $\widehat{\mathbf{G}} = \min_{\|\mathbf{G}\|_* \leqslant \gamma n, \|\mathbf{G}\|_\infty \leqslant \gamma} R_{\mathcal{S}}(\mathbf{G})$ and assume that $\widehat{\mathbf{G}}$ is rank d, with $n > 2d + 1$. Then,*

$$\frac{\|\widehat{\mathbf{G}} - \mathbf{G}^*\|_F^2}{n^2} = O\left(\frac{LC_{d,d}}{C_f^2}\sqrt{\frac{\gamma n \log n}{|\mathcal{S}|}}\right)$$

3.3 Experiments

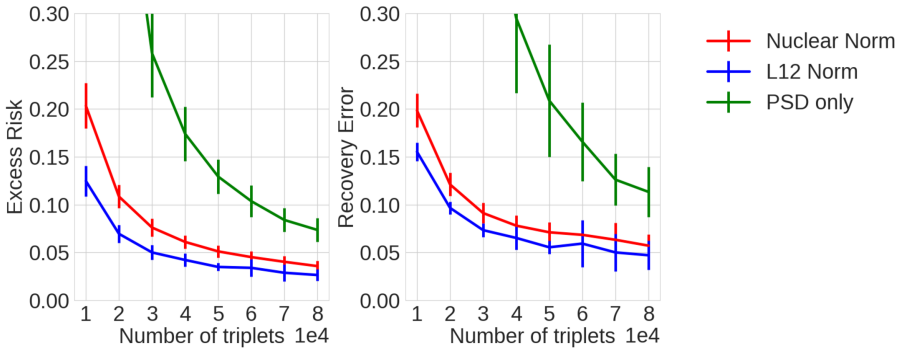
To validate our complexity and recovery guarantees, we ran the following simulations. We generate $x_1, \cdots, x_n \overset{\text{iid}}{\sim} \mathcal{N}(\mathbf{0}, \frac{1}{p}\mathbf{I})$, with $n = 200$, and $\mathbf{K}^* = \frac{p}{\sqrt{d}}\mathbf{U}\mathbf{U}^T$ for a random orthogonal matrix $\mathbf{U} \in \mathbb{R}^{p \times d}$ with unit norm columns. In Figure 3.2a, $\mathbf{K}^*$ has d nonzero rows/columns. In Figure 3.2b, $\mathbf{K}^*$ is a dense rank-d matrix. We compare the performance of nuclear norm and $\ell_{1,2}$ regularization in each setting against an unconstrained baseline where we only enforce that $\mathbf{K}$ be psd. Given a

fixed number of samples, each method is compared in terms of the relative excess risk, $\frac{R(\widehat{\mathbf{K}})-R(\mathbf{K}^*)}{R(\mathbf{K}^*)}$, and the relative squared recovery error, $\frac{\|\widehat{\mathbf{K}}-\mathbf{K}^*\|_F^2}{\|\mathbf{K}^*\|_F^2}$, averaged over 20 trials. The y-axes of both plots have been trimmed for readability.

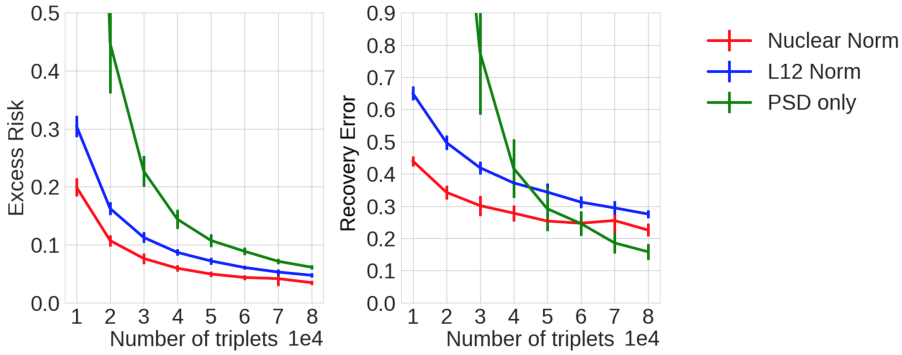
In the case that $\mathbf{K}^*$ is sparse, $\ell_{1,2}$ regularization outperforms nuclear norm regularization. However, in the case of dense low rank matrices, nuclear norm reularization is superior. Notably, as expected from our upper and lower bounds, the performances of the two approaches seem to be within constant factors of each other. Therefore, unless there is strong reason to believe that the underlying $\mathbf{K}^*$ is sparse, nuclear norm regularization achieves comparable performance with a less restrictive modeling assumption. Furthermore, in the two settings, both the nuclear norm and $\ell_{1,2}$ constrained methods outperform the unconstrained baseline, especially in the case where $\mathbf{K}^*$ is low rank and sparse.

To empirically validate our sample complexity results, we compute the number of samples averaged over 20 runs to achieve a relative excess risk of less than 0.1 in Figure 3.3. First, we fix $p = 100$ and increment d from 1 to 10. Then we fix $d = 10$ and increment p from 10 to 100 to clearly show the linear dependence of the sample complexity on d and p as demonstrated in Corollary 3.2. To our knowledge, these are the first results quantifying the sample complexity in terms of the number of features, p, and the embedding dimension, d.

Acknowledgments This work was partially supported by the NSF grants CCF-1218189 and IIS-1623605

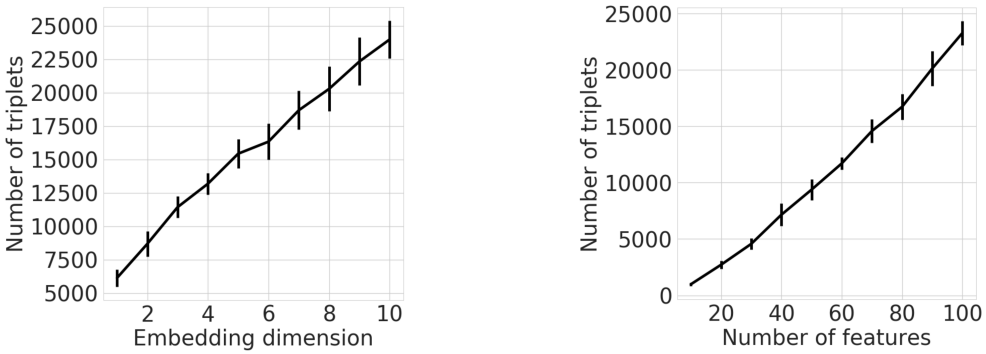


(a) Sparse low rank metric



(b) Dense low rank metric

Figure 3.2: $\ell_{1,2}$ and nuclear norm regularization performance



(a) d varying

(b) p varying

Figure 3.3: Number of samples to achieve relative excess risk < 0.1

3.A Proof of Results

3.A.1 Proof of Theorem 3.1

Our argument follows standard statistical learning theory techniques used in the classification literature. This framework is also similar to that used in the one bit matrix completion literature, see Davenport et al. (2014). The main ingredient in the proof is the use of a Matrix Bernstein to bound the Rademacher complexity of our class.

By the Bounded Difference inequality,

$$\begin{aligned} R(\widehat{K}) - R(K^\star) &= \mathbb{R}(\widehat{K}) - \widehat{R}(\widehat{K}) + \widehat{R}(\widehat{K}) - \widehat{R}(K^\star) + \widehat{R}(K^\star) - R(K^\star) \\ &\leqslant 2 \sup_{K \in \mathcal{K}_{\lambda,\gamma}} |\widehat{R}(K) - R(K)| \\ &\leqslant 2\mathbb{E}[\sup_{K \in \mathcal{K}_{\lambda,\gamma}} |\widehat{R}(K) - R(K)|] + \sqrt{\frac{2\beta^2 \log 2/\delta}{|\mathcal{S}|}}, \end{aligned}$$

where $\beta = \sup_{K \in \mathcal{K}_{\lambda,\gamma}} |\ell((y_t \langle M_t, K\rangle) - \ell((y_{t'} \langle M_{t'}, K\rangle)| \leqslant L\gamma$ since $\langle M_t, K\rangle \leqslant \gamma$. Using standard symmetrization and contraction lemmas, we can introduce Rademacher random variables $\varepsilon_t \in \{-1, 1\}$ for all $t \in \mathcal{T}$ so that

$$\begin{aligned} \mathbb{E}\left[\sup_{K \in \mathcal{K}_{\lambda,\gamma}} |\widehat{R}(K) - R(K)|\right] &\leqslant \mathbb{E}\frac{2L}{|\mathcal{S}|} \sup_{K \in \mathcal{K}_{\lambda,\gamma}} \left|\sum_{t \in \mathcal{S}} \varepsilon_t \langle M_t, K\rangle\right| \\ &\leqslant \mathbb{E}\frac{2L}{|\mathcal{S}|} \sup_{K \in \mathcal{K}_{\lambda,\gamma}} \|\sum_{t \in \mathcal{S}} \varepsilon_t M_t\| \|K\|_* \\ &\leqslant \mathbb{E}\frac{2L\lambda}{|\mathcal{S}|} \sup_{K \in \mathcal{K}_{\lambda,\gamma}} \|\sum_{t \in \mathcal{S}} \varepsilon_t M_t\| \end{aligned}$$

We employ a matrix Bernstein bound, Theorem 6.6.1 in Tropp (2015), to compute

$$\mathbb{E}\|\sum_{t\in\mathcal{S}}\varepsilon_t \mathbf{M}_t\| \leqslant \sqrt{140\frac{\|\mathbf{X}\mathbf{X}^T\|}{n}|\mathcal{S}|\log p} + 2\log p.$$

To see this, it suffices to bound $\left\|\sum_{t\in\mathcal{T}}\mathbf{M}_t^2\right\|$ which is done in Lemma 3.10. Plugging this in above gives

$$\mathbb{E}\frac{2L\lambda}{|\mathcal{S}|}\left\|\sum_{t\in\mathcal{S}}\varepsilon_t\mathbf{M}_t\right\| \leqslant 2L\left(\sqrt{\frac{140\lambda^2\frac{\|\mathbf{X}\mathbf{X}^T\|}{n}\log p}{|\mathcal{S}|}} + \frac{2\log p}{|\mathcal{S}|}\right)$$

Lemma 3.10.

$$\frac{1}{n\binom{n-1}{2}}\left\|\sum_{t\in\mathcal{T}}\mathbf{M}_t^2\right\| \leqslant 70\frac{\|\mathbf{X}\mathbf{X}^T\|}{n}$$

Proof. Let $\boldsymbol{e}_i$ be the i^{th} standard basis vector. For a triplet $t = (i, j, k)$, define

$$\mathbf{L}_t = \boldsymbol{e}_i\boldsymbol{e}_k^T + \boldsymbol{e}_k\boldsymbol{e}_i^T - \boldsymbol{e}_i\boldsymbol{e}_j^T - \boldsymbol{e}_j\boldsymbol{e}_i^T + \boldsymbol{e}_j\boldsymbol{e}_j^T - \boldsymbol{e}_k\boldsymbol{e}_k^T$$

(in particular $\mathbf{L}_t$ is the matrix corresponding to the operator $\mathbf{L}_t$ given in Section 2.3). A computation shows that $\langle\mathbf{L}_t, \mathbf{X}^T\mathbf{K}\mathbf{X}\rangle = \langle\mathbf{M}_t, \mathbf{K}\rangle$ and moreover $\mathbf{M}_t = \mathbf{X}\mathbf{L}_t\mathbf{X}^T$. By definition,

$$\begin{aligned}\sum_{t\in\mathcal{T}}\mathbf{M}_t^2 &= \sum_{t\in\mathcal{T}}\mathbf{X}\mathbf{L}_t\mathbf{X}^T\mathbf{X}\mathbf{L}_t\mathbf{X} \\ &= \mathbf{X}\left(\sum_{t\in\mathcal{T}}\mathbf{L}_t\mathbf{X}^T\mathbf{X}\mathbf{L}_t\right)\mathbf{X}^T\end{aligned}$$

We now focus our attention on simplifying the middle term. Firstly, note that we can assume that the $\mathbf{X}$'s are centered, i.e. $\mathbf{X}\mathbf{1}_n = \mathbf{0}$. To see this, note that the $\mathbf{L}_t$'s

are centered so in particular, $\mathbf{L}_t\mathbf{V} = \mathbf{L}_t$. Then

$$\mathbf{L}_t\mathbf{X}^T\mathbf{X}\mathbf{L}_t = \mathbf{L}_t\mathbf{V}^T\mathbf{X}^T\mathbf{X}\mathbf{V}\mathbf{L}_t = \mathbf{L}_t(\mathbf{X}\mathbf{V})^T(\mathbf{X}\mathbf{V})\mathbf{L}_T$$

so we can replace $\mathbf{X}$ with $\mathbf{X}\mathbf{V}$, i.e. we can center $\mathbf{X}$. Also note that note that centering $\mathbf{X}$ only diminishes the operator norm $\mathbf{X}\mathbf{X}^T$, so centering does not affect the statement of the bound, and furthermore a tighter statement is certainly possible by assuming that $\mathbf{X}$ is centered.

Using the reduction to a centered $\mathbf{X}$, a computation (omitted due to length) shows that

$$\left(\sum_{t\in T}\mathbf{L}_t\mathbf{X}^T\mathbf{X}\mathbf{L}_t\right)_{i,j} = \begin{cases} (2n-3)\|\mathbf{X}^T\mathbf{X}\|_* + (n^2-3n)\|x_i\|^2 & i = j \\ (n-4)\langle x_i, x_j\rangle - (n-2)\|x_j\|^2 - (n-2)\|x_i\|^2 - \|\mathbf{X}^T\mathbf{X}\|_* & i \neq j \end{cases}$$

To bound $\|\sum_{t\in\mathcal{T}}\mathbf{L}_t\mathbf{X}^T\mathbf{X}\mathbf{L}_t\| \leqslant 7n^2$, by Gershgorin's Circle Theorem we just have to bound the sums of the absolute values of the entries in each row. This ends up being,

$$\begin{aligned} &(2n-3+n-1)\|\mathbf{X}^T\mathbf{X}\|_* + (n^2-3n+(n-1)(n-2))\|x_i\|^2 + (n-2)\sum_{i\neq j}\|x_j\|^2 \\ &\qquad\qquad +(n-4)\sum_{i\neq j}|\langle x_i, x_j\rangle| \\ &\leqslant (2n-3+n-1+n-2)\|\mathbf{X}^T\mathbf{X}\|_* + (n^2-3n+(n-1)(n-2)-2)\|x_i\|^2 \\ &\qquad\qquad +(n-4)\sum_j\|x_i\|\|x_j\| \\ &\leqslant (4n-6)\|\mathbf{X}^T\mathbf{X}\|_* + (2n^2-6n)\|x_i\|^2 + n(n-4)\max_j\|x_j\|^2 \\ &\leqslant (4n-6)n\max_j\|x_j\|^2 + (2n^2-6n)\max_j\|x_j\|^2 + (n^2-4n)\max_j\|x_j\|^2 \\ &\leqslant 7n^2\max_j\|x_j\|^2 \end{aligned}$$

So $\|\sum_{t\in\mathcal{T}} \mathbf{L}_t\mathbf{X}^\top\mathbf{X}\mathbf{L}_t\| \leqslant 7n^2$ and

$$\frac{1}{n\binom{n-1}{2}}\left\|\sum_{t\in\mathcal{T}}\mathbf{X}\mathbf{L}_t\mathbf{X}^\top\mathbf{X}\mathbf{L}_t\mathbf{X}\right\| \leqslant 70\frac{\|\mathbf{X}\mathbf{X}^\top\|}{n}$$

using the fact that $\frac{2n^2}{(n-1)(n-2)} \leqslant 10$ for positive $n \geqslant 3$. □

3.A.2 Proof of Theorem 3.4

We will need the following lemma relating the KL-divergence to squared distance in this section and in the proof of Theorem 3.8.

Lemma 3.11. *Let* $y, z \in (0,1)$, *then*

$$2(z-y)^2 \leqslant \mathrm{KL}(z\|y) \leqslant \frac{(z-y)^2/2}{\inf_{x\in(0,1)} x(1-x)}$$

Proof. For $y, z \in (0,1)$ let $g(z) = z\log\frac{z}{y} + (1-z)\log\frac{1-z}{1-y}$. Then $g'(z) = \log\frac{z}{1-z} - \log\frac{y}{1-y}$ and $g''(z) = \frac{1}{z(1-z)}$. By Taylor's theorem, for some η in the interval between y and z, $g(z) = \frac{g''(\eta)}{2}(z-y)^2$. So for a lower bound,

$$g(z) \geqslant \frac{(z-y)^2/2}{\sup_{x\in(0,1)} x(1-x)} \geqslant 2(z-y)^2.$$

Similarly an upper bound is given by,

$$g(z) \leqslant \frac{(z-y)^2/2}{\inf_{x\in(0,1)} x(1-x)}$$

□

Now we resume the proof of Theorem 2.3. Fix $\mathbf{X} = \mathbf{I}$. Given triplet comparisons generated according to $\mathbf{K}$, we are interested in finding the minimax lower bound,

$$\inf_{\widehat{\mathbf{K}}} \sup_{\mathbf{K}\in\mathcal{K}'_{\lambda,\gamma}} \mathbb{E}[R(\widehat{\mathbf{K}})] - R(\mathbf{K})$$

Where as previously computed in Section 3.2.3

$$R(\widehat{\mathbf{K}})-R(\mathbf{K}) = \frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}} f(\langle \mathbf{M}_t, \mathbf{K}\rangle)\log\frac{f(\langle \mathbf{M}_t, \mathbf{K}\rangle)}{f(\langle \mathbf{M}_t, \widehat{\mathbf{K}}\rangle)}+(1-f(\langle \mathbf{M}_t, \mathbf{K}\rangle))\log\frac{1-f(\langle \mathbf{M}_t, \mathbf{K}\rangle)}{1-f(\langle \mathbf{M}_t, \widehat{\mathbf{K}}\rangle)}$$

Lemma 3.5 implies,

$$R(\widehat{\mathbf{K}}) - R(\mathbf{K}) \geqslant \frac{2C_f^2}{|\mathcal{T}|}\|\mathcal{M}(\mathbf{K}) - \mathcal{M}(\widehat{\mathbf{K}})\|_2^2.$$

where $C_f = \inf_{|x|\leqslant\gamma} f'(x)$. We will construct a set $\kappa \subset \mathcal{K}'_{\lambda,\gamma}$ so that for any two $\mathbf{K}^1, \mathbf{K}^2 \in \kappa$, with $K_1 \not= K_2$,

- $\frac{2C_f^2}{|\mathcal{T}|}\|\mathcal{M}(\mathbf{K}^1) - \mathcal{M}(\mathbf{K}^2)\|_F^2 \geqslant 4s_n^2$, for $\mathbf{K}^1 \neq \mathbf{K}^2$
- Let $P^{\mathcal{S}}_{\mathbf{K}}$ denote the sample distribution of a set of $|\mathcal{S}|$ samples conditioned on it being drawn from $\mathbf{K} \in \kappa$. Then we also require $KL(P^{\mathcal{S}}_{\mathbf{K}^1}\|P^{\mathcal{S}}_{\mathbf{K}^2}) \leqslant \frac{1}{16}\ln|\kappa|$

Following an argument similar to the proof of Theorem 2 in Abramovich and Grinshtein (2016), it will then follow from a variant of Fano's inequality, namely Lemma A.1 from Bunea et al. (2007), that

$$\inf_{\widehat{\mathbf{K}}}\sup_{\mathbf{K}\in\mathcal{K}'_{\lambda,\gamma}} \mathbb{E}[R(\widehat{\mathbf{K}})] - R(\mathbf{K}) \geqslant s_n^2.$$

By Lemma 8.3 of Rigollet and Tsybakov (2011), there exists a subset $\kappa \subset \mathcal{K}'_{\lambda,\gamma}$, and an absolute constant $0 < C_1 < 1$ such that

- $\ln|\kappa| \geqslant C_1 d \ln\frac{p}{d}$
- Each element of κ has sparsity d, is 0 away from the diagonal, and on the diagonal the elements are either 0 or γ, for a value of $\gamma \geqslant 0$ we will choose later.
- For all $\mathbf{K}^i, \mathbf{K}^j \in \kappa$, $\|\mathbf{K}^i - \mathbf{K}^j\|_0 \geqslant C_1 d$.

Therefore, for $\mathbf{K}^1, \mathbf{K}^2 \in \kappa$, we need only to show $\mathrm{KL}(\mathbf{K}^1 \| \mathbf{K}^2) \leqslant \frac{1}{16} \ln |\kappa|$. Using the fact that $\mathbf{X} = \mathbf{I}$,

$$\begin{aligned}\frac{2C_f^2}{|\mathcal{T}|}\|\mathcal{M}(\mathbf{K}^1) - \mathcal{M}(\mathbf{K}^2)\|_2^2 &\geqslant \frac{2C_f^2}{|\mathcal{T}|} p \sum_{j<k} ((\mathbf{K}^1_{kk} - \mathbf{K}^2_{kk}) - (\mathbf{K}^1_{jj} - \mathbf{K}^2_{jj}))^2 \\ &\geqslant \frac{2C_f^2 C_1 p d(p-2d)\gamma^2}{|\mathcal{T}|}\end{aligned}$$

To see the second to last inequality, note that there are at least $C_1 d(p-2d)$ pairs of indices j, k where $\mathbf{K}^1_{kk} \neq \mathbf{K}^2_{kk}$ but $\mathbf{K}^1_{jj} = \mathbf{K}^2_{jj}$, because $\mathbf{K}^1$ and $\mathbf{K}^2$ share at least $p-2d$ entries on their diagonal that are both 0. Each such entry contributes a γ^2 to the sum.

In particular choose,

$$s_n^2 = \frac{C_f^2 C_1 p d(p-2d)\gamma^2}{2|\mathcal{T}|}.$$

We proceed by selecting γ such that $\mathrm{KL}(P^{\mathcal{S}}_{\mathbf{K}^1} \| P^{\mathcal{S}}_{\mathbf{K}^2}) \leqslant \frac{1}{16} \ln |\kappa|$. Assume our samples are $\mathcal{S} = \{(t, y_t)\}$. Then since the samples are i.i.d.

$$\mathrm{KL}(P^{\mathcal{S}}_{\mathbf{K}^1} \| P^{\mathcal{S}}_{\mathbf{K}^2}) = \sum_{t \in \mathcal{S}} \mathrm{KL}(P_{\mathbf{K}^1}(t) \| P_{\mathbf{K}^2}(t))$$

where $P_{\mathbf{K}^i}(t)$ is the distribution of y_t conditioned on $\mathbf{K}^i$, in particular the probability of $y_t = -1$ is $f(\langle \mathbf{M}_t, \mathbf{K}^i \rangle)$.

We can bound each term of the sum above using the upper bound from Lemma 3.11.

$$\begin{aligned}\mathrm{KL}(P_{\mathbf{K}^1}(t) \| P_{\mathbf{K}^2}(t)) &\leqslant \frac{(f(\langle M_t, \mathbf{K}^1 \rangle) - f(\langle M_t, \mathbf{K}^2 \rangle))^2}{2 \inf_t f(\langle M_t, \mathbf{K}^2 \rangle)(1 - f(\langle M_t, \mathbf{K}^2 \rangle))} \\ &\leqslant \frac{(\langle M_t, \mathbf{K}^1 - \mathbf{K}^2 \rangle)^2 \sup_{|\nu| \leqslant \gamma} f'(\nu)^2}{2 \inf_{|x| \leqslant \gamma} f(x)(1 - f(x))}\end{aligned}$$

$$\leqslant \frac{\gamma^2 \sup_{|\nu|\leqslant\gamma} f'(\nu)^2}{2 \inf_{|x|\leqslant\gamma} f(x)(1-f(x))}$$

Summing over $t \in \mathcal{S}$, we require that

$$KL(P^{\mathcal{S}}_{\mathbf{K}^1} \| P^{\mathcal{S}}_{\mathbf{K}^2}) \leqslant \frac{\gamma^2 |\mathcal{S}| \sup_{|\nu|\leqslant\gamma} f'(\nu)^2}{2 \inf_{|x|\leqslant\gamma} f(x)(1-f(x))} \leqslant \frac{C_1}{16} d \ln \frac{p}{d} \leqslant \frac{1}{16} \ln |\kappa|,$$

so in particular, we will take

$$\frac{\gamma^2 \sup_{|\nu|\leqslant\gamma} f'(\nu)^2}{2 \inf_{|x|\leqslant\gamma} f(x)(1-f(x))} = \frac{C_1}{16|\mathcal{S}|} d \ln \frac{p}{d}$$

From this point on, let's take $\lambda = p$, and $d = \frac{p}{4}$. Now we have a few additional constraints on γ,

- Since $\|\mathbf{K}^i\|_{1,2} \leqslant \lambda$ for each $\mathbf{K}^i \in \kappa$, we require $\gamma d \leqslant \lambda$, so in particular $\gamma \leqslant 4$.
- In addition, we are going to require $\gamma \geqslant 1$ since we will need $p\gamma \geqslant \lambda$ (used below).

Based on these conditions, we just take $\gamma = 2$ and after simplification choose,

$$|S| := \frac{C_1 p \ln 4 \inf_{|x|\leqslant\gamma} f(x)(1-f(x))}{32\gamma^2 \sup_{|\nu|\leqslant\gamma} f'(\nu)^2}$$

Now we are finally in a position to use our choice of γ, d, λ and $|\mathcal{S}|$. We see that

$$\begin{aligned} s_n^2 = \frac{C_f^2 C_1 p d(p-2d)\gamma^2}{2|\mathcal{T}|} &= \frac{C_f^2 C_1 p^2 \gamma \lambda}{16|\mathcal{T}|} && \text{(since } p\gamma \geqslant \lambda\text{)} \\ &\geqslant \frac{C_f^2 C_1 \gamma \lambda}{8p} \\ &\geqslant \frac{C_f^2 \sqrt{\inf_{|x|\leqslant\gamma} f(x)(1-f(x))}}{8p\sqrt{\sup_{|\nu|\leqslant 2} f'(\nu)^2}} \sqrt{\frac{C_1^3 \ln 4}{32} \frac{p}{|\mathcal{S}|}} \end{aligned}$$

$$= \frac{C_f^2}{32}\sqrt{\frac{\inf_{|x|\leqslant\gamma} f(x)(1-f(x))}{\sup_{|v|\leqslant\gamma} f'(v)^2}}\sqrt{\frac{C_1^3 \ln 4}{2}\frac{\lambda^2\frac{\|\mathbf{X}\mathbf{X}^T\|}{n}}{|\mathcal{S}|}}$$

where the final equality follows from the fact that we have chosen $\mathbf{X} = \mathbf{I}_n$ so $n = p$. □

3.A.3 Proof of Lemma 3.5

Proof of Lemma 3.5. As computed prior to the statement of Theorem 3.8.

$$R(\widehat{\mathbf{K}}) - R(\mathbf{K}^*) = \frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}} KL(f(\langle \mathbf{M}_t, \mathbf{K}^*\rangle)\|f(\langle \mathbf{M}_t, \widehat{\mathbf{K}}\rangle))$$

Now using Lemma 3.11 with $z = f(\langle \mathbf{M}_t, \mathbf{K}^*\rangle)$ and $y = f(\langle \mathbf{M}_t, \widehat{\mathbf{K}}\rangle)$ we see

$$KL(f(\langle \mathbf{M}_t, \mathbf{K}^*\rangle)\|f(\langle \mathbf{M}_t, \widehat{\mathbf{K}}\rangle)) \geqslant 2C_f^2(\langle \mathbf{M}_t, \mathbf{K}^*\rangle - \langle \mathbf{M}_t, \widehat{\mathbf{K}}\rangle)^2$$

Summing over all $t \in \mathcal{T}$

$$\begin{aligned} R(\widehat{\mathbf{K}}) - R(\mathbf{K}^\star) &\geqslant \frac{2C_f^2}{|\mathcal{T}|}\sum_{t\in T}(\langle \mathbf{M}_t, \mathbf{K}^*\rangle - \langle \mathbf{M}_t, \widehat{\mathbf{K}}\rangle)^2 \\ &= \frac{2C_f^2}{|\mathcal{T}|}\sum_{t\in T}(\langle \mathbf{M}_t, \widehat{\mathbf{K}} - \mathbf{K}^\star\rangle)^2 = \frac{2C_f^2}{|\mathcal{T}|}\|\mathcal{M}(\widehat{\mathbf{K}}) - \mathcal{M}(\mathbf{K}^*)\|_2^2. \end{aligned}$$

□

3.A.4 Proof of Theorem 3.8

Before launching into the proof of Theorem 3.8, we first prove an auxiliary set of results that depend on the classical correspondence between centered Gram matrices and Euclidean distance matrices. For a more in depth discussion of this correspondence, we refer interested readers to Dattorro (2011). Let $\mathbb{S}_h^n$ be the subspace of symmetric hollow matrices, i.e. symmetric matrices with zero diagonal,

and let $\mathbb{S}_c^n$ be the subspace of centered Gram matrices, i.e. positive semi-definite matrices with $\mathbf{1}_n$ in their kernel.

Note that $\dim \mathbb{S}_h^n = \dim \mathbb{S}_c^n = \binom{n}{2}$. In fact these spaces are isomorphic with an explicit linear isomorphism given by the maps

$$\mathbb{S}_h^n \to \mathbb{S}_c^n : \mathbf{D} \to -\frac{1}{2}\boldsymbol{V}\mathbf{D}\boldsymbol{V}$$

with inverse

$$\mathbb{S}_c^n \to \mathbb{S}_h^n : \mathbf{G} \to \operatorname{diag}(\mathbf{G})\mathbf{1}_n^\top - 2\mathbf{G} + \mathbf{1}_n \operatorname{diag}(\mathbf{G})^\top$$

where again, $\boldsymbol{V} = \mathbf{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top$.

Given a set of centered points $\mathbf{X} \in \mathbb{R}^p$, then under the isomorphism above, the associated Gram matrix $\mathbf{G} \in \mathbb{S}_c^n$ maps to the squared distance matrix $\mathbf{D} \in \mathbb{S}_h^n$. In particular, a matrix in $\mathbb{S}_h^n$ is a valid Euclidean distance matrix if and only if $-\frac{1}{2}\boldsymbol{V}\mathbf{D}\boldsymbol{V}$ is a centered Gram matrix.

Given a triplet $t = (i, j, k) \in \mathcal{T}$, we can define an operator $\Delta_t(\mathbf{D}) := \mathbf{D}_{ij} - \mathbf{D}_{ik}$ and

$$\Delta(\mathbf{D}) := (\Delta_t(\mathbf{D}) | \text{ for } t \in \mathcal{T})$$

analogous to $\mathcal{L}$ and $\mathcal{M}$. In particular, for associated $\mathbf{D}$ and $\mathbf{G}$, $\Delta_t(\mathbf{D}) = \mathcal{L}_t(\mathbf{G})$ for all t so $\Delta(\mathbf{D}) = \mathcal{L}(\mathbf{G})$. We can now prove the key lemmas used in the proof of 3.8.

Lemma 3.12. *The null space of $\mathcal{L}$ is one dimensional, spanned by $\boldsymbol{V}$.*

Proof. Lemma 2 in Jain et al. (2016a) shows $\ker \Delta$ is one dimensional and is spanned by $\mathbf{J} = \mathbf{1}_n\mathbf{1}_n^\top - \mathbf{I}_n$. A computation shows that $-\frac{1}{2}\boldsymbol{V}\mathbf{J}\boldsymbol{V} = \frac{1}{2}\boldsymbol{V}$. Since $\mathcal{L}(\boldsymbol{V}) = \Delta(\mathbf{J}) = \mathbf{0}$, $\boldsymbol{V}$ spans $\ker \mathcal{L}$. □

We rely on an analogous statement for distance matrices given in Lemma 3 in Jain et al. (2016a).

Lemma 3.13. *Let $\mathbf{G} \in \mathbb{S}_c^n$ and $\mathbf{H}$ the component of $\mathbf{G}$ orthogonal $\boldsymbol{V}$ then $\|\mathcal{L}(\mathbf{H})\|^2 \geqslant n\|\mathbf{H}\|_F^2$.*

Proof. Again, let $\mathbf{D}$ be the symmetric hollow matrix corresponding to $\mathbf{G}$. We can take a decomposition of $\mathbf{D}$ into a component perpendicular to $\ker \Delta$

$$\mathbf{D} = \mathbf{C} + \sigma_{\mathbf{D}} \mathbf{J}.$$

Applying $-\frac{1}{2}\mathbf{V} \cdot \mathbf{V}$ to both sides we get,

$$\mathbf{G} = -\frac{1}{2}\mathbf{V}\mathbf{C}\mathbf{V} + \frac{\sigma_{\mathbf{D}}}{2}\mathbf{V}.$$

We claim that $\mathbf{H} = -\frac{1}{2}\mathbf{V}\mathbf{C}\mathbf{V}$ and $\sigma_{\mathbf{G}} = \sigma_{\mathbf{D}}/2$. It suffices to prove that $\mathbf{V}\mathbf{C}\mathbf{V}$ is perpendicular to $\mathbf{V}$. To see this note that $\langle \mathbf{V}\mathbf{C}\mathbf{V}, \mathbf{V} \rangle = \langle \mathbf{C}, \mathbf{V} \rangle = 0$, since $\mathbf{C}$ is hollow and perpendicular to $\mathbf{J}$.

We now apply Lemma 3 in Jain et al. (2016a) which shows that the minimal eigenvalue of Δ is n.

$$\begin{aligned}
\|\mathcal{L}(\mathbf{H})\|^2 &= \|\Delta(\mathbf{C})\|^2 \\
&\geqslant n\|\mathbf{C}\|_F^2 \qquad (\text{ since C is perpendicular to the kernel of } \Delta) \\
&\geqslant n\left\|-\frac{1}{2}\mathbf{V}\mathbf{C}\mathbf{V}\right\|_F^2 \qquad (\text{Since } \mathbf{V} \text{ is a projection.}) \\
&\geqslant n\|\mathbf{H}\|_F^2
\end{aligned}$$

□

Proof of Theorem 3.8. We begin by applying Lemma 3.7 in the specific case where $\mathbf{G}^* = \mathbf{X}^T\mathbf{K}^*\mathbf{X}$ and $\widehat{\mathbf{G}} = \mathbf{X}^T\widehat{\mathbf{K}}\mathbf{X}$ with $\mathbf{H}^*$ and $\widehat{\mathbf{H}}$ defined analogously to above. Firstly, by definition

$$\widehat{\mathbf{G}} - \mathbf{G}^* = \widehat{\mathbf{H}} - \mathbf{H}^* + (\sigma_{\widehat{\mathbf{G}}} - \sigma_{\mathbf{G}^*})\mathbf{V}$$

By orthogonality

$$\begin{aligned}
\|\widehat{\mathbf{G}} - \mathbf{G}^*\|_F^2 &= \|\widehat{\mathbf{H}} - \mathbf{H}^*\|_F^2 + (\sigma_{\widehat{\mathbf{G}}} - \sigma_{\mathbf{G}^*})^2\|\mathbf{V}\|_F^2 \\
&= \|\widehat{\mathbf{H}} - \mathbf{H}^*\|_F^2 + (n-1)(\sigma_{\widehat{\mathbf{G}}} - \sigma_{\mathbf{G}^*})^2 \qquad (\text{Since } \|\mathbf{V}\|_F^2 = n-1)
\end{aligned}$$

$$\leqslant \|\widehat{\mathbf{H}} - \mathbf{H}^*\|_F^2 + \frac{n-1}{(n-d-d'-1)}\|\widehat{\mathbf{H}} - \mathbf{H}^*\|_F^2$$

(By Lemma 3.7 $\sigma_{\widehat{\mathbf{G}}} - \sigma_{\mathbf{G}^*}$ is a repeated eigenvalue with multiplicity $n-d-d'-1$)

$$= C_{d,d'}\|\widehat{\mathbf{H}} - \mathbf{H}^*\|_F^2.$$

Now,

$$\begin{aligned}
\|\mathcal{M}(\widehat{\mathbf{K}}) - \mathcal{M}(\mathbf{K}^*)\|_2^2 &= \|\mathcal{L}(\mathbf{X}^T\mathbf{K}\mathbf{X}) - \mathcal{L}(\mathbf{X}^T\mathbf{K}^*\mathbf{X})\|^2 \\
&\geqslant n\|\widehat{\mathbf{H}} - \mathbf{H}^*\|_F^2 && \text{(Using Lemma 3.13)} \\
&\geqslant \|\widehat{\mathbf{G}} - \mathbf{G}^*\|_F^2 && \text{(From the above.)} \\
&= \frac{n}{C_{d,d'}}\|\mathbf{X}^T\widehat{\mathbf{K}}\mathbf{X} - \mathbf{X}^T\mathbf{K}^*\mathbf{X}\|_F^2 \\
&\geqslant \frac{n\sigma_{\min}(\mathbf{X}\mathbf{X}^T)^2}{C_{d,d'}}\|\widehat{\mathbf{K}} - \mathbf{K}^*\|_F^2
\end{aligned}$$

To see the last line, recall $\text{vec}(\mathbf{X}^T\mathbf{K}\mathbf{X}) = (\mathbf{X}^T \otimes \mathbf{X}^T)\text{vec}(\mathbf{K})$. Now, the minimal eigenvalue of $\mathbf{X}^T \otimes \mathbf{X}^T$ is $\sigma_{\min}(\mathbf{X}\mathbf{X}^T)$ which is nonzero since $\mathbf{X}$ is rank p.

So we see from Lemma 3.5, that

$$\frac{n\sigma_{\min}(\mathbf{X}\mathbf{X}^T)^2}{|\mathcal{T}|}\|\mathbf{K} - \widehat{\mathbf{K}}\|_F^2 \leqslant \frac{C_{d,d'}}{C_f^2}(R(\widehat{\mathbf{K}}) - R(\mathbf{K}^*))$$

The result now follows from Theorem 3.1. □

3.B Kernelized Metric Learning

Traditional Mahalanobis distance metric learning is equivalent to learning a linear mapping of the data such that Euclidean distance in the mapped space agrees with a set of labels, such as class labels or triplet comparisons. Often, we are interested in a richer set of mappings than linear ones. Indeed, this is the idea that underlies deep learning and kernel learning. In this section, we show how to extend Mahalanobis distance metric learning to the kernelized setting and then present a quick result

about kernelized triplet metric learning following extending the results of this chapter.

3.B.1 Warmup: Kernelized PCA

Here, we explain how to perform PCA in a reproducing kernel Hilbert space (RKHS). In doing so, we will arrive at a general representer theorem from Chatpatanasiri et al. (2010) which gives theoretical justification to extending Mahalanobis distance metric learning to the kernelized regime.

Setup: Consider n points in $\mathbb{R}^d$, $x_1, \cdots, x_n$, and assume we have a mapping ϕ from $\mathbb{R}^d$ to a D dimensional reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ for $D \in \mathbb{N} \cup \{\infty\}$. Further, assume that $\langle \phi(x_i), \phi(x_j) \rangle_{\mathcal{H}} = k(x_i, x_j)$ for a $k(\cdot, \cdot) : \mathcal{H} \times \mathcal{H} \to \mathbb{R}$. Lastly, will assume that the $\phi(x_i)$ are linearly independent.

Computing principal components in $\mathcal{H}$: Consider the subspace of $\mathcal{H}$ spanned by $\{\phi(x_1), \cdots, \phi(x_n)\}$. Let $\psi_1, \cdots \psi_n$ be the n principal components in this space. The term "kernelized PCA" is a slight misnomer as one does not compute the principal components $\psi_1, \cdots \psi_n$ themselves, but rather the projection of data onto them. This is important as the principal components live in the D dimensional space $\mathcal{H}$ and D may be intractably large or infinite. Nevertheless, one can use a kernel trick to efficiently compute these projections as follows:

1. Form the Grammian: $\mathbf{K} \in \mathbb{R}^{n \times n}$ such that $\mathbf{K}_{ij} = k(\phi(x_i), \phi(x_j))$.
2. Center the Grammian: $\tilde{\mathbf{K}} = \mathbf{K} - \frac{1}{n}\mathbf{1}_{n \times n}\mathbf{K} - \frac{1}{n}\mathbf{K}\mathbf{1}_{n \times n} + \frac{1}{n^2}\mathbf{1}_{n \times n}\mathbf{K}\mathbf{1}_{n \times n}$ where $\mathbf{1}_{n \times n}$ is the n by n matrix of all ones.
3. Compute all n eigenvectors of $\tilde{\mathbf{K}}$, $\alpha_1, \cdots, \alpha_n$ and form matrix $\mathbf{A} = [\alpha_1, \cdots, \alpha_n]$.
4. For any $x \in \mathbb{R}^d$ and any principal component ψ_j with eigenvector α_j, we have that $\langle \phi(x), \psi_j \rangle_{\mathcal{H}} = \sum_{i=1}^{n} \alpha_{i,j} k(x, x_i)$.

5. Therefore, for any $x \in \mathbb{R}^d$ we may represent $\phi(x)$ in terms of its projection onto $\psi_1, \cdots, \psi_n$ as

$$\varphi(x) = A^T \begin{bmatrix} k(x, x_1) \\ \vdots \\ k(x, x_n) \end{bmatrix} \tag{3.1}$$

It is important that we center the Grammian before computing eigenvectors as $x_1, \cdots, x_n$ being centered does not in general imply that $\phi(x_1), \cdots, \phi(x_n)$ are. It is also worth noting that the eigenvalues of the Grammian do not provide the same intuition in the kernelized case as they do in linear PCA. In the linear case, this implies that one direction captures more variation than another. In the kernelized case, for an unlucky choice of kernel, the variation in all directions may be the same, and finding a good kernel such that this does not occur may be challenging.

3.B.1.1 Representer theorems for Kernelized PCA

In this subsection, we quote two results from Chatpatanasiri et al. (2010) about optimization with respect to $\phi(x_1), \cdots, \phi(x_n) \in \mathcal{H}$ versus $\varphi(x_1), \cdots, \varphi(x_n) \in \mathbb{R}^n$.

Theorem 3.14 (Full-Rank Representer Theorem, Chatpatanasiri et al. (2010))**.** *Let $\{\tilde{\psi}_i\}_{i=1}^n$ be any set of points in $\mathcal{H}$ such that Span$(\{\tilde{\psi}_i\}_{i=1}^n)$ = Span$(\{\phi(x_i)\}_{i=1}^n)$ and let $\mathcal{H}'$ be a Hilbert space such that $\mathcal{H}$ and $\mathcal{H}'$ are separable. For any objective function f, the optimization*

$$\min_{L} f(\{\langle L\phi(x_i), L\phi(x_j)\rangle_{\mathcal{H}'}\}_{i,j\in[n]})$$

such that $L : \mathcal{H} \to \mathcal{H}'$ is a bounded linear map, has the same optimal value as

$$\min_{\tilde{L}\in\mathbb{R}^{n\times n}} f(\{\tilde{\varphi}(x_i)^T \tilde{L}^T \tilde{L} \tilde{\varphi}(x_j)\}_{i,j\in[n]})$$

where $\tilde{\varphi}(x) = [\langle \phi(x), \tilde{\psi}_1\rangle, \cdots, \langle \phi(x), \tilde{\psi}_n\rangle]^T \in \mathbb{R}^n$.

Theorem 3.15 (Low-Rank Representer Theorem, Chatpatanasiri et al. (2010))**.** *Let $\{\tilde{\psi}_i\}_{i=1}^n$ be any set of points in $\mathcal{H}$ such that Span$(\{\tilde{\psi}_i\}_{i=1}^n)$ = Span$(\{\phi(x_i)\}_{i=1}^n)$, and let*

$\tilde{\varphi}(x) = [\langle \phi(x), \tilde{\psi}_1 \rangle, \cdots, \langle \phi(x), \tilde{\psi}_n \rangle]^T \in \mathbb{R}^n$. *For any objective function* f*, the optimization*

$$\min_{L} f(\{\langle L\phi(x_i), L\phi(x_j) \rangle_{\mathcal{H}'}\}_{i,j \in [n]})$$

such that $L : \mathcal{H} \to \mathbb{R}^k$ *is a bounded linear map, has the same optimal value as*

$$\min_{\tilde{L} \in \mathbb{R}^{k \times n}} f(\{\tilde{\varphi}(x_i)^T \tilde{L}^T \tilde{L} \tilde{\varphi}(x_j)\}_{i,j \in [n]}).$$

Remark 3.16. *These theorems suggest that it is not actually important that the basis for* $\phi(x_1), \cdots, \phi(x_n)$ *is the principal component directions as the statement holds for arbitrary bases. Instead, the use of the principal components is beneficial as inner products with this basis can be easily computed.*

3.B.2 Using Kernelized PCA to Compute Kernelized Mahalanobis Distances

The above should give us hope that one can learn a kernelized-Mahalanobis distance as this family of distance functions can be written as a linear combination of weighted inner products as in Theorems 3.14 and 3.15. Mahalanobis distances in general may be written in terms of the square-root matrix of the semidefinite weighting matrix. Below we expand out squared Mahalanobis distance in $\mathcal{H}$ in terms of a bounded linear map L from $\mathcal{H}$ to $\mathbb{R}^D$.

$$\begin{aligned} d_L(\phi(x_i), \phi(x_j))^2 &= \|L\phi(x_i) - L\phi(x_j)\|^2 \\ &= \langle L\phi(x_i) - L\phi(x_j), L\phi(x_i) - L\phi(x_j) \rangle \\ &= \langle L\phi(x_i), L\phi(x_i) \rangle - 2\langle L\phi(x_i), L\phi(x_j) \rangle + \langle L\phi(x_j), L\phi(x_j) \rangle. \end{aligned}$$

Expand $L := UA^T\Phi^T$ for a linear map U from $\mathbb{R}^n$ to $\mathbb{R}^D$. Let A be as defined in kernelized PCA and $\Phi := [\phi(x_1), \cdots, \phi(x_n)]$, the matrix whose columns are $\phi(x_i)$. As the $\phi(x_i)$'s are linearly independent by assumption, Φ is full rank. Additionally, by definition of the kernel function $k(\cdot, \cdot)$, $\Phi^T\phi(x) = [k(x, x_1), \cdots, k(x, x_n)]^T$. A is

full rank by definition. Plugging this in,

$$\begin{aligned}
&\langle \mathbf{L}\phi(\mathbf{x}_i), \mathbf{L}\phi(\mathbf{x}_i)\rangle - 2\langle \mathbf{L}\phi(\mathbf{x}_i), \mathbf{L}\phi(\mathbf{x}_j)\rangle + \langle \mathbf{L}\phi(\mathbf{x}_j), \mathbf{L}\phi(\mathbf{x}_j)\rangle \\
&\qquad = \langle \mathbf{U}\mathbf{A}^T\Phi^T\phi(\mathbf{x}_i), \mathbf{U}\mathbf{A}^T\Phi^T\phi(\mathbf{x}_i)\rangle - 2\langle \mathbf{U}\mathbf{A}^T\Phi^T\phi(\mathbf{x}_i), \mathbf{U}\mathbf{A}^T\Phi^T\phi(\mathbf{x}_j)\rangle \\
&\qquad\qquad + \langle \mathbf{U}\mathbf{A}^T\Phi^T\phi(\mathbf{x}_j), \mathbf{U}\mathbf{A}^T\Phi^T\phi(\mathbf{x}_j)\rangle \\
&\qquad = \langle \mathbf{U}\varphi(\mathbf{x}_i), \mathbf{U}\varphi(\mathbf{x}_i)\rangle - 2\langle \mathbf{U}\varphi(\mathbf{x}_i), \mathbf{U}\varphi(\mathbf{x}_j)\rangle + \langle \mathbf{U}\varphi(\mathbf{x}_j), \mathbf{U}\varphi(\mathbf{x}_j)\rangle \\
&\qquad = \|\mathbf{U}\varphi(\mathbf{x}_i) - \mathbf{U}\varphi(\mathbf{x}_j)\|^2 \\
&\qquad = \|\varphi(\mathbf{x}_i) - \varphi(\mathbf{x}_j)\|_{\mathbf{M}}^2
\end{aligned}$$

for $\varphi(\mathbf{x})$ defined by kernelized PCA on $\phi(\mathbf{x}_1), \cdots, \phi(\mathbf{x}_n)$, and semidefnite $\mathbf{M} := \mathbf{U}^T\mathbf{U} \in \mathbb{R}^{n\times n}$. Therefore, we may use kernelized PCA to efficiently compute distances in $\mathbb{R}^n$ as opposed to in the ϕ space. Furthermore, theorems 3.14 and 3.15 guarantee we may do this with no loss in performance for any downstream application.

3.B.3 Learning low-dimensional kernelized metrics using Kernelized PCA

To demonstrate the power of this framework, we use the above result to extend Theorem 3.1 which establishes prediction error bounds for (linear) triplet metric learning. In this setting, we assume as above that we have n points $\mathbf{x}_1, \cdots, \mathbf{x}_n \in \mathbb{R}^d$. Additionally, we collect a set $\mathcal{S}$ of *triplets*. Each triple $t = (i, j, k)$ is sampled uniformly (with replacement) from the set of $n\binom{n-1}{2}$ unique triples and we observe a label $y_t = \pm 1$ which is a (possibly noisy) indication of the distance comparison

$$d_{\mathbf{L}}(\phi(\mathbf{x}_i), \phi(\mathbf{x}_j)) < d_{\mathbf{L}}(\phi(\mathbf{x}_i), \phi(\mathbf{x}_k))$$

for some unknown $\mathbf{L}$. Let $y_t = -1$ correspond to $d_{ij} < d_{ik}$ and $y_t = 1$ to the reverse. We wish to predict these $|\mathcal{S}|$ triplets as well as possible. Assume for any $\mathbf{L}'$ we have

an L-Lipschitz loss

$$\ell(y_t[d_{L'}(\phi(x_i), \phi(x_j))^2 - d_{L'}(\phi(x_i), \phi(x_k))^2])$$

which is a function of y_t times the difference of squared distances. For instance, $\ell(x) = \log(1 + \exp(-x))$ corresponds to the logistic loss. We seek a distance metric which minimizes our average loss over the data we have collected, S:

$$\widehat{R}_S(L') = \frac{1}{|S|} \sum_{t=(i,j,k), y_t \in S} \ell(y_t[d_{L'}(\phi(x_i), \phi(x_j))^2 - d_{L'}(\phi(x_i), \phi(x_k))^2]),$$

the *empirical risk* of a Mahalanobis metric on $\mathcal{H}$ defined by linear map L'. The empirical risk is an unbiased estimator of the *True Risk*:

$$R_S(L') = \mathbb{E}_{t, y_t} \left[\ell(y_t[d_{L'}(\phi(x_i), \phi(x_j))^2 - d_{L'}(\phi(x_i), \phi(x_k))^2]) \right],$$

the expected loss over the randomness in the selection of triplet t and any randomness in the label y_t.

Define $\widehat{L}$ as the bounded linear map that minimizes $\widehat{R}_S(L)$ (the empirical risk minimizer) and L^* as the bounded linear operator that minimizes $R_S(L)$ (the true risk minimizer). As $\widehat{L}$ is a linear operator on $\mathcal{H}$ which is D dimensional, it is intractable to compute in practice. However, the above framework for computing Mahalanobis distances via kernelized PCA guarantees that

$$d_{L'}(\phi(x_i), \phi(x_j))^2 - d_{L'}(\phi(x_i), \phi(x_k))^2]) = \|\varphi(x_i) - \varphi(x_j)\|_M^2 - \|\varphi(x_i) - \varphi(x_k)\|_M^2$$

for an appropriately chosen M. Therefore, define

$$\widehat{\widehat{R}}_S(M) = \frac{1}{|S|} \sum_{t=(i,j,k), y_t \in S} \ell(y_t[\|\varphi(x_i) - \varphi(x_j)\|_M^2 - \|\varphi(x_i) - \varphi(x_k)\|_M^2]).$$

The representer theorems guarantee that

$$\min_{L} \widehat{R}_S(L) = \min_{M} \widehat{\overline{R}}_S(M)$$

where the latter optimization is over $n \times n$ semidefinite matrices M. Let $\overline{M}$ denote a minimizer to the latter optimization. A similar statement can be made about the true risk as well. Let $\overline{R}(M)$ denote the true risk with respect to a matrix M and let M^* denote the minimizer of $\overline{R}(M)$.

Since we have now reduced the problem to linear metric learning over $\varphi(x_i), \cdots, \varphi(x_n)$, the results of Mason et al. (2017) apply. To apply the guarantees therein, we place two additional restrictions beyond semidefiniteness on the $\overline{M}$ that minimizes $\widehat{\overline{R}}_S(M)$. Firstly, we assume that

$$\left| \|\varphi(x_i) - \varphi(x_j)\|_{\overline{M}}^2 - \|\varphi(x_i) - \varphi(x_k)\|_{\overline{M}}^2 \right| \leqslant \gamma.$$

This is necesssary as a technical assumption for the proof. Secondly, we assume that $\|\overline{M}\|_* \leqslant \lambda$. This bound on the nuclear norm of $\overline{M}$ is a convex relaxation enforcing that M be low rank. This is equivalent to the assumption that the metric is captured by a low-dimensional combination of the $\phi(x_1), \cdots, \phi(x_n)$ in the high dimensional space. Namely, we lie on a low-dimensional subspace of the high-dimensional ϕ space.

Corollary 3.17 (Cor. to Theorem 3.1)**.** *Fix $\delta, \gamma, \lambda > 0$. Assume that $\max_{i \in [n]} \|\varphi(x_i)\| \leqslant 1$. Then with probability at least $1 - \delta$*

$$R(\widehat{L}) - R(L^*) = \overline{R}(\overline{M}) - \overline{R}(M^*) \leqslant 4L \leqslant \sqrt{\frac{140\lambda^2 \frac{\sigma_{max}}{n} \log(n)}{|S|}} + \frac{2\log(n)}{|S|} + L\sqrt{\frac{2\gamma^2 \log(2/\delta)}{|S|}}$$

where σ_{max} is the largest singular value of the matrix $[\varphi(x_1), \cdots, \varphi(x_n)]$.

Remark 3.18. *The constraint that $\max_{i \in [n]} \|\varphi(x_i)\| \leqslant 1$ is a mild one and can easily be relaxed by scaling the right hand side of the inequality.*

Remark 3.19. *Instead of a constraint that* $\max_{i\in[n]} \|\varphi(x_i)\| \leqslant 1$*, one could instead assume that* $\max_{i\in[n]} \|x_i\| \leqslant 1$ *and scale the RHS by* $\|\phi\|_{\mathcal{H}}$*, the operator norm of* ϕ *in* $\mathcal{H}$*.*

3.C Geometric Bounds for Large Margin Metric Learning from Labelled Data

3.C.1 Introduction

The previous results in this chapter all deal with learning metrics from triplet comparisons of the form "item i is closer to item j than it is to item k." A related setting to this is learning metrics from *labelled* data. This is common in many real-world problems that use metric learning to learn feature representations for tasks such as facial recognition Schroff et al. (2015). Indeed, many empirical works in metric learning such as Weinberger and Saul (2009) which proposed the `Large Margin Nearest Neighbors` algorithm assume access to labelled data. In general, we assume that we have a set of high dimensional feature vectors $x_1, \cdots, x_n \in \mathbb{R}^d$ with labels $y_1, \cdots, y_n$. Here, the goal is to learn a metric given by a semidefinite matrix M such that if $y_i = y_j \neq y_k$ for points x_i, x_j, and x_k, then $\|x_i - x_j\|_M < \|x_i - x_k\|_M$. This is spiritually similar to the setting of triplet metric learning considered previously in this chapter except now triplet labels are determined by the class labels themselves.

As the authors of Weinberger and Saul (2009) note, this is not always possible for any three randomly chosen x_i, x_j, and x_k and can lead to unstable learning. Instead, a better posed question is to learn a metric given by matrix M such that for any x_i with label y_i, and a margin parameter $\gamma > 0$, if $\|x_i - x_j\|_M \leqslant \gamma$, then $y_i = y_j$ for all x_j within γ of x_i according to the metric. We are especially interested in the case that this metric is low-dimensional. This corresponds to the case that there exists a low dimensional subspace such that any two points within γ of each other must share a label.

We consider a binary classification problem with two class-conditional distribu-

tions in $\mathbb{R}^d$. Assume we have a set of labeled points drawn from these distributions and that the distributions are separable (i.e., have disjoint supports). Furthermore, assume that the distributions are separable in an optimal low-dimensional subspace, but non-separable in the complementary orthogonal subspace. The minimum distance between the distributions is called the margin. We analyze the question of how many samples are necessary to learn a low-dimensional projection that preserves the margin, ideally by estimating the projection onto the optimal low-dimensional subspace.

3.C.2 Approaches for Dimensionality Reduction from Labelled Data

Linear Discriminant Analysis (LDA) is a common approach for dimensionality reduction from labelled data. Unfortunately, it makes somewhat strict modeling assumptions and is restricted to identifying subspaces that are 'number of classes' minus 1 dimensional Balakrishnama and Ganapathiraju (1998). In the case of binary classification for instance, this restricts to 1-dimensional subspaces. Additionally, LDA can be ill-suited for data that is not linearly separable. As Principle Component Analysis (PCA) is not applicable to labelled data, Bair et al. (2006) propose Supervised-PCA (SPCA). SPCA finds a matrix $\mathbf{U}$ that maximizes the dependence between y and $\mathbf{U}^T\mathbf{x}$ according to the Hilbert-Schmidt independence criterion. Finding $\mathbf{U}$ is done by a process very similar to PCA, and in the case that labels y are related to features $\mathbf{x}$ via a latent linear model, SPCA can recover the underlying linear model. Another method in this setting is Sufficient Dimensionality Reduction (SDR) which seeks to find a $\mathbf{U}$ that maximizes the mutual information between y and $\mathbf{U}^T\mathbf{x}$ Globerson and Tishby (2003). Finally, Sliced Inverse Regression is a classical approach for dimensionality reduction, but requires strong assumptions on the distribution of the $\mathbf{x}$'s (independent of the y values) Li (1991). None of these methods consider the effect of margin, however. McWhirter et al. (2018) study this same problem setting and propose the `SqeezeFit` algorithm which does account for the effect of margin. The algorithm achieves great empirical performance. Un-

fortunately, the theoretical results require an additional assumption that is unlikely to be satisfied by any real distribution: the practitioner is able to query multiple $\mathbf{x}$'s that have identical components projected onto the $\mathcal{U}$ subspace. Unfortunately, this is a probability 0 event if either $\mathbb{P}_{\mathcal{X}|\mathcal{Y}=0}$ or $\mathbb{P}_{\mathcal{X}|\mathcal{Y}=1}$ are continuous as they are in most real problems. Even for discrete probability distributions over a fixed set of $\mathbf{x}$'s, the probability that this condition holds is exponentially small.

In the case of binary classification that we focus on, this problem is strongly related to learning *linear k-juntas* as analyzed in Mossel et al. (2004). In this setting, one wishes to learn a boolean function on $\{0,1\}^d$ that maps to $\{0,1\}$. If the function depends on a subset of $k < d$ variables, $O\left(\binom{d}{k}\right)$ samples are needed. This is equivalent to the problem in Section 3.C.3 specialized to the case that $\mathbf{x}_1, \cdots, \mathbf{x}_n \subset \{0,1\}^d$ and $\mathbf{U}$ being a k-dimensional *axis-aligned* subspace of $\mathbb{R}^d$. In particular, this suggests that $O(d^k)$ samples are necessary for this problem, exponentially more than is needed for traditional triplet metric learning as shown in Theorem 3.4!

3.C.3 Problem Setup

Assume we are given a dataset of n labeled pairs $\{\mathbf{x}_i, y_i\}_{i=1}^n : \mathbf{x}_i \in \mathbb{R}^d =: \mathcal{X}\ y_i \in \{0,1\} =: \mathcal{Y}$ drawn from a joint distribution $\mathbb{P}_{\mathcal{X}\times\mathcal{Y}}$ on $\mathcal{X} \times \mathcal{Y}\ \forall\ i$. Let $\mathbb{P}_{\mathcal{X}|\mathcal{Y}=0}$ and $\mathbb{P}_{\mathcal{X}|\mathcal{Y}=1}$ denote the class conditional densities of the feature vectors $\mathcal{X}$ given the labels. We assume that there exists a known margin of at least $\gamma > 0$ between the supports of $\mathbb{P}_{\mathcal{X}|\mathcal{Y}=0}$ and $\mathbb{P}_{\mathcal{X}|\mathcal{Y}=1}$.

Definition 3.20 (Margin)**.** *The margin between two densities* $\mathbb{P}_1(\mathbf{x})$ *and* $\mathbb{P}_2(\mathbf{x})$ *on vectors* $\mathbf{x}$ *is*

$$\inf \|\mathbf{x} - \mathbf{x}'\|_2 \quad \textit{s.t. } \mathbf{x} \in \textit{Support}(\mathbb{P}_1) \quad \textit{and } \mathbf{x}' \in \textit{Support}(\mathbb{P}_2).$$

Further, we assume that there exists a k-dimensional subspace $\mathcal{U}$ of $\mathbb{R}^d$ with associated projection operator $P_{\mathcal{U}}$ such that:

1. Labels y are fully described by a projection of the features onto $\mathcal{U}$: $\mathbb{P}_{y_i|\mathbf{x}_i} = \mathbb{P}_{y_i|P_{\mathcal{U}}\mathbf{x}_i}$

2. Labels y are independent of the projection of features $\mathbf{x}$ onto $\mathcal{U}^\perp$, the orthogonal complement of $\mathcal{U}$.

3. The class conditional densities marginalized onto $\mathcal{U}$, $\mathbb{P}_{P_\mathcal{U} X|y=0}$ and $\mathbb{P}_{P_\mathcal{U} X|y=1}$, have margin $\gamma > 0$ between their supports.

4. Any subspace $\mathcal{U}' \subset \mathcal{U}$ of dimension strictly less than k achieves margin 0.

This corresponds to the hypothesis that the labels y are fully explained by the projection of feature vectors $\mathbf{x}$ onto a low-dimensional subspace and all other $d - k$ dimensions are noise independent of the label. The last assumption is to prevent pathological cases where a lower dimensional subspace than $\mathcal{U}$ satisfies the same conditions. The final assumption is a simple boundedness condition. We are interested in learning $\mathcal{U}$ as a means of pre-training for downstream machine learning tasks, such as non-parametric estimation or nearest neighbors classification which can suffer greatly from the curse of dimensionality.

Throughout we will make use of the concept of a *difference vector* between two oppositely labelled points. Given two samples, $(\mathbf{x}_1, y_1)$ and $(\mathbf{x}_2, y_2)$ such that $y_1 \neq y_2$, $\boldsymbol{z} := \mathbf{x}_1 - \mathbf{x}_2$ is a difference vector. Let

$$\mathcal{Z}(\{\mathbf{x}_i, y_i\}_{i=1}^n) := \{\mathbf{x}_i - \mathbf{x}_j : y_j \neq y_j\})$$

be the set of difference vectors. We can rewrite the margin condition on subspace $\mathcal{U}$ as $\|P_\mathcal{U} \boldsymbol{z}\| = \|\boldsymbol{z}\|_{P_\mathcal{U} P_\mathcal{U}^T} > \gamma$ for any difference vector $\boldsymbol{z}$ between a point in the supports of each class conditional density. We consider the same black box problem as in McWhirter et al. (2018).

$$\text{Minimize } \text{Rank}(P_\mathcal{V}) \quad \text{s.t. } \|\boldsymbol{z}\|_{P_\mathcal{V} P_\mathcal{V}^T} > \gamma \; \forall \, \boldsymbol{z} \in \mathcal{Z}(\{\mathbf{x}_i, y_i\}_{i=1}^n), \; P_\mathcal{V} = P_\mathcal{V}^T, \; P_\mathcal{V} = P_\mathcal{V}^2. \tag{3.2}$$

The final two conditions that $P_\mathcal{V} = P_\mathcal{V}^T$ and $P_\mathcal{V} = P_\mathcal{V}^2$ enforce that $P_\mathcal{V}$ is a projector and we say that $P_\mathcal{V}$ projects from $\mathbb{R}^d$ to a 2-dimensional subspace $\mathcal{V}$. Note that this optimization is always feasible as $P_\mathcal{U}$ satisfies all constraints and is optimal if one

had access to the class conditional densities. Throughout, we will make frequent use of the fact that for any $\mathbf{x} \in \mathbb{R}^d$, $\|\mathbf{U}\mathbf{x}\| = \|\mathbf{x}\|_{\mathbf{U}\mathbf{U}^\intercal}$. Hence, distance wish respect to a Mahalanobis distance and magnitude of a projected vector are equivalent concepts. We restrict to Mahalanobis metrics defined by projectors to avoid trivial solutions. In particular, given any set of data, dilating all distances sufficiently ensures that *every* pairwise distance is at least γ without need to estimate $\mathcal{U}$. The above problem is non-convex and developing efficient methods to optimize it is a separate avenue. Instead, we focus on the fundamental question of when can we guarantee that its solution is well-behaved, independent of being able to solve it. Precisely, we consider the following question:

Problem Statement: Given a dataset $\{\mathbf{x}_i, y_i\}_{i=1}^n$ with difference vectors $\mathcal{Z} = \{\mathbf{x}_i - \mathbf{x}_j : y_j \neq y_j\}) \overset{\text{i.i.d.}}{\sim} \mathbb{P}_{\mathcal{X}\times\mathcal{Y}}$. Let $P_{\mathcal{V}}$ be a solution to Problem (3.2). How many samples a sufficient to ensure that the metric $\|\cdot\|_{P_{\mathcal{V}}P_{\mathcal{V}}^\intercal}$ achieves nonzero margin?

3.C.4 Geometric Results for $k = 2$ relevant dimensions

Sample complexity and error bounds for large margin metric learning is in general an open problem. In this section, we present results in the special case that $\mathcal{U}$ is a 2-dimensional subspace (k=2) of an arbitrary d-dimensional feature space. We begin by stating a geometric condition on the difference vectors in set $\mathcal{Z}$ and prove that it is sufficient to ensure that a solution $P_{\mathcal{V}}$ to Problem (3.2) achieves margin $O(\gamma)$ (with respect to the supports of the class-conditional densities). Later, we will present a family of distributions that satisfies this condition with high probability if enough data has been collected.

Theorem 3.21. *Let $P_{\mathcal{V}}$ solve problem (3.2). Assume that for any $\mathbf{x}_1 \in$ Support$(\mathbb{P}_{\mathcal{X}|\mathcal{Y}=0})$ and $\mathbf{x}_2 \in$ Support$(\mathbb{P}_{\mathcal{X}|\mathcal{Y}=1})$, $\|P_{\mathcal{U}}(\mathbf{x}_1 - \mathbf{x}_2)\| \leqslant 1$. If there exists $\mathbf{z}_1$, $\mathbf{z}_2 \in \mathcal{Z}$ for any 2-dimensional subspace $\mathcal{W}$ of $\mathcal{U}^\perp$ such that*

$$\max\{\|P_{\mathcal{W}}\mathbf{z}_1\|, \|P_{\mathcal{W}}\mathbf{z}_2\|\} \leqslant \epsilon_1$$

and

$$\left|\left\langle \frac{P_{\mathcal{U}} z_1}{\|P_{\mathcal{U}} z_1\|}, \frac{P_{\mathcal{U}} z_2}{\|P_{\mathcal{U}} z_2\|} \right\rangle\right| \leqslant \epsilon_2,$$

then the margin between $\mathbb{P}_{P_{\mathcal{V}} X|y=0}$ *and* $\mathbb{P}_{P_{\mathcal{V}} X|y=1}$ *is at least*

$$\frac{\gamma}{2}\left(\sqrt{\frac{\gamma^2 - \epsilon_1^2}{1-\epsilon_1^2}} - \epsilon_2\right)$$

Remark 3.22. *This theorem provides sufficient conditions to ensure that the solution to problem (3.2) as a margin of at least* $\Omega(\gamma^2)$*. Hence,* $P_{\mathcal{V}}$ *perfectly separates the class conditional densities.*

Proof of Theorem 3.21. To prove this, we first require two lemmas. For brevity, we define $\mathbf{u}_i = P_{\mathcal{U}} z_i / \|P_{\mathcal{U}} z_i\|$ and $\mathbf{v}_i = P_{\mathcal{V}} z_i / \|P_{\mathcal{V}} z_i\|$ for $i = 1, 2$. The first ensures that if the data contains z_1 and z_2 such that $\mathbf{u}_1$ and $\mathbf{u}_2$ form a nearly orthogonal basis for $\mathcal{U}$ and the angle between $\mathbf{u}_i$ and $\mathbf{v}_i$ is small for $i = 1, 2$, then the margin between $\mathbb{P}_{P_{\mathcal{V}} X|y=0}$ and $\mathbb{P}_{P_{\mathcal{V}} X|y=1}$ is non-zero.

Lemma 3.23. *Let* $P_{\mathcal{U}}$ *and* $P_{\mathcal{V}}$ *be projectors onto* $\mathcal{U}$ *and* $\mathcal{V}$ *respectively where* $P_{\mathcal{V}}$ *solves problem (3.2). If* $|\mathbf{u}_1^T \mathbf{u}_2| \leqslant \epsilon_2$ *and* $\min\{|\mathbf{u}_1^T \mathbf{v}_1|, |\mathbf{u}_2^T \mathbf{v}_2|\} \geqslant \gamma'$ *for some* $\gamma' > 0$*, then the margin between the class conditional densities projected onto* $\mathcal{V}$*,* $\mathbb{P}_{P_{\mathcal{V}} X|y=0}$ *and* $\mathbb{P}_{P_{\mathcal{V}} X|y=1}$ *is at least* $\gamma \frac{\gamma' - \epsilon_2}{2}$

The above lemma relies on the assumption that $\min\{|\mathbf{u}_1^T \mathbf{v}_1|, |\mathbf{u}_2^T \mathbf{v}_2|\} \geqslant \gamma'$ for some $\gamma' > 0$. Next, we show that if $\max\{\|P_{\mathcal{W}} z_1\|, \|P_{\mathcal{W}} z_2\|\} \leqslant \epsilon_1$ for any 2-dimensional subspace $\mathcal{W}$ of $\mathcal{U}^{\perp}$, then this condition is satisfied.

Lemma 3.24. *Let* $P_{\mathcal{V}}$ *solve problem (3.2) and assume that* $\mathrm{Rank}(P_{\mathcal{V}}) = 2$*. Assume that for any* $\mathbf{x}_1 \in \mathrm{Support}(\mathbb{P}_{X|y=0})$ *and* $\mathbf{x}_2 \in \mathrm{Support}(\mathbb{P}_{X|y=1})$*,* $\|P_{\mathcal{U}}(\mathbf{x}_1 - \mathbf{x}_2)\| \leqslant 1$*. If there exists* $z_1, z_2 \in \mathcal{Z}$ *such that*

$$\max\{\|P_{\mathcal{W}} z_1\|, \|P_{\mathcal{W}} z_2\|\} \leqslant \epsilon_1$$

for any 2-dimensional subspace $\mathcal{W}$ of $\mathcal{U}^\perp$ then

$$\min_{i=1,2} |\mathbf{u}_i^T \mathbf{v}_i| \geqslant \sqrt{\frac{\gamma^2 - \epsilon_1^2}{1 - \epsilon_1^2}}.$$

Plugging in $\gamma' = \sqrt{\frac{\gamma^2-\epsilon_1^2}{1-\epsilon_1^2}}$ from Lemma 3.24 to Lemma 3.23 completes the proof of this Theorem.

□

Proof of Lemma 3.23. Choose any $\mathbf{x} \in \text{Support}(\mathbb{P}_{X|y=0})$ and $\mathbf{x}' \in \text{Support}(\mathbb{P}_{X|y=1})$. Let $\mathbf{z} = \mathbf{x} = \mathbf{x}'$.

$$\|P_{\mathcal{V}}\mathbf{z}\| \geqslant \|P_{\mathcal{V}}P_{\mathcal{U}}\mathbf{z}\| \geqslant \frac{1}{2}\left(|\mathbf{v}_1^T P_{\mathcal{U}}\mathbf{z}| + |\mathbf{v}_2^T P_{\mathcal{U}}\mathbf{z}|\right)$$

We may write $P_{\mathcal{U}}\mathbf{z} = a_1\mathbf{u}_1 + a_2\mathbf{u}_2$ for appropriately chosen a_1, a_2. Then,

$$\begin{aligned}
|\mathbf{v}_1^T P_{\mathcal{U}}\mathbf{z}| &= |\mathbf{v}_1^T(a_1\mathbf{u}_1 + a_2\mathbf{u}_2)| \\
&\geqslant |a_1||\mathbf{v}_1^T\mathbf{u}_1| - |a_2||\mathbf{v}_1^T\mathbf{u}_2| \\
&\geqslant |a_1|\gamma' - |a_2||\mathbf{u}_1^T\mathbf{u}_2| \\
&\geqslant |a_1|\gamma' - |a_2|\epsilon_2.
\end{aligned}$$

Similarly, $|\mathbf{v}_2^T P_{\mathcal{U}}\mathbf{z}| \geqslant |a_2|\gamma' - |a_1|\epsilon_2$. Putting these pieces together,

$$\begin{aligned}
\|P_{\mathcal{V}}\mathbf{z}\| &\geqslant \frac{|a_1| + |a_2|}{2}(\gamma' - \epsilon_2) \\
&= \frac{|a_1|\|\mathbf{u}_1\| + |a_2|\|\mathbf{u}_2\|}{2}(\gamma' - \epsilon_2) \\
&= \frac{\|a_1\mathbf{u}_1\| + \|a_2\mathbf{u}_2\|}{2}(\gamma' - \epsilon_2) \\
&\geqslant \frac{\|a_1\mathbf{u}_1 + a_2\mathbf{u}_2\|}{2}(\gamma' - \epsilon_2) \\
&= \frac{\gamma' - \epsilon}{2}\|P_{\mathcal{U}}\mathbf{z}\|
\end{aligned}$$

$$\geqslant \gamma \frac{\gamma' - \epsilon_2}{2}$$

where the final inequality follows from the assumption that the margin between $\mathbb{P}_{P_{\mathcal{U}}\mathbf{x}|y=0}$ and $\mathbb{P}_{P_{\mathcal{U}}\mathbf{x}|y=1}$ is at least γ. □

Proof of Lemma 3.24. First, note that $P_{\mathcal{V}}$ has rank at most 2 since $P_{\mathcal{U}}$ has rank 2 and satisfies all constraints of problem (3.2). Hence, $\mathcal{V}$ is supported on the Cartesian product of $\mathcal{U}$ and a noise subspace $\mathcal{W} \subset \mathcal{U}^{\perp}$. We may expand $P_{\mathcal{V}}z_i$ for $i = 1, 2$ as

$$P_{\mathcal{V}}z_i = \alpha_i P_{\mathcal{U}}z_i + \sqrt{1-\alpha_i^2}P_{\mathcal{W}}z_i.$$

Therefore,

$$\begin{aligned}\gamma \overset{(a)}{\leqslant} \|P_{\mathcal{V}}z_i\| &\leqslant \sqrt{\alpha_i^2\|P_{\mathcal{U}}z_i\|^2 + (1-\alpha_i^2)\|P_{\mathcal{W}}z_i\|^2} \\ &\overset{(b)}{\leqslant} \sqrt{\alpha_i^2 + (1-\alpha_i^2)\|P_{\mathcal{W}}z_i\|^2} \\ &\overset{(c)}{\leqslant} \sqrt{\alpha_i^2 + (1-\alpha_i^2)\epsilon_1^2}.\end{aligned}$$

Inequality (a) follows since $z_i \in \mathcal{Z}(\{\mathbf{x}_i, y_i\})$ and $P_{\mathcal{V}}$ is a solution to (3.2). Inequality (b) follows from the assumption that for any $\mathbf{x}_1 \in \text{Support}(\mathbb{P}_{\mathcal{X}|y=0})$ and $\mathbf{x}_2 \in \text{Support}(\mathbb{P}_{\mathcal{X}|y=1})$, $\|P_{\mathcal{U}}(\mathbf{x}_1 - \mathbf{x}_2)\| \leqslant 1$. Inequality (c) follows from the assumption that there exist $z_1, z_2 \in \mathcal{Z}$ such that $\max\{\|P_{\mathcal{W}}z_1\|, \|P_{\mathcal{W}}z_2\|\} \leqslant \epsilon_1$ for any noise subspace $\mathcal{W}$. Rearranging, we see that

$$\alpha_i \geqslant \sqrt{\frac{\gamma^2 - \epsilon_1^2}{1-\epsilon_1^2}}$$

Finally, note that α_i is the cosine of the angle between $P_{\mathcal{U}}z_i$ and $P_{\mathcal{W}}z_i$. Hence, $\alpha_i = \mathbf{u}_i^{\mathsf{T}}\mathbf{v}_i$, completing the proof. □

3.C.5 Geometric Results for $k > 2$ Relevant Dimensions

Assume that Dimension$(\mathcal{U}) = k$. For $k > 2$ dimensions, the conditions for learning $\mathcal{U}$ are unknown. Beyond learning $\mathcal{U}$, a simpler and still unanswered question is what conditions ensure that any solution $P_{\mathcal{V}}$ to problem (3.2) cannot be orthogonal to $P_{\mathcal{U}}$ (in the sense that $\mathcal{U}$ and $\mathcal{V}$ are not orthogonal subspaces). In this section, we provide a simple condition that ensures that a solution $P_{\mathcal{V}}$ cannot be orthogonal to $P_{\mathcal{U}}$ and bound the number of samples sufficient to ensure that this condition holds with high probability for any joint distribution $\mathbb{P}_{\mathcal{X}\times\mathcal{Y}}$ with γ margin.

Lemma 3.25. *Consider an* i.i.d. *sample* $\{\mathbf{x}_i, y_i\}_{i=1}^n$ *drawn from* $\mathbb{P}_{\mathcal{X}\times\mathcal{Y}}$ *with difference vectors* $\mathcal{Z}(\{\mathbf{x}_i, y_i\}_{i=1}^n)$. *Let* $P_{\mathcal{V}}$ *be any solution to problem* (3.2). *If*

$$\exists \mathbf{z}_i \in \mathcal{Z} : \|\mathbf{z}_i\|_{P_{\mathcal{W}} P_{\mathcal{W}}^T} < \gamma$$

for any $\mathcal{W} \subset \mathcal{U}^\perp$, *then* $P_{\mathcal{V}} P_{\mathcal{U}} \neq 0$. *This holds for any* $\mathbb{P}_{\mathcal{X}\times\mathcal{Y}}$ *with* γ *margin.*

Intuitively, this Lemma states that if in any noise subspace $\mathcal{W}$ there exists a $\mathbf{z}_i$ such that $\|\mathbf{z}_i\|_{P_{\mathcal{W}} P_{\mathcal{W}}^T} < \gamma$, then any $P_{\mathcal{W}}$ cannot solve problem (3.2).

Proof. Assume for contradiction that $P_{\mathcal{V}}$ solves problem (3.2) and that $\mathcal{V} \subset \mathcal{U}^\perp$. In this case, we have that $P_{\mathcal{V}} P_{\mathcal{U}} = 0$. By assumption, there exists a $\mathbf{z}_i \in \mathcal{Z}$ such that $\|\mathbf{z}_i\|_{P_{\mathcal{V}} P_{\mathcal{V}}^T} < \gamma$. However, this contradicts the assumption that $P_{\mathcal{V}}$ solves problem (3.2). In particular, this is true for any $\mathbb{P}_{\mathcal{X}\times\mathcal{Y}}$. □

Next, we upper bound the number of samples sufficient to ensure that the condition of Lemma 3.25 holds with high probability. For this, we require two additional assumptions on $P_{\mathcal{X}\times\mathcal{Y}}$ beyond those from Section 3.C.3.

1. Support$(P_{\mathcal{X}}) \subset \{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\| \leqslant 1\}$
2. $P_{\mathcal{Y}} = \text{Bernoulli}(1/2)$

Both assumptions could be changed to others depending on the problem setting. The first bounds the support of $P_{\mathcal{X}\times\mathcal{Y}}$ inside the unit ball in $\mathbb{R}^d$. Bounded support

is necessary for our proof technique, but the same technique is applicable to a different bounded region with only minor changes. The second condition is to ensure that we see examples with both the label $y = 0$ and $y = 1$. In particular, for a degenerate density where $\mathbb{P}(y = 1) = 1$, $\mathcal{Z} = \emptyset$ and any projector $P_{\mathcal{V}}$ solves problem (3.2). The probability of either label could be changed, and the technique we present would require only minimal changes. Intuitively, this assumption is equivalent to the assumption that the classes are balanced (as many points have label $y = 0$ as have $y = 1$) in expectation.

The technique proceeds via a covering argument. The probability that any two points share a label is $1/2$ by assumption. Hence, if one packs enough points, the probability that no two points within γ of each other shares a label becomes small. To make this precise, we employ a covering argument.

Definition 3.26 (ϵ-Covering number, Bartlett (2013))**.** *An ϵ-cover of a set T in metric space (M, d) is a set $\widehat{T} \subset T$ such that for any $t \in T$, there exists a $\hat{t} \in \widehat{T}$ such that $d(t, \hat{t}) \leqslant \epsilon$. The ϵ-covering number of T is the*

$$N(\epsilon, T, d) := \min\{|\widehat{T}| : \widehat{T} \text{ is an } \epsilon\text{-cover of } T\}$$

In the case that the metric $d(\cdot, \cdot)$ is the standard Euclidean metric, we denote $N(\epsilon, T, \|\cdot\|_2) = N(\epsilon, T)$.

Theorem 3.27. *Let $\{(x_i, y_i)\}_{i=1}^n \overset{\text{i.i.d.}}{\sim} P_{\mathcal{X}\times\mathcal{Y}}$ where $P_{\mathcal{X}\times\mathcal{Y}}$ satisfies both of the above assumptions. If*

$$n \geqslant \left(\frac{4}{\gamma} + 1\right)^2 + m,$$

then the probability that there exists $z_1 \in \mathcal{Z}(\{(x_i, y_i)\}_{i=1}^n)$ such that $\|P_{\mathcal{W}} z_1\|$ for every k dimensional subspace $\mathcal{W} \subset \mathcal{U}^\perp$ is at least

$$1 - \left(\frac{5}{\gamma}\right)^{2kd} 2^{-m/2}.$$

Corollary 3.28. *In the same setting as Theorem 3.27, if*

$$m \geqslant 3\log\left(\frac{1}{\delta}\right) + 6kd\log\left(\frac{5}{\gamma}\right)$$

then the probability that there exists $z_2 \in \mathcal{Z}(\{(x_i, y_i)\}_{i=1}^n)$ *such that* $\|P_{\mathcal{W}} z_1\| \leqslant \gamma$ *for every* k*-dimensional subspace* $\mathcal{W} \subset \mathcal{U}^\perp$ *is at least* $1 - \delta$*.*

Proof of Theorem 3.27. Let $E_{\mathcal{W}}(\epsilon)$ be the event

$$E_{\mathcal{W}}(\epsilon) = \mathbf{1}\{\nexists z_1 \in \mathcal{Z}(\{bx_i, y_i\}) : \|P_{\mathcal{W}} z_1\| \leqslant \epsilon\}.$$

We are interested in controlling

$$\mathbb{P}\left(\bigcup_{\mathcal{W}\in\mathcal{U}^\perp} E_{\mathcal{W}}(\epsilon)\right).$$

Note that we consider all possible k-dimensional noise subspaces $\boldsymbol{W} \in \mathcal{U}^\perp$. We begin by controlling $\mathbb{P}(E_{\mathcal{W}}(\epsilon))$ for a fixed $\mathcal{W}$ and then union bound. Let $\mathbb{P}_{P_{\mathcal{W}}\boldsymbol{x}|y=0}$ and $\mathbb{P}_{P_{\mathcal{W}}\boldsymbol{x}|y=1}$ denote the class conditional densities marginalized onto $\mathcal{W}$. For any two x_i, x_j with labels y_i, y_j, we say that they *collide* in $\mathcal{W}$ if $\|P_{\mathcal{W}}(x_i - x_j)\| < \epsilon/2$ but $y_i \neq y_j$. We seek to bound the probability that there is not a collisions in $\mathcal{W}$.

Let $B_1(\mathbb{R}^k)$ denote the Euclidean ball of radius 1 in $\mathbb{R}^k$. By the assumption on $P_{\mathcal{X}\times\mathcal{Y}}$, the supports of $\mathbb{P}_{P_{\mathcal{W}}\boldsymbol{x}|y=0}$ and $\mathbb{P}_{P_{\mathcal{W}}\boldsymbol{x}|y=1}$ are contained in $B_1(\mathbb{R}^k)$. If we collect at least

$$N(\epsilon_1/4, B_1(\mathbb{R}^k)) + m$$

samples from $\mathbb{P}_{\mathcal{X}\times\mathcal{Y}}$, there are $\lfloor (m+1)/2 \rfloor$ independent pairs of points within ϵ_1 of another deterministically.

To see this, note that if the m more points than $\epsilon/4$-covering number of $B_1(\mathbb{R}^k)$ have been sampled, then at least m must be within $\epsilon/4$ of one of the others by deterministically. In the worst case, all m points are within an $\epsilon_1/4$ radius ball of a *single* other point, forming $\binom{m+1}{2}$ pairs (and are within $\epsilon_1/2$ of each other). Note that points may appear in multiple pairs creating dependency between these

random variables. However, note that of the $\binom{m+1}{2}$ pairs, at least $\lfloor (m+1)/2 \rfloor$ are independent such that each individual point only appears in a single pair.

Let $\mathcal{S}$ be the set of pairs. We are interested in the probability that at least 1 pair collides. Since $\mathcal{W} \subset \mathcal{U}^\perp$, $\mathbb{P}_y$ is independent of $\mathbb{P}_{P_\mathcal{W} x}$, which denotes P_x marginalized onto $\mathcal{W}$. Furthermore, $\mathbb{P}_y = \text{Bernoulli}(1/2)$. Combining this with independence between y and $P_\mathcal{W} x$ for $(x, y) \sim \mathbb{P}_{\mathcal{X} \times \mathcal{Y}}$, the probability that a pair in $\mathcal{S}$ collides is $1/2$.

Therefore, the probability that there are *no collisions* in $\mathcal{S}$ is at most $2^{-|\mathcal{S}|} \leqslant 2^{-\lfloor (m+1)/2 \rfloor} \leqslant 2^{-m/2}$. Hence we have that

$$\mathbb{P}(E_\mathcal{W}(\epsilon/2)) \leqslant 2^{-m/2}.$$

The above analysis is with applies for collisions with respect to metric $\|\cdot\|_{P_\mathcal{W} P_\mathcal{W}^T}$ for a fixed $\mathcal{W}$. It remains to union bound over all $\mathcal{W} \subset \mathcal{U}^\perp$.

To do so, we again appeal to covering numbers, but in this case over k-dimensional projection matrices. Let $\mathcal{M}_k^d$ denote the set of Mahalanobis metrics on $\mathbb{R}^d$ such that for any $\mathbf{M} \in \mathcal{M}_k^d$, $\mathbf{M}$ can be written as $\mathbf{M} = \mathbf{U}\mathbf{U}^T$ where $\mathbf{U}$ is a projection from $\mathbb{R}^d$ to an k dimensional subspace:

$$\mathcal{M}_k^d = \{P_\mathcal{S} P_\mathcal{S}^T \in \mathbb{R}^{d \times d} : \text{Rank}(P_\mathcal{S} P_\mathcal{S}^T) = k,\ P_\mathcal{S} P_\mathcal{S}^T \succcurlyeq 0, (P_\mathcal{S} P_\mathcal{S}^T)^2 = P_\mathcal{S} P_\mathcal{S}^T\}.$$

In Lemma 3.29 we bound the distortion incurred by restricting to a covering set of metrics. In particular, let $\widehat{\mathcal{M}}$ be an $\epsilon^2/4$ cover of $\mathcal{M}_k^d$. By Lemma 3.29,

$$\bigcup_{\mathcal{W}: P_\mathcal{W} P_\mathcal{W}^T \in \widehat{\mathcal{M}}} E_\mathcal{W}(\epsilon/2) \implies \bigcup_{\mathcal{W} \in \mathcal{U}^\perp} E_\mathcal{W}(\epsilon).$$

Hence, we need only union bound over $\widehat{\mathcal{M}}$. We bound the covering number of $\mathcal{M}_k^d$ in the Lemma 3.30. Therefore

$$\begin{aligned}\mathbb{P}\left(\bigcup_{\mathcal{W}: P_\mathcal{W} P_\mathcal{W}^T \in \widehat{\mathcal{M}}} E_\mathcal{W}(\epsilon/2)\right) &\leqslant \sum_{P_\mathcal{W} P_\mathcal{W}^T \in \widehat{\mathcal{M}}} \mathbb{P}(E_\mathcal{W}(\epsilon/2)) \\ &\leqslant N(\epsilon, \mathcal{M}_k^d, \|\cdot\|) 2^{-m/2}\end{aligned}$$

$$\overset{\text{Lem3.30}}{\leqslant} \left(\frac{24}{\epsilon^2}\right)^{kd} 2^{-m/2}$$
$$\leqslant \left(\frac{5}{\epsilon}\right)^{2kd} 2^{-m/2}$$

The proof concludes by noting that the $\epsilon/2$-covering number of the unit ball in $\mathbb{R}^k$ is $\left(\frac{4}{\epsilon}+1\right)^k$ and plugging in $\epsilon = \gamma$ throughout. □

Lemma 3.29. *Let $\widehat{\mathcal{M}}$ be an ϵ-cover of $\mathcal{M}_r^d$ with respect to the spectral norm. For any $z \in \mathbb{R}^d$ with $\|z\| \leqslant 2$ and metric $\mathbf{M} \in \mathcal{M}_r^d$, there exists an $\overline{\mathbf{M}} \in \widehat{\mathcal{M}}$ such that*

$$|\|z\|_{\mathbf{M}} - \|z\|_{\overline{\mathbf{M}}}| \leqslant 2\sqrt{\epsilon}.$$

Lemma 3.30. *There exists a cover of $\mathcal{M}_r^d$ at a scale ϵ with respect to the spectral norm such that $N(\epsilon, \mathcal{M}_r^d, \|\cdot\|) \leqslant \left(\frac{6}{\epsilon}\right)^{dr}$*

Proof of Lemma 3.29.

$$\begin{aligned}
\|z\|_{\mathbf{M}} &= \sqrt{z^T \mathbf{M} z} \\
&= \sqrt{z^T \overline{\mathbf{M}} z + z^T \mathbf{M} z - z^T \overline{\mathbf{M}} z} && \text{(subadditivity)} \\
&\leqslant \sqrt{z^T \overline{\mathbf{M}} z} + \sqrt{|z^T \mathbf{M} z - z^T \overline{\mathbf{M}} z|} \\
&= \|z\|_{\overline{\mathbf{M}}} + \sqrt{|z^T (\mathbf{M} - \overline{\mathbf{M}}) z|} \\
&\leqslant \|z\|_{\overline{\mathbf{M}}} + \sqrt{\|zz^T\|_* \|\mathbf{M} - \overline{\mathbf{M}}\|} && \text{(Hölder's inequality)} \\
&\leqslant \|z\|_{\overline{\mathbf{M}}} + \sqrt{4\|\mathbf{M} - \overline{\mathbf{M}}\|} && (\|z\| \leqslant 2) \\
&= \|x - x'\|_{\overline{\mathbf{M}}} + \sqrt{4\|\mathbf{M} - \overline{\mathbf{M}}\|}
\end{aligned}$$

Reversing the roles of $\mathbf{M}$ and $\overline{\mathbf{M}}$ in the above computation, we get that

$$|\|z\|_{\mathbf{M}} - \|z\|_{\overline{\mathbf{M}}}| \leqslant 2\sqrt{\|\mathbf{M} - \overline{\mathbf{M}}\|} \leqslant 2\sqrt{\epsilon}$$

where the final inequality follows since $\widehat{\mathcal{M}}$ is an ϵ cover of $\mathcal{M}_r^d$. □

Proof of Lemma 3.30. This proof follows similarly to that of Candes and Plan (2011). We will construct an ϵ-cover of $\mathcal{M}_r^d$ with respect to the spectral norm, and will bound the cardinality of this set. By the spectral decomposition, for all $\mathbf{M} \in \mathcal{M}_r^d$, there exits matrices $\mathbf{U} \in \mathbb{R}^{d\times r}$ and $\mathbf{\Sigma} \in \mathbb{R}^{d\times d}$ such that $\mathbf{M} = \mathbf{U\Sigma U}^T$. Furthermore, the columns of $\mathbf{U}$ are orthonormal and $\mathbf{\Sigma}$ is diagonal with its first r diagonal entries equal to 1 and bounded by 1 and its remaining entries 0 since we have assumed $\mathbf{M} = \mathbf{M}^2$. We proceed by covering unitary matrices to form our cover.

We wish to cover matrices in $\mathbb{R}^{d\times r}$ whose columns are orthonormal, again with respect to the spectral norm. Let $\mathcal{U}_r^d$ be the set of such matrices. Define $\|\mathbf{X}\|_{2,\infty} := \max_i \|\mathbf{X}\|_2$, where $\mathbf{X}_i$ is the i^{th} column of $\mathbf{X}$. Define $\mathcal{Q}_r^d := \{\mathcal{X} \in \mathbb{R}^{d\times r} : \|\mathbf{X}\|_{2,\infty} \leqslant 1\}$. Then $\mathcal{U}_r^d \subset \mathcal{Q}_r^d$. For all matrices $\mathcal{X} \in \mathbb{R}^{m\times n}$, $\|\mathbf{X}\|_F \geqslant \|\mathbf{X}\|$. Then the Frobenius norm ball around $\mathbf{X}$ of radius $\epsilon > 0$ is a subset of the spectral norm ball of radius ϵ around $\mathbf{X}$. Therefore,

$$N(\epsilon, \mathcal{Q}_r^d, \|\cdot\|) \leqslant N(\epsilon, \mathcal{Q}_r^d, \|\cdot\|_F) \; \forall \epsilon > 0$$

Using the proof of Lemma 3.1 from Candes and Plan (2011), there exists an $\epsilon/2$ Frobenius covering of cardinality at most $\left(\frac{6}{\epsilon}\right)^{dr}$, which via duality of the spectral an Frobenius norms, is also a valid spectral norm $\epsilon/2$ covering of $\mathcal{Q}_r^d$. Denote this covering as $\overline{\mathcal{Q}_r^d}$.

Then we define the ϵ cover of $\mathcal{M}_r^d$ with respect to the spectral norm as

$$\overline{\mathcal{M}_r^d} := \{\overline{\mathbf{U}}\mathbf{\Sigma}\overline{\mathbf{U}}^T : \overline{\mathbf{U}} \in \overline{\mathcal{Q}_r^d}\}.$$

Then $|\overline{\mathcal{M}_r^d}| \leqslant |\overline{\mathcal{S}_r^d}| \leqslant \left(\frac{6}{\epsilon}\right)^{dr}$. It remains to show that this is a valid cover. Choose $\mathbf{M} \in \mathcal{M}_r^d$, and let $\overline{\mathbf{M}} = \arg\min_{\mathbf{X}\in\overline{\mathcal{M}_r^d}} \|\mathbf{M} - \mathbf{X}\|$

$$\begin{aligned}\|\mathbf{U\Sigma U}^T - \overline{\mathbf{U}}\mathbf{\Sigma}\overline{\mathbf{U}}^T\| &= \|\mathbf{U\Sigma U}^T - \overline{\mathbf{U}}\mathbf{\Sigma}\mathbf{U}^T + \overline{\mathbf{U}}\mathbf{\Sigma}\mathbf{U}^T - \overline{\mathbf{U}}\mathbf{\Sigma}\overline{\mathbf{U}}^T\| \\ &\leqslant \|\mathbf{U\Sigma U}^T - \overline{\mathbf{U}}\mathbf{\Sigma}\mathbf{U}^T\| + \|\overline{\mathbf{U}}\mathbf{\Sigma}\mathbf{U}^T - \overline{\mathbf{U}}\mathbf{\Sigma}\overline{\mathbf{U}}^T\|\end{aligned}$$

To bound the first term,

$$\begin{aligned}\|\mathbf{U}\mathbf{\Sigma}\mathbf{U}^\top - \overline{\mathbf{U}}\mathbf{\Sigma}\mathbf{U}^\top\| &= \|(\mathbf{U} - \overline{\mathbf{U}})\mathbf{\Sigma}\| \\ &\leqslant \|\mathbf{U} - \overline{\mathbf{U}}\|\|\mathbf{\Sigma}\| \\ &\leqslant \frac{\epsilon}{2}\end{aligned}$$

The second term can be bounded similarly. Therefore, each term is bounded by $\epsilon/2$, so $\|\mathbf{M} - \overline{\mathbf{M}}\| \leqslant \epsilon$. □

3.C.6 A Generative Family of Distributions for $k = 2$

In this section we provide a generative family of data distributions that satisfy the assumptions of Theorem 3.21. We seek to ensure that there exists a pair of $z_1, z_2 \in \mathcal{Z}(\{\mathbf{x}_i, \mathbf{x}_j\})$ for any 2-dimensional subspace $\mathcal{W}$ of $\mathcal{U}^\perp$ such that

1. $\max\{\|P_{\mathcal{W}} z_1\|, \|P_{\mathcal{W}} z_2\|\} \leqslant \epsilon_1$
2. $\left|\left\langle \frac{P_{\mathcal{U}} z_1}{\|P_{\mathcal{U}} z_1\|}, \frac{P_{\mathcal{U}} z_2}{\|P_{\mathcal{U}} z_2\|} \right\rangle\right| \leqslant \epsilon_2$.

In Theorem 3.32 we bound how many samples are sufficient to ensure the first condition. In Theorem 3.34 we bound the number of samples necessary for the second condition. As one condition is on $\mathcal{W} \subset \mathcal{U}^\perp$ and the other is on $\mathcal{U}$, the two are independent. Hence, to satisfy the conditions of Theorem 3.21, one need only collect enough samples to satisfy both bounds such that both events co-occur as shown in Corollary 3.35.

Fix an arbitrary 2-dimensional subspace $\mathcal{U}$ of $\mathbb{R}^d$ and consider the following class of joint probability distributions $\mathbb{P}_{\mathcal{X}\times\mathcal{Y}}$.

1. $\mathbb{P}_{\mathcal{X}\times\mathcal{Y}}$ satisfies all assumptions in section 3.C.3
2. Support$(\mathbb{P}_{\mathcal{X}}) \subset \{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\| \leqslant 1\}$
3. $\mathbb{P}_{P_{\mathcal{U}}\mathbf{x}|\mathcal{Y}=0}$ and $\mathbb{P}_{P_{\mathcal{U}}\mathbf{x}|\mathcal{Y}=1}$ are both isotropic in $\mathcal{U}$ as defined in Definition.
4. $\mathbb{P}_{\mathcal{Y}} = \text{Bernoulli}(1/2)$.

In particular, we show that if $\mathbb{P}_{P_{\mathcal{U}}x|y=0}$ and $\mathbb{P}_{P_{\mathcal{U}}x|y=1}$ are isotropic in $\mathcal{U}$ both classes are equally likely, then if sufficiently many samples are drawn, then the conditions are satisfied with high probability.

Definition 3.31. *[Isotropic distribution in a subspace, adapted from Definition 1 of Eaton et al. (1981)] A distribution f_x on $\mathbb{R}^d$ is isotropic on a subspace $\mathcal{U}$ if $f_{\Gamma P_{\mathcal{U}}x} = f_{P_{\mathcal{U}}x}$ where $f_{P_{\mathcal{U}}x}$ is the marginal of f_x on $\mathcal{U}$ and Γ is any rotation in the $\mathcal{U}$ subspace.*

3.C.6.1 Ensuring the first condition

In this section, we analyze how many samples are needed to ensure that there exists $z_1, z_2 \in \mathcal{Z}$ such that $\max\{\|P_{\mathcal{W}}z_1\|, \|P_{\mathcal{W}}z_2\|\} \leqslant \epsilon_1$ for any 2-dimensional subspace $\mathcal{W}$ of $\mathcal{U}^\perp$. Similar to the proof of Theorem 3.27, this proceeds via a covering argument.

Theorem 3.32. *Fix $\epsilon_1 \leqslant \gamma$. Let $\{(x_i, y_i)\}_{i=1}^n \overset{\text{i.i.d.}}{\sim} P_{\mathcal{X}\times\mathcal{Y}}$ where $P_{\mathcal{X}\times\mathcal{Y}}$ follows the 4 above assumptions. If*

$$n \geqslant \left(\frac{4}{\epsilon_1} + 1\right)^2 + m,$$

then the probability that there exists z_1 and $z_2 \in \mathcal{Z}(\{(x_i, y_i)\}_{i=1}^n)$ such that $\max\{\|P_{\mathcal{W}}z_1\|, \|P_{\mathcal{W}}z_2\|\} \leqslant \epsilon_1$ for every $\mathcal{W} \subset \mathcal{U}^\perp$ is at least

$$1 - (m+3)\left(\frac{5}{\epsilon_1}\right)^{4d} 2^{-(m/2+1)}.$$

Corollary 3.33. *In the same setting as Theorem 3.32, if*

$$m \geqslant 6\log\left(\frac{1}{\delta}\right) + 24d\log\left(\frac{5}{\epsilon_1}\right) + 34$$

then the probability that there exists z_1 and $z_2 \in \mathcal{Z}(\{(x_i, y_i)\}_{i=1}^n)$ such that $\max\{\|P_{\mathcal{W}}z_1\|, \|P_{\mathcal{W}}z_2\|\} \leqslant \epsilon_1$ for every $\mathcal{W} \subset \mathcal{U}^\perp$ is at least $1 - \delta$.

Proof of Corollary 3.33. We seek a sufficiency condition on m to imply that

$$(m+3)\left(\frac{5}{\epsilon_1}\right)^{4d} 2^{-(m/2+1)} \leqslant \delta$$

This is equivalent to the condition that

$$m \log(\sqrt{2}) - \log(m+3) \geqslant \log(1/\delta) + 4d \log(5/\epsilon_1) - \log(2).$$

By inspection, we see that $m \geqslant 2$ as the above is not satisfied for $m < 2$. For $m \geqslant 2$, $\log(m+3) \leqslant 3\log(m)$. Hence, the above is implied by

$$m \log(\sqrt{2}) - 3\log(m) \geqslant \log(1/\delta) + 4d \log(5/\epsilon_1) - \log(2).$$

Plugging in Proposition 4 of Antos et al. (2010), the above holds for

$$m \geqslant 6 \log\left(\frac{1}{\delta}\right) + 24d \log\left(\frac{5}{\epsilon_1}\right) + 34$$

completing the proof. □

Proof of Theorem 3.32. We proceed similarly to the proof of Theorem 3.27 except that we wish to guarantee that at least two z's exist. Let $E_W(\epsilon)$ be the event

$$E_W(\epsilon) = \mathbf{1}\{\nexists z_2, z_2 \in Z(\{bx_i, y_i\}) : \max\{\|P_W z_1\|, \|P_W z_2\|\} \leqslant \epsilon\}.$$

We are interested in controlling $\mathbb{P}\left(\bigcup_{W \in \mathcal{U}^\perp} E_W\right)$. We begin by controlling $P(E_W(\epsilon))$ for a fixed $\mathcal{W}$ and then union bound.

Let $B_1(\mathbb{R}^2)$ denote the Euclidean ball of radius 1 in $\mathbb{R}^2$. By the assumption on $\mathbb{P}_X$, the supports of $\mathbb{P}_{P_W x|y=0}$ and $\mathbb{P}_{P_W x|y=1}$ are contained in $B_1(\mathbb{R}^2)$. If we collect at least $N(\epsilon_1/4, B_1(\mathbb{R}^2)) + m$ samples from $\mathbb{P}_{X \times Y}$, there are $\lfloor (m+1)/2 \rfloor$ independent pairs of points within ϵ_1 of another deterministically. Since $\mathbb{P}_Y = \text{Bernoulli}(1/2)$ and label y is independent of $P_W x$ for $(x, y) \sim P_{X \times Y}$, the probability that a pair in S collides is $1/2$.

Therefore, the probability that there are *no collisions* in S is at most $2^{-\lfloor (m+1)/2 \rfloor} \leqslant 2^{-m/2}$. The probability there there is exactly 1 collision is at most

$$\binom{\lfloor (m+1)/2 \rfloor}{1} 2^{-\lfloor (m+1)/2 \rfloor} = \lfloor (m+1)/2 \rfloor 2^{-\lfloor (m+1)/2 \rfloor} \leqslant (m+1) 2^{-(m/2+1)}.$$

Hence we have that

$$P(E_{\mathcal{W}}(\epsilon/2)) \leqslant 2^{-m/2} + (m+1)2^{-(m/2+1)} = (m+3)2^{-(m/2+1)}.$$

The above analysis is with applies for collisions with respect to metric $\|\cdot\|_{P_{\mathcal{W}}P_{\mathcal{W}}^T}$ for a fixed $\mathcal{W}$. It remains to union bound over all $\mathcal{W} \subset \mathcal{U}^\perp$. We wish to control the probability To do so, we again appeal to covering numbers, but in this case over 2-dimensional projection matrices. Let $\mathcal{M}_r^d$ denote the set of Mahalanobis metrics defined by projections as in the proof of Theorem 3.27. Let $\widehat{\mathcal{M}}$ be an $\epsilon_1^2/4$ cover of $\mathcal{M}_2^d$. By Lemma 3.29,

$$\bigcup_{\mathcal{W}:P_{\mathcal{W}}P_{\mathcal{W}}^T \in \widehat{\mathcal{M}}} E_{\mathcal{W}}(\epsilon_1/2) \implies \bigcup_{\mathcal{W}\in\mathcal{U}^\perp} E_{\mathcal{W}}(\epsilon_1).$$

Hence, we need only union bound over $\widehat{\mathcal{M}}$. Therefore

$$\begin{aligned}
\mathbb{P}\left(\bigcup_{\mathcal{W}:P_{\mathcal{W}}P_{\mathcal{W}}^T \in \widehat{\mathcal{M}}} E_{\mathcal{W}}(\epsilon_1/2)\right) &\leqslant \sum_{P_{\mathcal{W}}P_{\mathcal{W}}^T \in \widehat{\mathcal{M}}} P(E_{\mathcal{W}}(\epsilon_1/2)) \\
&\leqslant (m+3)N(\epsilon, \mathcal{M}_2^d, \|\cdot\|)2^{-(m/2+1)} \\
&\overset{\text{Lem3.30}}{\leqslant} (m+3)\left(\frac{24}{\epsilon_1^2}\right)^{2d} 2^{-(m/2+1)} \\
&\leqslant (m+3)\left(\frac{5}{\epsilon_1}\right)^{4d} 2^{-(m/2+1)}
\end{aligned}$$

The proof concludes by noting that the ϵ-covering number of the unit ball in $\mathbb{R}^2$ is $\left(\frac{2}{\epsilon}+1\right)^2$ and plugging in $\epsilon = \epsilon_1/2$. □

3.C.6.2 Ensuring the second condition

In this section, we bound how large the set $\mathcal{Z}(\{x_i, y_i\})$ must be to ensure that the condition

$$\left|\left\langle \frac{P_{\mathcal{U}}z_1}{\|P_{\mathcal{U}}z_1\|}, \frac{P_{\mathcal{U}}z_2}{\|P_{\mathcal{U}}z_2\|}\right\rangle\right| \leqslant \epsilon_2$$

must hold with high probability for a $z_1, z_2 \in \mathcal{Z}$.

Theorem 3.34. *Fix $\epsilon_2 > 0$. Let $\{(x_i, y_i)\}_{i=1}^n \overset{\text{i.i.d.}}{\sim} P_{\mathcal{X}\times\mathcal{Y}}$ where $P_{\mathcal{X}\times\mathcal{Y}}$ follows the 4 above assumptions.*

$$\mathbb{P}\left(\nexists z_1, z_2 \in \mathcal{Z} : \left|\left\langle \frac{P_{\mathcal{U}} z_1}{\|P_{\mathcal{U}} z_1\|}, \frac{P_{\mathcal{U}} z_2}{\|P_{\mathcal{U}} z_2\|} \right\rangle\right| \leqslant \epsilon_2\right) \leqslant (1 - p_{\epsilon_2})^{\min(n_0, n_1)-1}$$

where $n_0 := |\{x_i : y_i = 0\}|$, $n_1 := |\{x_i : y_i = 1\}|$, and

$$p_{\epsilon_2} := 2\left(\frac{\arccos(-\epsilon_2) - \arccos(\epsilon_2)}{\pi}\right).$$

Corollary 3.35. *Fix $\epsilon_1 < \gamma$ and $\epsilon_2 > 0$. Let $\{(x_i, y_i)\}_{i=1}^n \overset{\text{i.i.d.}}{\sim} P_{\mathcal{X}\times\mathcal{Y}}$ where $P_{\mathcal{X}\times\mathcal{Y}}$ follows the 4 above assumptions. If*

$$\min(n_0, n_1) \geqslant 1 + \frac{\log(2/\delta)}{\log(1/(1 - p_{\epsilon_2}))}$$

and

$$n_0 + n_1 \geqslant \left(\frac{4}{\epsilon_1} + 1\right)^2 + 6\log\left(\frac{2}{\delta}\right) + 24d\log\left(\frac{5}{\epsilon_1}\right) + 34$$

then with probability at least $1 - \delta$, there exists a pair of $z_1, z_2 \in \mathcal{Z}(\{x_i, x_j\})$ for any 2-dimensional subspace $\mathcal{W}$ of $\mathcal{U}^\perp$ such that $\max\{\|P_{\mathcal{W}} z_1\|, \|P_{\mathcal{W}} z_2\|\} \leqslant \epsilon_1$ and $\left|\left\langle \frac{P_{\mathcal{U}} z_1}{\|P_{\mathcal{U}} z_1\|}, \frac{P_{\mathcal{U}} z_2}{\|P_{\mathcal{U}} z_2\|} \right\rangle\right| \leqslant \epsilon_2$.

Proof of Theorem 3.34. Let $\mathcal{Z}'(\{x_i, y_i\}) \subset \mathcal{Z}(\{x_i, y_i\})$ be the subset of the set of difference vectors such that each x_i only appears in a single difference vector z. This is done so all difference vectors are independent. We bound how large $\mathcal{Z}'$ must be for the condition to occur. Since $\mathcal{Z}' \subset \mathcal{Z}$, this implies that $\mathcal{Z}$ also satisfies the condition.

Let $x \sim \mathbb{P}_{\mathcal{U}x|y=0}$ and $x' \sim \mathbb{P}_{\mathcal{U}x|y=1}$. Let $z \overset{D}{=} x - x'$ where '$\overset{D}{=}$' denotes equality in distribution. Let $\mathbb{P}_z$ denote the associated density function. We begin by showing that $\mathbb{P}_z$ is also isotropic in $\mathcal{U}$. To see this, let $P_{\mathcal{U}}$ denote the projector onto $\mathcal{U}$, and consider any rotation Γ in the $\mathcal{U}$ subspace. Note that $\mathbb{P}_{P_{\mathcal{U}}x|y=0}$ and $\mathbb{P}_{P_{\mathcal{U}}x|y=1}$ being isotropic in $\mathcal{U}$ implies that $P_{\mathcal{U}}x \overset{D}{=} \Gamma P_{\mathcal{U}}x$ and $P_{\mathcal{U}}x' \overset{D}{=} \Gamma P_{\mathcal{U}}x'$ by Definition 3.31.

Therefore,

$$P_{\mathcal{U}}\boldsymbol{z} \overset{D}{=} P_{\mathcal{U}}(\boldsymbol{x}-\boldsymbol{x}') \overset{D}{=} P_{\mathcal{U}}\boldsymbol{x} - P_{\mathcal{U}}\boldsymbol{x}' \overset{D}{=} \Gamma P_{\mathcal{U}}\boldsymbol{x} - \Gamma P_{\mathcal{U}}\boldsymbol{x}' \overset{D}{=} \Gamma P_{\mathcal{U}}(\boldsymbol{x}-\boldsymbol{x}') \overset{D}{=} \Gamma P_{\mathcal{U}}\boldsymbol{Z}.$$

By Definition 3.31, this implies that $\mathbb{P}_z$ is isotropic. Therefore, fixing any $\boldsymbol{u} \in \mathcal{U}$ and $\boldsymbol{z} \sim \mathbb{P}_z$, the angle between $\boldsymbol{u}$ and $P_{\mathcal{U}}\boldsymbol{z}$ is a Uniform$[-\pi, \pi]$ random variable. Hence, if we fix an arbitrary direction $\boldsymbol{v} \in \mathbb{R}^d$.

$$\begin{aligned}
&\mathbb{P}_z\left(\left|\left\langle \frac{P_{\mathcal{U}}\boldsymbol{v}}{\|P_{\mathcal{U}}\boldsymbol{v}\|}, \frac{P_{\mathcal{U}}\boldsymbol{z}}{\|P_{\mathcal{U}}\boldsymbol{z}\|}\right\rangle\right| \leqslant \epsilon_2\right) \\
&\quad = \mathbb{P}_z\left(0 \leqslant \left\langle \frac{P_{\mathcal{U}}\boldsymbol{v}}{\|P_{\mathcal{U}}\boldsymbol{v}\|}, \frac{P_{\mathcal{U}}\boldsymbol{z}}{\|P_{\mathcal{U}}\boldsymbol{z}\|}\right\rangle \leqslant \epsilon_2 \text{ or } 0 \geqslant \left\langle \frac{P_{\mathcal{U}}\boldsymbol{v}}{\|P_{\mathcal{U}}\boldsymbol{v}\|}, \frac{P_{\mathcal{U}}\boldsymbol{z}}{\|P_{\mathcal{U}}\boldsymbol{z}\|}\right\rangle \geqslant -\epsilon_2\right) \\
&\quad = \mathbb{P}_z\left(0 \leqslant \theta_{\boldsymbol{v}\boldsymbol{z}} \leqslant \arccos(\epsilon_2) \text{ or } 0 \geqslant \theta_{\boldsymbol{v}\boldsymbol{z}} \geqslant \arccos(-\epsilon_2)\right) \\
&\quad = \mathbb{P}_z\left(\arccos(-\epsilon_2) \leqslant \theta_{\boldsymbol{v}\boldsymbol{z}} \leqslant \arccos(\epsilon_2)\right) \\
&\quad = 2\left(\frac{\arccos(-\epsilon_2) - \arccos(\epsilon_2)}{\pi}\right) =: p_{\epsilon_2}.
\end{aligned}$$

Therefore, for a set of independent directions $\mathcal{Z}'$,

$$\mathbb{P}\left(\exists \boldsymbol{z}_i \notin \mathcal{Z}' : \left|\left\langle \frac{P_{\mathcal{U}}\boldsymbol{v}}{\|P_{\mathcal{U}}\boldsymbol{v}\|}, \frac{P_{\mathcal{U}}\boldsymbol{z}_i}{\|P_{\mathcal{U}}\boldsymbol{z}_i\|}\right\rangle\right| \leqslant \epsilon_2\right) = (1 - p_{\epsilon_2})^{|\mathcal{Z}'|}.$$

Next we extend this analysis to random directions in $\mathcal{U}$ by integrating over all such directions and noting that the distribution of such directions is uniform by the isotropic assumption.

$$\begin{aligned}
&\mathbb{P}\left(\nexists \boldsymbol{z}_1, \boldsymbol{z}_2 \in \mathcal{Z}' : \left|\left\langle \frac{P_{\mathcal{U}}\boldsymbol{z}_1}{\|P_{\mathcal{U}}\boldsymbol{z}_1\|}, \frac{P_{\mathcal{U}}\boldsymbol{z}_2}{\|P_{\mathcal{U}}\boldsymbol{z}_2\|}\right\rangle\right| \leqslant \epsilon_2\right) \\
&\quad = \mathbb{E}\left[\mathbb{1}\left(\nexists \boldsymbol{z}_1, \boldsymbol{z}_2 \in \mathcal{Z}' : \left|\left\langle \frac{P_{\mathcal{U}}\boldsymbol{z}_1}{\|P_{\mathcal{U}}\boldsymbol{z}_1\|}, \frac{P_{\mathcal{U}}\boldsymbol{z}_2}{\|P_{\mathcal{U}}\boldsymbol{z}_2\|}\right\rangle\right| \leqslant \epsilon_2\right)\right] \\
&\quad = \mathbb{E}_{\boldsymbol{z}_1}\left[\mathbb{E}_{\boldsymbol{z}_2}\left[\mathbb{1}\left(\nexists \boldsymbol{z}_1, \boldsymbol{z}_2 \in \mathcal{Z}' : \left|\left\langle \frac{P_{\mathcal{U}}\boldsymbol{z}_1}{\|P_{\mathcal{U}}\boldsymbol{z}_1\|}, \frac{P_{\mathcal{U}}\boldsymbol{z}_2}{\|P_{\mathcal{U}}\boldsymbol{z}_2\|}\right\rangle\right| \leqslant \epsilon_2\right) \middle| \boldsymbol{z}_1\right]\right] \\
&\quad = \mathbb{E}_{\boldsymbol{z}_1}\left[(1 - p_{\epsilon_2})^{|\mathcal{Z}'|-1}\right]
\end{aligned}$$

$$= (1 - p_{\epsilon_2})^{|\mathcal{Z}'|-1}$$

where the exponent of $|\mathcal{Z}'| - 1$ is due to the fact that if one fixes a single z_i there are $|\mathcal{Z}'| - 1$ elements of $\mathcal{Z}'$ remaining. It remains to compute $|\mathcal{Z}'|$ in terms of $\mathcal{Z}$. Let $n_0 := |\{x_i : y_i = 0\}|$ and $n_1 := |\{x_i : y_i = 1\}|$. $|\mathcal{Z}| = n_0 n_1$ and $|\mathcal{Z}'| = \min(n_0, n_1)$. $\square$

4 APPLICATIONS OF METRIC LEARNING TO COGNITIVE SCIENCE AND EDUCATION

4.1 Introduction

Visual representations are ubiquitous in science, technology, engineering, and math (STEM) domains (Ainsworth, 2008; (US), 2006). For example, chemistry instruction on bonding typically uses the visuals shown in Figure 4.1. While we typically assume that such visuals help students learn because they make abstract concepts more accessible, they can also impede students' learning if students do not know how the visuals show information (Rau, 2017). To successfully learn with visuals, students need representational competencies: knowledge about how visual representations show information (Ainsworth, 2006; Gilbert, 2005). For example, a chemistry student needs to learn that dots in Lewis structures (e.g., Figure 4.1a) show electrons and white spheres in ball-and-stick models (e.g., Figure 4.1b) show hydrogen atoms.

Educational technologies can help students learn by adding instructional support for representational competencies to problem-solving activities. A particular

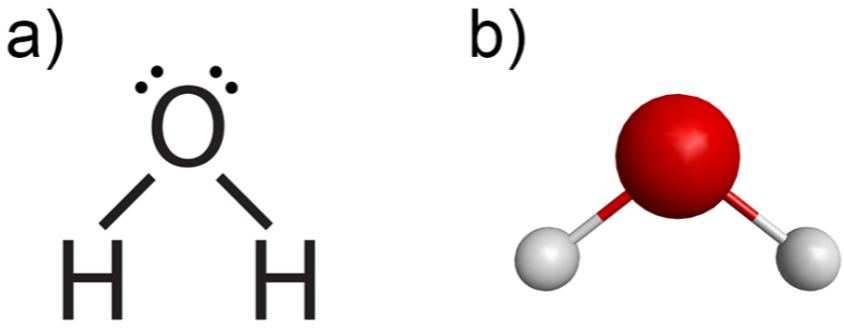


Figure 4.1: Commonly used visual representations of chemical molecules. A: Lewis structure of water. B: ball-and-stick model of water.

advantage of educational technologies is that they can adapt such supports to individual student's knowledge about the visuals (Koedinger et al., 2006; VanLehn, 2011). To do so, technologies use a cognitive model that infers whether the student has learned target knowledge based on their interactions with the visuals. Tailoring instruction to students' representational competencies can enhance their learning about the representations (Tuckey et al., 1991) and of domain knowledge (Davidowitz and Chittleborough, 2009).

However, current educational technologies have two critical limitations. First, most technologies support only *conceptual representational competencies*: the ability to map visual features to concepts and to use visuals to reason about concepts (Bodemer et al., 2004; Stieff, 2005; van der Meij and de Jong, 2011; Wu et al., 2001). For example, chemists can explain how the number of lines and dots shown in a 2D Lewis structure allow inferences about bonding. Such competencies are learned via explicit, verbally mediated processes that are best supported by prompting students to explain how visuals show concepts (Koedinger et al., 2012).

In addition, learning with visuals involves *perceptual representational competencies*: the ability to quickly and effortlessly see meaning in visuals (Gibson, 2000; Goldstone and Barsalou, 1998; Goldstone et al., 2010; Kellman et al., 2010). For example, chemists can immediately see that both visuals in Figure 4.1 show water, without having to make effortful inferences about this. Perceptual representational competencies are acquired via implicit induction processes that involve non-verbal pattern recognition (Koedinger et al., 2012). Perceptual representational competencies play an important role in students' learning because they free cognitive resources for higher-order complex reasoning, allowing students to use the visuals to learn new domain knowledge (Goldstone et al., 1997b; Rau, 2017).

A second limitation of current educational technologies is that–if they do incorporate supports for perceptual competencies–they do so without a cognitive model of these competencies but instead rely on performance measures (Kellman and Garrigan, 2009). That is, they treat each visual as an independent skill for students to learn (e.g., a Lewis structure of water, a Lewis structure of carbon dioxide), instead of considering the visual features students have to learn to perceive (e.g., dots,

lines, letters). As a result, existing educational technologies cannot trace students' acquisition of perceptual competencies or provide tailored feedback if students fail to attend to particular visual features. Given that decades of research show that cognitive models can increase the effectiveness and efficiency of educational technologies (Anderson et al., 1990; van der Meij and de Jong, 2011), we need to address this limitation and develop adequate measures for perceptual competencies to create adaptive supports for them.

We address this limitation with a method to assess students' perceptual competencies.

4.2 Prior Research

4.2.1 Learning with Visual Representations

Visuals are a specific type of external representation. External representations are objects that stand for something other than themselves: a *referent*, which can be a concrete object or an abstract concept (Peirce, 1931). Representations in instructional materials are defined as *external* representations because they are external to the viewer. By contrast, *internal* representations are mental objects that students can manipulate through imagination. Internal representations are building blocks of *mental models*, which constitute students' knowledge of a particular topic or domain. External representations can be *symbolic* or visual. For example, text or equations are symbolic external representations that have arbitrary (or convention-based) mappings to the referent (Schnotz, 2005). By contrast, *visual representations* are external representations that that consist of icons that have similarity-based mappings to the referent (e.g., diagrams, simulations, or physical models) (Schnotz, 2005).

Several theories describe how students learn with visuals. Mayer's (Mayer and Mayer, 2005) Cognitive Theory of Multimedia Learning (CTML) and (Schnotz, 2005; Schnotz and Bannert, 2003) Integrated Model of Text and Picture Comprehension (ITPC) draw on information processing theory (Baddeley, 1992, 2012; Chandler

and Sweller, 1991; Paivio, 1990) to describe learning from external representations as the integration of new information into a mental model. We focus on processes that pertain to visual representations.

First, students select relevant sensory information from the visual for further processing in working memory. To this end, students use perceptual processes that capture visuo-spatial patterns of the representation in working memory (Schnotz, 2005). This process is affected by conceptual competencies about relevant visual features, which enables top-down thematic selection of visual features (Goldstone et al., 1997a; Harel, 2016). Hence, the selection of relevant sensory information involves both perceptual and conceptual competencies, allowing students to select features based on learned perceptual cues that are linked to domain-relevant concepts.

Second, students *organize* this information into an *internal representation* that depicts the information. Because visuals have similarity-based mappings to referents, their structure can be directly mapped to internal representations that are also analogs of the referent (Gentner et al., 2003; Gentner and Markman, 1997; Schnotz, 2005). In forming the internal representation, students engage the perceptual processes of pattern recognition based on visual cues. They engage conceptual processes to map visual cues to concepts. The resulting internal representation is depictive in that its organization corresponds to the visuo-spatial organization of the external visual (Schnotz, 2005). Thus, the formation of an internal representation involves both perceptual and conceptual competencies, yielding a perceptual analog of the external visual representation that is linked to conceptual knowledge the domain.

Third, students integrate the information of the internal representations into a *mental model* of the domain knowledge (e.g., schemas). To this end, students integrate map the analog features of the internal representation to concepts retrieved from long-term memory. This third step constitutes learning: students learn content by integrating the internal representation into a mental model of domain (Hegarty and Just, 1993; Mayer and Mayer, 2005; Schnotz, 2005; Wylie and Chi, 2014). Thus, mental model formation involves perceptual and conceptual competencies.

In sum, students' learning with visuals involves both conceptual and perceptual competencies (Rau, 2017). While this brief review illustrates that conceptual and perceptual competencies are inter-related (Goldstone et al., 1997a; Harel, 2016), we discuss each of them separately to highlight that they are learned via different processes and hence require different assessments (Goldstone et al., 1997a; Kellman and Massey, 2013; Koedinger et al., 2012).

4.2.1.1 Conceptual Representational Competencies

Experts can map visuals to concepts, make inferences based on visual representations, and choose a particular visual for a task because it shows relevant concepts (Ainsworth, 2006; Rau, 2017; Schnotz, 2005). For example, a chemist can use Lewis structures and ball-and-stick models to show how the geometry and its lone electrons explain properties of water. According to the CTML and ITPC (Mayer and Mayer, 2005; Schnotz, 2005), conceptual competencies are involved when students select information by identifying meaningful features, when they form internal representations by mapping the features to concepts, and when they integrate internal representations with conceptual knowledge. According to cognitive learning theories, the acquisition of conceptual representational competencies involves learning about general principles of how a visual shows concepts (DeLoache, 2000; Eilam, 2012; Uttal and O'Doherty, 2008). Furthermore, because most STEM domains use multiple visuals, the acquisition of conceptual competencies involves understanding how one visual constrains the interpretation of a second visual and how they complement one another (Ainsworth, 2006, 2014).

The cognitive science literature (e.g., (Koedinger et al., 2012)) suggests that students learn conceptual competencies via sense-making processes. Sense-making processes are explicit processes in that students have to willfully engage in them (Chi et al., 1994; Sherin et al., 2000). They are verbally mediated because they involve explanations (Chi et al., 1989; Gentner et al., 2003; Koedinger et al., 2012). The importance of sense-making processes for learning with visuals is widely recognized. For example, (Ainsworth, 2006) and (Schnotz, 2005) describe sense-

making processes in terms of structure mapping that allows students to distinguish relevant and irrelevant features and to determine which information is (or is not) shown in different visuals (Gentner et al., 2003). Sense-making processes are also involved when students explain why a visual can help solve a given problem (Acevedo Nistal et al., 2013, 2014; Disessa, 2004)

4.2.1.2 Perceptual Representational Competencies

Further, experts see connections among visuals, translate among visuals, and integrate information across visuals with little time and cognitive effort (Dreyfus, 2004; Gibson, 1969, 2000; Richman et al., 1996). For example, chemists can see "at a glance" that a Lewis structure and ball-and-stick model both show water. Such perceptual expertise frees cognitive resources for higher-order reasoning (Goldstone and Barsalou, 1998; Richman et al., 1996) and is considered an important goal in STEM education (Airey and Linder, 2009; Kozma and Russell, 2005; Pape and Tchoshanov, 2001). According to the CTML and ITCP, perceptual competencies are characterized by high efficiency in forming accurate internal representations (Mayer and Mayer, 2005; Schnotz, 2005). Further, the ability to automatically combine information from different visuals without little mental effort (Chase and Simon, 1973; Kellman and Garrigan, 2009; Kellman and Massey, 2013) results from efficiency in mapping analog internal representations of visuals to one another (Mayer and Mayer, 2005; Schnotz, 2005).

The cognitive science literature (e.g., (Gibson, 2000; Goldstone et al., 1997b; Koedinger et al., 2012)) suggests that students acquire perceptual expertise via perceptual-induction processes. These processes are inductive because students can infer how visual features map to concepts through experience with many examples (Fahle et al., 2002; Gibson, 2000; Goldstone et al., 1997b; Kellman and Massey, 2013). Students gain efficiency in seeing meaning in visuals via perceptual chunking: rather than mapping specific features to concepts, they learn to treat each analog feature as one perceptual chunk that relates to multiple concepts. Perceptual-induction processes are considered to be non-verbal because they do not require

explicit reasoning (Koedinger et al., 2012; Richman et al., 1996; Fiore, 1997). They are implicit because they happen unintentionally and sometimes unconsciously (Frensch and Rünger, 2003; Shanks et al., 2005). Hence, they do not rely on students' deliberate direction of conscious attention.

4.2.1.3 Enhancing domain knowledge by supporting representational competencies

In sum, conceptual and perceptual representational competencies are learned via different processes but mutually enhance one another. Empirical evidence for this claim comes from studies showing that interventions that support sense-making and perceptual-induction processes have complementary effects on students' learning (Rau, 2017). Several studies show that sense-making support enhances students' learning of domain knowledge in engineering (van der Meij and de Jong, 2011), biology (Seufert, 2003), math (Rau et al., 2015a), physics (Gutwill et al., 1999), and chemistry (Chiu and Linn, 2012).

Less research has focused on perceptual-induction support. Kellman and colleagues developed interventions that expose students to many short tasks where they have to rapidly translate among representations. For example, a student may be asked to select one of several pie charts that shows the same fraction as a number line. Tasks are sequenced to expose students to systematic variation, often in the form of contrasting cases, so that irrelevant features vary but relevant features appear across several tasks (Kellman and Massey, 2013; Massey et al., 2013). Experiments show that these interventions enhance learning in math and chemistry (Kellman et al., 2008; Wise et al., 2000). Finally, experiments on math and chemistry learning show that combining perceptual-induction support and sense-making support yields higher learning gains on domain knowledge tests compared to sense-making support alone (Rau et al., 2015a; Rau, 2017, 2016).

4.2.2 Adaptive Educational Technologies for Representational Competencies

Much research on learning with visuals has been done in the context of educational technologies because they make it easy to integrate instructional supports for representational competencies into problem-solving activities. Many technologies include supports for representational competencies (Ainsworth et al., 2002; Linn et al., 2015; Linn and Slotta, 2000; van der Meij, 2007; van der Meij and de Jong, 2011). However, current technologies have two important limitations. First, they typically do not adapt such support to the individual student's level of representational competencies. This limitation results from the fact that these technologies do not contain a cognitive model of students' learning of representational competencies. One prominent exception is the External Representation Selection Tutor (ERST), (Cox and Brna, 2016; Grawemeyer, 2006). ERST incorporates a cognitive model of students' conceptual representational competencies, in particular, their knowledge about which visual to use for which type of problem. It provides adaptive feedback on students' choice of visual and has been shown to improve representational competencies and domain knowledge.

A second limitation is that only a few educational technologies support perceptual representational competencies. Kellman and colleagues spearheaded efforts to develop such technologies (Massey et al., 2013; Wise et al., 2000), which provide single-step translation problems that expose students to numerous visuals. However, these technologies do not take full advantage of adaptive capabilities that educational technologies can achieve. Specifically, Kellman and colleagues' technologies are adaptive in that they trace students' improvement in accuracy and speed. Until students achieve a mastery threshold, the same translation problems are repeated, hence treating each problem as an independent skill. Thus, existing technologies that support perceptual competencies lack a cognitive model that maps each translation problem to a latent skill that describes mappings between visual features that students can apply to a variety of visuals. Without such a cognitive model, the technology cannot adequately adapt to students' learning of

perceptual competencies. We propose that the main reason for this limitation lies in traditional methods for the assessment of representational competencies.

4.2.2.1 Assessments of representational competencies

Prior research has yielded different assessments of conceptual and perceptual representational competencies. To assess conceptual competencies, research has used tests that assess students' ability to explain how visuals show concepts (e.g., (Luxford, 2013; Rau et al., 2015b)). Further, research has used think-alouds to assess how students learn these competencies during problem solving (e.g., ainsworth2003effects, ploetzner2008successful). Also, research has used eye-tracking to assess students' visual attention to conceptually relevant features (for an overview, see (Alemdag and Cagiltay, 2018)). Eye-tracking research is based on the so-called eye-mind assumption (Hegarty and Just, 1993; Underwood and Everatt, 1992), which states that the duration of eye-gaze fixations reflects the duration of cognitive processes that students use on the information they are looking at. Consequently, most eye-tracking research on multimedia learning has focused on measures that reflect intentional direction of visual attention and assumes that these measures reflect conceptual processes. The most prominent measures include fixation duration and switching between stimuli (for recent meta-reviews, see (Alemdag and Cagiltay, 2018; Gegenfurtner et al., 2011; Lai et al., 2013)). For example, long fixations on a stimulus are assumed to reflect deep processing of the information (e.g., (Lai et al., 2013; Mason et al., 2013; Schmidt-Weigand et al., 2010)). Further, frequent switching between stimuli is assumed to reflect integration of information across the stimuli (e.g., (Alemdag and Cagiltay, 2018; Johnson and Mayer, 2012; Stalbovs et al., 2015)). Indeed, these measures correlate with students' conceptual competencies (Rau et al., 2015a; Van Gog and Scheiter, 2010). In sum, these measures rely on verbalization (e.g., think-alouds) or behaviors that can be mapped to verbal processes (e.g., eye-tracking).

To assess perceptual competencies, research has relied on measures of accuracy and efficiency in recognizing, classifying, or categorizing visuals (Hill and Sharma,

2015; Kellman et al., 2008; Kellman and Garrigan, 2009; Rau, 2016)). However, these measures cannot determine which features drive students' gains in accuracy and efficiency. Further, research has used gestures as a measure of how students learn perceptual competencies. Specifically, gestures can reveal which visual features they attend to while interacting with visuals (e.g., (Airey and Linder, 2009; Bieda and Nathan, 2009)). Also, teachers' use of gestures reveal which features they aim to draw students' attention to (e.g., (Alibali et al., 2014; Cope et al., 2015)). However, as gestures typically accompany speech, these measures are related to students' and teachers' explicit use of visual features in conceptual reasoning. Hence, they cannot reliably distinguish perceptual from conceptual competencies. Finally, research has used eye-tracking measures of students' efficiency in processing visuals (e.g., (Goldstone et al., 2010; Jarodzka et al., 2012)). Here, decreased fixation durations indicate increased efficiency, which contradicts the use of increased fixation durations as an indicator of increased conceptual competencies. Further, eye-tracking measures have been criticized for the assumption that cognitive processing is exclusively related to foveal location (Irwin, 2004). It is possible that peripheral vision plays a role in perceptual competencies. In sum, these measures cannot capture the implicit impact of visual features because they involve verbalization (e.g., gestures), do not distinguish visual features (e.g., accuracy and efficiency), or assume explicit attention (e.g., eye tracking).

4.2.2.2 Methods for cognitive model development

The lack of adequate methods to assess perceptual competencies is paralleled by a neglect of perceptual-induction processes in the development of cognitive models for educational technologies. To develop cognitive models, researchers typically analyze how experts and students solve tasks (Koedinger et al., 2006; Rau, 2016; Rau et al., 2015a). Such cognitive task analyses involve asking experts and students to "think aloud"; that is, to verbalize their thought processes (Clark, Richard E and Feldon, David E, and Van Merrienboer, Jeroen JG and Yates, Kenneth and Early, Sean, 2007; Schraagen et al., 2000). The main idea underlying this method

is that think-alouds provide a readout of working memory (Ericsson and Simon, 1984). Think-aloud protocols can then be analyzed for conceptual and procedural knowledge used to solve a problem (Clark, Richard E and Feldon, David E, and Van Merrienboer, Jeroen JG and Yates, Kenneth and Early, Sean, 2007; Schraagen et al., 2000). A cognitive model then captures conceptual and procedural knowledge prevalent among experts and traces students' progress towards expert thinking (Rau, 2016).

Think-aloud methods have been augmented with eye tracking, for example in cued retrospective reports (Conati et al., 2005; De Koning et al., 2010; Van Gog et al., 2005). Because talking can interfere with eye tracking, students' eye gaze is recorded while they solve problems quietly. Then, they view their gaze recordings and think aloud retrospectively. While this method allows gathering some information about processing efficiency, it emphasizes verbal knowledge. Further, the issues mentioned above of assessing perceptual competencies with eye tracking persist.

In sum, think-aloud methods are suitable for conceptual and procedural knowledge that can be verbalized but they likely cannot capture implicit and nonverbal knowledge.

4.2.3 Similarity Learning Methods

To identify an alternative method to assess implicit, nonverbal perceptual competencies, we draw on techniques for metric learning from triplet similarity judgments. Figure 4.2 shows an example of a triplet judgment with Lewis structure representations of chemical molecules. Given the top image (the "target molecule"), participants are asked to click on one of the bottom two images (the "choice molecules") that they perceive as most similar to the top image. Participants receive many such triplet similarity judgments in a row.

This approach draws on findings that people are better at providing ordinal (i.e., comparative) responses than at providing fine-grained quantitative judgments or ratings (Kruskal, 1964a). We assume that participants' perceived similarity among visuals is a function of the visual features present in each image. Hence,

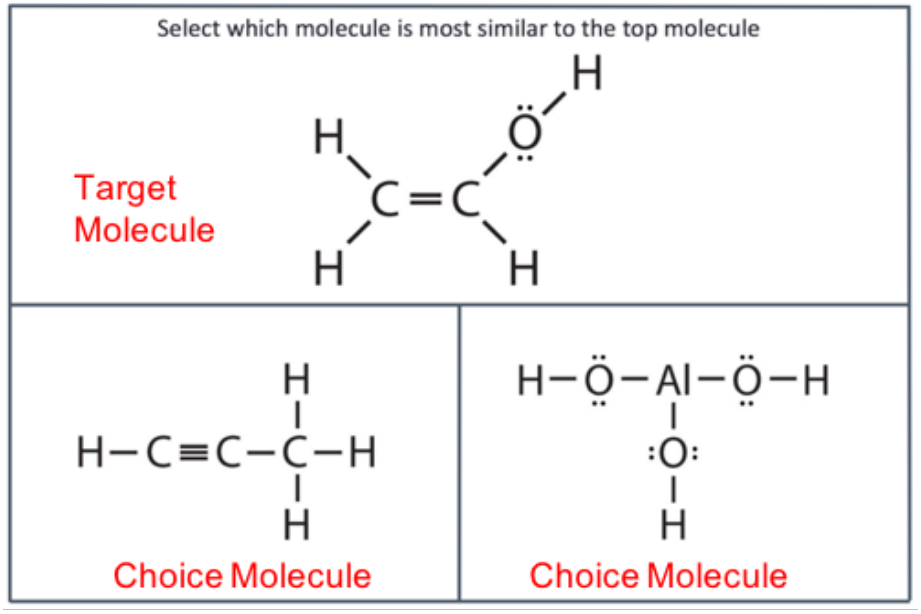


Figure 4.2: Example triplet judgment task with Lewis structures, as used in Experiment 1. Participants are given a target molecule (top) and asked to click on one of the two choice molecules (bottom) that are most similar to the target molecule.

if we know which features account for students' perception of similarity, we can predict participants' similarity judgments. In the example shown in Figure 4.2, if participants' similarity judgments are strongly affected by the presence of the letter C, then the molecule on the left is more likely to be selected. However, if participants' similarity judgments are more affected the presence of the letter O, the molecule on the right is more likely to be selected. These conjectures do not rely on the assumption that students are aware of these features driving their perception, the assumption that students explicitly attend to the features, or even that they foveally fixate on them.

Our method involves three steps. First, we code a corpus of visuals based on simple features they contained (e.g., the presence or absence of lines between two letters). Second, we collect triplet similarity judgments for each visual of the form "molecule i is more similar to j than it is to k," which provides information about the relative perceptual similarity of the molecules shown in a visual. Third,

we analyze which features drive participants' similarity judgments using metric learning, a branch of machine learning interested in learning notions of distance that correspond to the similarity between items. Before we detail each step, we review how metric learning allow detecting correlations between features and similarity judgments.

4.2.3.1 Computational metric learning approach

To detect which visual features drive participants' similarity judgments, we apply the model developed in previous work (Mason et al., 2017). Simply put, the model seeks to learn a notion of distance between the visuals parametrized by their features such that the most visually similar visuals are nearest to each other. To achieve this goal, the model learns an ordinal embedding that spatially represents the similarity of molecule representations as distances. Formally, the model describes the ith visual by a q-dimensional feature vector $\mathbf{x}_i$. Our goal is to learn a distance metric parametrized by a symmetric positive-semidefinite matrix $\mathbf{K}$, called a *kernel matrix*. The kernel matrix $\mathbf{K}$ is chosen such that triplet similarity judgments of the form "visual i is more similar to visual j than to k" are consistent with the distance metric defined as $d_{\mathbf{K}}(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{K}(\mathbf{x}_i - \mathbf{x}_j)$. Further, $\mathbf{K}$ is assumed to be sparse, giving weight to only a subset of the features, corresponding to the belief that a small number of features significantly influence human similarity judgments (Shepard, 1980). By comparing the weight given to each feature in this subset, we rank the features that are most important in determining similarity among the visuals.

Note that this method is not biased towards selecting features that are more prevalent in the dataset. As in (Mason et al., 2017) uses differences of distances with respect to kernel $\mathbf{K}$ to predict similarity. As shown in the definition of $d_{\mathbf{K}}(\mathbf{x}_i, \mathbf{x}_j)$, distance and hence the prediction of similarity relies on *differences* of the feature vectors $\mathbf{x}_j$. For a feature to have the chance of being predictive, it must vary between feature vectors. For example, a feature that is present in all molecules but takes the same value for all molecules is ignored, except for its interaction with other features.

Additionally, by focusing on the difference of feature vectors, the absence of a given feature can also be used to predict similarity or dissimilarity. Finally, the model seeks to reweight features only based on their ability to predict triplet judgements and hence does not assume to attention to any specific features. Reweighting features also has the benefit that it allows the model to further correct for features being more common by giving them less weight if they are not predictive of similarity.

There are three advantages of this model that make it suitable for modeling perceptual competencies. First, by Theorem 2.1 in (Mason et al., 2017), the number of triplet responses that are necessary and sufficient to learn $\mathbf{K}$ is known: if q is the number of features that describe each representation, d is the number of relevant features that affect students' similarity judgments, and n is the number of molecules students are comparing, then $\mathbf{K}$ can be uniquely identified with $O(dq \log n)$ triplets where $O(\cdot)$ hides constant factors. This ensures that sufficient data is collected to learn $\mathbf{K}$ and detect relevant features. Second, Theorem 2.7 in (Mason et al., 2017) guarantees that the metric is uniquely determined by the triplet responses for a generative model. This ensures that a ranking of the importance of features for participants' triplet judgments uniquely describes their perceived similarity and can hence be confidently used for a cognitive model of perceptual learning. Third, it may be reasonable to expect that participants agree on some triplets more than others, and students may not be fully self-consistent. Other conventional methods that require consistency place a stringent requirement on the data unlikely to be satisfied in practice. The model in (Mason et al., 2017) makes no such assumption. Instead, it directly models he level of disagreement in different triplets and can use this information to better estimate the kernel matrix $\mathbf{K}$. The resulting kernel yields the best predictions on average for the collected data, and correspondingly yields the best explanation on average of students' judgments.

4.2.3.2 Efficiency of metric learning method

As mentioned, sample size (i.e., the number of similarity judgments) needed to learn $\mathbf{K}$ is a practical concern. Active machine learning research has investigated

how to reduce the data needed to achieve good model performance by querying the most informative data-points (see (Settles, 2009) for an overview). Several algorithms have been proposed for triplet-based active machine learning. While some algorithms rely on uncertainty- sampling based approaches to select the most informative queries with respect to a learned model, others rely on an information theoretic approach has been proposed (Tamuz et al., 2011) to select triplets for queries. Both are agnostic to the features (i.e., they do not have access to the features) and instead directly focus on participants' judgments. Notably, none of the algorithms guarantee a number of samples necessary and sufficient to learn K. In fact, the sample complexity of triplet-based metric learning was an open problem until (Mason et al., 2017) demonstrated a lower bound of $O(qd)$ samples being necessary for any algorithm to learn the kernel matrix, $\mathbf{K}$. Since our method requires only $O(qd \log n)$ triplets, it is essentially optimal up to logarithmic factors using random sampling. This does not preclude, however, the possibility of constant and logarithmic factor gains in performance of active versus random sampling, which can have significant impact in real world settings.

In our experiments, we use an implementation of the algorithm proposed by (Tamuz et al., 2011), developed originally as part of the NEXT system (Jamieson et al., 2015). This package showed empirical success versus random sampling in previous, offline experiments (Heim et al., 2015), which we expect to generalize to our context. The results by (Heim et al., 2015) are referred to as *offline* in the context of active learning research, because the authors first collected all $O(n^3)$ possible samples for their dataset, and then allowed the algorithm to query from this set of collected samples as if it was progressively seeing the data. This mitigates some practical concerns of actively querying people such as server and network latency. Our experiments take place in the online setting, where the algorithm progressively updates and queries new samples in real time from participants, but we make use of the same implementation of the same algorithm, which controls the sampling as in (Heim et al., 2015).

4.3 Research Questions

In sum, perceptual representational competencies are acquired via implicit, non-verbal learning processes that result from the induction of relevant visual features through experience with many visuals. We propose a similarity learning method that draws on metric learning to assess participants' perceptual representational competencies without explicitly asking them to explain visual features or assuming that they explicitly attend to these features. We chose chemistry as a domain for this study for several reasons. First, our goal is to develop a method that is applicable to realistic educational scenarios. Much prior research on perceptual learning has used artificial visual stimuli that vary only one or two feature dimensions (e.g., Gabor patches; see (Fahle et al., 2002)). Real visuals are more complex in that they vary on many feature dimensions, and the visuals used in chemistry are a good example. Second, the visuals used in chemistry share several characteristics with visuals commonly used in other STEM domains. For example, they encode spatial information that is relevant to domain-specific concepts, combine symbols, shapes, and color, and they recur throughout instruction. Third, perceptual fluency trainings have been shown to be effective for this population (e.g., (Rau, 2018)), and therefore chemistry students are a representative target population for perceptual fluency trainings.

In sum, to test if the similarity learning method can assess chemistry students' perceptual competencies, we address the following research questions:

1. Which visual features drive chemistry students' perception of similarity among visual representations, as assessed with similarity learning?
2. How do features identified for chemistry students compare to features that novices and experts are expected to attend to, based on prior research that used traditional methods?

While chemistry students are not completely novice to the visuals, they also do not have decades of experience with the visuals that is characteristic of perceptual expertise (Kellman and Massey, 2013; Fiore, 1997). Therefore, we expect that

students will attend to broader features that are not specific to particular molecules–as is common among novices–rather than to features that are specific to a given molecule and that reflect top-down processes through which conceptual knowledge affects perception–as is common among experts. Further, we examine ways to enhance the efficiency of this method by addressing:

3. Does the use of an active learning algorithm improve the efficiency of the method?

While there are many proposed active algorithms, replicating their successes in real world settings remains a challenge (Jamieson et al., 2015). To maximize our chance of success, we employ an implementation of the algorithm proposed by (Tamuz et al., 2011), which independently showed empirical success in an offline setting (Heim et al., 2015) similar to our online, real-time sampling setting. We used the same implementation to select which triplets to sample with the same parameters. Showing that active learning can improve the efficiency of learning from triplets would be some of the first evidence of its efficacy in real world settings where data is sampled in real time and a significant contribution to the field of active learning.

4.4 Experiment 1

Experiment 1 was designed to address research questions 1 and 2. To this end, we collected similarity judgments of Lewis structures (see Figure 4.1a) from undergraduate chemistry students and applied the similarity learning method as described above to these judgments.

4.4.1 Method

4.4.1.1 Participants

A total of 614 freshmen undergraduate students from a general chemistry course for science majors at a large US university participated in the experiment. The

course prerequisite was high-school chemistry. Students were invited to participate in the survey in the middle of the semester, three weeks had covered the basics of covalent bonding, which included information about common functional groups such as alcohols and carboxylic acids. Hence, all students had some knowledge of chemistry and had seen Lewis structures before, but they had not had the level of experience that is characteristic of expertise. Participants were invited via email with a request from the instructor to help with a research project on learning with visual representations. They were not offered any incentive. Participants were allowed to quit the survey at any time.

4.4.1.2 Materials

Participants took a brief online survey asking them to make similarity judgments between Lewis structures showing different molecules. The survey asked them to make triplet similarity judgments of the form shown in Figure 4.2 for three Lewis structures at a time. Each participant received 50 triplets, which they completed at home. They were asked not to use course materials or other support. For each triplet, they had to choose which of the choice molecules was most similar to the target molecule. To create the triplets, we selected 212 molecules that commonly appear in textbooks. From this set, we then selected 50 molecules uniformly at random without replacement. All triplets of three unique molecules were sampled uniformly at random with replacement from the set of $58,880$ possible triplets from the 50 molecules, totaling $26,180$ samples. As per Theorem 2.1 of (Mason et al., 2017), this sample size satisfies the necessary and sufficient number of triplet responses for the rank/sparsity of $\mathbf{K}$.

Further, for each of the 50 molecules, we hand-coded visual features of their Lewis structure representations. Specifically, we coded for 106 specific features that describe anything a naive viewer could see, such as the number of distinct letters, the number of bonds, etc. (see Table 4.1). These features are specific to a given molecule in the sense that they uniquely and sufficiently distinguish between molecules. We quantified these features in feature vectors, which encoded whether

the feature was present or absent. These features include counts of individual letters as well as information about specific bonds present in each molecule. In addition, we created four summary features that reflect our interpretation of the expert-novice literature, which documents that students tend to focus on broader characteristics that are not specific to particular molecules. The summary features are the sum of specific features of a given type; for example, the total number of letters in a Lewis structure is the sum of the number of distinct letters. In total, there were $q = 110$ unique features describing each molecule. To guarantee that our method satisfies the conditions of Theorems 2.3 and 2.7 in (Mason et al., 2017) that guarantee a unique and optimal ranking of features by this procedure, we selected a subset of 50 features of these 110 features that are relevant to the greatest number of molecules. Our prior work shows that reducing the number of features from 110 to 50 features has no effect on the results because many of the features that are left out pertain to only a small number of the molecules and are hence unlikely to explain perceived similarity of the whole set of visuals. Figure 4.3 shows feature vectors for two visuals (red), including their summary features (yellow).

We collected participants' triplet similarity judgments using the NEXT open-source system for active machine learning (Jamieson et al., 2015). NEXT has modules for triplet experiments and runs on top of Amazon Web Services. Further, NEXT provides an easy interface for active data collection, as shown in Figure 4.2.

4.4.1.3 Analysis

To detect which visual features drive chemistry students' perception of similarity (research question 1) and to compare these to features to expected novice-expert differences (research question 2), our goal was to learn a ranking of the features based on how strongly they correlate with students' triplet similarity judgments. As described above, we achieve this goal by learning a symmetric positive-semidefinite kernel matrix $\mathbf{K}$ that defines a distance metric from triplet similarity judgments of the form "molecule i is more similar to molecule j than to k."

Specifically, for a given Lewis structure, let $\mathbf{x}_i$ be its feature vector (i.e., red

Feature Type	**Number of features per molecule**	**Prevalence of feature across molecule**
Specific: Charge	1	8
Specific: Number of dots	1	475
Specific: Number of distinct letters (e.g., number of C's)	14	361
Specific: Number of single bonds between two letters	20	260
Specific: Number of double bonds between two letters	6	22
Specific: Number of triple bonds between two letters	2	6
Specific: Number of 90-degree bond angles between any two bonds	20	232
Specific: Number of 120-degree bond angles between any two bonds	15	76
Specific: Number of 180-degree bond angles between any two bonds	27	132
Summary: Total number of connections (i.e., double bond and triple bonds count as one connection)	1	293
Summary: Total number of bonds (i.e., double bond counts double, triple bonds counts triple)	1	315
Summary: Total number of different letters	1	122
Summary: Total number of letters	1	365

Table 4.1: Summary of hand-coded visual features for Lewis structure representations for the 50 molecules used in Experiment 1. Features include four summary features and 106 specific features.

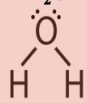
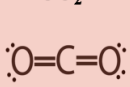
		Feature vector $x_{i=1}$	Feature vector $x_{i=2}$	... $x_{i=50}$
	Molecule representation →	**H_2O** 	**CO_2** 	
	↓ Features			
	single lines	2	4	
	dots	4	8	
	connections	2	2	
	bondType_single,O,H	2		
	bondType_single,C,O			
	bondType_double,C,O		2	
	bondAngle_O{H,H},90	1		
	bondAngle_C{O,O},180		1	
	... i=110			
Educated guess features	number of letters	3	3	
	number distinct letters	2	2	

Figure 4.3: Feature vectors of molecules represented as Lewis structures. Columns show example feature vectors for H_2O and CO_2 Lewis structure representations (red). Rows show features, including surface features (yellow).

columns in Figure 4.3). We model each triplet similarity judgment as a relative distance constraint of the form $d_{\mathbf{K}}(x_i, x_j) < d_{\mathbf{K}}(x_i, x_k)$, where $d_{\mathbf{K}}(x_i, x_j) := (x_i - x_j)^T \mathbf{K}(x_i - x_j)$. This distance function weights differences between some features more than others such that relative distances are as consistent as possible with participants' similarity judgments. Features with more weight (i.e., larger values in $\mathbf{K}$) have stronger correlations with participants' similarity judgments. Importantly, we consider a method for learning $\mathbf{K}$ that does not require knowledge of d and is not biased towards any feature.

For each triplet $t = (i, j, k)$, in the set $\mathcal{S} = \{t : (i, j, k)\}$ of queried triplets, there is an associated label $y_t = \pm 1$ indicating a participant's response (e.g., $y_t = -1$ means "i is more similar to j than to k"). The distance function predicts similarity judgments according to

$$\hat{y}_t = \begin{cases} -1 & x \leqslant d_{\mathbf{K}}(\mathbf{x}_i, \mathbf{x}_j) < d_{\mathbf{K}}(\mathbf{x}_i, \mathbf{x}_k) \\ 1 & 0 \leqslant d_{\mathbf{K}}(\mathbf{x}_i, \mathbf{x}_j) > d_{\mathbf{K}}(\mathbf{x}_i, \mathbf{x}_k) \end{cases}$$

for a learned kernel matrix $\mathbf{K}$, and if $\hat{y}_t = y_t$, the distance function agrees with the participant's response for triplet t. We assume that there exists a latent, generative kernel matrix $\mathbf{K}^*$, and that participants' responses are noisy indications of relative distances with respect to the distance function defined by $\mathbf{K}^*$. Note that their responses may be contradictory and $\mathcal{S}$ may contain duplicate triplets as the queries are generated uniformly at random.

Further, based on the assumption that few features drive participants' judgments, we restrict the learned kernel matrix to be row-sparse, such that only few, $d < q = 50$, features receive weight in $\mathbf{K}$. That is, the distance function only considers differences between a small set of the features to determine the distance between representations. Ideally for a range of choices for d, we would search over all subsets of d features and learn matrices $\mathbf{K}$ that satisfy as many of the triplet constraints as possible. Unfortunately, this optimization is computationally infeasible. Instead, for the set of triplets, $\mathcal{S}$ with associated labels y_t, we optimize the following convex relaxation:

$$\mathbf{K} = \arg\min_{\mathcal{K}_\lambda} \frac{1}{\mathcal{S}} \sum_{t \in S} \log\left(1 + \exp\left(y_t \left(d_{\mathbf{K}}(\mathbf{x}_i, \mathbf{x}_j) - d_{\mathbf{K}}(\mathbf{x}_i, \mathbf{x}_k)\right)\right)\right)$$

over the convex set

$$\mathcal{K}_\lambda := \{\mathbf{K} \in \mathbb{R}^{q \times q} : \mathbf{K} \text{ symmetric PSD}, \|\mathbf{K}\|_{1,2} \leqslant \lambda\}$$

where $\|\mathbf{K}\|_{1,2} := \sum_{i=1}^{q} \left(\sum_{j=1}^{q} \mathbf{K}_{i,j}^2\right)^{1/2}$. This norm encourages solutions that give weight to fewer features without explicitly enforcing a specific number. By increasing and decreasing λ, we can control the number of features in the solution. We solve this optimization via projected gradient descent and choose λ by splitting our dataset into training, validation, and test sets. In particular, we used 80% of the collected queries to learn kernel matrices, $\mathbf{K}$, for a range of different values of λ. We

then tested each kernel on the validation set to find the optimal value of λ. With the selected value of λ, we relearn $\mathbf{K}$ using both the training and validation sets as training data and evaluated the performance on the held-out test set. Additionally, as a form of preprocessing, we normalized all features to have 0 mean, further aligning the procedure with the conditions of Theorem 2.7 in (Mason et al., 2017) for our choice of loss function. This theorem guarantees that the $\mathbf{K}$ we recover as a result of this convex optimization is the unique optimum and hence that the corresponding ranking of features uniquely describes participants' judgments.

4.4.2 Results

4.4.2.1 Prior Checks

We evaluated the accuracy of the model using ten-fold cross validation. Our model achieved an average 72% prediction accuracy of students' similarity judgments. Note that in general, triplet queries were not repeated for different students, which makes it difficult to compute an absolute metric of consistency among participants. From the 50 molecules we considered, there were $58,880$ unique triplets that could have been queried, and not deliberately repeating triplets allowed us to maximize our coverage of this set. Some triplets were repeated due to the random sampling procedure, but 90% of triplets were unique. Due to the large size of the training and test sets, the effect of repeated triplets is minimal. In particular, all theoretical guarantees of the model discussed in (Mason et al., 2017) are from the perspective of random sampling and do not preclude the possibility of repeated samples. That said, the prediction accuracy of this procedure serves as a proxy for estimating the consistency of participants' responses as well as a participant's self-consistency. Note that consistency in the context of triplet judgements is richer than agreement on repeated samples because triplet judgments imply transitive information about the molecules of the form "if molecule i is more similar to j than to k, and i is more similar to k than to i, then we can infer that i is more similar to j than to i." Prediction accuracy captures this richer notion of consistency. In particular, if there was perfect consistency, the model would achieve accuracy close to 100%, whereas

if there was little to no consistency, the model would achieve accuracy near 50%. Thus, the 72% accuracy of our model indicates that there was consensus over which visuals were more or less similar, but also that there were some disagreements among students' similarity judgments.

Further, we validated the claim outlined in Section 4.2.3 that the method is not biased towards selecting more common features. An alternative method to predict triplets, as opposed to learning a kernel matrix, is to use the standard Euclidean distance directly on the feature vectors. More common features have a greater impact on this distance metric. Hence, if our method was biased towards selecting common features, and the most common features explain participants' judgments, then the Euclidean distance method should do well and perform similarly to our model in terms of prediction accuracy. Instead, the Euclidean distance method only achieves 50.3% accuracy, which is markedly worse than the 72% percent accuracy of our method and barely above a model that lacks any consistency.. Note that this merely highlights the importance of feature selection for predicting similarity judgments. It would be possible to use multidimensional scaling to predict similarity judgments; but this would not address research questions 1 and 2. Thus, our method is not simply selecting common features but reweights features that explaining students' similarity judgments.

4.4.2.2 Similarity Judgments

To identify which visual features account for students' similarity judgments, we estimated the weights for each feature in matrix $\mathbf{K}$. The stronger a feature's weight in matrix $\mathbf{K}$, the more this feature affected students' similarity judgments. To compute the weight for a given feature, we calculated the Euclidean norm of that feature's row in the kernel matrix $\mathbf{K}$. Note that since $\mathbf{K}$ is symmetric, it is equivalent to computing this for columns. This norm correlates to the amount that a given feature influences the overall distance metric, with features with greater corresponding norms in $\mathbf{K}$ having greater influence in determining similarity between visuals. For rows whose norm is 0, the corresponding feature is given no weight in determining similarity.

Thus, a feature's weight corresponds to its saliency in students' perception of visuals.

First, we used this matrix to investigate which features drive students' perception of similarity among visuals (research question 1). Table 4.2 shows the top ten ranked features selected in our kernel matrix **K**. This ranking shows that the presence and counts of specific atoms (e.g., sulfur and oxygen) affected similarity judgments. Further, the most prevalent features relate to specific features of bonds. Specifically, with respect to bond types, students focus on whether specific atoms are bonded to one another by single, double, or triple bonds. Finally, similarity judgments rely strongly on angles between specific atoms in chains (e.g., carbon-oxygen-hydrogen chains).

Second, we used matrix **K** to investigate how features identified by similarity learning correspond to features that novices and experts are expected to attend to (research question 2). No summary features were among the highest ranked features. The three summary features (total letters, connections, different letters), ranked 32, 39, and 40, and the last two had little weight in **K**. The observation that several of the highest ranked specific features were related to chemical functional groups aligns with features we expected to be characteristic of expert perception.

4.4.3 Discussion

With respect to visual features that drive students' similarity judgments (research question 1), our results indicate that students' similarity judgments are strongly affected by the presence or absence of specific atoms. This finding may reflect the fact that Lewis structures make atom identity very salient through the use of letters. We note that the most highly ranked bond types contain atoms between less frequent atoms, such as sulfur and nitrogen. Hence, the saliency of atom identity may cause students' perceptual learning processes to be sensitive to specific atoms that are present in fewer molecules, rather than common atoms that appear in most molecules, such as carbon and hydrogen.

Further, we found that students' similarity judgments are not only affected by

Ranking	Feature name	Average Weight
1	Number of sulfur atoms	11.8%
2	180 degree bond angle in fluorine-carbon -fluorine chain	10.0%
3	Sulfur-oxygen double bond type	7.4%
4	Carbon-nitrogen single bond type	6.7%
5	180 degree bond angle in carbon-oxygen -hydrogen chain	6.1%
6	180 degree bond angle in hydrogen-carbon -nitrogen chain	5.3%
7	90 degree bond angle in carbon-carbon -oxygen chain	4.3%
8	Number of oxygen atoms	4.1%
9	Nitrogen-oxygen double bond type	3.9%
10	Carbon-carbon triple bond type	3.8%

Table 4.2: Top ten ranked visual features in Experiment 1 on Lewis structure representations.

individual atoms, but also by groups of atoms. For instance, while the feature pertaining to the number of carbon atoms is not ranked highly, carbon is present in most of the bond angle and bond type features present in the top ten ranked features. This indicates that students' perceptual learning processes are attuned to how carbon interacts with other atoms, rather than its presence directly. This result may reflect that most molecules contain carbon, hydrogen, and oxygen, and consequently bonds amongst these three may be less informative when comparing the similarity of molecules. More broadly, this ranking suggests that students perceptually process bonding when judging the similarity of Lewis structures, a

key concept these visuals depict.

With respect to the comparison of visual features assessed by similarity learning to those suggested by prior expert-novice research (research question 2), our results suggest that students' similarity judgments are more strongly affected by specific visual features than we had expected. Whereas the summary features captured broader information about the visuals (e.g., the number of atoms in the molecule), students instead processed more fine-grained information about the visuals such as the presence of specific atoms and their interactions with other specific atoms via bonding. The impact of more specific features may allow students to compare the molecules more so than the broader features. For example, it is possible that for difficult triplets (i.e., triplets where all three molecules are somewhat dissimilar), students may answer the triplet query by focusing on the presence or absence of a specific feature (e.g., "these three molecules are all kind of different, but these two contain fluorine"). Thus, the impact of specific features on similarity judgments may reflect that these are more informative than the summary features.

Finally, when inspecting the specific features related to bond angles, we observed that they are indicative of functional groups that characterize the type of molecule. For instance, carbon-oxygen-hydrogen chains indicate alcohols, hydrogen-carbon-nitrogen chains indicate amides, and triple bonds indicate unstable molecules. This suggests that students' perceptual processes may reflect conceptual knowledge about how Lewis structures show molecules. Hence, chemistry students seem to exhibit stronger top-down processes through which conceptual knowledge affects perception than we would expect for students who are completely novice to the visuals. Thus, metric learning reveals useful information about how chemistry students perceive visuals, without requiring them to explicitly explain how they do so.

A limitation of our method is that it requires a rather large number of samples. Hence, Experiment 2 investigates if active machine learning can improve the efficiency of this method.

4.5 Experiment 2

Experiment 2 addresses research questions 1-3. To this end, we collected similarity judgements of ball-and-stick models (see Figure 4.1-b) using random and active sampling from undergraduate chemistry students and used the same similarity learning method as above.

4.5.1 Methods

4.5.1.1 Participants

A total of 489 freshmen undergraduate students were recruited from the same general chemistry course as Experiment 1. The course was taught by a different instructor than the course we used in Experiment 1, but thanks to a standardization of the course, the course materials were identical (i.e., content covered, sequence of content, representations used, lecture slides, syllabus, etc.). Students were invited at the same time during the semester as in Experiment 1. All students had some knowledge of chemistry and had seen ball-and-stick models before, but they had not had the level of experience that is characteristic of expertise. Participants were recruited in the same way as in Experiment 1 and did not receive incentives.

4.5.1.2 Materials

Participants again took part in a brief online survey asking them to make similarity judgments between visual representations. We generated 50 ball-and-stick model representations of the same molecules as in Experiment 1 using WebMO, a modelling software for chemical molecules. As in Experiment 1, each ball-and-stick model representation had a hand-coded $q = 132$ dimensional feature vector which from which we took a subset of $q = 50$ features that included the summary features by the same method as in Experiment 1, described above (see Table 4.3). Figure 4.4 shows feature vectors for two visuals (red), including their summary features (yellow). We collected 9005 randomly sampled triplet similarity judgments as well as

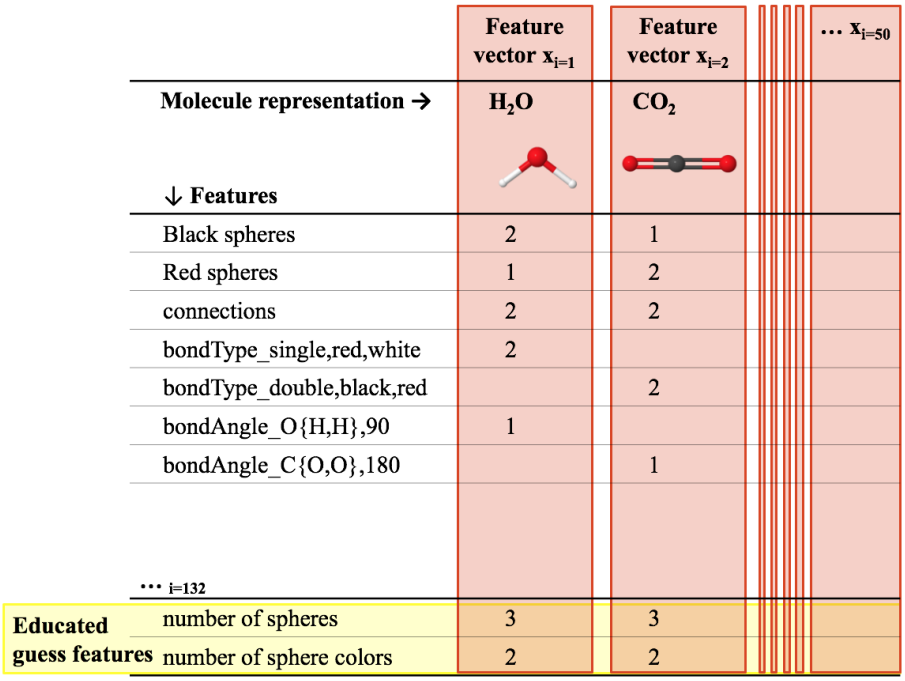


Figure 4.4: Feature vectors of molecules represented as ball-and-stick models. Columns show example feature vectors for H_2O and CO_2 ball-and-stick model representations (red). Rows show features, including educated guess features (yellow).

8792 actively sampled triplets, totaling 17,797 samples. An example triplet with three ball-and-stick model is shown in Figure 4.5.

As with Experiment 1, we used the NEXT system (Jamieson et al., 2015) to collect participants' similarity judgments, and responses were collected via an online survey that asked students not to use any outside materials. A difference to Experiment 1 was that we used active sampling, so as to address research question 3. Active triplets were sampled according to the CrowdKernel algorithm for active triplet embedding (Tamuz et al., 2011). To select the next triplet to query, the algorithm computes an information gain criterion that estimates which triplets provide the most information based on past responses. Importantly, as opposed to random sampling, which gathers independent samples, active sampling is inherently sequential, collecting dependent samples to minimize redundant information. Ideally, this

Feature type	Number of features per molecule	Prevalence of feature across molecules
Specific: Number of distinct sphere colors	15	367
Specific: Number of single bonds between any two spheres	19	291
Specific: Number of double bonds between any two spheres	5	21
Specific: Number of triple bonds between any two spheres	2	6
Specific: Bond lengths of single bonds between any two spheres	19	99
Specific: Bond lengths of double bonds between any two spheres	5	17
Specific: Bond lengths of triple bonds between any two spheres	2	6
Specific: Number of 109-degree bond angles between any two bonds	22	364
Specific: Number of 120-degree bond angles between any two bonds	19	85
Specific: Number of 180-degree bond angles between any two bonds	6	13
Specific: Atomic radii of spheres of a given color	15	122
Summary: Total number of connections	1	321
Summary: Total number of different sphere colors	1	121
Summary: Total number of spheres	1	367

Table 4.3: Summary of hand-coded visual features for ball-and-stick model representations for the 50 molecules used in Experiment 2. Features include three summary features and 129 specific

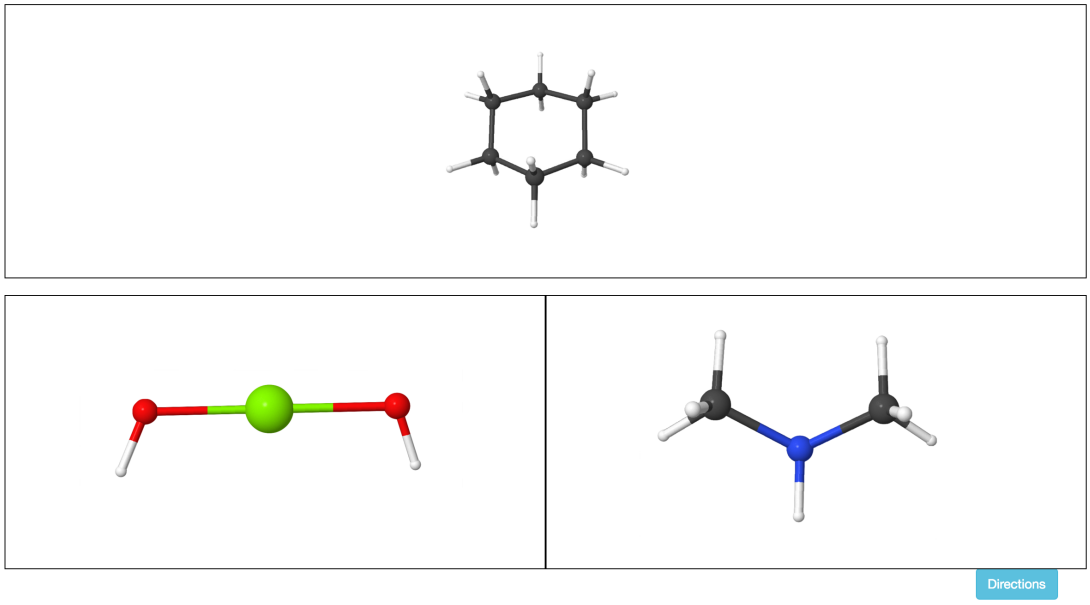


Figure 4.5: Example triplet judgment task with ball-and-stick models, as used in Experiment 2. Participants are given a target molecule (top) and asked to click on one of the two choice molecules (bottom) that are most similar to the target molecule.

would allow the algorithm to achieve a more precise estimate of $\mathbf{K}$ in fewer samples. We used the standard implementation of this algorithm provided in the NEXT system. Because searching over the full set of triplets is computationally expensive and because it is important to minimize downtime when querying participants, the implementation instead computes the information gain criterion over a randomly drawn subset of the full set of triplets and returns the most informative triplet from that subset. In spite of this issue, the implementation has been shown to achieve gains in efficiency with respect to random sampling (Heim et al., 2015).

4.5.1.3 Analysis

To detect which features drive students' perception of similarity among visuals (research question 1) and to compare them to expected features (research question

2), we used the same procedure as in Experiment 1, except that for Experiment 2, $\mathbf{K}$ is learned from a combination of both actively and randomly sampled triplets, so as to make maximum use of our data. In addition, we compared the performance of an active learning for triplet selection to randomly sampling triplets in terms of learning an embedding based on perceptual similarity (research question 3). To this end, we used the algorithm outlined in (Jain et al., 2016b) for the randomly sampled triplets to learn a Euclidean embedding from participants' similarity judgments. The embedding maximally satisfies the provided constraints from both actively and randomly sampled triplets. We chose this method instead of other methods (e.g., (Agarwal et al., 2007; Jamieson and Nowak, 2011; Kruskal, 1964b; Van Der Maaten and Weinberger, 2012)) because it is theoretically motivated and the only method to give optimality guarantees. To learn an embedding of the actively sampled triplets, we used the algorithm outlined by (Tamuz et al., 2011), which–although it lacks the guarantees of the algorithm by (Jain et al., 2016b)–guarantees asymptotically recovering an optimal solution. Moreover, this algorithm is better suited for learning from actively sampled triplets according to the CrowdKernel model (Tamuz et al., 2011) than the algorithm by (Jain et al., 2016b). It is worth noting, however, that both algorithms have the same optimal solution in the generative setting that a true embedding exists from which we sample noiseless triplets and attempt to reconstruct the embedding. Further, if triplet responses are noisy as is common in realistic contexts, both recover similar embeddings.

To compare how the accuracy of the active and random sampling embeddings changes as a function of the sample size used for training, we computed errors of each embedding for different training set sizes. We computed errors as the proportion of triplet constraints the learned embedding violates, given the training set size that was used to learn the embedding. If active learning is more efficient, we should see fewer errors (i.e., fewer constraints violated on the unseen test set) for an embedding learned from a training set of actively sampled triplets than for randomly sampled ones. Specifically, of the 9005 triplet similarity judgments, we used up to 6000 for training and 3005 for testing. For example, to determine the relative accuracy of the active and random sampling embeddings for a training set

size of 500 triplets, we trained each with 500 triplets and tested them on 3005 triplets to compute their errors. Likewise, to determine the relative accuracy of the active and random sampling embeddings for a training set size of 1000 triplets, we trained each with 1000 triplets and tested them on 3005 triplets to compute their errors. We repeated this procedure at intervals of 500 triplets between 500 to 6000 triplets. The training sets from the randomly sampled triplets were selected uniformly at random without replacement. The training sets from the actively sampled triplets were grown progressively in the order that the active algorithm queried triplets to mimic the sequential, dependent procedure of the active algorithm.

We opted not to use an alternate method for assessing the performance of the active learning method, which would be to perform metric learning (see Experiment 1) by learning two matrices $\mathbf{K}_1$, $\mathbf{K}_2$ from similarity judgments generated by the random and the active algorithms. From there, with the notion of distance $d_{\mathbf{K}}(\mathbf{x}_i, \mathbf{x}_j)$ as defined above, we could likewise compare the proportion of held out triplets each $\mathbf{K}$ satisfies when learned from actively versus randomly sampled triplets. We also did not compare performance on feature selection directly because it is difficult to measure accuracy in this setting without a ground truth set of important features. Instead, we chose to compare performance on ordinal embedding for three reasons. First, ordinal embedding compared to low-dimensional metric learning is a better studied problem. There are efficient and precise parameter-free algorithms that yield simpler and computationally efficient analyses. This is important because the goal of this research is to develop a cognitive model for educational technologies, which would frequently assess student performance. Second, our active sampling method is agnostic to the features of the visuals. However, the geometry of the features that characterize each visual impacts the ability to learn $\mathbf{K}$ (see Theorem 2.1 by (Mason et al., 2017)). By performing ordinal embedding as opposed to metric learning, we mitigate the confounding factor of the geometry of the feature vectors impacting our ability to learn a matrix $\mathbf{K}$ that satisfies triplet constraints. Instead, we can compare the quality of the embeddings themselves. Third, ordinal embedding from triplets is a restricted version of the more general problem of metric learning from triplets (see (Mason et al., 2017)). Hence, if actively sampled triplets generate

an embedding that more readily generalizes to unseen triplets than an embedding generated from random triplets, it implies that it is possible to learn a matrix **K** with similarly superior generalization performance in the metric learning problem. In sum, comparing the actively and randomly sampled triplets by their ability to generate ordinal embeddings that generalize to unseen data is a better method for comparing actively and randomly sampled datasets than comparing the quality of learned metrics or selected features directly.

4.5.2 Results

Using ten-fold cross validation, we achieved an average 63% prediction accuracy of students' similarity judgments on the ball-and-stick models. This finding indicates that there was consensus about the similarity among visuals, but also that there were some disagreements among students' similarity judgments. Further, there was greater disagreement for ball-and-stick triplets than for Lewis structure triplets. To identify which visual features account for students' similarity judgments, we estimated the weights for each feature in matrix **K** as in Experiment 1.

First, we investigated which features drive perception of similarity among visuals (research question 1). Table 4 shows the top ten ranked features selected in our kernel matrix **K**.

Most notably, this ranking indicates that a single feature dominates the metric far more strongly than was the case for the Lewis structure representations. This indicates not only that students' similarity judgments are strongly affected by this feature, but also that there is a greater degree of consensus between students about how this feature corresponds to similarity between ball-and-stick models. Further, we notice that students' similarity judgments of the ball-and- stick models are strongly affected by specific features, such as the number of carbon atoms or beryllium atoms. Also, their judgments are driven by specific features related to bonding, specifically bond types between specific atoms and bond angles in specific atom chains. In addition, summary features such as the total number of atoms and number of bonds also impacted students' similarity judgments. These summary

Ranking	Feature Name	Average Weight
1	109.5 degree bond angle in hydrogen-oxygen-carbon chain	27.3%
2	Length of a single bond between oxygen and hydrogen	7.0%
3	Length of a single bond between carbon and carbon	6.3%
4	Number of carbon atoms	5.1%
5	Length of a single bond between oxygen and nitrogen	4.2%
6	Number of bonds	3.9%
7	Number of beryllium atoms	3.5%
8	120 degree bond angle in flourine-carbon-carbon chain	3.4%
9	Atomic radii of hydrogen atoms	3.0%
10	Number of total atoms	2.6%

Table 4.4: Top ten ranked visual features in Experiment 2 on ball-and-stick model representations.

features are ranked amongst the top ten features but are weighted less strongly than the more specific features.

The ranking visual features also allows us to investigate how the features assessed with similarity learning correspond to visual features that novices and experts are expected to attend to (research question 2). As noted above, summary features were predictive of students' similarity judgments, but not as strongly as the specific visual features that uniquely describe specific molecules. Hence, while students' similarity judgments were affected by broad features, they were more strongly affected by specific features that experts would be expected to attend to.

Finally, we investigated whether active learning improves the efficiency of the method (research question 3). Figure 4.6 shows a comparison of the errors of active versus random sampling in predicting triplet judgments by training sample sizes.

To analyze this data, we computed t-tests for different training set sizes in intervals of 500 samples, yielding 12 t-tests (corresponding to the 12 data points

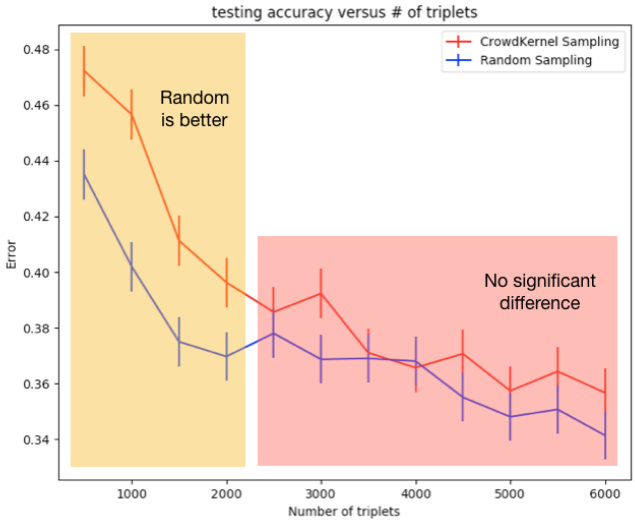


Figure 4.6: Errors of random sampling (blue) and active (blue) sampling methods as a function of the number of triplets used in the training set. The y-axis shows the percentage of triplet queries that are not satisfied in the embedding learned for with the training set of each size on the x-axis. Lower values correspond to better performance with respect to this metric. Error bars were computed using a binomial proportion confidence interval for one standard deviation.

in Figure 4.6). Each t-test compared the performance of the model learned by the active or random method on predicting students' responses to 3,005 triplet queries that were held out for testing. Table 4.5 shows the Bonferoni-adjusted p-values from the t-tests. For training set sizes lower than 1,500, we found significant advantages of the random sampling method, but not for larger training set sizes.

4.5.3 Discussion

With respect to visual features that drive chemistry students' similarity judgments (research question 1), our results indicate that students' perceptual learning processes are attuned to specific features such as the presence or absence of specific atoms as well as on features related to interactions among specific atoms via bond type and bond angles. These results align with our findings on students' perceptual

Training set size	t-value	p-value (unadjusted)	p-value (Bonferoni adjusted)
500	$t(3005) = 2.904$	0.004	0.044
1000	$t(3005) = 4.28$	<.001	<.001
1500	$t(3005) = 2.88$	0.004	0.048
2000	$t(3005) = 2.123$	0.034	0.405
2500	$t(3005) = 0.611$	0.542	1.0
3000	$t(3005) = 1.887$	0.059	0.711
3500	$t(3005) = 0.16$	0.873	1.0
4000	$t(3005) = -0.187$	0.851	1.0
4500	$t(3005) = 1.261$	0.207	1.0
5000	$t(3005) = 0.756$	0.45	1.0
5500	$t(3005) = 1.103$	0.27	1.0
6000	$t(3005) = 1.245$	0.213	1.0

Table 4.5: Results from t-tests comparing active and random sampling methods by training set size.

processing of Lewis structures in Experiment 1. We observe that the bonding-related features include a combination of atoms present in many molecules (e.g., carbon), as well as atoms that are more distinctive, such as chlorine and fluorine. Further, when inspecting the atoms in bonding-related features, we note that they are indicative of chemical functional groups and hence seem to carry conceptual meaning. These findings align with our findings from Experiment 1.

A difference between the findings from the experiments is, however, that the ball-and-stick model features contain more common atoms than the Lewis structure features from Experiment 1. A possible explanation is that ball-and-stick models show atom identity by color, which may make it more difficult for students to map them to elements than Lewis structures that show atom identity by letters that resemble the element name (e.g., yellow for sulfur is less intuitive than S for sulfur). It is likely that students are more familiar with the color used for common atoms such as carbon, oxygen, and hydrogen than with the colors used for uncommon atoms. Hence, when presented with ball-and-stick models, their perceptual learning processes may be more strongly affected by common atoms

than when presented with Lewis structures. This may explain why the top ten features for ball-and-stick models contained more common atoms than those for Lewis structures.

A further difference to Experiment 1 regards how the features we identified compare to features that we expected novices and experts to attend to (research question 2). While Experiment 1 had shown that students' perceptual processing of Lewis structures is mostly driven by specific features, Experiment 2 showed that students' perceptual processing of ball-and-stick models was also driven by broader summary features, such as the total number of atoms and the total number of bonds–although less than by the specific features. Again, the fact that ball-and-stick models show atom identity by color may explain this difference. First, identifying atom identity with ball-and-stick models may require more experience because mappings to atom identity is less salient than for Lewis structures. Second, because Lewis structures are more prevalent in chemistry instruction, students may have less experience with ball-and-stick models. If students lack experience with how ball-and-stick models show atoms, their perception may have to rely on broader features such as the total number of atoms.

With respect to whether the use of active learning improves the efficiency of the method (research question 3), our results are somewhat disappointing. First, in the initial stages of sampling (less than $1,500$ triplets in the training set), we find an advantage of the random sampling. We explain this result in the following way. Before the active algorithm can ask potentially more informative queries, it must first build an initial estimate of the model it is attempting to learn. This is referred to as the "cold start problem" (Su and Khoshgoftaar, 2009). To do this, the model begins by deterministically asking queries where each visual is the "target molecule" for a set number of queries with different visuals as the "choice molecules." As a result, in the initial stages of sampling triplets, the active algorithm may be prone to asking correlated and thus less informative queries as compared to random sampling. This may explain why the random sampling method performs better at the initial stages of sampling. Second, for larger numbers of training set sizes ($2,000$ triplets or more in the training set), we find no differences between the

methods. Here, the active algorithm has begun asking the queries it believes to be most informative as opposed to queries with a given target molecule. Note that the algorithm's guess at which query is most informative is always with respect to the embedding it has learned thus far. We hypothesize that for moderate numbers of triplets, the embedding learned thus far is somewhat inaccurate so the gain with respect to random sampling may be small if present at all. It is possible that with a larger number of samples, the active algorithm may eventually outperform random sampling, but the effect of the cold-start problem is such that this effect is not visible for the number of samples collected in Experiment 2. It is also possible that the performance gain is very small due to that fact that participants' responses tend to be noisy and even self-contradictory, which makes learning an embedding challenging, independent of the sampling method and cold-start problem. In sum, our results show that there seems to be no advantage in actively sampling similarity judgments in our setting, despite successes in related settings (e.g., (Heim et al., 2015)).

4.6 General Discussion

This article makes two contributions to research on perceptual representational competencies. First, we show that we can assess perceptual representational competencies without verbalization or the assumption of explicit attention. We applied this method to two different types of visuals and were able to identify which features drive students' perception of visuals of chemical molecules. Our results show that similarity learning methods can identify and rank the impact of visual features on chemistry students' perceptual learning processes when they engage with visuals. Hence, this method allowed us to examine the nature of the features that affect perceptual processing. Both experiments showed that students' perceptual processing is affected by specific features that describe uncommon atoms and interactions among common and uncommon atoms via bonding. This finding provides further evidence for the interrelated nature of conceptual and perceptual representational competencies. Given that students' perception of similarity is affected by features

that are conceptually meaningful because they describe chemical functional groups, students have moved beyond being driven by broader features that novices may be expected to attend to. Instead, their perceptual processing system seems to be informed by top-down processes that reflect conceptual knowledge about chemical molecules. In contrast to prior methods for assessing representational competencies, we uncovered these top-down processes without asking students to verbally explain, which could have falsely prompted conceptual knowledge.

Second, our findings suggest that comparing perceptual processing of different visuals can provide insights into their relative difficulty. Experiment 2 showed that chemistry students' perceptual processing is affected by specific as well as broad features of ball-and-stick models. The specific features align with those from Experiment 1 in that they describe chemical functional groups that reflect conceptual knowledge about molecules. Yet, compared to the Lewis structure features, the ball-and-stick model features contained more common atoms. A reason for this finding may be that Lewis structures more saliently represent atom identity, which makes it easier for students to identify uncommon atoms, whereas ball-and-stick models denote atom identity with colors that may make it more difficult to identify atoms. Further, while none of the features that were drivers of perceptions of Lewis structures were broader summary features, the important features for ball-and-stick models included summary features. Again, the fact that ball-and-stick models may be more difficult for students than Lewis structures may explain this result. Ball-and-stick models make atom identity less salient and students tend to have less experience with them. If students have less experience in identifying atoms based on the ball-and-stick model color scheme, their perceptual processing system may have to rely on broader features to judge the similarity of visuals.

These two contributions should be interpreted in light of our procedure for coding the feature vectors. Our method assumes that the features that drive students' similarity judgments are represented in the feature codes. While we are confident that our feature vectors reflect objective characteristics of the visuals because coding the features did not require any knowledge about the molecules or the visuals, we

note that coding of complex stimuli always contains a degree of subjectivity. This may particularly be true of the summary features that we created based on literature suggesting that students attend to broader characteristics rather than features of specific molecules. Our interpretation of this literature may be subjective and could have led us to construct particular summary features that may not accurately reflect how chemistry students in our sample perceive similarity among feature vectors.

Further, our research makes two contributions to machine learning. The first contribution to machine learning is less optimistic. We had expected that active sampling would improve the efficiency of the similarity learning method, thereby allowing us to assess perceptual competencies with fewer samples; but, this was not the case. The active learning method yielded lower accuracy than the random sampling method for small samples (less than $1,500$ samples in the training set). A positive side of this finding is that extant methods of random sampling, which are easier to implement than active sampling methods, yield superior results, especially for small samples that are likely more prevalent in educational research.

A second contribution to machine learning is that we provide the first real-world demonstration of a new mathematical theory for feature selection based on metric learning based on the method by (Mason et al., 2017). This method is the first to allow practitioners to both quantify the accuracy of the learned perceptual embeddings and guarantee unique recovery of the kernel matrix, $\mathbf{K}$. Therefore, our ranking of features likewise is unique and optimal. Specifically, we applied bounds on the accuracy of estimating low-dimensional metrics learned from small numbers of comparative judgments. Further, we test the practical performance of the empirical risk minimization framework outlined in our prior theoretical work (Mason et al., 2017) and demonstrate its feasibility for problems with moderate numbers of features and large numbers of samples. Specifically, we explored the application of group Lasso regularized metric learning algorithms for automatically selecting the most perceptually salient features by learning a low- dimensional metric. Thus, our experiment empirically validates the low-dimensional metric learning approach with similarity judgments of undergraduate chemistry students, as well as new and provably accurate machine learning methods to assess how

visual features predict or encode perceptual similarity judgments.

4.7 Limitations and Future Directions

A limitation of our research results from our choice of population. We focused on chemistry students because they are a target population for perceptual fluency trainings. But this choice also implies that students had prior exposure to the visuals and likely had high motivation to learn about them. Hence, our conclusions that students' perceptual competencies have moved beyond those of novices because the visual features align with chemical functional groups that we expect experts to attend to is–although it is founded in the expert-novice literature on conceptual and perceptual knowledge–to some extent speculative.

Further, a limitation of the similarity learning method is that it requires relatively large samples. Ideally, cognitive models rely on assessments of individual students' competencies. Currently, our method can be used to assess perceptual representational competencies for subpopulations (e.g., freshmen chemistry students) but cannot be used to assess an individual's competencies. Future research will investigate whether using more images per query (e.g., quadruplets instead of triplets) would increase the information value of each query without increasing the difficulty of the task for participants. Further, it is possible that combining cohort information (e.g., freshmen students tend to focus on...) with active sampling yields efficiency gains that can assess individual students' perceptual competencies.

A final limitation of our research results from its focus on visuals of chemical molecules. While we consider the fact that we used realistic visuals a strength of our research because it extends prior research on perceptual learning that often relies on artificial visuals that have only one or two feature dimensions (e.g., (Fahle et al., 2002)), the complexity of the visuals we considered may not be representative of *all* realistic stimuli. Indeed, the complexity of the visuals we considered is evidenced by the fact that they required 110 (Lewis structures) or 132 (ball-and-stick models) features to describe their information content. This complexity may explain why students' similarity judgments were somewhat contradictory, which reduced the

accuracy of our models. While many visuals in most STEM domains are complex (e.g., drawings of cells in biology, circuit diagrams in physics), there are certainly simpler visuals (e.g., line graphs in math), and future research should determine if our findings generalize to these visuals.

4.8 Conclusion

To the best of our knowledge, the present experiments are the first to test a method to assess students' implicit perceptual competencies without requiring explicit verbalization or explicit attention. In spite of its limitations, this method allowed us to identify and quantify the impact of visual features on chemistry students' perceptual processing of visuals.

Applying this method to two types of visuals that are common in chemistry allowed us to examine how students' perceptions of the visuals that reflect differences in how saliently the visuals depict specific information about chemical molecules. Further, the visual features revealed that students' perception is informed top-down by conceptual knowledge about the visuals, hence providing new evidence for the interrelated nature of perceptual and conceptual processes. While future research has to address limitations of this method, this research takes a significant step towards the development of cognitive models that can assess students' perceptual competencies via implicit measures. This is important because most educational technologies focus on conceptual competencies and fail to incorporate adaptive supports for perceptual competencies–a limitation we attribute to lack of methods to assess implicit knowledge. Thus, the present experiments may advance adaptive instruction for implicit knowledge.

5 LEARNING NEAREST NEIGHBOR GRAPHS FROM NOISY DISTANCE SAMPLES

5.1 Introduction

In modern machine learning applications, we frequently seek to learn proximity/ similarity relationships between a set of items given only noisy access to pairwise distances. For instance, practitioners wishing to estimate internet topology frequently collect one-way-delay measurements to estimate the distance between a pair of hosts (Eriksson et al., 2010). Such measurements are affected by physical constraints as well as server load, and are often noisy. Researchers studying movement in hospitals from WiFi localization data likewise contend with noisy distance measurements due to both temporal variability and varying signal strengths inside the building (Booth et al., 2019). Additionally, human judgments are commonly modeled as noisy distances (Shepard, 1962; Kruskal, 1964b). As an example, Amazon Discover asks customers their preferences about different products and uses this information to recommend new items it believes are similar based on this feedback. We are often primarily interested in the *closest* or *most similar* item to a given one– e.g., the closest server, the closest doctor, the most similar product. The particular item of interest may not be known *a priori*. Internet traffic can fluctuate, different patients may suddenly need attention, and customers may be looking for different products. To handle this, we must learn the closest/ most similar item for *each* item. This chapter introduces the problem of learning the *Nearest Neighbor Graph* that connects each item to its nearest neighbor from noisy distance measurements.

Problem Statement: Consider a set of n points $\mathcal{X} = \{x_1, \cdots, x_n\}$ in a metric space. The metric is unknown, but we can query a stochastic oracle for an estimate of any pairwise distance. In as few queries as possible, we seek to learn a nearest neighbor graph of $\mathcal{X}$ that is correct with probability $1 - \delta$, where each x_i is a vertex and has a directed edge to its nearest neighbor $x_{i^*} \in \mathcal{X} \setminus \{x_i\}$.

5.1.1 Related work

Nearest neighbor problems (from noiseless measurements) are well studied and we direct the reader to Bhatia et al. (2010) for a survey. Clarkson (1983); Vaidya (1989); Sankaranarayanan et al. (2007) all provide theory and algorithms to learn the nearest neighbor graph which apply in the noiseless regime. Note that the problem in the noiseless setting is *very* different. If noise corrupts measurements, the methods from the noiseless setting can suffer persistent errors. There has been recent interest in introducing noise via subsampling for a variety of distance problems LeJeune et al. (2019); Bagaria et al. (2017, 2018), though the noise here is not actually part of the data but introduced for efficiency. In our algorithm, we use the triangle inequality to get tighter estimates of noisy distances in a process equivalent to the classical Floyd–Warshall Floyd (1962); Cormen et al. (2009). This has strong connections to the metric repair literature (Brickell et al., 2008; Gilbert and Jain, 2017) where one seeks to alter a set of noisy distance measurements as little as possible to learn a metric satisfying the standard axioms. (Singla et al., 2016) similarly uses the triangle inequality to bound unknown distances in a related but noiseless setting. In the specific case of noisy distances corresponding to human judgments, a number of algorithms have been proposed to handle related problems, most notably Euclidean embedding techniques, e.g. (Jain et al., 2016b; Van Der Maaten and Weinberger, 2012; Kruskal, 1964b). To reduce the load on human subjects, several attempts at an active method for learning Euclidean embeddings have been made but have only seen limited success Jamieson et al. (2015). Among the culprits is the strict and often unrealistic modeling assumption that the metric be Euclidean and low dimensional.

5.1.2 Main contributions

In this chapter, we introduce the problem of identifying the *nearest neighbor graph* from noisy distance samples and propose `ANNTri`, an active algorithm, to solve it for general metrics. We empirically and theoretically analyze its complexity to show improved performance over a passive and an active baseline. In favorable settings, such as when the data forms clusters, `ANNTri` needs only $\mathcal{O}(n \log(n)/\Delta^2)$ queries, where Δ accounts for the effect of noise. Furthermore, we show that `ANNTri` achieves superior performance compared to methods which require much stronger assumptions. We highlight two such examples. In Fig. 5.2c, for an embedding in $\mathbb{R}^2$, `ANNTri` outperforms the common technique of triangulation that works by estimating each point's distance to a set of anchors. In Fig. 5.3b, we show that `ANNTri` likewise outperforms Euclidean embedding for predicting which images are most similar from a set of similarity judgments collected on Amazon Mechanical Turk. The rest of the chapter is organized as follows. In Section 7.2, we further setup the problem. In Sections 5.3 and 5.4 we present the algorithm and analyze its theoretical properties. In Section 5.5 we show `ANNTri`'s empirical performance on both simulated and real data. In particular, we highlight its efficiency in learning from human judgments.

5.2 Problem setup and summary of our approach

We denote distances as $d_{i,j}$ where $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_{\geqslant 0}$ is a distance function satisfying the standard axioms and define $x_{i^*} := \arg\min_{x \in \mathcal{X} \setminus \{x_i\}} d(x_i, x)$. Though the distances are unknown, we are able to draw independent samples of its true value according to a stochastic distance oracle, i.e. querying

$$Q(i,j) \quad \text{yields a realization of} \quad d_{i,j} + \eta, \tag{5.1}$$

where η is a zero-mean subGaussian random variable assumed to have scale parameter $\sigma = 1$. We let $\hat{d}_{i,j}(t)$ denote the empirical mean of the values returned by $Q(i,j)$ queries made until time t. The number of $Q(i,j)$ queries made until time

t is denoted as $T_{i,j}(t)$. A possible approach to obtain the nearest neighbor graph is to repeatedly query all $\binom{n}{2}$ pairs and report $x_{i^*}(t) = \arg\min_{j \neq i} \hat{d}_{i,j}(t)$ for all $i \in [n]$. But since we only wish to learn $x_{i^*} \forall i$, if $d_{i,k} \gg d_{i,i^*}$, we do not need to query $Q(i,k)$ as many times as $Q(i,i^*)$. To improve our query efficiency, we could instead adaptively sample to focus queries on distances that we estimate are smaller. A simple adaptive method to find the nearest neighbor graph would be to iterate over $x_1, x_2, \ldots, x_n$ and use a best-arm identification algorithm to find x_{i^*} in the i^{th} round.[1] However, this procedure treats each round independently, ignoring properties of metric spaces that allow information to be shared between rounds.

- Due to symmetry, for any $i < j$ the queries $Q(i,j)$ and $Q(j,i)$ follow the same law, and we can *reuse* values of $Q(i,j)$ collected in the i^{th} round while finding x_{j^*} in the j^{th} round.
- Using concentration bounds on $d_{i,j}$ and $d_{i,k}$ from samples of $Q(i,j)$ and $Q(i,k)$ collected in the i^{th} round, we can bound $d_{j,k}$ via the triangle inequality. As a result, we may be able to state $x_k \neq x_{j^*}$ without even querying $Q(j,k)$.

Our proposed algorithm `ANNTri` uses all the above ideas to find the nearest neighbor graph of $\mathcal{X}$. For general $\mathcal{X}$, the sample complexity of `ANNTri` contains a problem-dependent term that involves the order in which the nearest neighbors are found. For an $\mathcal{X}$ consisting of sufficiently well separated clusters, this order-dependence for the sample complexity does not exist.

5.3 Algorithm

Our proposed algorithm (Algorithm 1) `ANNTri` finds the nearest neighbor graph of $\mathcal{X}$ with probability $1 - \delta$. It iterates over $x_j \in \mathcal{X}$ in order of their subscript index and finds x_{j^*} in the j^{th} 'round'. All bounds, counts of samples, and empirical means are stored in $n \times n$ symmetric matrices in order to share information between

[1] We could also proceed in a non-iterative manner, by adaptively choosing which among $\binom{n}{2}$ pairs to query next. However this has worse empirical performance and same theoretical guarantees as the in-order approach.

Algorithm 1 `ANNTri`

Require: n, procedure `SETri`, 2, confidence δ

Initialize $\hat{d}, T$ as $n \times n$ matrices of zeros, $U, U^{\triangle}$ as $n \times n$ matrices where each entry is ∞, $L, L^{\triangle}$ as $n \times n$ matrices where each entry is $-\infty$, NN as a length n array
for $j = 1$ to n **do**
  for [**do**find tightest triangle bounds]$i = 1$ to n
    for all $k \neq i$ **do**
      Set $U^{\triangle}[i,k],\ U^{\triangle}[k,i], \leftarrow \min_{\ell} U^{\triangle_\ell}_{i,k}$, see (5.7)
      Set $L^{\triangle}[i,k],\ L^{\triangle}[k,i] \leftarrow \max_{\ell} L^{\triangle_\ell}_{i,k}$, see (5.8)
    end for
  end for
  $NN[j] = \texttt{SETri}(j, \hat{d}, U, U^{\triangle}, L, L^{\triangle}, T, \xi = \delta/n)$
end for
return The nearest neighbor graph adjacency list NN

different rounds. We use Python array/Matlab notation to indicate individual entries in the matrices, for e.g., $\hat{d}[i,j] = \hat{d}_{i,j}(t)$. The number of $Q(i,j)$ queries made is queried is stored in the $(i,j)^{th}$ entry of T. Matrices U and L record upper and lower confidence bounds on $d_{i,j}$. $U^{\triangle}$ and $L^{\triangle}$ record the associated triangle inequality bounds. Symmetry is ensured by updating the $(j,i)^{th}$ entry at the same time as the $(i,j)^{th}$ entry for each of the above matrices. In the j^{th} round, `ANNTri` finds the correct x_{j^*} with probability $1 - \delta/n$ by calling `SETri` (Algorithm 2), a modification of the successive elimination algorithm for best-arm identification. In contrast to standard successive elimination, at each time step `SETri` only samples those points in the active set that have the fewest number of samples.

5.3.1 Confidence bounds on the distances

Using the subGaussian assumption on the noise random process, we can use Hoeffding's inequality and a union bound over time to get the following confidence

Algorithm 2 `SETri`

Require: index j, callable oracle $Q(\cdot,\cdot)$ (5.1), six $n \times n$ matrices: $\hat{d}$, U, $U^{\triangle}$, L, $L^{\triangle}$, T, confidence ξ

Initialize active set $\mathcal{A}_j \leftarrow \{a \neq j : \max\{L[a,j], L^{\triangle}[a,j]\} < \min_k \min\{U[j,k], U^{\triangle}[j,k]\}\}$
while $|\mathcal{A}_j| > 1$ **do**
  for all $i \in \mathcal{A}_j$ such that $T[i,j] = \min_{k\in\mathcal{A}_j} T[i,k]$ (only query points with fewest samples) **do**
    Update $\hat{d}[i,j],\ \hat{d}[j,i] \leftarrow (\hat{d}[i,j] \cdot T[i,j] + Q(i,j))/(T[i,j]+1)$
    Update $T[i,j],\ T[j,i] \leftarrow T[i,j] + 1$
    Update $U[i,j],\ U[j,i] \leftarrow \hat{d}[i,j] + C_\xi(T[i,j])$
    Update $L[i,j],\ L[j,i] \leftarrow \hat{d}[i,j] - C_\xi(T[i,j])$
  end for
  Update $\mathcal{A}_j \leftarrow \{a \neq j : \max\{L[a,j], L^{\triangle}[a,j]\} < \min_k \min\{U[j,k], U^{\triangle}[j,k]\}\}$
end while
return The index i for which $x_i \in \mathcal{A}_j$

intervals on the distance $d_{j,k}$:

$$|\hat{d}_{j,k}(t) - d_{j,k}| \leqslant \sqrt{2\frac{\log(4n^2(T_{j,k}(t))^2/\delta)}{T_{j,k}(t)}} =: C_{\delta/n}(T_{j,k}(t)), \tag{5.2}$$

which hold for all points $x_k \in \mathcal{X} \setminus \{x_j\}$ at all times t with probability $1 - \delta/n$, i.e.

$$\mathbb{P}(\forall t \in \mathbb{N}, \forall i \neq j, d_{i,j} \in [L_{i,j}(t), U_{i,j}(t)]) \geqslant 1 - \delta/n, \tag{5.3}$$

where $L_{i,j}(t) := \hat{d}_{i,j}(t) - C_{\delta/n}(T_{i,j}(t))$ and $U_{i,j}(t) := \hat{d}_{i,j}(t) + C_{\delta/n}(T_{i,j}(t))$. Even-Dar et al. (2006) use the above procedure to derive the following upper bound for the number of oracle queries used to find x_{j^*}:

$$\mathcal{O}\left(\sum_{k\neq j} \frac{\log(n^2/(\delta\Delta_{j,k}))}{\Delta_{j,k}^2}\right), \tag{5.4}$$

where for any $x_k \notin \{x_j, x_{j^*}\}$ the suboptimality gap $\Delta_{j,k} := d_{j,k} - d_{j,j^*}$ characterizes how hard it is to rule out x_k from being the nearest neighbor. We also set $\Delta_{j,j^*} := \min_{k \notin \{j,j^*\}} \Delta_{j,k}$. Note that one can use tighter confidence bounds as detailed in Garivier (2013) and Jamieson and Nowak (2014) to obtain sharper bounds on the sample complexity of this subroutine.

5.3.2 Computing the triangle bounds and active set $\mathcal{A}_j(t)$

Since $\mathcal{A}_j(\cdot)$ is only computed within `SETri`, we abuse notation and use its argument t to indicate the time counter private to `SETri`. Thus, the initial active set computed by `SETri` when called in the j^{th} round is denoted $\mathcal{A}_j(0)$. During the j^{th} round, the active set $\mathcal{A}_j(t)$ contains all points that have not been eliminated from being the nearest neighbor of x_j at time t. We define x_j's active set at time t as

$$\mathcal{A}_j(t) := \{a \neq j : \max\{L_{a,j}(t), L^{\triangle}_{a,j}(t)\} < \min_k \min\{U_{j,k}(t), U^{\triangle}_{j,k}(t)\}\}. \tag{5.5}$$

Assuming $L^{\triangle}_{a,j}(t)$ and $U^{\triangle}_{j,k}(t)$ are valid lower and upper bounds on $d_{a,j}, d_{j,k}$ respectively, (5.5) states that point x_a is active if its lower bound is less than the minimum upper bound for $d_{j,k}$ over all choices of $x_k \neq x_j$. Next, for any (j,k) we construct triangle bounds $L^{\triangle}, U^{\triangle}$ on the distance $d_{j,k}$. Intuitively, for some reals g, g', h, h', if $d_{i,j} \in [g, g']$ and $d_{i,k} \in [h, h']$ then $d_{j,k} \leqslant g' + h'$, and

$$d_{j,k} \geqslant |d_{i,j} - d_{i,k}| = \max\{d_{i,j}, d_{i,k}\} - \min\{d_{i,j}, d_{i,k}\} \geqslant (\max\{g,h\} - \min\{g', h'\})_+ \tag{5.6}$$

where $(s)_+ := \max\{s, 0\}$. The lower bound can be seen as true by Fig. 5.B.1 in the Appendix. Lemma 5.1 uses these ideas to form upper and lower bounds on distances by the triangle inequality.

Lemma 5.1. *For all* $k \neq 1$, *set* $U^{\triangle_1}_{1,k}(t) = U^{\triangle}_{1,k}(t) := U_{1,k}(t)$. *For any* $i < j$ *define*

$$U^{\triangle_i}_{j,k}(t) := \min_{\max\{i_1,i_2\}<i} (\min\{U_{i,j}(t), U^{\triangle_{i_1}}_{i,j}(t)\} + \min\{U_{i,k}(t), U^{\triangle_{i_2}}_{i,k}(t)\}). \tag{5.7}$$

For all $k \neq 1$, *set* $L_{1,k}^{\triangle_1}(t) = L_{1,k}^{\triangle}(t) := L_{1,k}(t)$. *For any* $i < j$ *define*

$$L_{j,k}^{\triangle_i}(t) := \max_{\max\{i_1,i_2,i_3,i_4\}<i} \Big(\max\{L_{i,j}(t), L_{i,j}^{\triangle_{i_1}}(t), L_{i,k}(t), L_{i,k}^{\triangle_{i_2}}(t)\} - \min\{U_{i,j}(t), U_{i,j}^{\triangle_{i_3}}(t), U_{i,k}(t), U_{i,k}^{\triangle_{i_4}}(t)\} \Big)_+, \tag{5.8}$$

where $(s)_+ := \max\{s, 0\}$. *If all the bounds obtained by* `SETri` *in rounds* $i < j$ *are correct then*

$$d_{j,k} \in \left[L_{j,k}^{\triangle}(t), U_{j,k}^{\triangle}(t)\right], \quad \textit{where} \quad L_{j,k}^{\triangle}(t) := \max_{i<j} L_{j,k}^{\triangle_i}(t) \quad \textit{and} \quad U_{j,k}^{\triangle}(t) := \min_{i<j} U_{j,k}^{\triangle_i}(t).$$

The proof is in Appendix 5.B.1. `ANNTri` has access to two sources of bounds on distances: concentration bounds and triangle inequality bounds, and as can be seen in Lemma 5.1, the former affects the latter. Furthermore, triangle bounds are computed from other triangle bounds, leading to the recursive definitions of $L_{j,k}^{\triangle_i}$ and $U_{j,k}^{\triangle_i}$. Because of these facts, triangle bounds are dependent on the order in which `ANNTri` finds each nearest neighbor. These bounds can be computed using dynamic programming and brute force search over all possible i_1, i_2, i_3, i_4 is not necessary. We note that the above recursion is similar to the Floyd-Warshall algorithm for finding shortest paths between all pairs of vertices in a weighted graph Floyd (1962); Cormen et al. (2009). The results in Singla et al. (2016) show that the triangle bounds obtained in this manner have the minimum L_1 norm between the upper and lower bound matrices.

5.4 Analysis

All omitted proofs of this section can be found in the Appendix Section 5.B.

Theorem 5.2. *`ANNTri` finds the nearest neighbor for each point in* $\mathcal{X}$ *with probability* $1 - \delta$.

5.4.1 A simplified algorithm

The following Lemma indicates which points must be eliminated initially in the j^{th} round.

Lemma 5.3. *If* $\exists i: \ 2U_{i,j} < L_{i,k}$, *then* $x_k \notin \mathcal{A}_j(0)$ *for* ***ANNTri****.*
Proof. $2U_{i,j} < L_{i,k} \iff U_{i,j} < L_{i,k} - U_{i,j} \leqslant L_{j,k}^{\triangle_i}$ □

Next, we define `ANNEasy`, a simplified version of `ANNTri` that is more amenable to analysis. Here, we say that x_k is eliminated in the j^{th} round of `ANNEasy` if i) k<j and $\exists i: U_{i,j} < L_{j,k}$ (symmetry from past samples) or ii) $\exists i: 2U_{i,j} < L_{i,k}$ (Lemma 5.3). Therefore, x_j's active set for `ANNEasy` is

$$\mathcal{A}_j = \{a{\neq}j : L_{a,k} \leqslant 2U_{j,k} \ \forall k \ \text{ and } L_{a,j} < \min_k U_{j,k}\}. \tag{5.9}$$

To define `ANNEasy` in code, we remove lines 3-8 of `ANNTri` (Algorithm 1), and call a subroutine `SEEasy` in place of `SETri`. `SEEasy` matches `SETri` (Algorithm 2) except that lines 1 and 9 are replaced with (5.9) instead. We provide full pseudocode of both `ANNEasy` and `SEEasy` in the Appendix 5.A.1.1. Though `ANNEasy` is a simplification for analysis, we note that it empirically captures much of the same behavior of `ANNTri`. In the Appendix 5.A.1.2 we provide an empirical comparison of the two.

5.4.2 Complexity of `ANNEasy`

We now turn our attention to account for the effect of the triangle inequality in `ANNEasy`.

Lemma 5.4. *For any* $x_k \in \mathcal{X}$ *if the following conditions hold for some* $i < j$, *then* $x_k \notin \mathcal{A}_j(0)$.

$$6C_{\delta/n}(1) \leqslant d_{i,k} - 2d_{i,j} \quad \textit{and} \quad \{j,k\} \cap (\cup_{m<i}\{\ell : 2d_{m,i} < d_{m,\ell}\}) = \emptyset. \tag{5.10}$$

The first condition characterizes which x_k's must satisfy the condition in Lemma 5.3 for the j^{th} round. The second guarantees that x_k was sampled in the i^{th} round, a necessary condition for forming triangle bounds using x_i.

Theorem 5.5. *Conditioned on the event that all confidence bounds are valid at all times,* `ANNEasy` *learns the nearest neighbor graph of $\mathcal{X}$ in the following number of calls to the distance oracle:*

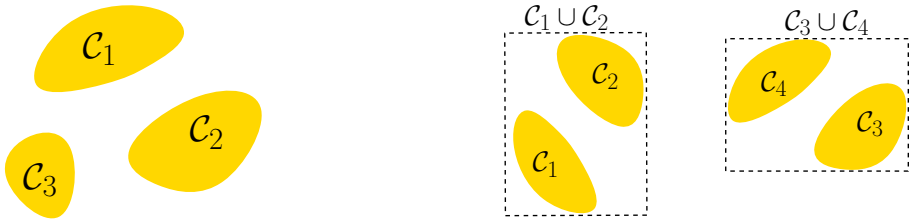
$$\mathcal{O}\left(\sum_{j=1}^{n}\sum_{k>j}\mathbb{1}_{[A_{j,k}]}H_{j,k}+\sum_{k<j}\mathbb{1}_{[A_{j,k}]}(H_{j,k}-\mathbb{1}_{[A_{k,j}]}H_{k,j})_{+}\right). \tag{5.11}$$

In the above expression $H_{j,k}:=\frac{\log(n^2/(\delta\Delta_{j,k}))}{\Delta_{j,k}^2}$ *and* $\mathbb{1}_{[A_{j,k}]}:=1$*, if* x_k *does not satisfy the triangle inequality elimination conditions of* (5.10) $\forall i<j$*, and* 0 *otherwise.*

In Theorem 5.13, in the Appendix, we state the sample complexity when triangle inequality bounds are ignored by `ANNTri`, and this upper bounds (5.11). Whether a point can be eliminated by the triangle inequality depends both on the underlying distances and the order in which `ANNTri` finds each nearest neighbor (*c.f.* Lemma 5.4). In general, this dependence on the order is necessary to ensure that past samples exist and may be used to form upper and lower bounds. Furthermore, it is worth noting that even without noise the triangle inequality may not always help. A simple example is any arrangement of points such that $0<r\leqslant d_{j,k}<2r\ \forall j,k$. To see this, consider triangle bounds on any distance $d_{j,k}$ due to any $x_i, x_{i'}\in\mathcal{X}\backslash\{x_j,x_k\}$. Then $|d_{i,j}-d_{i,k}|\leqslant r<2r\leqslant d_{i',j}+d_{i',k}\ \forall i,i'$ so $L_{i,j}^{\triangle}<U_{j,k}^{\triangle}\ \forall i,j,k$. Thus no triangle upper bounds separate from triangle lower bounds so no elimination via the triangle inequality occurs. In such cases, it is necessary to sample all $\mathcal{O}(n^2)$ distances. However, in more favorable settings where data may be split into clusters, the sample complexity can be much lower by using triangle inequality.

5.4.3 Adaptive gains via the triangle inequality

We highlight two settings where `ANNTri` provably achieves sample complexity better than $\mathcal{O}(n^2)$ independent of the order of the rounds. Consider a dataset containing c clusters of n/c points each as in Fig. 5.1a. Denote the m^{th} cluster as $\mathcal{C}_m$ and



(a) Clustered data (b) Hierarchical clusters

Figure 5.1: Example datasets where triangle inequalities lead to provable gains.

suppose the distances between the points are such that

$$\{x_k : d_{i,k} < 6C_{\delta/n}(1) + 2d_{i,j}\} \subseteq \mathcal{C}_m \ \forall i, j \in \mathcal{C}_m. \tag{5.12}$$

The above condition is ensured if the distance between any two points belonging to different clusters is at least a (δ, n)-dependent constant plus twice the diameter of any cluster.

Theorem 5.6. *Consider a dataset of $\sqrt{n}$ clusters which satisfy the condition in (5.12). Then `ANNEasy` learns the correct nearest neighbor graph of $\mathcal{X}$ with probability at least $1 - \delta$ in*

$$\mathcal{O}\left(n^{3/2}\overline{\Delta^{-2}}\right) \tag{5.13}$$

queries where $\overline{\Delta^{-2}} := \frac{1}{n^{3/2}} \sum_{i=1}^{\sqrt{n}} \sum_{j,k \in \mathcal{C}_i} \log(n^2/(\delta\Delta_{j,k}))\Delta_{j,k}^{-2}$ is the average number of samples distances between points in the same cluster.

By contrast, random sampling requires $\mathcal{O}(n^2\Delta_{\min}^{-2})$ where

$$\Delta_{\min}^{-2} := \min_{j,k} \log(n^2/(\delta\Delta_{j,k}))\Delta_{j,k}^{-2} \geqslant \overline{\Delta^{-2}}.$$

In fact, the value in (5.11) be be even lower if unions of clusters also satisfy (5.12). In this case, the triangle inequality can be used to separate *groups* of clusters. For example, in Fig. 5.1b, if $\mathcal{C}_1 \cup \mathcal{C}_2$ and $\mathcal{C}_3 \cup \mathcal{C}_4$ satisfy (5.12) along with $\mathcal{C}_1, \cdots, \mathcal{C}_4$, then the triangle inequality can separate $\mathcal{C}_1 \cup \mathcal{C}_2$ and $\mathcal{C}_3 \cup \mathcal{C}_4$. This process can be generalized to consider a dataset that can be split recursively into into subclusters

following a binary tree of k levels. At each level, the clusters are assumed to satisfy (5.12). We refer to such a dataset as *hierarchical in* (5.12).

Theorem 5.7. *Consider a dataset* $\mathcal{X} = \cup_{i=1}^{n/\nu} \mathcal{C}_i$ *of* n/ν *clusters of size* $\nu = \mathcal{O}(\log(n))$ *that is hierarchical in* (5.12). *Then* `ANNEasy` *learns the correct nearest neighbor graph of* $\mathcal{X}$ *with probability at least* $1 - \delta$ *in*

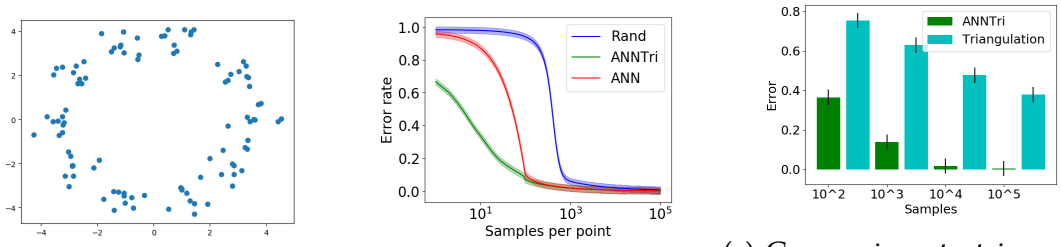
$$\mathcal{O}\left(n \log(n) \overline{\Delta^{-2}}\right) \tag{5.14}$$

queries where $\overline{\Delta^{-2}} := \frac{1}{n\nu} \sum_{i=1}^{n/\nu} \sum_{j,k \in \mathcal{C}_i} \log(n^2/(\delta \Delta_{j,k})) \Delta_{j,k}^{-2}$ *is the average number of samples distances between points in the same cluster.*

Expression (5.14) matches known lower bounds of $\mathcal{O}(n \log(n))$ on the sample complexity for learning the nearest neighbor graph from noiseless samples (Vaidya, 1989), the additional penalty of $\overline{\Delta^{-2}}$ is due to the effect of noise in the samples. In Appendix 5.C, we state the sample complexity in the average case, as opposed to the high probability statements above. The analog of the cluster condition (5.12) there does not involve constants and is solely in terms of pairwise distances (*c.f.* (5.33)).

5.5 Experiments

Here we evaluate the performance of `ANNTri` on simulated and real data. To construct the tightest possible confidence bounds for `SETri`, we use the law of the iterated logarithm as in Jamieson and Nowak (2014) with parameters $\epsilon = 0.7$ and $\delta = 0.1$. Our analysis bounds the number of queries made to the oracle. We visualize the performance by tracking the empirical *error rate* with the number of queries made per point. For a given point x_i, we say that a method makes an error at the t^{th} sample if it fails to return x_{i^*} as the nearest neighbor, that is, $x_{i^*} \neq \arg\min_j \hat{d}[i, j]$. Throughout, we will compare `ANNTri` against random sampling. Additionally, to highlight the effect of the triangle inequality, we will compare our method against the same active procedure, but ignoring triangle inequality bounds (referred to as



(a) Example embedding (b) Error curves (c) Comparison to triangulation

Figure 5.2: Comparison of `ANNTri` to `ANN` and `Random` for 10 clusters of 10 points separated by 10% of their diameter with $\sigma = 0.1$. `ANNTri` identifies clusters of nearby points more easily.

`ANN` in plots). All baselines may reuse samples via symmetry as well. We plot all curves with 95% confidence regions shaded.

5.5.1 Simulated Experiments

We test the effectiveness of our method, we generate an embedding of 10 clusters of 10 points spread around a circle such that each cluster is separated by at least 10% of its diameter in $\mathbb{R}^2$ as in shown in Fig. 5.2a. We consider Gaussian noise with $\sigma = 0.1$. In Fig. 5.2b, we present average error rates of `ANNTri`, `ANN`, and `Random` plotted on a log scale. `ANNTri` quickly learns x_{i^*} and has lower error with 0 samples due to initial elimination by the triangle inequality. The error curves are averaged over 4000 repetitions. All rounds were capped at 10^5 samples for efficiency.

5.5.1.1 Comparison to triangulation

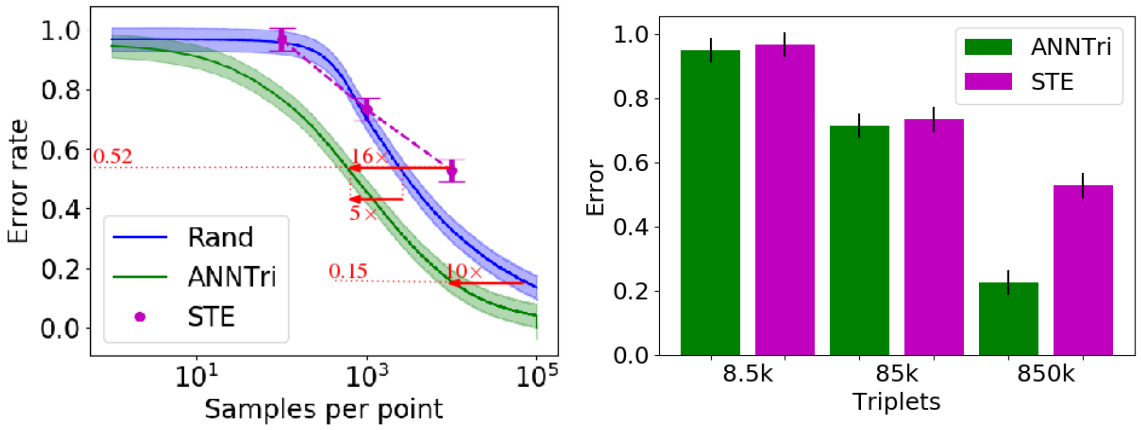
An alternative way a practitioner may use to obtain the nearest neighbor graph might be to estimate distances with respect to a few anchor points and then triangulate to learn the rest. Eriksson et al. (2010) provide a comprehensive example, and we summarize in Appendix 5.A.2 for completeness. The triangulation method is naïve for two reasons. First, it requires *much* stronger modeling assumptions than `ANNTri`— namely that the metric is Euclidean and the points are in a low-dimensional of known dimension. Forcing Euclidean structure can lead to

unpredictable errors if the underlying metric might not be Euclidean, such as in data from human judgments. Second, this procedure may be more noise sensitive because it estimates squared distances. In the example in Section 5.A.2, this leads to the additive noise being sub-exponential rather than subGaussian. In Fig. 5.2c, we show that even in a favorable setting where distances are truly sampled from a low-dimensional Euclidean embedding and pairwise distances between anchors are known exactly, triangulation still performs poorly compared to `ANNTri`. We consider the same 2-dimensional embedding of points as in Fig. 5.2a for a noise variance of $\sigma = 1$ and compare the `ANNTri` and triangulation for different numbers of samples.

5.5.2 Human judgment experiments

5.5.2.1 Setup

Here we consider the problem of learning from human judgments. For this experiment, we used a set $\mathcal{X}$ of 85 images of shoes drawn from the UT Zappos50k dataset Yu and Grauman (2014, 2017) and seek to learn which shoes are most visually similar. To do this, we consider queries of the form "between i, j, and k, which two are most similar?". We show example queries in Figs. 5.A.2a and 5.A.2b in the Appendix. Each query maps to a pair of triplet judgments of the form "is j or k more similar to i?". For instance, if i and j are chosen, then we may imply the judgments "i is more similar to j than to k" and "j is more similar to i than to k". We therefore construct these queries from a set of triplets collected from participants on Mechanical Turk by Heim et al. (2015). The set contains multiple samples of all $85\binom{84}{2}$ unique triples so that the probability of any triplet response can be estimated. We expect that i^* is most commonly selected as being more similar to i than any third point k. We take distance to correspond to the fraction of times that two images i, j are judged as being more similar to each other than a different pair in a triplet query (i, j, k). Let $E^j_{i,k}$ be the event that the pair i, k are chosen as most similar amongst i, j, and k. Accordingly, we define the 'distance' between images i



(a) Sample complexity gains

(b) Comparison to STE

Figure 5.3: Performance of `ANNTri` on the Zappos dataset. `ANNTri` achieves superior performance over STE in identifying nearest neighbors and has $5-10$x gains in sample efficiency over random.

and j as

$$d_{i,j} := \arg\min_{j \neq i} \mathbb{E}_{k \sim \text{Unif}(\mathcal{X} \setminus \{i,j\})} \mathbb{E}[\mathbb{1}_{E^j_{i,k}} | k]$$

where k is drawn uniformly from the remaining 83 images in $\mathcal{X}\setminus\{i,j\}$. For a fixed value of k,

$$\mathbb{E}[\mathbb{1}_{E^j_{i,k}} | k] = \mathbb{P}(E^j_{i,k}) = \mathbb{P}(\text{“}i \text{ more similar to } j \text{ than to } k\text{”}) \mathbb{P}(\text{“}j \text{ more similar to } i \text{ than to } k\text{”}).$$

where the probabilities are the empirical probabilities of the associated triplets in the dataset. This distance is a quasi-metric on our dataset as it does not always satisfy the triangle inequality; but satisfies it with a multiplicative constant: $d_{i,j} \leqslant 1.47(d_{i,k} + d_{j,k})\ \forall i, j, k$. Relaxing metrics to quasi-metrics has a rich history in the classical nearest neighbors literature Houle and Nett (2015); Tschopp et al. (2011); Goyal et al. (2008), and `ANNTri` can be trivially modified to handle quasi-metrics. However, we empirically note that $< 1\%$ of the distances violate the ordinary triangle inequality here so we ignore this point in our evaluation.

5.5.2.2 Results

When ANNTri or any baseline queries $Q(i, j)$ from the oracle, we randomly sample a third point $k \in \mathcal{X} \backslash \{i, j\}$ and flip a coin with probability $\mathbb{P}(E^j_{i,k})$. The resulting sample is an unbiased estimate of the distance between i and j. In Fig. 5.3a, we compare the error rate averaged over 1000 trials of ANNTri compared to Random and STE. We also plot associated gains in sample complexity by ANNTri. In particular, we see gains of $5 - 10$x over random sampling, and gains up to 16x relative to ordinal embedding. ANNTri also shows 2x gains over ANN in sample complexity (see Fig. 5.A.3 in Appendix).

Additionally, a standard way of learning from triplet data is to perform ordinal embedding. With a learned embedding, the nearest neighbor graph may easily be computed. In Fig. 5.3b, we compare ANNTri against the state of the art STE algorithm Van Der Maaten and Weinberger (2012) for estimating Euclidean embeddings from triplets, and select the embedding dimension of $d = 16$ via cross validation. To normalize the number of samples, we first perform ANNTri with a given max budget of samples and record the total number needed. Then we select a random set of triplets of the same size and learn an embedding in $\mathbb{R}^{16}$ via STE. We compare both methods on the fraction of nearest neighbors predicted correctly. On the x axis, we show the total number of triplets given to each method. For small dataset sizes, there is little difference, however, for larger dataset sizes, ANNTri significantly outperforms STE. Given that ANNTri is active, it is reasonable to wonder if STE would perform better with an actively sampled dataset, such as (Tamuz et al., 2011). Many of these methods are computationally intensive and lack empirical support (Jamieson et al., 2015), but we can embed using the full set of triplets to mitigate the effect of the subsampling procedure. Doing so, STE achieves 52% error, within the confidence bounds of the largest subsample shown in Fig. 5.3b. In particular, more data and more carefully selected datasets, may not correct for the bias induced by forcing Euclidean structure.

5.6 Conclusion

In this chapter, we solve the nearest neighbor graph problem by adaptively querying distances. Our method makes no assumptions beyond standard metric properties and is empirically shown to achieve sample complexity gains over passive sampling. In the case of clustered data, we show provable gains and achieve optimal rates in favorable settings.

5.A Additional experimental results and details

5.A.1 Differences between `ANNTri` and `ANNEasy`

5.A.1.1 Pseudocode for `ANNEasy` and `SEEasy`

We begin by providing pseudocode for both `ANNEasy` and `SEEasy` as described in Section 5.4.1 in Algorithms 3 and 4.

5.A.1.2 Empirical differences in performance for `ANNTri` and `ANNEasy`

In Figure 5.A.1 we compare the empirical performance of `ANNTri` and `ANNEasy`. We compare their performance in the same setting as Figure 5.2a with 10 clusters of 10 points separated by their at least 10% of their diameter. The curves are averaged over 4000 independent trials and plotted with 95% confidence regions. As is indicated in the plot, `ANNEasy` has similar behavior as `ANNTri`, but achieves slightly worse performance.

5.A.2 Triangulation

In this section, we provide a brief review of triangulation to estimate Euclidean embeddings, similar to the presentation in (Eriksson et al., 2010). The method is summarized as follows. Let $\mathcal{X}$ be a set of n points in Euclidean d space and

Algorithm 3 `ANNEasy`

Require: n, procedure `SEEasy`, 4, confidence δ

1: Initialize $\hat{d}$, T as $n \times n$ matrices of zeros, U as $n \times n$ matrix where each entry is ∞, L as $n \times n$ matrix where each entry is $-\infty$, NN as a length n array
2: **for** $j = 1$ to n **do**
3: $\quad$ NN[j] = `SEEasy`$(j, \hat{d}, U, L, T, \xi = \delta/n)$
4: **end for**
5: **return** The nearest neighbor graph adjacency list NN

Algorithm 4 `SEEasy`

Require: index j, callable oracle $Q(\cdot,\cdot)$ (5.1), 4 $n \times n$ matrices: $\hat{d}$, U, L, T, confidence ξ

Initialize the active set $\mathcal{A}_j \leftarrow \{a \neq j : L[a,k] \leqslant 2U[j,k]\ \forall k$ and $L[a,j] < \min_k U[j,k]\}$
while $|\mathcal{A}_j| > 1$ **do**
  for all [doonly query points with fewest samples$]i \in \mathcal{A}_j$ such that $T[i,j] = \min_{k\in\mathcal{A}_j} T[i,k]$
    Update $\hat{d}[i,j],\ \hat{d}[j,i] \leftarrow (\hat{d}[i,j] \cdot T[i,j] + Q(i,j))/(T[i,j]+1)$
    Update $T[i,j],\ T[j,i] \leftarrow T[i,j]+1$
    Update $U[i,j],\ U[j,i] \leftarrow \hat{d}[i,j] + C_\xi(T[i,j])$
    Update $L[i,j],\ L[j,i] \leftarrow \hat{d}[i,j] - C_\xi(T[i,j])$
  end for
  Update $\mathcal{A}_j \leftarrow \{a \neq j : L[a,k] \leqslant 2U[j,k]\ \forall k$ and $L[a,j] < \min_k U[a,k]\}$
end while
return The index i for which $x_i \in \mathcal{A}_j$

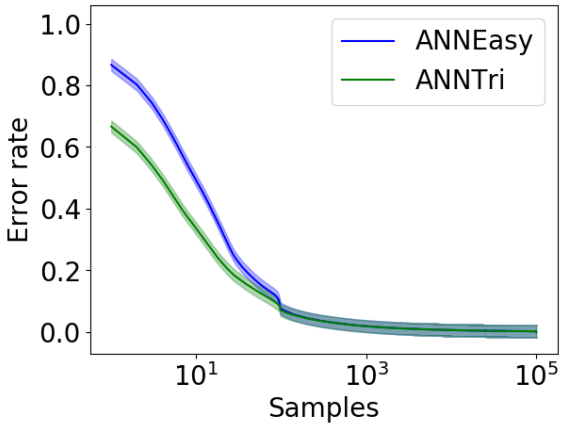


Figure 5.A.1: Comparison of error in identifying x_{i^*} `ANNTri` and the `ANNEasy` for 10 clusters of 10 points separated by 10% of their diameter with $\sigma = 0.1$.

$\mathbf{D}$ be the associated Euclidean distance matrix where each entry is the square of the associated Euclidean distance. Let A be a set of anchor points. Without loss of generality, we take $A := \{x_1, \cdots, x_{d+2}\}$. The $+2$ is to correct for the fact that Euclidean distance matrices have rank $d+2$. Let $\mathbf{A} := \mathbf{D}[1:d+2, 1:d+2]$ and

(a) An easy query

(b) A harder query

Figure 5.A.2: Two example zappos queries.

$\mathbf{L} := \mathbf{D}[1 : d+2, 1 : n]$. Then it can easily be verified that $\mathbf{D} = \mathbf{L}\mathbf{A}^{-1}\mathbf{L}^T$. To learn the entries in $\mathbf{L}$ (as well as $\mathbf{A}$), sample the distance from each of the n points to the $d+2$ anchors as many times as there is budget for and square the results. The empirical mean is a plugin estimator of the associated entry in $\mathbf{L}$ and $\mathbf{A}$, and we take $\widehat{\mathbf{L}}$ and $\widehat{\mathbf{A}}$ to be their unbiased estimates. Therefore $\widehat{\mathbf{D}} := \widehat{\mathbf{L}}\widehat{\mathbf{A}}^{-1}\widehat{\mathbf{L}}^T$ is an unbiased estimate of $\mathbf{D}$. With $\widehat{\mathbf{D}}$, the nearest neighbor graph can easily be computed.

5.A.3 Additional experimental results for Zappos dataset

In Fig. 5.A.2 we show two example queries of the form "which pair are most similar of these three?". Some queries are more straightforward whereas some are more subjective.

Additionally, in Fig. 5.A.3, we show the performance of `ANNTri`, `ANN`, and `Random` in identifying nearest neighbors from the Zappos data. In this setting, there is less of an advantage to using the triangle inequality due to the highly noisy and subjective nature of human judgments. Despite this, we still see a slight advantage to `ANNTri` over `ANN`. In particular, for moderate accuracy, there is a gain sample complexity of around 2x.

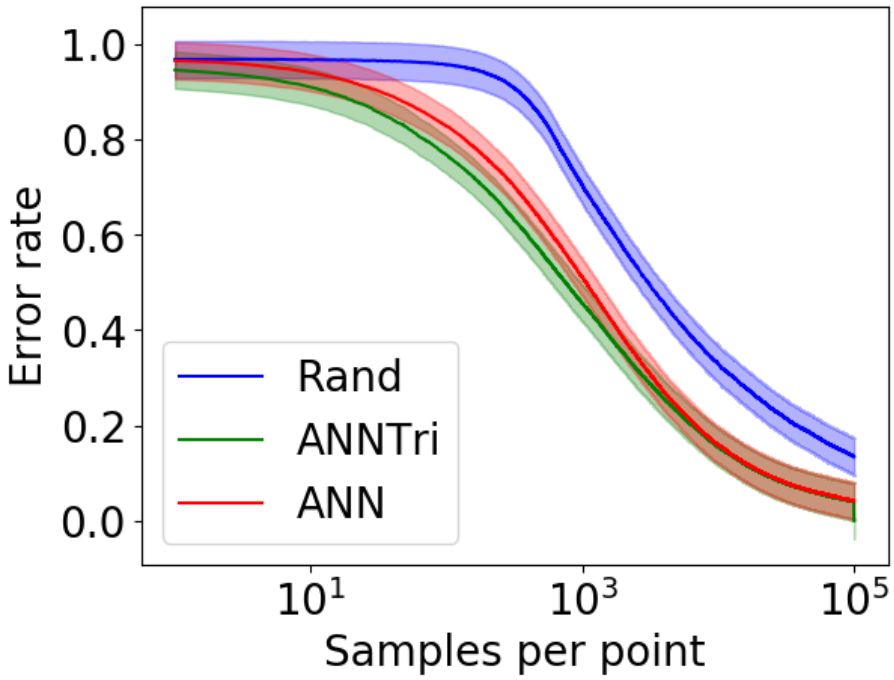


Figure 5.A.3: Error rates for nearest neighbor identification on Zappos Data

5.B Proofs and technical lemmas

5.B.1 Proof of Lemma 5.1

By symmetry for all $i < j$, we have existing samples of $Q(i,j)$ and $Q(i,k)$ and we use bounds based on these samples as well as past triangle inequality upper bounds on $d_{i,j}$ and $d_{i,k}$ due to $i_1 < i$ and $i_2 < i$ respectively. The upper bound is derived as follows:

$$d_{j,k} \leqslant d_{i,j} + d_{i,k} \leqslant \min\{U_{i,j}(t), U_{i,j}^{\triangle_{i_1}}(t)\} + \min\{U_{i,k}(t), U_{i,k}^{\triangle_{i_2}}(t)\} =: U_{j,k}^{\triangle_i}$$

Since we may form bounds based on all $i < j$ for which we have both samples of $Q(i,j)$ and $Q(i,k)$, we may optimize over i to get the tightest possible triangle inequality bounds on $d_{j,k}$.

Lower bounds are derived similarly. Again, intuitively, we may use past samples

of both $Q(i,j)$ and $Q(i,k)$ and associated bounds to derive a lower bound on $d_{j,k}$. The form is slightly more complicated here since we have to worry about both upper and lower bounds on $d_{i,j}$ and $d_{i,k}$. These bounds may either be from concentration bounds based on past samples directly or past triangle inequality upper and lower bounds on these distances due to points $i_1 - i_4 < i$.

$$\begin{aligned}
d_{j,k} \geqslant & |d_{i,j} - d_{i,k}| \\
= & \max\{d_{i,j}, d_{i,k}\} - \min\{d_{i,j}, d_{i,k}\} \\
\geqslant & (\max\{\max\{L_{i,j}(t), L_{i,j}^{\triangle_{i_1}}(t)\},\ \max\{L_{i,k}(t), L_{i,k}^{\triangle_{i_2}}(t)\}\} \\
& - \min\{\min\{U_{i,j}(t), U_{i,j}^{\triangle_{i_3}}(t)\},\ \min\{U_{i,k}(t), U_{i,k}^{\triangle_{i_4}}(t)\}\})_+ \\
= & (\max\{L_{i,j}(t), L_{i,j}^{\triangle_{i_1}}(t), L_{i,k}(t), L_{i,k}^{\triangle_{i_2}}(t)\} \\
& - \min\{U_{i,j}(t), U_{i,j}^{\triangle_{i_3}}(t), U_{i,k}(t), U_{i,k}^{\triangle_{i_4}}(t)\})_+
\end{aligned}$$

where $(s)_+ := \max\{s, 0\}$ and $i_1, i_2, i_3, i_4 < i$, (not necessarily unique) are chosen to optimize the bound. Similar to the upper bound, this holds with respect to any $i < j$ and we optimize over i. To ease presentation, let $UB'[i,j] := \min\{U_{i,j}, \min_{l<i} U_{i,j}^{\triangle_l}\}$ and $LB'[i,j] := \max\{L_{i,j}, \max_{l<i} L_{i,j}^{\triangle_l}\}$ be the tightest upper and lower bounds for $d_{i,j}$. For the lower bound, note that if the argument of $(\cdot)_+$ is negative, then any

$$\begin{aligned}
s \in & [\max\{LB'[i,j], LB'[i,k]\}, \min\{UB'[i,j], UB'[i,k]\}] \\
= & [LB'[i,j], UB'[i,j]] \cap [LB'[i,k], UB'[i,k]] \neq \emptyset
\end{aligned}$$

can be the value of both $d_{i,j}$ and $d_{j,k}$ as it lies in both their confidence intervals. Then points x_j, x_k can possibly be at the same location in the metric space, in which case $d_{j,k} = 0$. On the other hand if the RHS is positive, then x_j and x_k cannot be at the same location as $d_{i,j} \neq d_{i,k}$. In fact, the smallest possible value for $d_{j,k}$ occurs if x_i, x_j, x_k are collinear. This can be seen to be true from Figure 5.B.1. We finish with a quick lemma noting what can and cannot be eliminated via the triangle inequality.

Lemma 5.8. *Conditioned on the good event that all bounds are correct at all times, the*

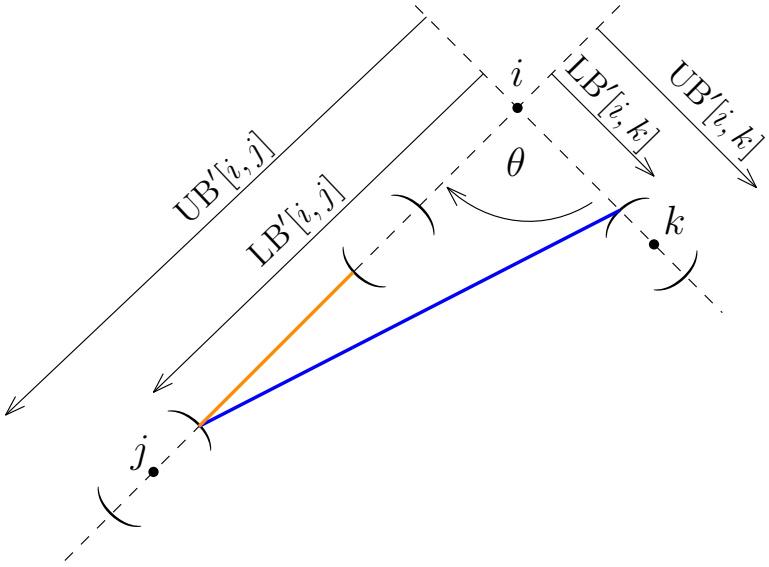


Figure 5.B.1: Pictorial justification for the lower bound in (5.6). True positions of points i, j, k are shown along with the upper and lower bounds for $d_{i,j}, d_{i,k}$ that are known to the algorithm. If the angle θ between $\vec{ij}$ and $\vec{ik}$ is known, the blue segment shows the lowest possible value for $d_{j,k}$ based on the bounds. The orange segment is the value in the RHS of (5.6). Without any information about θ, the three points could be collinear, in which case $d_{j,k}$ could equal the length of the orange segment.

triangle inequality cannot be used to to separate the two closest points to any given third point.

Proof. Consider finding x_{i^*}. Let $d_{i,i^*} \leqslant d_{i,j} \leqslant d_{i,k} \forall k \neq i^*, j$. By the triangle inequality, $d_{i,i^*} \leqslant d_{i,j} + d_{j,i^*}$ Clearly, the RHS is no smaller than $d_{i,j}$. Since we are conditioning on all bounds being correct at all times, no upper bound on d_{i,i^*} from the triangle inequality can ever be smaller that $d_{i,j}$. Rearranging the inequality, we see that $d_{i,i^*} - d_{j,i^*} \leqslant d_{i,j}$. The LHS is no larger than d_{i,i^*}, and d_{i,i^*} is the only distance wrt x_i that is smaller than $d_{i,j}$ by assumption. Therefore, no lower bound on $d_{i,j}$ due to the triangle inequality is greater than $d_{i,i^*} < d_{i,j}$. □

5.B.2 Helper Lemmas

Lemma 5.9. *Let* $t \in \mathbb{N}$ *index the rounds of the procedure* `SETri` *in finding* x_{i^*}*. Suppose all confidence intervals are valid, i.e.,* (5.3) *is true. Then* $\forall j \neq i$ *and all* t*,*

$$L_{i,j}(t) \geqslant d_{i,j} - 2C_{\delta/n}(T_{i,j}(t)) \quad \textit{and} \quad U_{i,j}(t) \leqslant d_{i,j} + 2C_{\delta/n}(T_{i,j}(t)). \tag{5.15}$$

Proof. If the good event (5.3) is true then for any pair (i, j) and time t we have

$$\hat{d}_{i,j}(t) < d_{i,j} + C_{\delta/n}(T_{i,j}(t)) \implies U_{i,j}(t) := \hat{d}_{i,j}(t) + C_{\delta/n}(T_{i,j}(t)) \leqslant d_{i,j} + 2C_{\delta/n}(T_{i,j}(t)).$$

A similar calculation can be done for $L_{i,j}(t)$ as well. □

Lemma 5.10. *Let* $j > i$*, and let* t_j *be the time when* x_j *is last sampled in the* i^{th} *round and equivalently for* t_k*. Assume without loss of generality that* $d_{i,j} < d_{i,k}$*. If* $d_{i,j}$ *and* $d_{i,k}$ *are such that*

$$4C_{\delta/n}(T_{i,j}(t_j)) + 2C_{\delta/n}(T_{i,k}(t_k)) \leqslant d_{i,k} - 2d_{i,j} \tag{5.16}$$

then `SETri` *can eliminate* $d_{j,k}$ *without sampling it, i.e.,* $x_k \notin \mathcal{A}_j(0)$*.*

Proof. Focusing on the number of $Q(i, j)$ queries, we have that

$$2U_{i,j}(t_j) = 2(\hat{d}_{i,j}(t_j) + C_{\delta/n}(T_{i,j}(t_j))) \leqslant 2(d_{i,j} + 2C_{\delta/n}(T_{i,j}(t_j))), \tag{5.17}$$

the inequality in (5.17) is due to Lemma 5.9, and using the number of $Q(i, k)$ queries,

$$L_{i,k}(t_k) \geqslant \hat{d}_{i,k}(t_k) - C_{\delta/n}(T_{i,k}(t_k)) \geqslant d_{i,k} - 2C_{\delta/n}(T_{i,k}(t_k)). \tag{5.18}$$

The first inequality in (5.18) is because if $k < j$ then there may have been more $Q(k, i)$ queries beyond the t_k number of $Q(i, k)$ queries made while finding x_{i^*}. Rearranging the equation in the Lemma statement,

$$2d_{i,j} + 4C_{\delta/n}(T_{i,j}(t_j)) \leqslant d_{i,k} - 2C_{\delta/n}(T_{i,k}(t_k)),$$

which implies that $2U_{i,j} \leqslant L_{i,k}$ from (5.17), (5.18). Hence from Lemma 5.3 $x_k \notin \mathcal{A}_j(0)$. □

Lemma 5.11. *There exists a dataset $\mathcal{P}$ containing 2ν points such that for all $x_p \in \mathcal{P}$ and $\alpha > 0$ the set of suboptimality gaps $\Delta_{p,p'}$ is*

$$\left\{1 - \left(\frac{s-1}{\nu-1}\right)^{\alpha} : s \in \{1, 2, \ldots, \nu-1\}\right\}. \tag{5.19}$$

Proof. Note that there are $\nu - 1$ values given in (5.19) while there are $2\nu - 2$ points in the cluster, excluding x_p and x_{p^*}. Each value in (5.19) is the suboptimality gap for two distinct points in $\mathcal{P} \setminus \{x_p, x_{p^*}\}$. We can construct such a dataset $\mathcal{P}$ in the following manner.

We index these points as $p, p_1, p_2, \ldots, p_{2\nu-1}$. Suppose the pairwise distance values are such that

$$\begin{gathered} d_{p,p_1} > d_{p,p_2} > \cdots > d_{p,p_{\nu-1}} > d_{p,p_\nu} =: d_{p,p^*}, \text{ and } d_{p,p_{\nu+1}} < d_{p,p_{\nu+2}} < \cdots < d_{p,p_{2\nu-1}} \\ \text{such that} \\ \forall s \in \{1, 2, \ldots, \nu-1\} \text{ we have that } d_{p,p_{\nu-s}} = d_{p,p_{\nu+s}} \implies d_{p,p_i} = d_{p,p_{2\nu-i}}. \end{gathered} \tag{5.20}$$

We can then construct a $2\nu \times 2\nu$ distance matrix D in the following manner. The first row of D is

$$D[0,:] := \begin{bmatrix} 0 & d_{p,p_1} & d_{p,p_2} & \cdots & d_{p,p_{\nu-1}} & d_{p,p^*} & d_{p,p_{\nu+1}} & \cdots & d_{p,p_{2\nu-2}} & d_{p,p_{2\nu-1}} \end{bmatrix}.$$

The ith row of D is obtained by carrying out i circular shifts on the initial row $D[0,:]$ shown above. Thus D is a circulant matrix and we can see $D[i,j]$ and $D[j,i]$ to be as follows.

$$D[i,j] = \begin{cases} d_{p,p_{j-i}} & \text{if } j > i \\ d_{p,p_{2\nu-(i-j)}} & \text{if } j < i, \end{cases} \quad \text{and} \quad D[j,i] = \begin{cases} d_{p,p_{i-j}} & \text{if } i > j \\ d_{p,p_{2\nu-(j-i)}} & \text{if } i < j. \end{cases}$$

Then using (5.20) we have that $D[i,j] = D[j,i]$ for all $i \neq j$ and the diagonal entries are all 0. Thus D is symmetric. In addition, the distance values of the points to any point in the cluster take the same set of values. Suppose $d_{p,p^*} =: r > 0$ and $d_{p,p_1} = 2r$. Choose an $\alpha > 0$ and let

$$d_{p,p_{2\nu-i}} = d_{p,p_i} := r\left(2 - \left(\frac{s-1}{\nu-1}\right)^{\alpha}\right), \ \forall s \in \{1,2,\ldots,\nu-1\}.$$

Then $D[i,j] \leqslant D[i,k] + D[k,j]$ for any three distinct i,j,k as the sum of any two elements is greater than $2r$, which is the largest element in D. Thus the distance values in D satisfy the triangle inequality, and D is a valid distance matrix. The suboptimality gaps for any point in the cluster is $\Delta_{p,p_i} = d_{p,p_i} - d_{p,p^*} = r(1 - ({}^{(i-1)}/_{(\nu-1)})^{\alpha})$, choosing $r = 1$ finishes the required construction. □

5.B.3 Proof of Theorem 5.2

Proof. `ANNTri` makes an error in finding the nearest neighbor for some point with probability $\mathbb{P}(\texttt{SETri}$ is wrong for some $x_j, j \in \{1,2,\ldots,n\})$. We show that probability is at most $n\xi = \delta$, where the confidence level ξ for each execution of `SETri` is set to be δ/n. We use induction on $s \in \mathbb{N}$ to obtain that

$$\mathbb{P}(\forall j \in \{1,2,\ldots,s\}, k \neq j, \max\{L_{j,k}(t), L^{\triangle}_{j,k}(t)\} \leqslant d_{j,k} \leqslant \min\{U_{j,k}(t), U^{\triangle}_{j,k}(t)\}) \geqslant 1 - s\xi. \tag{5.21}$$

Consider the base case, i.e., point x_1. From the initialization of `ANNTri` 1,

$$\min\{U_{1,k}(t), U^{\triangle}_{1,k}(t)\} = U_{1,k}(t), \min\{L_{1,k}(t), L^{\triangle}_{1,k}(t)\} = L_{1,k}(t)$$

for all $k \neq 1$. Using (5.3) we have $L_{1,k}(t) \leqslant d_{1,k} \leqslant U_{1,k}(t)$ with probability $1 - \delta/n$, and since ξ is δ/n the base case is true. Assume the hypothesis (5.21) is true for some s. We show that it is true for $s+1$ as well. We can bound the error event as

follows.

$$\mathbb{P}(\exists j \in \{1,\ldots,s+1\}, k \neq j : d_{j,k} \notin [\max\{L_{j,k}(t), L^{\triangle}_{j,k}(t)\}, \min\{U_{j,k}(t), U^{\triangle}_{j,k}(t)\}]) \tag{5.22}$$

$$\begin{aligned} &= \mathbb{P}(\exists j \in \{1,\ldots,s\}, k \neq j : d_{j,k} \notin [\max\{L_{j,k}(t), L^{\triangle}_{j,k}(t)\}, \min\{U_{j,k}(t), U^{\triangle}_{j,k}(t)\}]) \\ &\quad + \mathbb{P}\Big(\{k \neq s+1 : d_{s+1,k} \notin [\max\{L_{s+1,k}(t), L^{\triangle}_{s+1,k}(t)\}, \min\{U_{s+1,k}(t), U^{\triangle}_{s+1,k}(t)\}]\} \\ &\qquad \cap \{\forall j \in \{1,2,\ldots,s\}, k \neq j, \max\{L_{j,k}(t), L^{\triangle}_{j,k}(t)\} \leqslant d_{j,k} \leqslant \min\{U_{j,k}(t), U^{\triangle}_{j,k}(t)\}\}\Big) \end{aligned}$$

From (5.21) the first summand in the RHS of (5.22) is at most $s\xi$. In the event corresponding to the second term, all the bounds used by `SETri` for $d_{j,k}, j \leqslant s, k \neq j$ are correct. Since $U^{\triangle}_{s+1,\cdot}$ and $L^{\triangle}_{s+1,\cdot}$ are both deterministically obtained (see (5.7), (5.8)) from them, they are correct as well. Thus

$$\begin{aligned} &\mathbb{P}(\max\{L_{s+1,k}(t), L^{\triangle}_{s+1,k}(t)\} \leqslant d_{s+1,k} \leqslant \min\{U_{s+1,k}(t), U^{\triangle}_{s+1,k}(t)\}) \\ &= \mathbb{P}(L_{s+1,k}(t) \leqslant d_{s+1,k} \leqslant U_{s+1,k}(t)) \geqslant 1-\xi. \end{aligned}$$

Hence the second summand in the RHS of (5.22) is at most ξ. This proves (5.21) for $s+1$ and completes the induction.

Thus with probability $1 - n\xi = 1 - \delta$, the bounds obtained by `SETri` for finding $x_{j^*}, j \in \{1,\ldots,n\}$ are all correct. We show that `ANNTri` correctly finds all nearest neighbors if the bounds are correct. For if not, suppose `SETri` returns the wrong nearest neighbor of x_j which happens only if x_{j^*} is not the last point in the active set. $x_{j^*} \notin \mathcal{A}$ because some other point $x_k \in \mathcal{A}$ eliminates it. Then $d_{j,k} < \min\{U_{j,k}, U^{\triangle}_{j,k}\} < \max\{L_{j,j^*}, L^{\triangle}_{j,j^*}\} < d_{j,j^*}$, which contradicts the fact that j^* is the nearest neighbor. □

5.B.4 Proof of Lemma 5.4

Proof. Consider a point $x_i, i < j$ which satisfies the first part of (5.10). If $x_j \in \mathcal{A}_i(0)$ and $x_k \in \mathcal{A}_i(0)$, then neither x_j and x_k were eliminated without sampling when

SEEasyi was called for x_i and hence $T_{i,j} \geqslant 1$ and $T_{i,k} \geqslant 1$. Then we have that

$$4C_{\delta/n}(T_{i,j}(t_j)) + 2C_{\delta/n}(T_{i,k}(t_k)) \leqslant 6C_{\delta/n}(1) \leqslant d_{i,k} - 2d_{i,j}$$

and $x_k \notin \mathcal{A}_j(0)$ by Lemma 5.10. The second part of (5.10) ensures that $\{x_j, x_k\} \subseteq \mathcal{A}_i(0)$ as shown next. The points eliminated from being the nearest neighbor of x_i using triangle inequality are $\mathcal{A}_i(0)^\complement = \cup_{m<i}\{\ell : 2U_{m,i} < L_{m,\ell}\}$. If the bounds obtained by SEEasy for all $m < i$ are correct,

$$\{\ell : 2U_{m,i} < L_{m,\ell}\} \subseteq \{\ell : 2d_{m,i} < d_{m,\ell}\} \implies \mathcal{A}_i(0)^\complement \subseteq \cup_{m<i}\{\ell : 2d_{m,i} < d_{m,\ell}\}.$$

Hence if the second condition of (5.10) is satisfied, then $\{j, k\} \subseteq \mathcal{A}_i(0)$ and we are done. □

5.B.5 Proof of Theorem 5.5

Proof. Let x_j be the point on which SEEasy is called. Consider the case $j < k$. If $\mathbb{1}_{[A_{j,k}]} = 0$ then $x_k \notin \mathcal{A}_j(0)$ and no $Q(j,k)$ queries are made. Otherwise, x_k can be in the active set and from (5.4) at most $H_{j,k}$ samples of $d_{j,k}$ are taken. Now consider the case $k < j$. Samples of $d_{j,k}$ are only queried if $x_k \in \mathcal{A}_j(0)$. If $x_j \notin \mathcal{A}_k(0)$, i.e., x_j was eliminated when SEEasy was called for x_k then no $Q(k,j)$ queries made at that round. Again from (5.4) at most $H_{j,k}$ samples of $d_{j,k}$ are taken by SEEasy while finding x_{j^*}. If however $\mathbb{1}_{[A_{k,j}]} = 1$, then $Q(k,j)$ queries were made while finding x_{k^*} and let the number of those samples be $\#Q(k,j)$. Because of the sampling procedure of SEEasy, at most $(H_{j,k} - \#Q(k,j))_+$ queries are made for $d_{j,k}$. The total number of $Q(j,k)$ and $Q(k,j)$ queries is $\max\{H_{j,k}, \#Q(k,j)\}$, and since $\#Q(k,j) \leqslant H_{k,j}$, we get the result. □

5.B.6 Details for Section 5.4.3

In this section, we consider a case where ANNTri achieves complexity that scales like $O(n^{1.5})$ as well as $O(n\log(n))$, the known optimal rate for the all nearest neighbors

problem for noiseless data. To do this, we first prove a lemma about the complexity of learning with clustered data. In particular, we show that if the data comes from two well separated clusters, then the complexity of learning the nearest neighbor graph can be bounded as the complexity of learning the nearest neighbors of two points looking at the full dataset and the complexity of learning the remaining nearest neighbors graphs on each of the clusters.

Lemma 5.12. *Consider $\mathcal{X} = \mathcal{C}_1 \cup \mathcal{C}_2$ where $\mathcal{C}_1$ and $\mathcal{C}_2$ both satisfy 5.12 for all i, j. Then* `ANNEasy` *learns the nearest neighbor graph of $\mathcal{X}$ with probability at least $1 - \delta$ in at most*

$$\mathcal{O}\left(|\mathcal{C}_1| + |\mathcal{C}_2| + \mathcal{H}_{\mathcal{C}_1} + \mathcal{H}_{\mathcal{C}_2}\right) \tag{5.23}$$

samples independent of the order in which it finds nearest neighbors where $\mathcal{H}_{\mathcal{C}_i}$ denotes the complexity of learning the nearest neighbor graph of cluster $\mathcal{C}_i$ as bounded by 5.13.

The above lemma implies that for the first point explored in each cluster, it is necessary to look at all other points in the dataset, but for all other points, it is only necessary to search within that point's respective cluster.

Proof. Choose a random order of points and fix it. Without loss of generality, we assume that $x_1 \in \mathcal{C}_1$.Let j_2 be the first point visited in C_2. Throughout, we will ignore reused samples since they only contribute at most a factor of 2 to the sample complexity as can be seen by Theorems 5.5 and 5.13 and we seek an upper bound. Via standard analysis for successive elimination, x_{1^*} can be be found in $\mathcal{O}\left(\sum_{j=2}^{n} H_{1,j}\right) = |\mathcal{C}_2| + \mathcal{O}\left(\sum_{j \in \mathcal{C}_1 \setminus \{x_1\}} H_{1,j}\right)$ samples with probability at least $1 - \delta/n$. For all $i = 2, \cdots, j_2 - 1$,

$$\mathcal{A}_i(0)^c \supset \{\mathcal{A}_1(0) \cap \{k : d_{i,k} \geqslant 6d_{i,j} - 3d_{1,1^*}\}\} \supset \{\mathcal{X} \setminus \{x_1\} \cap \mathcal{C}_2\} = \mathcal{C}_2$$

which implies that $x_{i^*} \in \mathcal{C}_1$. For x_{j_2} we may trivially say that $\mathcal{A}_{j_2}(0)^c \supset \{j_2\}$ so $x_{j_2^*}$ can be learned in $\mathcal{O}\left(\sum_{l \neq j_2} H_{i,j}\right) = |\mathcal{C}_1| + \mathcal{O}\left(\sum_{j \in \mathcal{C}_2 \setminus \{x_{j_2}\}} H_{j_2,j}\right)$ samples with probability at least $1 - \delta/n$. We conclude by showing that for all remaining x_i, if $x_i \in \mathcal{C}_1$, then $\mathcal{A}_i(0) \subset \mathcal{C}_1$ and if $x_i \in \mathcal{C}_2$, then $\mathcal{A}_i(0) \subset \mathcal{C}_2$. Consider the case that $x_1 \in \mathcal{C}_1$. Suppose

that $\exists x_j \in \mathcal{A}_i(0) \cap \mathcal{C}_2$. Then $2U_{1,i} > L_{1,j}$.

$$U_{i,1} = \hat{d}_{1,i} + C_{\delta/n}(T_{1,i}) \leqslant d_{1,i} + 2C_{\delta/n}(T_{1,i}) = d_{1,i} + 2C_{\delta/n}(1)$$

where the first inequality holds by 5.9. Similarly,

$$L_{1,j} = \hat{d}_{1,j} - C_{\delta/n}(T_{1,j}) \geqslant d_{1,j} - 2C_{\delta/n}(T_{1,j}) \geqslant d_{1,j} - 2C_{\delta/n}(1)$$

Then $2(d_{1,i} + 2C_{\delta/n}(1)) \geqslant 2U_{1,i} > L_{1,j} \geqslant d_{1,j} - 2C_{\delta/n}(1) \implies d_{1,j} < 2d_{1,i} + 6C_{\delta/n}(1) \implies j \in \mathcal{C}_1$ which is a contradiction. A similar proof holds for $x_i \in \mathcal{C}_2$. It remains to argue that j_2 can be any number between 2 (by assumption that $x_1 \in \mathcal{C}_1$) and $|\mathcal{C}_1| + 1$ without affecting the bound on the complexity. By the assumption that $\mathcal{C}_1$ and $\mathcal{C}_2$ satisfy 5.12, out of cluster points can be eliminated in a single sample. Therefore, for any j_2, $\sum_{l \in \mathcal{C}_1} H_{j_2,l} = |\mathcal{C}_1|$. Then we have that the total complexity is $\mathcal{O}\left(|\mathcal{C}_1| + |\mathcal{C}_2| + \mathcal{H}_{\mathcal{C}_1} + \mathcal{H}_{\mathcal{C}_2}\right)\ \forall j_2$. Since we have considered general orders of finding each nearest neighbor, we are done. □

5.B.6.1 Proof of Theorem 5.6

Proof. By assumption, the dataset $\mathcal{X} = \cup_{i=1}^{c} \mathcal{C}_i$ with each cluster satisfies Equation 5.12. Therefore, for all m, $\mathcal{X} = \mathcal{C}_m \cup (\cup_{j \neq m} \mathcal{C}_i)$. By applying Lemma 5.12, iteratively, we bound the complexity in terms of the the complexity of learning the nearest neighbor graph of $\mathcal{C}_m$, the complexity of learning the nearest neighbor graph of $\cup_{j \neq m} \mathcal{C}_i$, and an additive penalty of n which accounts for the samples taken between the two. Since $\mathcal{X}$ is a union of c clusters, this process may repeat c times. Therefore the total complexity can be bounded as

$$\mathcal{O}\left(cn + \sum_{i=1}^{c} \sum_{j,k \in \mathcal{C}_i} H_{j,k}\right)$$

Taking $c = \sqrt{n}$, we see that the above sum is $\mathcal{O}\left(n^{1.5}\overline{\Delta^{-2}}\right)$ where

$$\overline{\Delta^{-2}} = \frac{1}{c * n} \sum_{i=1}^{c} \sum_{j,k \in \mathcal{C}_i} H_{j,k}$$

is the average number of times intra-cluster distances are sampled. By contrast, the complexity for random sampling is $\mathcal{O}(n^2\Delta_{\min}^{-2})$ where $\Delta_{\min}^{-2} := \min_{j,k} H_{j,k}$. Comparing the two, we see that the latter is larger by at least a factor of $\mathcal{O}(\sqrt{n})$. □

5.B.6.2 Proof of Lemma 5.7

Next we use Lemma 5.12 to show that for datasets such that the clusters nest, we can achieve complexity scaling in $\mathcal{O}(n \log(n)\overline{\Delta^{-2}})$. In particular, we will recursively apply Lemma 5.12 to show that clusters can be broken into subclusters and initial active sets shrink in diadic splits.

Proof. Before we prove the theorem, we begin by introducing some notation to make this proof concise. Recall that we have assumed that $\mathcal{X}$ can be written as a hierarchy of clusters and sub clusters that form a balanced tree. We will denote the root of the tree with the full dataset as the 0^{th} level and each split in that level with be indexed by $i = 1, \cdots, 2^\ell$ where $\ell = 0, \cdots, \log(n/\nu) - 1$ denotes the level. For notational ease, we take $\mathcal{C}_{0,1} \equiv \mathcal{X}$. $\mathcal{C}_{\ell,i}$ denotes the i^{th} cluster at the ℓ^{th} level of the tree which may be split into subclusters if $\ell < \log(n/\nu) - 1$. The idea will be to traverse the tree and split clusters into subclusters while keeping track of the number of between cluster samples that were be necessary due to the bound in Lemma 5.12. We let $\mathcal{H}_{\mathcal{C}_{\ell,i}}$ denote complexity of learning the nearest neighbor graph of $\mathcal{C}_{\ell,i}$.

Randomize the order and fix it. We will proceed by recursively applying Lemma 5.12 to bound the complexity of learning the full nearest neighbor graph of a cluster in terms of learning it for each subcluster plus an additive penalty. By Lemma 5.12 the complexity of finding the nearest neighbor graph of $\mathcal{X}$ can be upper

bounded as

$$\mathcal{O}\left(|\mathcal{C}_{1,1}| + |\mathcal{C}_{1,2}| + \mathcal{H}_{\mathcal{C}_{1,1}} + \mathcal{H}_{\mathcal{C}_{1,2}}\right) = \mathcal{O}\left(n + \mathcal{H}_{\mathcal{C}_{1,1}} + \mathcal{H}_{\mathcal{C}_{1,2}}\right).$$

We may again apply Lemma 5.12 to $\mathcal{C}_{1,1}$ and $\mathcal{C}_{1,2}$. to bound their complexities as $\mathcal{O}\left(\frac{n}{2} + \mathcal{H}_{\mathcal{C}_{2,1}} + \mathcal{H}_{\mathcal{C}_{2,2}}\right)$ and $\mathcal{O}\left(\frac{n}{2} + \mathcal{H}_{\mathcal{C}_{2,3}} + \mathcal{H}_{\mathcal{C}_{2,4}}\right)$ respectively where $\mathcal{C}_{1,1} = \mathcal{C}_{2,1} \cup \mathcal{C}_{2,2}$ and $\mathcal{C}_{1,2} = \mathcal{C}_{2,3} \cup \mathcal{C}_{2,4}$. Therefore, similar to the above level, the total additive penalty for samples between clusters is n for the level. We may continue this process of splitting and paying the penalty of $n/2^\ell \times 2^\ell$ between cluster samples down to the bottom level $\ell = \log(n/\nu)$ with clusters of size ν.

Therefore, we may write the complexity as

$$\mathcal{O}\left(n \log\left(\frac{n}{\nu}\right) + \sum_{i=1}^{n/\nu} \sum_{j,k \in \mathcal{C}_{\log(n/\nu),i}} H_{j,k}\right). \tag{5.24}$$

Ignoring logarithmic factors, each complexity term $H_{j,k}$ is of the order $\mathcal{O}(\Delta_{j,k}^{-2})$. Therefore the entire summation is of the order

$$\mathcal{O}\left(n \log\left(\frac{n}{\nu}\right) + n\nu\overline{\Delta^{-2}}\right)$$

where $\overline{\Delta^{-2}} := \frac{1}{n\nu}\sum_{i=1}^{n/\nu}\sum_{j,k\in\mathcal{C}_i} \log(n^2/(\delta\Delta_{j,k}))\Delta_{j,k}^{-2}$ is the average complexity. Recalling that $\nu = \mathcal{O}(\log(n))$, we are done. □

5.B.7 Sample complexity without using triangle inequality

Theorem 5.13. *With probability $1-\delta$, the number of oracle queries made by `ANNTri` and `ANNEasy` if all triangle bounds are ignored is at most*

$$\mathcal{O}\left(\sum_{i<j} \max\left\{\frac{\log(n^2/(\delta\Delta_{i,j}))}{\Delta_{i,j}^2}, \frac{\log(n^2/(\delta\Delta_{j,i}))}{\Delta_{j,i}^2}\right\}\right). \tag{5.25}$$

In the experiments, the process of using `ANNTri` and ignoring triangle inequality

bounds is referred to as ANN.

Proof. In the case that triange bounds are ignored, ANNTri and ANNEasy are the same. Consider the i^{th} round where we seek to identify x_{i^*} with probability $1-\delta/n$. ANNTri has found x_{ℓ^*} for all $\ell < i$, in particular, it has evaluated $\hat{d}_{\ell,i}, U_{\ell,i}, L_{\ell,i}$. For every $x_j \neq x_{i^*}, x_j \in \mathcal{A}_i(0)$, we can bound the number of $Q(i,j)$ queries in the following manner. Suppose $j > i$ and $i^* > i$, so that at the beginning of the i^{th} round we have that $T_{i,j} = T_{i,i^*} = 0$. From (5.3), with probability $1-\delta/n$, x_{i^*} is the last point in the active set. The point x_j is eliminated from the active set at time t_j if the following is true.

$$\begin{aligned} U_{i,i^*}(t_j) \overset{(a)}{\leqslant} d_{i,i^*} + 2C_{\delta/n}(T_{i^*}(t_j)) < d_{i,j} - 2C_{\delta/n}(T_j(t_j)) \overset{(b)}{\leqslant} L_{i,j}(t_j), \\ \implies 4C_{\delta/n}(t_j) < d_{i,j} - d_{i,i^*} = \Delta_{i,j}. \end{aligned} \tag{5.26}$$

Inequalities (a), (b) are due to Lemma 5.9, and the fact that if j is eliminated at time t_j, then $T_{i,j}(t_j) = t_j$. From the property of the $C_{\delta/n}(\cdot)$ function, (5.26) is ensured when the number of samples of $d_{i,j}$ is

$$t_j \leqslant \left\lceil \kappa \frac{\log(n^2/(\delta\Delta_{i,j}/4))}{(\Delta_{i,j}/4)^2} \right\rceil.$$

We now consider the cases when at least one of i^*, j are less than i.

$i^* > i, j < i$: In this case, at the beginning of the i^{th} round $T_{i,j}$ is equal to the number of $Q(j,i)$ queries made (denoted as $\#Q(j,i)$) while finding x_{j^*}:

$$\#Q(j,i) \leqslant \left\lceil \kappa \frac{\log(n^2/(\delta\Delta_{j,i}/4))}{(\Delta_{j,i}/4)^2} \right\rceil.$$

If $\#Q(j,i) > t_j$, then no further $Q(i,j)$ queries are made in the i^{th} round, as argued next. Because the sampling procedure of SETri queries all points who have the minimum number of samples at current time, if a query $Q(i,j)$ is made at time $t+1$,

that implies $T_{i,i^*}(t) = \#Q(j,i)$. But then j is not in the active set at time t as

$$U_{i,i^*}(\#Q(j,i)) < U_{i,i^*}(t_j) < d_{i,j} - 2C_{\delta/n}(t_j) < d_{i,j} - 2C_{\delta/n}(\#Q(j,i)) = L_{i,j}(\#Q(j,i))$$

and hence $Q(i,j)$ is not made. If $\#Q(j,i) < t_j$, then x_j is eliminated when $t_j - \#Q(j,i)$ more samples of $d_{i,j}$ have been queried. Thus the total number of samples of $d_{i,j}$ is at most $\max\{t_j, \#Q(j,i)\}$.

The other two cases of 1) $i^* < i, j > i$, and 2) $i^* < i, j < i$ can be handled similarly. □

5.C Average case performance of `ANNEasy`

We can obtain a different expression for the number of oracle queries if all the random quantities during a run of the algorithm take their expected values. In particular, Lemma 5.4 can be relaxed to the following.

Lemma 5.14. *If all bounds obtained by* `SEEasy` *are correct and all the random quantities take their expected values, then for some* $i < j$ *such that* $x_j \neq x_{i^*} \neq x_k$ *if we have that*

$$d_{i,k} > 6d_{i,j} - 3d_{i,i^*}, \quad \textit{and} \quad \{j,k\} \cap (\cup_{m<i}\{\ell : 2d_{m,i} < d_{m,\ell}\}) = \emptyset, \tag{5.27}$$

then $2U_{i,j} < L_{i,k}$ *and hence* $x_k \notin \mathcal{A}_j(0)$.

Proof. In the good event, the point x_{i^*} is the last element in the active set $\mathcal{A}_i$ and points x_j, x_k have been eliminated from $\mathcal{A}_i$ at some prior times t_j, t_k respectively. Both $t_j > 0$ and $t_k > 0$ as $\{x_j, x_k\} \subset \mathcal{A}_i(0)$ is ensured by the second part of the condition, as shown in the proof of Lemma 5.4. At time t_j, we have that

$$\min_\ell \hat{d}_{i,\ell} + C_{\delta/n}(t_j) \leqslant \min_\ell U_{i,\ell} \leqslant L_{i,j} \leqslant \hat{d}_{i,j} - C_{\delta/n}(t_j). \tag{5.28}$$

If all the random quantities take their expected values, then $\hat{d}_{i,\ell} = d_{i,\ell} \forall \ell \neq i$ and

we have that

$$d_{i,i^*} + C_{\delta/n}(t_j) \leqslant d_{i,j} - C_{\delta/n}(t_j) \implies C_{\delta/n}(t_j) \leqslant \Delta_{i,j}/2. \tag{5.29}$$

Under the assumption, $\hat{d}_{i,j} = d_{i,j}$ and using the definition of its upper and lower confidence bounds, we get that $\mathbb{E}[L_{i,j}] \geqslant d_{i,j} - \Delta_{i,j}/2$ and $\mathbb{E}[U_{i,j}] \leqslant d_{i,j} + \Delta_{i,j}/2$. Similar bounds are true for x_k. Then

$$\begin{aligned} d_{i,k} > 6d_{i,j} - 3d_{i,i^*} \implies d_{i,k} - \frac{d_{i,k} - d_{i,i^*}}{2} d_{i,k} - \frac{\Delta_{i,k}}{2} \\ > 2\left(d_{i,j} + \frac{d_{i,j} - d_{i,i^*}}{2}\right) = 2\left(d_{i,j} + \frac{\Delta_{i,j}}{2}\right), \end{aligned}$$

which implies that $L_{i,k} = \mathbb{E}[L_{i,k}] > 2\mathbb{E}[U_{i,j}] = 2U_{i,j}$ and $x_k \notin \mathcal{A}_j(0)$. □

If all the random quantities take their expected value, then using Lemma 5.14 and the elimination criterion of `ANNEasy` (Lemma 5.3), the complement of the initial active set $\mathcal{A}_j(0)$ (also called the elimination set) can be characterized in the following manner.

$$\begin{aligned} \mathcal{A}_j(0)^{\complement} &= \cup_{i<j : j \in \mathcal{A}_i(0)} \{\mathcal{A}_i(0) \cap \{k : 2U_{i,j} < L_{i,k}\}\} \\ &\supseteq \cup_{i<j : j \in \mathcal{A}_i(0)} \{\mathcal{A}_i(0) \cap \{k : d_{i,k} > 6d_{i,j} - 3d_{i,i^*}\}\}. \end{aligned} \tag{5.30}$$

Replacing the indicator $\mathbb{1}_{[\mathcal{A}_{j,k}]}$ in Theorem 5.5 with an indicator for the non-membership of point x_k in the set (5.30) gives us an upper bound to the sample complexity of `ANNEasy` that is valid when all random quantities take their expected values.

To gain an idea of the savings achieved by our algorithm in comparison to the random sampling, we evaluate the sample complexity expressions for an example dataset. The dataset we look at consists of c clusters, each cluster containing $n/c > 1$ points. The points are indexed such that the mth cluster is $\mathcal{C}_m := \{x_{\underline{m}}, x_{1+\underline{m}}, \ldots, x_{\overline{m}}\}$, where

$$\underline{m} := 1 + (m-1)n/c \tag{5.31}$$

and

$$\overline{m} := mn/c \tag{5.32}$$

for all $m \in [c]$. Suppose the distances between the points are such that for any pair $\{x_i, x_j\} \subseteq \mathcal{C}_m$, the set of points

$$\{x_k : d_{i,k} < 6d_{i,j} - 3d_{i,i^*}\} \subseteq \mathcal{C}_m. \tag{5.33}$$

The above condition is ensured if the smallest distance between two points belonging to different clusters is at least six times the diameter of any cluster.

Lemma 5.15. *Consider a dataset which satisfies the condition in* (5.33)*. If all random quantities take their expected values,* `ANNEasy` *uses* $O(\sqrt{n})$ *fewer oracle queries than the random sampling baseline to learn the nearest neighbor graph.*

Proof. In the following we assume that all random quantities take their expected values. We can find the points that are definitely eliminated using the triangle inequality when `ANNEasy` is called using (5.30). The elimination set $\mathcal{A}_1(0)^\complement = \{x_1\}$. For a point $x_i \in \mathcal{C}_1 \notin \mathcal{E}_1$, from (5.30), (5.33) we get that

$$\mathcal{A}_i(0)^\complement \supseteq \{\mathcal{A}_1(0) \cap \{k : d_{1,k} > 6d_{1,i} - 3d_{1,1^*}\}\} \supseteq \{(\mathcal{X} \setminus \{x_1\}) \cap \mathcal{C}_1^\complement\} = \mathcal{C}_1^\complement.$$

Thus $\mathcal{A}_i(0) \subseteq \mathcal{C}_1$ for all $x_i \in \mathcal{C}_1$. Point $x_{\underline{m}}$ is the first point processed by `ANNEasy` in the mth cluster. Suppose there exists a point $x_j \in \mathcal{C}_m \cap \mathcal{A}_{\underline{m}}(0)^\complement$, we show next that leads to a contradiction. Since $x_j \notin \mathcal{A}_{\underline{m}}(0)$, there is a point $x_i \in \mathcal{C}_{m'}$ with $i < j$, $m' < m$ such that $2U_{i,\underline{m}} < L_{i,j}$. Let $\mathrm{Diam}(\mathcal{C}_m) := \max_{x_\ell, x_k \in \mathcal{C}_m} d_{\ell,k}$ be the diameter of cluster $\mathcal{C}_m$ (similarly for $\mathcal{C}_{m'}$) and let $D(\mathcal{C}_{m'}, \mathcal{C}_m) := \min_{x_\ell \in \mathcal{C}_{m'}, x_k \in \mathcal{C}_m} d_{\ell,k}$ be the minimum inter-cluster distance. Since the random quantities take their expected values, we have that

$$U_{i,\underline{m}} \geqslant d_{i,\underline{m}} + \frac{d_{i,\underline{m}} - d_{i,i^*}}{2} \implies 2U_{i,\underline{m}} \geqslant 3D(\mathcal{C}_{m'}, \mathcal{C}_m) - \mathrm{Diam}(\mathcal{C}_{m'}),$$
$$L_{i,j} \leqslant d_{i,j} - \frac{d_{i,j} - d_{i,i^*}}{2} \implies L_{i,j} \leqslant \frac{\mathrm{Diam}(\mathcal{C}_{m'}) + D(\mathcal{C}_{m'}, \mathcal{C}_m) + \mathrm{Diam}(\mathcal{C}_m)}{2} + \frac{\mathrm{Diam}(\mathcal{C}_{m'})}{2}.$$

Using $2U_{i,\underline{m}} < L_{i,j}$ with the above inequalities implies that $2.5D(\mathcal{C}_{m'}, \mathcal{C}_m) < 2D(\mathcal{C}_{m'}) + 0.5D(\mathcal{C}_m)$, which is a contradiction as from (5.33) we have that $D(\mathcal{C}_{m'}, \mathcal{C}_m) \geqslant \max\{3D(\mathcal{C}_{m'}), 3D(\mathcal{C}_m)\}$. Thus we have that $\mathcal{C}_m \cap \mathcal{A}_{\underline{m}}(0)^{\complement} = \emptyset$. For any $x_j \in \mathcal{C}_m, j \neq \underline{m}$ we have that $x_j \in \mathcal{A}_{\underline{m}}(0)$ and hence from (5.30),

$$\mathcal{A}_j(0)^{\complement} \supseteq \{\mathcal{A}_{\underline{m}}(0) \cap \{k : d_{\underline{m},k} > 6d_{\underline{m},j} - 3d_{\underline{m},m^*}\}\} \supseteq \mathcal{C}_m^{\complement}.$$

Based on the above discussion, we have a lower bound on the number of points present in the elimination set $\mathcal{A}_j(0)^{\complement}$ for any $x_j \in \mathcal{C}_m$. By choosing the following values for the indicator in (5.25)

$$\mathbb{1}_{[A_{j,k}]} = \begin{cases} 0 & \text{if } x_j \in \mathcal{C}_m \setminus \{x_{\underline{m}}\} \text{ and } x_k \notin \mathcal{C}_m, \\ 1 & \text{otherwise,} \end{cases}$$

we get the following upper bound to the number of oracle queries, where $x_{\overline{m}}$ is the last point in $\mathcal{C}_m$.

$$\mathcal{O}\Bigg(\sum_{m=1}^{c}\Bigg(\sum_{k>\underline{m}} H_{\underline{m},k} + \sum_{k<\underline{m}} H_{\underline{m},k} - \sum_{\ell=1}^{m-1} H_{\underline{m},\ell} + \sum_{\ell=1}^{m-1}(H_{\underline{m},\ell} - H_{\ell,\underline{m}})_+ + \sum_{p>\underline{m}}^{\overline{m}}\sum_{q>p}^{\overline{m}} \max\{H_{p,q}, H_{q,p}\}\Bigg)\Bigg) \tag{5.34}$$

where $\underline{m}$ and $\overline{m}$ are defined in (5.31) and are functions of m. The number of terms in the sum above is $\mathcal{O}(cn + (n/c)^2)$. A uniform sampling baseline approach would have $\mathcal{O}(n^2)$ terms in its sample complexity. Letting $c = \sqrt{n}$ gives our result. □

The above lemma ensures that we have $\mathcal{O}(\sqrt{n})$ fewer terms in the sample complexity expression for `ANNEasy` compared to random sampling if the dataset satisfies (5.33). We can get a more precise characterization of the savings in query complexity in terms of the $\Delta_{p,q}$ values. For instance, using a single-parameter model for the distribution of $\Delta_{p,q}$ as done in Jamieson et al. (2013), we can directly use their Corollary 1 in our context.

Lemma 5.16. *Consider a clustered dataset* $\mathcal{X} = \cup_{m=1}^{c} \mathcal{C}_m$ *whose points satisfy* (5.33). *Each cluster contains an even number* $2\nu := n/c$ *of points. For any* $m \in [c]$ *and* $x_j \in \mathcal{C}_m$, *suppose the suboptimality gaps* $\Delta_{j,k}$ *for all* $x_k \in \mathcal{C}_m$ *take one of the following values, parametrized by an* $\alpha > 0$*:*

$$1 - \left(\frac{s-1}{\nu-1}\right)^{\alpha}, \qquad \text{where} \qquad s \in \{1, 2, \ldots, \nu-1\}. \tag{5.35}$$

Note that there are $\nu - 1$ *values given in* (5.35) *while there are* $2\nu - 2$ *points in the cluster, excluding* x_j *and* x_{j^*}. *Each value in* (5.35) *is the suboptimality gap for two distinct points in* $\mathcal{C}_m$. *Ignoring* log*-factors, if* $\alpha = 1$ `ANNTri` *finds all nearest neighbors with probability* $1 - \delta$ *in* $O(n(\nu^2 + n))$ *calls to the oracle, while uniform sampling requires* $O(n^2\nu^2)$ *calls for the same guarantee.*

Proof. By putting the clusters far from each other, one can see that there exist $\mathcal{X} = \cup_{m=1}^{c} \mathcal{C}_m$ whose points satisfy (5.33). Lemma 5.11 shows by explicit construction that the condition on the suboptimality gaps within each cluster as stated in (5.35) can also be satisfied. Note that (5.35) is the same parametrization as equation 3 in Jamieson et al. (2013).

Consider the points in the mth cluster, i.e., points $x_{\underline{m}}$ through $x_{\overline{m}}$. The elimination set $\mathcal{A}_{\underline{m}}(0)^{\complement}$ can be the singleton $\{x_{\underline{m}}\}$, but by Lemma 5.14 for all $x_p \in \mathcal{C}_m \setminus \{x_{\underline{m}}\}, \mathcal{A}_p(0)^{\complement} \supseteq \mathcal{C}_m^{\complement}$. Finding x_{p^*} is a best arm identification problem among points within the cluster $\mathcal{C}_m$. The last term in (5.34) counts the total number of oracle queries made by `ANNEasy` to identify the nearest neighbors of all $x_p \in \mathcal{C}_m \setminus \{x_{\underline{m}}\}$. Thus the number of oracle queries made by `ANNEasy` for identifying x_{p^*} is at most $\sum_{q \neq p} H_{p,q}$, while uniform sampling will make $nH_{p,p'}$ queries, where $p' := \arg\min_{q \neq p^*} \Delta_{p,q}$.

Ignoring log-factors, the sample complexity for finding x_{p^*} for an $x_p \in \mathcal{C}_m$ by `ANNEasy` is

$$\tilde{\mathcal{O}}\left(\sum_{i=1}^{2\nu-1} \Delta_{p,p_i}^{-2} + \sum_{x_\ell \notin \mathcal{C}_m} \Delta_{p,\ell}^{-2}\right) = \tilde{\mathcal{O}}\left(\sum_{i=1}^{2\nu-1} \Delta_{p,p_i}^{-2} + n - 2\nu\right).$$

Corollary 1 of Jamieson et al. (2013) lists the value of that sum for different choices of α, for e.g., if $\alpha = 1$ then the sample complexity is $\tilde{\mathcal{O}}(\nu^2 + n - 2\nu)$. On the other hand, for finding x_{p^*} uniform sampling would make $\tilde{\mathcal{O}}(n\Delta_{p,p'}^{-2},)$, i.e., $\tilde{\mathcal{O}}(n(\nu-1)^2)$ queries. By construction of the dataset, finding the nearest neighbor of each point in $\mathcal{X}$ is equally hard. Thus `ANNTri` would make $\tilde{\mathcal{O}}(n(\nu^2 + n - 2\nu))$ queries while uniform sampling would take $\tilde{\mathcal{O}}(n^2\nu^2)$ queries. □

Note that our problem setting is inherently different from the noiseless setting where all x_{i^*}'s can trivially be learned in $\binom{n}{2}$ samples. Due to the presence of noise in our queries, many distances must be repeatedly queried so $\binom{n}{2}$ samples is insufficient.

6 FINDING ALL ϵ-GOOD ARMS IN STOCHASTIC BANDITS

6.1 Introduction

We propose a new multi-armed bandit problem where the objective is to return *all* arms that are ϵ-good relative to the best-arm. Concretely, if the arms have means $\mu_1, \cdots, \mu_n$, with $\mu_1 = \max_{1 \leqslant i \leqslant n} \mu_i$, then the goal is to return the set $\{i : \mu_i \geqslant \mu_1 - \epsilon\}$ in the additive case, and $\{i : \mu_i \geqslant (1-\epsilon)\mu_1\}$ in the multiplicative case. The ALL-ϵ problem is a novel setting in the bandits literature, adjacent to two other methods for finding many good arms: TOP-k where the goal is to return the arms with the k highest means, and threshold bandits where the goal is to identify all arms above a fixed threshold. Building on a metaphor given by Locatelli et al. (2016), if TOP-k is a "contest" and thresholding bandits is an "exam", ALL-ϵ organically decides which arms are "above the bar" relative to the highest score. We argue that the ALL-ϵ problem formulation is more appropriate in many applications, and we show that it presents some unique challenges that make its solution distinct from TOP-k and threshold bandits.

A Natural and Robust Objective. A motivating example is drug discovery, where pharmacologists want to identify a set of highly-potent drug candidates from potentially millions of compounds using various *in vitro* and *in silico* assays, and only the selected undergo more expansive testing Christmann-Franck et al. (2016). Since performing the assays can be costly, one would like to use an adaptive, sequential experiment design that requires fewer experiments than a fixed experiment design. In sequential experiment design, it is important to fix the objective at the beginning as that choice affects the experimentation process. Both the objectives of finding the top-k performing drugs, or all drugs above a threshold can result in failure. In TOP-k, choosing k too small may miss potent compounds, and choosing k too large may yield many ineffective compounds and require an excessively large number of experiments. Setting a threshold suffers from the same issues - with the additional concern that if it is set too high, potentially no drug discoveries are made. In contrast, the ALL-ϵ objective of finding all arms whose potency is within

20% of the best avoids these concerns by giving a robust and natural guarantee: *no* significantly suboptimal arms will be returned and but *every* near-optimal arm will be discovered.

A second motivating example is selecting crowd-workers for a specialized crowdsourcing task, such as labelling images of dogs by breed. In this case, expertise is required to perform this task with high accuracy, and it is important that the data collected from workers is accurate since it will be used for downstream applications such as training neural networks. In order to select crowd-workers to label a dataset, it is reasonable for practitioners to first ask candidates some test questions where the true label is known. As an incentive to answer test questions, it may be necessary to pay per question asked. ALL-ϵ, efficiently finds every near-optimal worker without needing to know how many good workers are available or what performances they achieve.

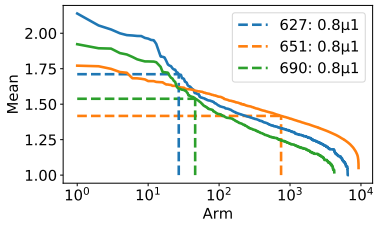


Figure 6.1: Mean ratings from contests 627, 651, 690

We emphasize that unlike TOP-k or thresholding which require some prior knowledge about the distribution of arms to guarantee a good set of returned arms, choosing the arms relative to the best is a natural, distribution-free metric for finding good arms. As an example, we consider the New Yorker Cartoon Caption Contest (NYCCC). Each week, contestants submit thousands of supposedly funny captions for a cartoon (see Appendix 6.A), which are rated from 1 (unfunny) to 3 (funny) through a crowdsourcing process. The New Yorker editors select final winners from a set with the highest average crowd-ratings (typically over 1 million ratings per contest). The number of truly funny captions varies from week to week, and this makes setting a choice of k or fixed threshold difficult. In Figure 6.1, we plot the distribution of ratings from 3 different contests. Horizontal lines depict a reasonable threshold of $0.8\mu_1$ in each and vertical lines show the number of arms that exceed this threshold. Both of these quantities vary over weeks and these differences can be stark. In contest 627, only $k = 27$ arms are within 20% of μ_1, but $k = 748$ are in contest 651. Additionally, a fixed threshold of $\tau = 1.5$, admits captions within

30% of the best in contest 627, but only those within 15% of the best in contest 651. These examples show that it would be imprudent, and indeed, incorrect to choose a value of k or a threshold based on past contests– the far more principled decision is to optimize for the objective of finding the captions that are within a percentage of the best every week.

Though the ALL-ϵ objective is natural and easy to state, it has not been studied in the literature. As we will show, admitting arms relative to the best makes the ALL-ϵ problem inherently more challenging than either TOP-k or thresholding. In particular, it is not easily possible to adapt TOP-k or thresholding algorithms to achieve the instance dependent lower bound for ALL-ϵ. In this work, we provide a careful investigation of the ALL-ϵ problem including theoretical and empirical guarantees.

6.1.1 Problem Statement and Notation

Fix $\epsilon > 0$ and a failure probability $\delta > 0$. Let $\nu := \{\rho_1, \cdots, \rho_n\}$ be an instance of n distributions (or arms) with 1-sub-Gaussian distributions having *unknown* means $\mu_1 \geqslant \cdots \geqslant \mu_n$. We now formally define our notions of additive and multiplicative ϵ-good arms.

Definition 6.1 (additive ϵ-good)**.** *For a given $\epsilon > 0$, arm i is additive ϵ-good if $\mu_i \geqslant \mu_1 - \epsilon$.*

Definition 6.2 (multiplicative ϵ-good)**.** *For a given $\epsilon > 0$, arm i is multiplicative ϵ-good if $\mu_i \geqslant (1-\epsilon)\mu_1$.*

Additionally, we define the sets

$$G_\epsilon(\nu) := \{i : \mu_i \geqslant \mu_1 - \epsilon\} \text{ and } M_\epsilon(\nu) := \{i : \mu_i \geqslant (1-\epsilon)\mu_1\} \tag{6.1}$$

to be the sets of additive and multiplicative ϵ-good arms respectively. Where clear, we take $G_\epsilon = G_\epsilon(\nu)$ and $M_\epsilon = M_\epsilon(\nu)$. Consider an algorithm that at each time s selects an arm $I_s \in [n]$ based on the history $\mathcal{F}_{s-1} = \sigma(I_1, X_1, \cdots, I_{s-1}, X_{s-1})$, and

observes a reward $X_s \overset{\text{iid}}{\sim} \rho_{I_s}$. The objective of the algorithm is to return G_ϵ or M_ϵ using as few total samples as possible.

Definition 6.3. *(ALL-ϵ problem). An algorithm for the ALL-ϵ problem is δ-PAC if (a) the algorithm has a finite stopping time τ with respect to $\mathcal{F}_t$, (b) at time τ it recommends a set $\widehat{G}$ such that with probability at least $1-\delta$, $\widehat{G} = G_\epsilon$ in the additive case, or $\widehat{G} = M_\epsilon$ in the multiplicative case.*

Notation: For any arm $i \in [n]$, let $\widehat{\mu}_i(t)$ denote the empirical mean after t pulls. For all $i \in [n]$, define the suboptimality gap $\Delta_i := \mu_1 - \mu_i$. Without loss of generality, we denote $k = |G_\epsilon|$ (resp. $k = |M_\epsilon|$). Throughout, we will keep track of the quantity $\alpha_\epsilon := \min_{i \in G_\epsilon} \mu_i - (\mu_1 - \epsilon)$ which is the distance from the smallest additive ϵ-good arm, denoted μ_k, to the threshold $\mu_1 - \epsilon$. Additionally, if G_ϵ^c is non-empty, we consider $\beta_\epsilon = \min_{i \in G_\epsilon^c}(\mu_1 - \epsilon) - \mu_i$, the distance of the largest arm that is not additive ϵ-good, denoted μ_{k+1}, to the threshold. Equivalently, in the case of returning multiplicative ϵ arms, we define $\tilde{\alpha}_\epsilon := \min_{i \in M_\epsilon} \mu_i - (1-\epsilon)\mu_1$, $\tilde{\beta}_\epsilon := \min_{i \in M_\epsilon^c}(1-\epsilon)\mu_1 - \mu_i$, μ_k, and μ_{k+1} to be the smallest differences of arms in M_ϵ and M_ϵ^c to $(1-\epsilon)\mu_1$ respectively. For our sample complexity results, we also consider a relaxed version of the ALL-ϵ problem, where for a user-given slack $\gamma \geqslant 0$, we allow our algorithm to return $\widehat{G}$ that satisfies $G_\epsilon \subset \widehat{G} \subset G_{\epsilon+\gamma}$ in the additive case, or $M_\epsilon \subset \widehat{G} \subset M_{\epsilon+\gamma}$ in the multiplicative case. As we will see, this prevents large or potentially unbounded sample complexities when arms' means are very close to or equal $\mu_1 - \epsilon$.

6.1.2 Contributions and Summary of Main Results

In this chapter we propose the new problem of finding *all* ϵ-good arms and give a precise characterization of its complexity. Our contribution is threefold:

- Information-theoretic lower bounds for the ALL-ϵ problem.
- A novel algorithm, $(\text{ST})^2$, that is nearly optimal, is easy to implement, and has excellent empirical performance on real-world data.

- An instance optimal algorithm, FAREAST.

We now summarize our results in the additive setting (the multiplicative setting is analogous).

Lower Bound and Algorithms. As a preview of our results, we highlight the impact of three key quantities that affect the sample complexity: the user provided ϵ and the instance dependent quantities α_ϵ and β_ϵ, (see Figure 6.2). In this case, Theorem 6.4 implies that any δ-PAC algorithm requires an expected number of samples exceeding

$$\sum_{i=1}^{n} \max\left\{\frac{1}{(\mu_1 - \epsilon - \mu_i)^2}, \frac{1}{(\mu_1 + \alpha_\epsilon - \mu_i)^2}\right\} \log\left(\frac{1}{\delta}\right). \tag{6.2}$$

We provide two algorithms, $(\text{ST})^2$ and FAREAST for the ALL-ϵ problem. Our starting point, $(\text{ST})^2$ is a novel combination of UCB Auer et al. (2002) and LUCB Kalyanakrishnan et al. (2012) and is easier to implement and has good empirical performance. $(\text{ST})^2$ is nearly optimal, however in some instances does not achieve the lower bound. To overcome this gap, we provide an instance optimal algorithm FAREAST which achieves the lower bound, however suffers from larger constants and is not always better in practical applications.

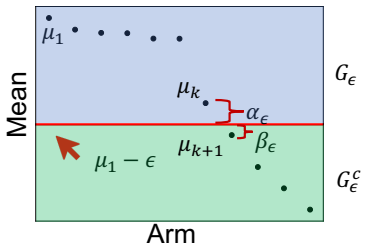


Figure 6.2: An example instance

To highlight the difficulty of developing optimal algorithms for the ALL-ϵ problem, we quickly discuss a naive elimination approach that uniformly samples all arms and eliminates arms once they are known to be above or below $\mu_1 - \epsilon$ and not the best arm. Intuitively, such an algorithm would keep pulling arms until $\mu_1 - \epsilon$ is estimated to an accuracy of $O(\min(\alpha_\epsilon, \beta_\epsilon))$ to resolve the arms around the threshold (see Figure 6.2). An elimination algorithm pays a high cost of exploration - potentially over pulling arms close to μ_1 compared to the lower bound until a time when $\mu_1 - \epsilon$ is estimated sufficiently well. Our algorithm FAREAST provides a novel

approach to overcome the issues with this approach. However, as we will show in Section 6.4, in certain instances a dependence on $\sum_{i=1}^{n}(\mu_1 + \beta_\epsilon - \mu_i)^{-2}$ is present in *moderate confidence*, i.e., it is not multiplied by $\log(1/\delta)$, unlike the lower bound in equation (6.2) and becomes negligible compared to other terms as $\delta \to 0$.

Empirical results. We demonstrate the empirical success of $(\texttt{ST})^2$ on a real world dataset of 9250 captions from the NYCCC. In Fig. 6.4a, we compare $(\texttt{ST})^2$ to other methods that have been used to run this contest. We show that $(\texttt{ST})^2$ is better able to detect which arms have means within 10% of the best. The plot demonstrates the sub-optimality of using existing sampling schemes such as UCB or LUCB with an incorrect k for the ALL-ϵ problem, providing an additional empirical validation for the study of this chapter.

6.1.3 Connections to prior Bandit art

Our problem is related to several prior pure-exploration settings in the multi-armed bandit literature, including TOP-k bandits, and threshold bandits.

TOP-k. In the TOP-k problem, the goal is to identify the set $\{\mu_1, \cdots, \mu_k\}$ with probability greater than $1-\delta$ Kalyanakrishnan et al. (2012); Bubeck et al. (2013); Kaufmann et al. (2016); Gabillon et al. (2012); Ren et al. (2019); Simchowitz et al. (2017). The ALL-ϵ problem reduces to the setting of the TOP-k problem with $k = |G_\epsilon|$ when $|G_\epsilon|$ is known. In particular, lower bounds for the TOP-k problem apply to our setting. A lower bound (with precise logarithmic factors) given in Simchowitz et al. (2017) is $\sum_{i=1}^{k}(\mu_i - \mu_{k+1})^{-2}\log((n-k)/\delta) + \sum_{i=k+1}^{n}(\mu_i - \mu_k)^{-2}\log(k/\delta)$. In general, this is smaller than our lower bound in Theorem 6.4 since $\mu_k \geqslant \mu_1 - \epsilon \geqslant \mu_{k+1}$. A particular case of this problem is best-arm identification when $k = 1$.

Approximate versions of the TOP-k problem have also been considered where the goal is to return a set of arms S with $|S| = k$ and such that with probability greater than $1-\delta$, each $i \in S$ satisfies $\mu_i \geqslant \mu_k - \epsilon$ Kalyanakrishnan et al. (2012); Karnin et al. (2013). In the case where $k = 1$, this is also known as the problem of identifying an (single) ϵ-good arm Mannor and Tsitsiklis (2004); Even-Dar et al. (2002); Kalyanakrishnan et al. (2012); Even-Dar et al. (2006); Kalyanakrishnan and Stone

(2010); Katz-Samuels and Jamieson (2020); Degenne and Koolen (2019); Gabillon et al. (2012); Kaufmann and Kalyanakrishnan (2013); Karnin et al. (2013); Simchowitz et al. (2017) which has received a large amount of interest. If $|G_\epsilon| = k$, Kaufmann et al. (2016), demonstrate a lower bound of $O((k\epsilon^{-2} + \sum_{i=k+1}^{n}(\mu_1 - \mu_i)^{-2})\log(1/\delta))$ samples in expectation to find such an arm and Karnin et al. (2013) provide an algorithm that matches this to doubly logarithmic factors, though methods such as Kalyanakrishnan et al. (2012); Simchowitz et al. (2017); Chaudhuri and Kalyanakrishnan (2017, 2019) achieve better empirical performance. A particular instance of interest is when it is known that one arm is at mean ϵ, and the rest are at mean zero. In this setting, Mannor and Tsitsiklis (2004) show a lower bound on the sample complexity of $O(n/\epsilon^2 + 1/\epsilon^2 \log(1/\delta))$ highlighting that the dependence on n only occurs in *moderate confidence*, i.e., for a fixed value of δ. They also provide a matching upper bound that motivates our procedure in FAREAST. Finally Katz-Samuels and Jamieson (2020) considers the *unverifiable* regime where there are potentially many ϵ-good arms. In such cases, sample-efficient algorithms exist that return an ϵ-good arm with high probability, but *verifying* it is ϵ-good requires far more samples. Extending these ideas to the setting of ALL-ϵ is a goal of future work.

Threshold Bandits. In the threshold bandit problem, we are given a threshold τ and the goal is to identify the set of arms whose means are greater than the threshold Locatelli et al. (2016); Kano et al. (2019). If the value of μ_1 were known, then ALL-ϵ problem would reduce to a threshold bandit with $\tau = \mu_1 - \epsilon$. A naive sequential sampling scheme that stops sampling an arm when its upper or lower confidence bound clears the threshold has sample complexity $O(\sum_{i=1}^{n}(\mu_i - \tau)^{-2}\log(n/\delta))$. Up to factors of $\log(n)$, this can be shown to be a lower bound for threshold bandits as well, and as a result is bounded above by the result Theorem 6.4. Hence, ALL-ϵ is intrinsically more difficult than threshold bandits. A naive approach to the ALL-ϵ problem is to first identify the index and mean of the best arm using a best-arm identification algorithm and then utilize it to build an estimate of the threshold $\mu_1 - \epsilon$. In general, this two-step procedure is sub-optimal if there are many arms close to the best-arm in which case identifying the best-arm is both unnecessary and expends unnecessary samples. In the fixed confidence setting, threshold bandits is

closely related to that of multiple hypothesis testing, and recent work Jamieson and Jain (2018) achieves tight upper and lower bounds for this problem including tighter logarithmic factors similar to those for Top-k. If μ_1 is known, then the additive all-ϵ problem reduces to the FWER (family-wise error rate) and FWPD (family-wise probability of detection) setting in Jamieson and Jain (2018). Finally, in the fixed budget setting, Locatelli et al. (2016) proposes an optimal anytime method APT whose sampling strategy we use as a comparison in Section 3.3.

6.2 Lower Bound

Theorem 6.4. *(additive and multiplicative lower bounds) Fix $\delta, \epsilon > 0$. Consider n arms, such that the i^{th} is distributed according to $\mathcal{N}(\mu_i, 1)$. Any δ-PAC algorithm for the additive setting satisfies*

$$\mathbb{E}[\tau] \geqslant 2 \sum_{i=1}^{n} \max \left\{ \frac{1}{(\mu_1 - \epsilon - \mu_i)^2}, \frac{1}{(\mu_1 + \alpha_\epsilon - \mu_i)^2} \right\} \log\left(\frac{1}{2.4\delta}\right)$$

and if $\mu_1 > 0$, any δ-PAC algorithm for the multiplicative algorithm satisfies,

$$\mathbb{E}[\tau] \geqslant 2 \sum_{i=1}^{n} \max \left\{ \frac{1}{((1-\epsilon)\mu_1 - \mu_i)^2}, \frac{1}{(\mu_1 + \frac{\tilde{\alpha}_\epsilon}{1-\epsilon} - \mu_i)^2} \right\} \log\left(\frac{1}{2.4\delta}\right).$$

The bounds are different but share a common interpretation. Consider the additive case. First, every arm must be sampled inversely proportional to its squared distance to $\mu_1 - \epsilon$. In a manner similar to thresholding Locatelli et al. (2016), even if $\mu_1 - \epsilon$ was known, these number of samples are necessary to decide if an arm's mean is above or below that quantity. This leads to the first term in the $\max\{\cdot, \cdot\}$. The second term in the $\max\{\cdot, \cdot\}$ states that every arm must be sampled inversely proportional to its squared distance to $\mu_1 + \alpha_\epsilon$. Recall that $\alpha_\epsilon = \mu_k - (\mu_1 - \epsilon)$ is the margin by which arm k is good. Hence, to verify that $k \in G_\epsilon$, it is also necessary to confirm that all means are below $\mu_1 + \alpha_\epsilon$, as $\mu_1 + \alpha_\epsilon - \epsilon \geqslant \mu_k$ which would imply that k is bad. This represents the necessity of estimating the threshold, and leads to

the second term. For arms in G_ϵ^c, comparing against $\mu_1 - \epsilon$ is always more difficult, but for arms in G_ϵ, either constraint may be more challenging to ensure. We state the bound for Gaussian distributions, but the same technique can be used to prove equivalent results for other distributions. Lastly, we note that it is possible to prove bounds with tighter logarithmic terms. For an instance where $O(n^\phi)$ arms have mean 2ϵ for $\phi \in (0, 1)$, and the remaining have mean 0, Theorem 1 of Malloy and Nowak (2014) suggests that $\Omega(n/\epsilon^2 \log(n/\delta))$ samples are necessary, exceeding the above bounds by a factor of $\log(n)$.

6.3 An Optimism Algorithm for ALL-ϵ

We propose algorithm 5 called $(\texttt{ST})^2$, (**S**ample the **T**hreshold, **S**plit the **T**hreshold) to return a set containing all ϵ-good arms and none worse than $(\epsilon + \gamma)$-good with probability $1 - \delta$. Intuitively, $(\texttt{ST})^2$ runs `UCB` and `LUCB1` in parallel. At all times, $(\texttt{ST})^2$ pulls three arms. We pull the arm with the highest upper confidence bound, similarly to the UCB algorithm, Auer et al. (2002), to refine an estimate of the threshold using the highest empirical mean (**S**ample the **T**hreshold). Using the empirical estimate of the threshold, we pull an arm above it and an arm below it whose confidence bounds cross it, similar to `LUCB1`, Kalyanakrishnan et al. (2012) (**S**plit the **T**hreshold). Using these bounds, $(\texttt{ST})^2$ forms upper and lower bounds on the true threshold, i.e. $\mu_1 - \epsilon$ (resp. $(1-\epsilon)\mu_1$) and terminates when it can declare that all arms are either in $G_{\epsilon+\gamma}$ or G_ϵ^c. To do so, $(\texttt{ST})^2$ maintains anytime confidence widths, $C_{\delta/n}(t)$ such that for an empirical mean $\widehat{\mu}_i(t)$ of t samples, we have $\mathbb{P}(\bigcup_{t=1}^{\infty} |\widehat{\mu}_i(t) - \mu_i| > C_{\delta/n}(t)) \leqslant \delta/n$. For this work, we take $C_\delta(t) = \sqrt{\frac{c_\phi \log(\log_2(2t)/\delta)}{t}}$ for a constant c_ϕ. It suffices to take $c_\phi = 4$, though tighter bounds are known and should be used in practice, e.g. Jamieson et al. (2014); Kaufmann et al. (2016); Howard et al. (2018).

Algorithm 5 $(\mathrm{ST})^2$: **S**ample the **T**hreshold, **S**plit the **T**hreshold

Require: $\epsilon, \delta > 0, \gamma \geqslant 0$, instance ν
1: Pull each arm once, initialize $T_i \leftarrow 1$, update $\widehat{\mu}_i$ for each $i \in \{1, 2, \ldots, n\}$
2: Empirically good arms: $\widehat{G} = \{i : \widehat{\mu}_i \geqslant \max_j \widehat{\mu}_j - \epsilon\}$, $\widehat{G} = \{i : \widehat{\mu}_i \geqslant (1-\epsilon)\max_j \widehat{\mu}_j\}$
3: $U_t = \max_j \widehat{\mu}_j(T_j) + C_{\delta/n}(T_j) - \epsilon - \gamma$ and $L_t = \max_j \widehat{\mu}_j(T_j) - C_{\delta/n}(T_j) - \epsilon$
4: $U_t = (1-\epsilon-\gamma)\left(\max_j \widehat{\mu}_j(t) + C_{\delta/n}(T_j)\right)$ and $L_t = (1-\epsilon)\left(\max_j \widehat{\mu}_j(t) - C_{\delta/n}(T_j)\right)$
5: Known arms: $K = \{i : \widehat{\mu}_i(T_i) + C_{\delta/n}(T_i) < L_t \text{ or } \widehat{\mu}_i(T_i) - C_{\delta/n}(T_i) > U_t\}$
6: **while** $K \neq [n]$ **do**
7: Pull arm $i_1(t) = \arg\min_{i \in \widehat{G} \setminus K} \widehat{\mu}_i(T_i) - C_{\delta/n}(T_i)$, update $T_{i_1}, \widehat{\mu}_{i_1}$
8: Pull arm $i_2(t) = \arg\max_{i \in \widehat{G}^c \setminus K} \widehat{\mu}_i(T_i) + C_{\delta/n}(T_i)$, update $T_{i_2}, \widehat{\mu}_{i_2}$
9: Pull arm $i^*(t) = \arg\max_i \widehat{\mu}_i(T_i) + C_{\delta/n}(T_i)$, update $T_{i^*}, \widehat{\mu}_{i^*}$
10: Update bounds L_t, U_t, sets $\widehat{G}$, K
11: **end while**
12: **return** The set of good arms $\{i : \widehat{\mu}_i(T_i) - C_{\delta/n}(T_i) > U_t\}$

6.3.1 Theoretical guarantees

Next we present a pair of theorems on the sample complexity of $(\mathrm{ST})^2$. For clarity, we omit doubly logarithmic terms and defer such statements to Appendix 6.B. Below we denote $a \wedge b := \min\{a, b\}$.

Theorem 6.5 (Additive Case). *Fix $\epsilon > 0$, $0 < \delta \leqslant 1/2$, $\gamma \leqslant 16$ and an instance ν such that $\max(\Delta_i, |\epsilon - \Delta_i|) \leqslant 8$ for all i. With probability at least $1-\delta$, there is a constant c_1 such that $(ST)^2$ returns a set $\widehat{G}$ such that $G_\epsilon \subset \widehat{G} \subset G_{(\epsilon+\gamma)}$ in at most the following number of samples.*

$$c_1 \log\left(\frac{n}{\delta}\right) \sum_{i=1}^{n} \max\left\{\frac{1}{(\mu_1 - \epsilon - \mu_i)^2}, \frac{1}{(\mu_1 + \alpha_\epsilon - \mu_i)^2}, \frac{1}{(\mu_1 + \beta_\epsilon - \mu_i)^2}\right\} \wedge \frac{1}{\gamma^2} \tag{6.3}$$

Given a positive slack γ, we are allowed to return an arm that is $(\epsilon+\gamma)$-good. Thus a confidence width less than $\Omega(\gamma)$ on any arm is not needed, resulting in the $1/\gamma^2$ term in Theorem 6.5. In particular this prevents unbounded sample complexities if there is an arm at the threshold $\mu_1 - \epsilon$. For $\gamma = 0$, the first two terms inside the max are also present in the lower bound (Theorem 6.4). When α_ϵ

is within a constant factor of β_ϵ, the second and third term in the max have the same order, and the upper bound matches the lower bound up to a $\log(n)$ factor.

If $\beta_\epsilon \ll \alpha_\epsilon$, (6.3) has a different scaling than the lower bound. In such restrictive settings the upper bound above can be significantly larger than the lower bound. In the next section, we provide an algorithm that overcomes these issues and is optimal over all parameter regimes. The multiplicative case has different terms but follows the same intuition.

Theorem 6.6 (Multiplicative Case)**.** *Fix* $\epsilon \in (0, 1/2]$, $\gamma \in [0, \min(16/\mu_1, 1/2)]$ *and* $0 < \delta \leqslant 1/2$ *and an instance* ν *such that* $\mu_1 \geqslant 0$ *and* $\max(\Delta_i, |\epsilon\mu_1 - \Delta_i|) \leqslant 2$ *for all* i*. With probability at least* $1-\delta$*, for a constant* c_1 $(ST)^2$ *returns a set* G *such that* $M_\epsilon \subset G \subset M_{(\epsilon+\gamma)}$ *with sample complexity:*

$$c_1 \log\left(\frac{n}{\delta}\right) \sum_{i=1}^{n} \max\left\{ \frac{1}{((1-\epsilon)\mu_1 - \mu_i)^2}, \frac{1}{(\mu_1 + \frac{\tilde{\alpha}_\epsilon}{1-\epsilon} - \mu_i)^2}, \frac{1}{(\mu_1 + \frac{\tilde{\beta}_\epsilon}{1-\epsilon} - \mu_i)^2} \right\} \wedge \frac{1}{\gamma^2\mu_1^2}.$$

6.4 Surprising Complexity of Finding All ϵ-Good arms

When α_ϵ and β_ϵ are not of the same order, $(ST)^2$ is not optimal. In this section we present an algorithm that is optimal for all parameter regimes. We focus on the additive case here, and defer the multiplicative case to Appendix 6.F. We first state an improved sample complexity lower bound for a family of problem instances that makes explicit the *moderate confidence* terms.

Theorem 6.7. *Fix* $\delta \leqslant 1/16$, $n \geqslant 2/\delta$, *and* $\epsilon > 0$*. Let* ν *be an instance of* n *arms such that the* i^{th} *is distributed as* $\mathcal{N}(\mu_i, 1)$, $|G_{2\beta_\epsilon}| = 1$, *and* $\beta_\epsilon < \epsilon/2$*. Select a permutation* $\pi : [n] \to [n]$ *uniformly from the set of* $n!$ *permutations, and consider the permuted instance* $\pi(\nu)$*. Any algorithm that returns* $G_\epsilon(\pi(\nu))$ *on* $\pi(\nu)$ *correctly with probability at least* $1-\delta$ *requires at least the following number of samples in expectation over randomness in* ν

and π for a universal constant c_2.

$$\left[c_2 \sum_{i=1}^{n} \max\left\{\frac{1}{(\mu_1 - \epsilon - \mu_i)^2}, \frac{1}{(\mu_1 + \alpha_\epsilon - \mu_i)^2}\right\} \log\left(\frac{1}{2.4\delta}\right)\right] + c_2 \sum_{i=1}^{n} \frac{1}{(\mu_1 + \beta_\epsilon - \mu_i)^2} \tag{6.4}$$

Proof. (Sketch) To give a tight lower bound in the setting where $|G_{2\beta_\epsilon}| = 1$ and $\beta_\epsilon < \epsilon/2$, we break our argument into pieces performing a series of reductions that link the ALL-ϵ problem to a hypothesis test, and then the hypothesis test to the problem of identifying the best-arm. We apply the Simulator technique from Simchowitz et al. (2017) to compute precise moderate confidence bounds. Other works that prove strong lower bounds in moderate confidence include Chen et al. (2017). We extend the Simulator technique via a novel reduction to composite hypothesis testing in order to connect to ALL-ϵ. In all cases, we consider sample complexity in expectation with respect to the randomness in the outcomes and a randomly chosen permutation of the means.

Step 1. Finding an isolated best arm: Consider the problem of finding the best arm where $\mu_1 = \beta > 0$ and $\mu_2, \cdots, \mu_n \leqslant -\beta$. This relates to the problem of finding a β-good arm when μ_1 is known, studied by Mannor and Tsitsiklis (2004). We use the Simulator technique, Simchowitz et al. (2017), to show that any algorithm requires $\Omega\left(\sum_{i=2}^{n} \Delta_i^{-2}\right)$ samples in expectation. This can be significantly larger than the asymptotically optimal rate of $O(\beta^{-2}\log(1/\delta))$ (which was proven by Mannor and Tsitsiklis (2004)) for non-asymptotic δ, e.g. $\delta = 0.05$.

Step 2. Deciding if Any mean is positive: We then consider a composite hypothesis test on n distributions where the null hypothesis, H_0, is that the mean of each distribution is less that $-\beta$ and the alternate hypothesis, H_1, is that there exists a *single* distribution i^* with mean β and the remainder have mean less than $-\beta$. Importantly, an algorithm does not need to declare which arm is i^*, otherwise the bound from step 1 applies immediately. Instead, to link this to step 1, we develop a novel extension of the simulator technique and use this to show that if an algorithm can solve this composite hypothesis test in fewer than $o\left(\sum_{i=2}^{n} \Delta_i^{-2}\right)$ samples, then

one may design a method to solve the problem in step 1 in $o\left(\sum_{i=2}^{n} \Delta_i^{-2}\right)$ samples which is a contradiction. Hence any algorithm for this hypothesis test requires $\Omega\left(\sum_{i=2}^{n} \Delta_i^{-2}\right)$ samples in expectation.

Step 3: Reducing ALL-ϵ to Step 2: Finally, we show that a generic algorithm for ALL-ϵ can be used to solve the hypothesis test in step 2. Hence the lower bound from step 2 applies to finding all ϵ-good arms as well. In the case of the instances considered in the theorem statement, $O\left(\sum_{i=2}^{n} \Delta_i^{-2}\right) = O\left(\sum_{i=2}^{n} (\mu_1 + \beta_\epsilon - \mu_i)^{-2}\right)$. Combining this bound, which is independent of δ with the result from Theorem 6.4 gives the result. □

Theorem 6.7 states that an additional $\Omega(\sum_{i=1}^{n} (\mu_1 + \beta_\epsilon - \mu_i)^{-2})$ samples are necessary for instances where no arm is within $2\beta_\epsilon$ of μ_1 compared to the lower bound Theorem 6.4. Somewhat surprisingly, these samples are *necessary in moderate confidence*, independent of δ and negligible as $\delta \to 0$. For non-asymptotic values of δ, such as the common choice of $\delta = .05$ in scientific applications, this term is present and can even dominate the sample complexity when $\beta_\epsilon \ll \alpha_\epsilon$. As an extreme example, if $\mu_1 = \beta > 0$, $\mu_2 \cdots, \mu_{n-1} = -\beta$, $\mu_n = -\epsilon$, the first term in 6.4 scales like $((n-1)/\epsilon^2 + 1/\beta^2)\log(1/\delta)$ but the second term scales like n/β^2, which is $O(n)$ larger than the first term for small β and fixed δ. Furthermore, we point out that Theorem 6.7 highlights that $(\texttt{ST})^2$ is optimal on these instances up to a log factor! The algorithm we present next, FAREAST, improves $(\texttt{ST})^2$'s dependence on δ and matches the lower bound in Theorem 6.7 for certain instances. Though moderate confidence terms can dominate the sample complexity in practice, few works have focused on understanding their effect.

6.4.1 FAREAST

We focus on the additive case with $\gamma = 0$ in Algorithm 6.4.1, FAREAST, and defer the more general case (multiplicative and $\gamma > 0$) to Algorithm 6.F.1 in the supplementary. FAREAST matches the instance dependent lower bound in Theorem 6.4 as $\delta \to 0$. At a high level, FAREAST (**F**ast **A**rm **R**emoval **E**limination **A**lgorithm for a **S**ampled **T**hreshold) proceeds in rounds r and maintains sets $\widehat{G}_r$ and $\widehat{B}_r$ of arms

thus far declared to be good or bad. It sorts unknown arms into either set through use of a good filter to detect arms in G_ϵ and a bad filter to detect arms in G_ϵ^c.

Good Filter: The good filter is a simple elimination scheme. It maintains an upper bound U_t and lower bound L_t on $\mu_1 - \epsilon$. If an arm's upper bound drops below L_t (line 20), the good filter eliminates that arm, otherwise, if an arm's lower bound rises above U_t (19), the good filter adds the arm to $\widehat{G}_r$, but only eliminates this arm if its upper bound falls below the highest lower bound. This ensures that μ_1 is never eliminated and U_t and L_t are always valid bounds [1]. As the sampling is split across rounds, the good filter always samples the least sampled arm, breaking ties arbitrarily. The number of samples given to the good filter in each round is such that both filters receive identically many samples. This prevents the good filter from over-sampling bad arms and vice versa. In our proof we show that in an unknown round, $\widehat{G}_r = G_\epsilon$, ie all good arms have been found, having used fewer than $O\left(\sum_{i=1}^{n} \max\left\{(\mu_1-\epsilon-\mu_i)^{-2}, (\mu_1+\alpha_\epsilon-\mu_i)^{-2}\right\}\log(n/\delta)\right)$ samples, matching the lower bound.

`FAREAST` cannot yet terminate, however, as it must also verify that any remaining arms are in G_ϵ^c.

Bad Filter: The bad filter removes arms that are not ϵ-good. To show an arm i is in G_ϵ^c, it suffices to find any j such that $\mu_j - \mu_i > \epsilon$. To motivate the idea of lines 9-12, consider the following procedure in the special case where $\beta_i = \mu_1 - \epsilon - \mu_i$ is known. In each round we first run `Median-Elimination`, Even-Dar et al. (2002), with failure probability 1/16, to find an arm $\hat{i}$ that is $\beta_i/2$-good in $O(n/\beta_i^2)$ samples[2]. We then pull both i and $\widehat{i}$ roughly $O(1/\beta_i^2 \log(1/\delta))$ times and can check whether $\mu_{\hat{i}} - \mu_i > \epsilon$ with probability greater than $1-\delta$. This procedure relies on `Median-Elimination` succeeding, which happens with probability 15/16. In the case that it fails and we declare $\mu_{\hat{i}} - \mu_i < \epsilon$, we merely repeat this process until it succeeds– on average $O(1)$ times. This gives an expected sample complexity of $O(n/\beta_i^2 + 1/\beta_i^2 \log(1/\delta))$ for any $i \in G_\epsilon^c$. Of course, β_i is unknown to the algorithm. Instead, in each round r,

[1] This scheme works as an independent algorithm, we analyze it in Appendix 6.F.5.

[2] `Median-Elimination` is used for ease of analysis. One can use `LUCB`Kalyanakrishnan et al. (2012) or another method instead.

the bad filter guesses that $\beta_i \geqslant 2^{-r}$ for all unknown arms $i \notin \widehat{G}_r \cup \widehat{B}_r$ and performs the above procedure. The following theorem demonstrates that this algorithm matches our lower bounds asymptotically as $\delta \to 0$.

Theorem 6.8. *Fix $0 < \epsilon, 0 < \delta < 1/8$, and an instance ν of n arms such that $\max(\Delta_i, |\epsilon - \Delta_i|) \leqslant 8$ for all i. There exists an event E such that $\mathbb{P}(E) \geqslant 1 - \delta$ and on E, FAREAST terminates and returns G_ϵ. Letting T denote the number of samples taken, for a constant c_3*

$$\mathbb{E}[\mathbb{1}_E T] \leqslant \left[c_3 \sum_{i=1}^{n} \max\left\{ \frac{1}{(\mu_1 - \epsilon - \mu_i)^2}, \frac{1}{(\mu_1 + \alpha_\epsilon - \mu_i)^2} \right\} \log\left(\frac{n}{\delta}\right) \right] + c_3 \sum_{i \in G_\epsilon^c} \frac{c'' n}{(\mu_1 - \epsilon - \mu_i)^2}.$$

Additionally for $\gamma \leqslant 16$ FAREAST terminates on E and returns a set $\widehat{G}$ such that $G_\epsilon \subset \widehat{G} \subset G_{\epsilon+\gamma}$ in a number of samples no more than a constant times (6.3), the complexity of (ST)^2.

Algorithm 6.4.1: additive FAREAST with $\gamma = 0$

Input: ϵ, δ, instance ν

Let $\widehat{G}_0 = \emptyset$ be the set of arms declared as good and $\widehat{B}_0 = \emptyset$ the set of arms declared as bad.

Let $\mathcal{A} = [n]$ be the active set, $N_i = 0$ track the total number of samples of arm i by the Good Filter.

Let $t = 0$ denote the total number of times that line 16 is true in the Good Filter.

for $r = 1, 2, \cdots$

  Let $\delta_r = \delta/2r^2$, $\tau_r = \left\lceil 2^{2r+3} \log\left(\frac{8n}{\delta_r}\right)\right\rceil$, Initialize $\widehat{G}_r = \widehat{G}_{r-1}$ and $\widehat{B}_r = \widehat{B}_{r-1}$

  // Bad Filter: find bad arms in G_ϵ^c

  Let $i_r = \texttt{MedianElimination}(\nu, 2^{-r}, 1/16)$, sample i_r τ_r times and compute $\widehat{\mu}_{i_r}$

  for $i \notin \widehat{G}_{r-1} \cup \widehat{B}_{r-1}$**:**

    Sample μ_i τ_r times and compute $\widehat{\mu}_i$

    If $\widehat{\mu}_{i_r} - \widehat{\mu}_i \geqslant \epsilon + 2^{-r+1}$**:** Add i to $\widehat{B}_r$ `// Bad arm detected`

  // Good Filter: find good arms in G_ϵ

  for $s = 1, \cdots, H_{\mathbf{ME}}(n, 2^{-r}, 1/16) + (|(\widehat{G}_{r-1} \cup \widehat{B}_{r-1})^c| + 1)\tau_r$**:**

    Pull arm $I_s \in \arg\min_{j \in \mathcal{A}}\{N_j\}$ and set $N_{I_s} \leftarrow N_{I_s} + 1$.

    if $\min_{j\in\mathcal{A}}\{N_j\} = \max_{j\in\mathcal{A}}\{N_j\}$**:**

      Update $t = t+1$. Let $U_t = \max_{j\in\mathcal{A}} \widehat{\mu}_i(t) + C_{\delta/2n}(t) - \epsilon$ and $L_t = \max_{j\in\mathcal{A}} \widehat{\mu}_i(t) - C_{\delta/2n}(t) - \epsilon$

      for $i \in \mathcal{A}$**:**

        if $\widehat{\mu}_i(t) - C_{\delta/2n}(t) \geqslant U_t$**:** Add i to $\widehat{G}_r$ `// Good arm detected`

        if $\widehat{\mu}_i(t) + C_{\delta/2n}(t) \leqslant L_t$**:** Remove i from $\mathcal{A}$ and add i to $\widehat{B}_r$ `// Bad arms removed`

        if $i \in \widehat{G}_r$ **and** $\widehat{\mu}_i(t) + C_{\delta/2n}(t) \leqslant \max_{j\in\mathcal{A}} \widehat{\mu}(t) - C_{\delta/2n}(t)$**:** `// Good arms removed`

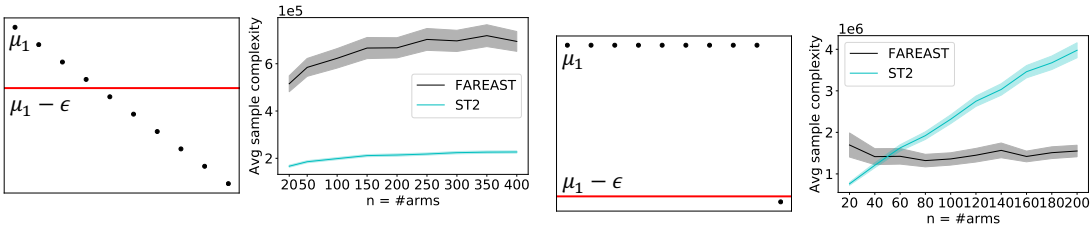
          Remove i from $\mathcal{A}$

        if $\mathcal{A} \subset \widehat{G}_r$ or $\widehat{G}_r \cup \widehat{B}_r = [n]$**: Return the set** $\widehat{G}_r$

6.5 Empirical Performance

We begin by comparing $(\texttt{ST})^2$ and FAREAST on simulated data. FAREAST is asymptotically optimal, but suffers worse constant factors compared to $(\texttt{ST})^2$ [3]. $(\texttt{ST})^2$ is optimal *except* when $\beta_\epsilon \ll \alpha_\epsilon$. We compare $(\texttt{ST})^2$ and FAREAST on two instances in the additive case, shown in Figure 6.3. All arms are Gaussian with $\sigma = 1$. In the first example on the left, $\delta = 0.1$, $\alpha_\epsilon = \beta_\epsilon = 0.05$. Both $(\texttt{ST})^2$ and FAREAST are optimal in this setting; we show the scaling of their sample complexity as the number of arms increases while keeping the threshold, α_ϵ, and β_ϵ constant. In the second example, $\alpha_\epsilon = \epsilon = 0.99$, and $\beta = 0.01$. When $1/\beta_\epsilon^2 \gg n/\epsilon^2$, Theorem 6.4

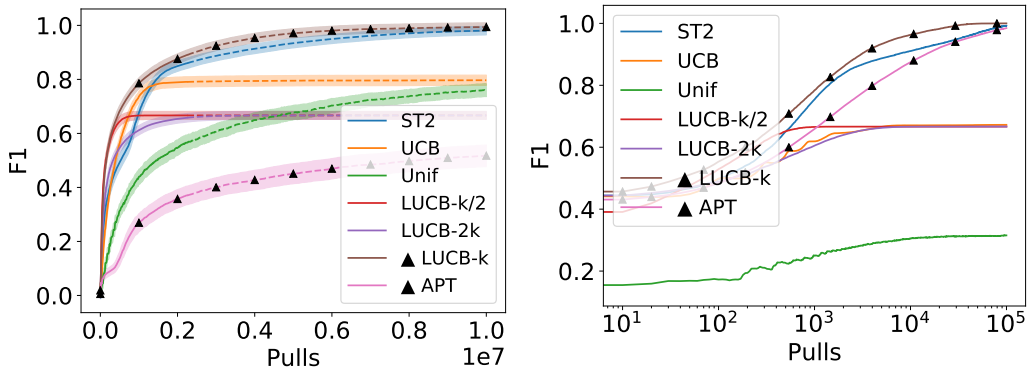
[3] Implementations of all algorithms and baselines used in this chapter are available on GitHub.



(a) A typical setting of diverse means (black dots).

(b) A more challenging setting

Figure 6.3: Comparison of $(\texttt{ST})^2$ and FAREAST averaged over 250 trials plotted with 3 standard errors.



(a) NYCCC with $\epsilon = 0.1$

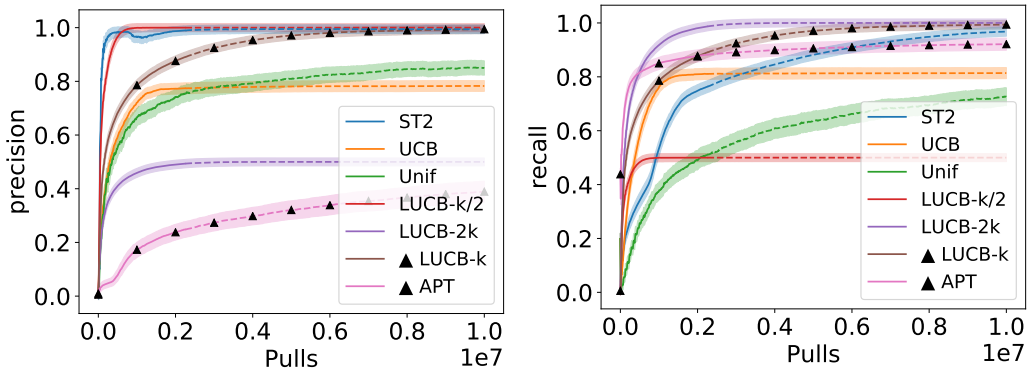
(b) Cancer drug discovery with $\epsilon = 0.8$

Figure 6.4: F1 scores averaged over 600 trials with 95% confidence widths for each dataset.

suggests that $O(1/\beta_\epsilon^2 \log(1/\delta))$ samples are necessary, independent of n. Indeed, in Figure 6.3, for $\delta = 0.01$, the average complexity of FAREAST is constant, but $(\texttt{ST})^2$ scales linearly with n as Theorem 6.5 suggests. Finally, a naive uniform sampling strategy performed very poorly - additional experiments including the uniform sampling method and with $\gamma > 0$ are in the Appendix 6.A.

6.5.1 Finding all ϵ-good arms in real world data – *fast*

As discussed in the introduction, in many applications such as the New Yorker Cartoon Caption Contest (NYCCC), the ALL-ϵ objective returns a set of good arms



(a) Precision curves for $\epsilon = 0.1$

(b) Recall curves for $\epsilon = 0.1$

Figure 6.5: Precision and recall averaged over 600 trials with 95% confidence widths on NYCCC data.

which can then be screened further to choose a favorite. We considered Contest 651, which had 9250 captions whose means we estimated from a total of 2.2 million ratings. We set $\epsilon = 0.1$ and focus on the multiplicative setting, i.e., the objective of recovering all captions within 10% of the funniest one. In this experiment, we contrast $(\texttt{ST})^2$ with several other methods including two *oracle* methods (marked with ▲): `LUCB1` Kalyanakrishnan et al. (2012) with k set to the number of ϵ-good arms (here it was 46), and a threshold-bandit, APT Locatelli et al. (2016) given the value of $0.9\mu_1$. We focus on a common practical requirement, each algorithm's ability to balance precision and recall as it samples. With every new sample, each method recommends an empirical set of ϵ-good arms based on the empirical means, and we consider the F1 score of this set[4]. We focus on the F1 score as it is practically relevant and provides a continuous measure of performance of each method. $\text{F1} = 1$ indicates that an algorithm has found all ϵ-good arms. As can be seen in Figure 6.4a, $(\texttt{ST})^2$ outperforms all baselines including the oracle `APT`, and almost matches the performance of the Top-k oracle! We transition from a solid line to a dashed one at 2.2M pulls to mark the number of samples drawn in the real contest from which we gather the data. To illustrate the importance of knowing the correct value of k, we also plot `LUCB1` given $k = 46/2 = 23$ and $k = 46 \times 2 = 92$, settings where

[4]F1 is the harmonic mean of precision (fraction of captions returned that are actually good) and recall (fraction of all good captions that are actually returned).

the experimenter under or over estimates the number of ϵ good arms by as little as a factor of 2. Both cases result in a poor performance. We have also included UCB, currently being used for the contestTanczos et al. (2017); the plot shows that UCB is not able to estimate the ϵ-good set. In Figure 6.5, we show precision and recall curves for each method on the NYCCC data. $(\texttt{ST})^2$ achieves near-perfect precision quickly, matched only by UCB. APT's poor performance is a consequence of having low-precision, shown in Figure 6.5a. $(\texttt{ST})^2$ achieves high recall more slowly, but is still competitive with other methods. In practical experiments, high precision early on may be more important than high recall, as it guarantees that practitioners can trust the declarations that the algorithm has made, even if some arms are yet to be found. In the Supplementary we show plots for more values of ϵ. Additionally, motivated by drug discovery, we performed an experiment on a dataset Drewry et al. (2017) of 189 inhibitors whose activities were tested against ACVRL1, a kinase associated with cancer Bocci et al. (2019). In this experiment, we use the multiplicative case of ALL-ϵ with $\epsilon = 0.8$ and $\delta = 0.001$, to promote high precision. In this experiment as well, $(\texttt{ST})^2$ performs best (Figure 6.4b), with only the oracle methods are competitive with it. We plot on a log-scale to emphasize the early regime.

6.6 Broader Impacts

The application of machine learning (ML) in domains such as advertising, biology, or medicine brings the possibility of utilizing large computational power and large datasets to solve new problems. It is tempting to use powerful, if not fully understood, ML tools to maximize scientific discovery. However, at times the gap between a tool's theoretical guarantees and its practical performance can lead to sub-optimal behavior. This is especially true in *adaptive data collection* where misspecifying the model or desired output (e.g., "return the top k performing compounds" vs. "return all compounds with a potency about a given threshold") may bias data collection and hinder post-hoc consideration of different objectives. In this chapter we highlight several such instances in real-life data collection using multi-armed

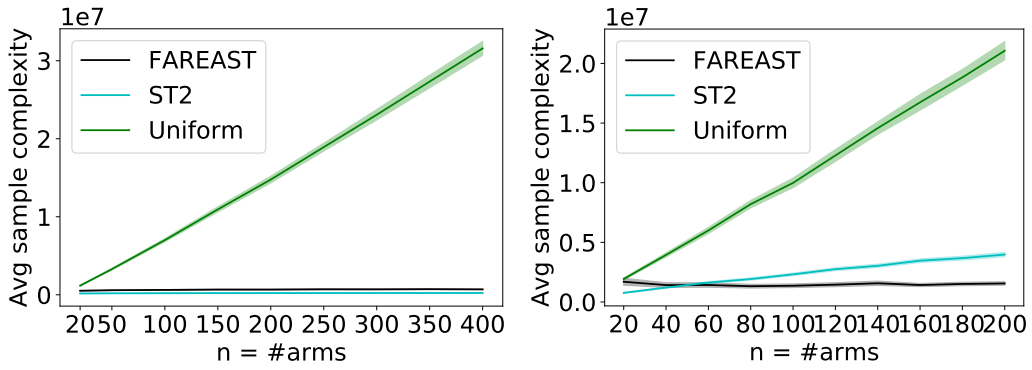
bandits where such a phenomenon occurs. We believe that the objective studied in this work, that of returning all arms whose mean is quantifiably near-best, more naturally aligns with practical objectives as diverse as finding funny captions to performing medical tests. We point out that methods from adaptive data collection and multi-armed bandits can also be used on content-recommendation platforms such as social media or news aggregator sites. In these scenarios, time and again, we have seen that recommendation systems can be greedy, attempting purely to maximize clickthrough with a long term effect of a less informed public. Adjacent to one of the main themes of this chapter, we recommend that practitioners not just focus on the objective of recommendation for immediate profit maximization but rather keep track of a more holistic set of metrics. We are excited to see our work used in practical applications and believe it can have a major impact on driving the process of scientific discovery.

6.A Additional Experimental Results

Practical change made to FAREAST for simulations: We make one change to FAREAST that we recommend for practitioners wishing to use FAREAST that improve its empirical performance. In particular, Median-Elimination may instead be replaced by another method, such as LUCB1, Kalyanakrishnan et al. (2012), to find ϵ-good arms. LUCB1, for instance, has better constant factors and enjoys improved empirical performance versus Median-Elimination. The use of Median-Elimination in this algorithm serves to ease both notation and analysis since it's sample complexity is deterministic. To modify the algorithm, simply track the number of samples given to the bad filter in total, which can be a random variable, and give the good filter the same number in that round. The proof then follows identically, with only the moderate confidence term changing in the result.

Additional Simulations Results As mentioned in the Experiments, Section 6.5, we omitted curves comparing against uniform sampling as they make the plots hard to read with uniform performing much more poorly. For completeness, we include them in Figure 6.A.1. Clearly, uniform sampling performs much more poorly than either active method, as expected.

Additionally, we include experiments with $\gamma > 0$ here. For small γ, the only valid solution is G_ϵ (resp. M_ϵ) itself. However, for larger γ, there are many valid solutions. Indeed, any G such that $G_\epsilon \subset G \subset G_{\epsilon+\gamma}$ is valid. To analyze the effect of γ on both $(\text{ST})^2$ and FAREAST, we consider the same type of instances studied in Figure 6.3b. Here, $n-1$ arms have means equal to μ_1, and a single arm is in G_ϵ^c. Again, we take $\epsilon = 0.99$ and $\beta_\epsilon = 0.01$, and additionally, set $n = 150$ arms. Recall that in this setting, FAREAST outperforms $(\text{ST})^2$, as shown in Figure 6.3b. As we increase γ, the problem becomes easier. We increase γ on an exponential scale, beginning with $\gamma \approx \epsilon/100$ and ending with $\gamma \approx \epsilon/2$. Indeed, for smaller values of γ, FAREAST is superior as it finds the exact solution fastest. For larger γ, $(\text{ST})^2$ is able to terminated more quickly. In Figure 6.A.2 we plot these results.



(a) Plot in Figure 6.3a with uniform sampling included.

(b) Plot in Figure 6.3b with uniform sampling included.

Figure 6.A.1: Simulation results with uniform sampling included.

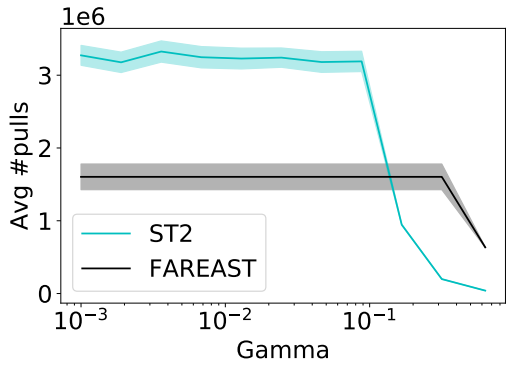


Figure 6.A.2: $(\texttt{ST})^2$ and FAREAST with different values of γ

Metrics we consider for real data experiments: For all methods, we track their precision, recall and F1 score with respect to the true set of ϵ-good arms. To compute these metrics, at each time, the algorithm outputs a set that it guesses are the ϵ-good arms based on the data it has gathered thus far. For UCB, Uniform, and $(\texttt{ST})^2$, this is based directly on empirical means, i.e., $\widehat{G} = \{i : \widehat{\mu}_i \geqslant \max_j \widehat{\mu}_j - \epsilon\}$ or $\widehat{G} = \{i : \widehat{\mu}_i \geqslant \max_j (1-\epsilon)\widehat{\mu}_j\}$ in the multiplicative case. Oracle methods may use their additional information to return the set. In particular, APT returns all arms whose empirical means exceed $(1-\epsilon)\mu_1$ (using knowledge of μ_1) and LUCB1

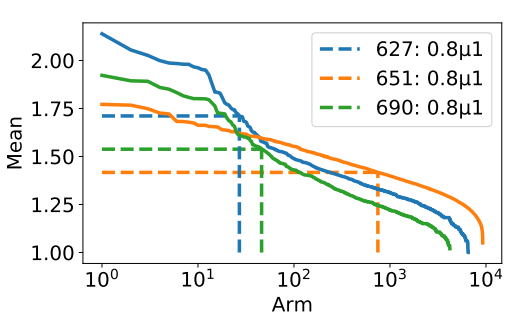
returns the k largest empirical means (using knowledge that $|M_\epsilon| = k$. Let TP (true positives) denote the number of arms that an algorithm declares as ϵ-good that truly are. Let FN (false negatives denote) the number of arms that an algorithm declares as *not* ϵ-good when in fact they are. Recall, $r \in [0,1]$, is computed as $r = \frac{TP}{TP+FN}$. Intuitively, recall is the total number of ϵ-good arms that the algorithm detects. Precision, $p \in [0,1]$, by contrast is the the fraction of the arms that an algorithm predicts as ϵ-good that truly are. It is computed as $p = \max(TP/|\widehat{G}|, 1)$ where the $\max()$ is necessary to avoid the trivial case that $\widehat{G} = \emptyset$. Finally, the F1 is the harmonic mean of precision and recall: $F1 = \frac{2pr}{p+r}$. It balances how precise an algorithm is with how many discoveries it makes. In many cases, F1 may a more relevant metric than the others, as it avoids trivial edge cases. For instance, an algorithm that always declare every arm as ϵ-good independent of the data, achieves perfect recall because it has 0 false negatives. Similarly, an algorithm that never declares any arms as ϵ-good, again independent of data, achieves perfect precision. Both methods, despite seemingly good performance with respect to their individual metrics, are undesirable in practice. In particular, both would achieve low F1 scores.

The New Yorker Caption Contest: In this section we provide additional experimental results adjoining those in Section 6.5. The data can be downloaded at https://github.com/nextml/caption-contest-data. We chose contest 651 for our experiments, but hundreds of others are available. Captions are rated on a scale of 1 to 3 ("unfunny", "somewhat funny", or "funny"). It is desirable to find all captions that are nearly as good as the best. However, setting a fixed number of captions or fraction of captions to accept is undesirable as the number of truly funny captions varies from week to week and represents a small fraction of the submissions. For instance, in the contest that ran the week of 3/14/16, only 8 captions were rated within 20% of the funniest caption. In the following week, by contrast, 187 captions were. Similarly, a choosing a fixed threshold of what it means for a caption to be funny is unrealistic. In the same two contests, first week saw 3% of captions be rated at least 1.5 out of 3 whereas the second saw $< 0.1\%$. For this reason, finding all ϵ-good arms is more natural. We consider finding all multiplicative ϵ-good arms

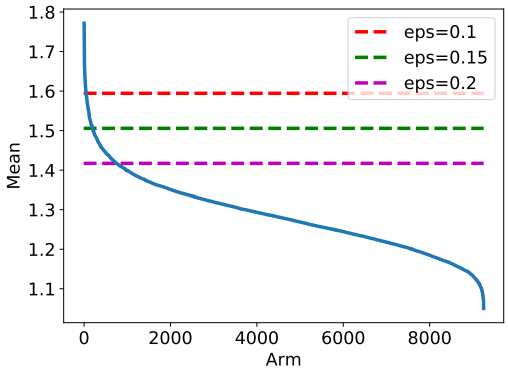
Figure 6.A.3: The user interface for the caption contest with the caption for contest 651. "Unfunny" = 1, "Somewhat funny" = 2, "Funny" = 3

with $\epsilon = 0.1, 0.15, 0.2$. To keep the comparison fair, all methods use the same confidence widths from Howard et al. (2018). In Figure 6.A.4b we plot the average rating of each caption in sorted order with horizontal lines corresponding to $(1-0.2)\mu_1$, $(1-0.15)\mu_1$, and $(1-0.1)\mu_1$. The arms with means above this line are 0.2, 0.15, and 0.1 ϵ-good. The oracle methods tend to achieve high recall, but low precision, and this is especially true for the threshold oracle, APT. In Figures 6.A.5, 6.A.6, 6.A.7 we plot F1, Precision, and Recall curves for all methods tested on $\epsilon = 0.2, 0.15, 0.1$ respectively. As before, all curves are averaged over 600 independent repetitions and plotted with 95% confidence intervals. It is evident from these curves, that $(\text{ST})^2$ performs especially well with regard to precision, though it achieves lower recall than some other baselines.

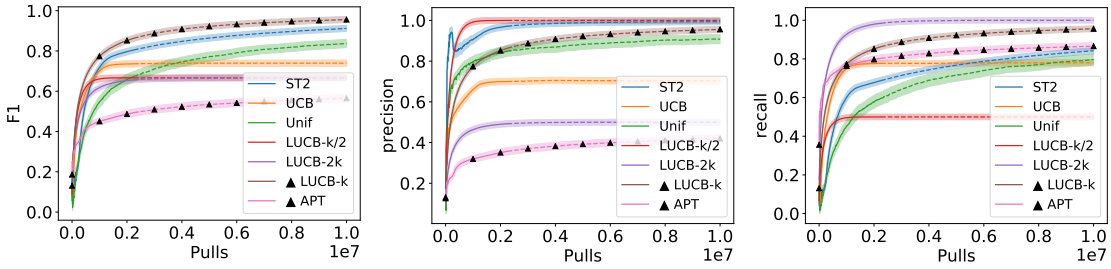
Protein Kinase Inhibitors for Cancer Drug Discovery: Additionally, we consider a second, medically focused experiment. In 2013, researchers at GlaxoSmithKline published a dataset of protein kinase inhibitors different kinases (PKIS1), primarily from humans Dranchak et al. (2013). Kinases are a family of enzymes



(a) Sorted means different contests, 627, 651, 690

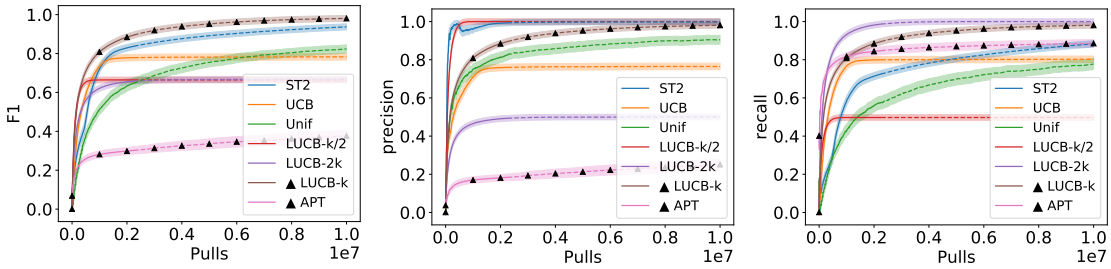


(b) Sorted means for caption contest 651



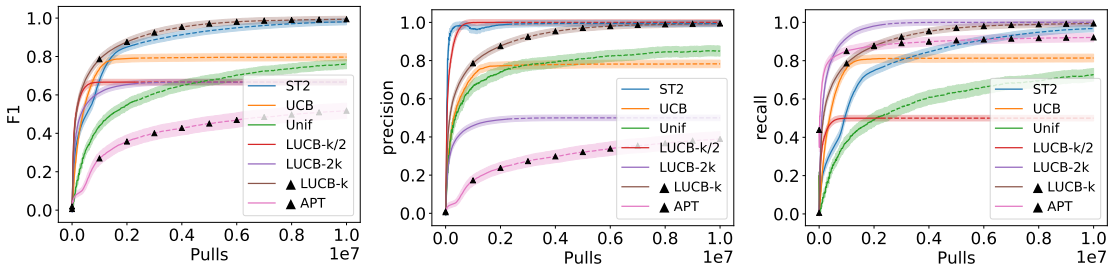
(a) F1 (b) Precision (c) Recall

Figure 6.A.5: F1, Precision, and Recall scores on the New Yorker Caption Contest with $\epsilon = 0.2$



(a) F1 (b) Precision (c) Recall

Figure 6.A.6: F1, Precision, and Recall scores on the New Yorker Caption Contest with $\epsilon = 0.15$

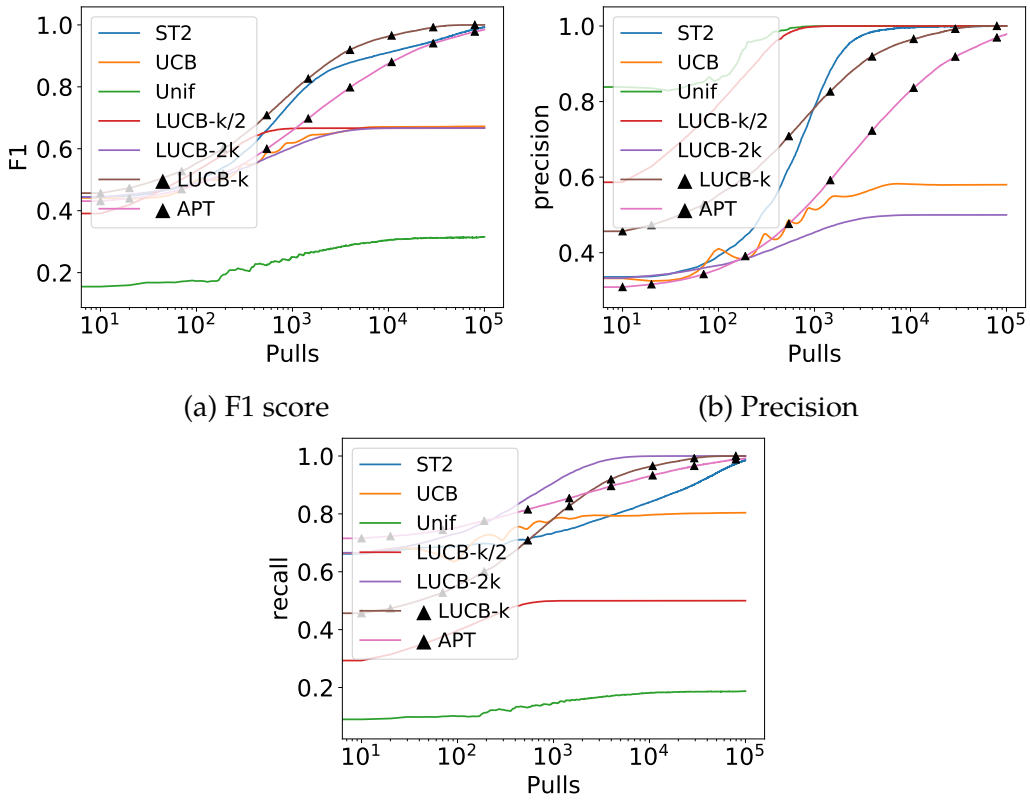


(a) F1 (b) Precision (c) Recall

Figure 6.A.7: F1, Precision, and Recall scores on the New Yorker Caption Contest with $\epsilon = 0.1$

present in many cells and researchers are interested in developing targeted kinase inhibitors to as a new way to treat cancer Christmann-Franck et al. (2016). The dataset contains numerous measures of how strongly each inhibitor reacts with each kinase. A second, larger dataset (PKIS2) was expanded on by Drewry et al. (2017)[5]. For the purpose of our experiment, we selected a single Kinase in the dataset, ACVRL1, which researchers have linked to numerous types of cancer, most prominently bladder and prostate cancers Bocci et al. (2019). PKIS2 contains 641 different compounds that were tested as being potential kinase inhibitors, though not every compound was tested against every kinase. In particular, 189 were tested against ACVRL1. For each compound, there is an associated average "percent inhibition" that is reported. All numbers are between 0 and 1 and averaged across multiple trials in a single assay. We subtract each number from 1 to compute the percent control, representing how effective any method is relative to a control, an important metric for estimating how effective that compound is against the target, ACRVL1. A meta-analysis, done by Christmann-Franck et al. (2016), reported that these values have log-normal distributions with variance less than 1. Therefore, we compute the log of each percent control and may sample from a normal distribution with that mean and variance 1. As before, we plot F1, precision, and recall for all methods. To simulate being in a medical research regime where a higher level of

[5]The dataset can be downloaded at the following link: https://doi.org/10.1371/journal.pone.0181585.s004.



(a) F1 score

(b) Precision

(c) Recall

Figure 6.A.8: Precision and Recall curves for the PKIS2 cancer drug discovery experiment with $\epsilon = 0.8$

precision is often desired, we take $\delta = 0.001$. We test each method on returning all multiplicative ϵ-good arms with $\epsilon = 0.8$ and plot the results in Figure 6.A.8. Note that these curves are plotted on a log-scale to emphasize the early regime of this experiment. It is likewise true here that the oracle baselines perform better on recall than they do on precision. $(\texttt{ST})^2$ again performs well with respect to precision, and is more competitive with respect to recall in this experiment. Finally, $(\texttt{ST})^2$ is competitive versus oracle methods on F1 score and greatly outperforms UCB and uniform sampling.

6.B $(\mathtt{ST})^2$, An optimism based algorithm for all-ϵ

Algorithm 6 The $(\mathtt{ST})^2$ Algorithm

Require: Instance ν, $\epsilon > 0$, $\delta \in (0, 1/2]$, $\gamma \geqslant 0$ ($\epsilon \in (0, 1/2]$, and $\gamma \in [0, \min(16/\mu_1, 1/2)]$)

Pull each arm once, initialize $T_i \leftarrow 1$, update $\hat{\mu}_i$ for each $i \in \{1, 2, \ldots, n\}$

Empirically good arms: $\widehat{G} = \{i : \hat{\mu}_i \geqslant \max_j \hat{\mu}_j - \epsilon\}$ or $\widehat{G} = \{i : \hat{\mu}_i \geqslant (1 - \epsilon) \max_j \hat{\mu}_j\}$

$U_t = \max_j \hat{\mu}_j(T_j) + C_{\delta/n}(T_j) - \epsilon - \gamma$ or $U_t = (1 - \epsilon - \gamma)\left(\max_j \hat{\mu}_j(t) + C_{\delta/n}(T_j)\right)$

$L_t = \max_j \hat{\mu}_j(T_j) - C_{\delta/n}(T_j) - \epsilon$ or $L_t = (1 - \epsilon)\left(\max_j \hat{\mu}_j(t) - C_{\delta/n}(T_j)\right)$

Known arms: $K = \{i : \hat{\mu}_i(T_i) + C_{\delta/n}(T_i) < L_t \text{ or } \hat{\mu}_i(T_i) - C_{\delta/n}(T_i) > U_t\}$

while $K \neq [n]$ **do**

  Pull arm $i_1(t) = \arg\min_{i \in \widehat{G} \setminus K} \hat{\mu}_i(T_i) - C_{\delta/n}(T_i)$, update $T_{i_1}, \hat{\mu}_{i_1}$

  Pull arm $i_2(t) = \arg\max_{i \in \widehat{G}_\epsilon^c \setminus K} \hat{\mu}_i(T_i) + C_{\delta/n}(T_i)$, update $T_{i_2}, \hat{\mu}_{i_2}$

  Pull arm $i^*(t) = \arg\max_i \hat{\mu}_i(T_i) + C_{\delta/n}(T_i)$, update $T_{i^*}, \hat{\mu}_{i^*}$

  Update bounds L_t, U_t, sets $\widehat{G}$, K

end whilereturn The set of good arms $\{i : \hat{\mu}_i(T_i) - C_{\delta/n}(T_i) > U_t\}$

6.B.1 Optimism with additive γ

Theorem 6.9. *Fix $\epsilon \geqslant 0$, $0 < \delta \leqslant 1/2$, $\gamma \in [0, 16]$ and an instance ν such that $\max(\Delta_i, |\epsilon - \Delta_i|) \leqslant 8$ for all i. In the case that $G_\epsilon = [n]$, let $\alpha_\epsilon = \min(\alpha_\epsilon, \beta_\epsilon)$. With probability at least $1 - \delta$, $(ST)^2$ correctly returns a set G such that $G_\epsilon \subset G \subset G_{\epsilon+\gamma}$ in at most*

$$12 \sum_{i=1}^{n} \min \left\{ \max \left\{ \frac{1024}{(\mu_1 - \epsilon - \mu_i)^2} \log\left(\frac{2n}{\delta} \log_2\left(\frac{3072n}{\delta(\mu_1 - \epsilon - \mu_i)^2}\right)\right), \right.\right.$$
$$\frac{4096}{(\mu_1 + \alpha_\epsilon - \mu_i)^2} \log\left(\frac{2n}{\delta} \log_2\left(\frac{12288n}{\delta(\mu_1 + \alpha_\epsilon - \mu_i)^2}\right)\right),$$
$$\left.\left. \frac{4096}{(\mu_1 + \beta_\epsilon - \mu_i)^2} \log\left(\frac{2n}{\delta} \log_2\left(\frac{12288n}{\delta(\mu_1 + \beta_\epsilon - \mu_i)^2}\right)\right)\right\}, \right.$$
$$\left. \frac{1}{\gamma^2} \log\left(\frac{2n}{\delta} \log_2\left(\frac{3072n}{\delta\gamma^2}\right)\right)\right\}$$

samples.

Proof. Throughout the proof, recall that $\Delta_i = \mu_1 - \mu_i$ for all i, $\alpha_\epsilon = \min_{i \in G_\epsilon} \mu_i - (\mu_1 - \epsilon)$, and $\beta_\epsilon = \min_{i \in G_\epsilon^c} (\mu_1 - \epsilon) - \mu_i$. Additionally, at any time t, we will take $T_j(t)$ to denote the number of samples of arm j up to time t.

Define the event

$$\mathcal{E} = \left\{ \bigcap_{i \in [n]} \bigcap_{t \in \mathbb{N}} |\hat{\mu}_i(t) - \mu_i| \leqslant C_{\delta/n}(t) \right\}.$$

Using standard anytime confidence bound results, and recalling that that $C_\delta(t) := \sqrt{\frac{4 \log(\log_2(2t)/\delta)}{t}}$, we have

$$\begin{aligned} \mathbb{P}(\mathcal{E}^c) &= \mathbb{P}\left(\bigcup_{i \in [n]} \bigcup_{t \in \mathbb{N}} |\hat{\mu}_i - \mu_i| > C_{\delta/n}(t) \right) \\ &\leqslant \sum_{i=1}^{n} \mathbb{P}\left(\bigcup_{t \in \mathbb{N}} |\hat{\mu}_i - \mu_i| > C_{\delta/n}(t) \right) \leqslant \sum_{i=1}^{n} \frac{\delta}{n} = \delta \end{aligned}$$

Hence, $\mathbb{P}(\mathcal{E}) \geqslant 1 - \delta$. Throughout, we will make use of a function $h(x, \delta)$ such that if $t \geqslant h(x, \delta)$, then $C_\delta(t) \leqslant |x|$. We bound $h(\cdot, \cdot)$ in Lemma 6.32. $h(\cdot, \cdot)$ is assumed to decrease monotonically in both arguments and is symmetric in its first argument.

6.B.1.1 Step 0: Correctness

We begin by showing that on $\mathcal{E}$, if $(\mathtt{ST})^2$ terminates, it returns a set G such that $G_\epsilon \subset G \subset G_{\epsilon+\gamma}$. Since $\mathbb{P}(\mathcal{E}) \geqslant 1 - \delta$, this implies that $(\mathtt{ST})^2$ is correct with high probability.

Claim 0: On Event $\mathcal{E}$, at all times t, $U_t \geqslant \mu_1 - \epsilon - \gamma$.

Proof.

$$U_t = \max_j \hat{\mu}_j(T_j(t)) + C_{\delta/n}(T_j(t)) - \epsilon - \gamma \geqslant \hat{\mu}_1(T_1(t)) + C_{\delta/n}(T_1(t)) - \epsilon - \gamma$$

$$\overset{\mathcal{E}}{\geqslant} \mu_1 - \epsilon - \gamma$$

□

Claim 1: On Event $\mathcal{E}$, at all times t, $L_t \leqslant \mu_1 - \epsilon$.

Proof.

$$L_t = \max_j \hat{\mu}_j(T_j(t)) - C_{\delta/n}(T_j(t)) - \epsilon \overset{\mathcal{E}}{\leqslant} \max_j \mu_j - \epsilon = \mu_1 - \epsilon$$

□

Claim 2: On event $\mathcal{E}$, if there is a time t such that $\hat{\mu}_i(T_i(t)) - C_{\delta/n}(T_i(t)) > U_t$, then $i \in G_{\epsilon+\gamma}$.

Proof. Assume for some t, $\hat{\mu}_i(T_i(t)) - C_{\delta/n}(T_i(t)) > U_t$. Then

$$\mu_i \overset{\mathcal{E}}{\geqslant} \hat{\mu}_i(T_i(t)) - C_{\delta/n}(T_i(t)) \geqslant U_t \overset{\text{Claim 0}}{\geqslant} \mu_1 - \epsilon - \gamma$$

which implies $i \in G_{\epsilon+\gamma}$ □

Claim 3: On event $\mathcal{E}$, if there is a time t such that $\hat{\mu}_i(T_i(t)) + C_{\delta/n}(T_i(t)) < L_t$, then $i \in G_\epsilon^c$.

Proof. Assume that is a t for which $\hat{\mu}_i(T_i(t)) + C_{\delta/n}(T_i(t)) < L_t$. Then

$$\mu_i \overset{\mathcal{E}}{\leqslant} \hat{\mu}_i(T_i(t)) + C_{\delta/n}(T_i(t)) \leqslant L_t \overset{\text{Claim 1}}{\leqslant} \mu_1 - \epsilon$$

which implies $i \in G_\epsilon^c$. □

$(\texttt{ST})^2$ terminates at any time t such that simultaneously for all arms i, either $\hat{\mu}_i(T_i(t)) + C_{\delta/n}(T_i(t)) > U_t$ or $\hat{\mu}_i(T_i(t)) - C_{\delta/n}(T_i(t)) < L_t$. On $\mathcal{E}$, by Claim 3, $G_\epsilon \subset \{i : \hat{\mu}_i(T_i(t)) + C_{\delta/n}(T_i(t)) > U_t\}$. On $\mathcal{E}$, by Claim 2, $\{i : \hat{\mu}_i(T_i(t)) + C_{\delta/n}(T_i(t)) > U_t\} \subset G_{\epsilon+\gamma}$. Hence, on the event $\mathcal{E}$. $(\texttt{ST})^2$ returns a set G such that $G_\epsilon \subset G \subset G_{\epsilon+\gamma}$.

6.B.1.2 Step 1: Complexity of estimating the threshold, $\mu_1 - \epsilon$

Let STOP denote the termination event that for all arms i, either $\hat{\mu}_i(T_i(t))+C_{\delta/n}(T_i(t)) > U_t$ or $\hat{\mu}_i(T_i(t)) - C_{\delta/n}(T_i(t)) < L_t$. Let ω denote the quantity

$$\omega := \max\{\gamma, \min(\alpha_\epsilon, \beta_\epsilon)\}.$$

Let T denote the random variable of the total number of rounds before $(\mathtt{ST})^2$ terminates. At most 3 samples are drawn in any round. Hence, the total sample complexity is bounded by $3T$. We may write T as

$$T := |\{t : \neg\text{STOP}\}| = |\{t : \neg\text{STOP and } i^* \notin G_\omega\}| + |\{t : \neg\text{STOP and } i^* \in G_\omega\}|$$

Next, we bound the first event in this decomposition.

Claim 0: On $\mathcal{E}$,
$|\{t : \neg\text{STOP and } i^* \notin G_\omega\}| \leqslant \sum_{i\in G_\omega^c} \min\left\{h\left(\frac{\gamma}{2}, \frac{\delta}{n}\right), \min\left[h\left(\frac{\Delta_i}{2}, \frac{\delta}{n}\right), h\left(\frac{\min(\alpha_\epsilon,\beta_\epsilon)}{2}, \frac{\delta}{n}\right)\right]\right\}$.

Proof. If for each $i \in G_\omega^c$, $\mu_i + 2C_{\delta/n}(T_i(t)) < \mu_1$ is true, which is ensured when $T_i(t) > h\left(\Delta_i/2, \frac{\delta}{n}\right)$ for all $i \in G_\omega^c$, then

$$\hat{\mu}_i(T_i(t)) + C_{\delta/n}(T_i(t)) \overset{\mathcal{E}}{\leqslant} \mu_i + 2C_{\delta/n}(T_i(t)) < \mu_1 \overset{\mathcal{E}}{\leqslant} \hat{\mu}_1(T_1(t)) + C_{\delta/n}(T_1(t))$$

which implies that $i \neq i^*$. Additionally, since $i \in G_\omega^c$ by assumption, we have that $\mu_1 - \omega - \mu_i \geqslant 0$, which reduces to $\Delta_i \geqslant \omega$. Since $\omega = \max(\gamma, \min(\alpha_\epsilon, \beta_\epsilon))$, it is likewise true that

$$h\left(\frac{\Delta_i}{2}, \frac{\delta}{n}\right) = \min\left[h\left(\frac{\gamma}{2}, \frac{\delta}{n}\right), \min\left\{h\left(\frac{\Delta_i}{2}, \frac{\delta}{n}\right), h\left(\frac{\min(\alpha_\epsilon, \beta_\epsilon)}{2}, \frac{\delta}{n}\right)\right]\right\}.$$

Summing over all $i \in G_\omega^c$ achieves the result. □

We may decompose the set $\{t : \neg\text{STOP and } i^* \in G_\omega\}$ as

$$\left\{t : \neg\text{STOP and } i^* \in G_\omega \text{ and } C_{\delta/n}(T_{i^*}(t)) > \frac{\omega}{16}\right\}$$
$$\cup \left\{t : \neg\text{STOP and } i^* \in G_\omega \text{ and } C_{\delta/n}(T_{i^*}(t)) \leqslant \frac{\omega}{16}\right\}$$

Claim 1: $\left|\left\{t : \neg\text{STOP and } i^* \in G_\omega \text{ and } C_{\delta/n}(T_{i^*}(t)) > \frac{\omega}{16}\right\}\right|$
$\leqslant \sum_{i\in G_\omega} \min\left\{h\left(\frac{\gamma}{16}, \frac{\delta}{n}\right), \min\left[h\left(\frac{\Delta_i}{8}, \frac{\delta}{n}\right), h\left(\frac{\min(\alpha_\epsilon,\beta_\epsilon)}{16}, \frac{\delta}{n}\right)\right]\right\}$

Proof. $C_{\delta/n}(T_i(t)) \leqslant \frac{\omega}{16}$ is true when $T_i(t) \geqslant h\left(\frac{\omega}{16}, \frac{\delta}{n}\right)$. Since $i^* \in G_\omega$, $\mu_i - (\mu_1 - \omega) \geqslant 0$, which implies $\Delta_i \leqslant \omega$. By definition, $\omega = \min(\gamma, \min(\alpha_\epsilon, \beta_\epsilon))$. Hence, by monotonicity of $h(\cdot,\cdot)$,

$$\begin{aligned} h\left(\frac{\omega}{16}, \frac{\delta}{n}\right) &= \min\left[h\left(\frac{\Delta_i}{16}, \frac{\delta}{n}\right), h\left(\frac{\omega}{16}, \frac{\delta}{n}\right)\right] \\ &= \min\left\{h\left(\frac{\gamma}{16}, \frac{\delta}{n}\right), \min\left[h\left(\frac{\Delta_i}{16}, \frac{\delta}{n}\right), h\left(\frac{\min(\alpha_\epsilon, \beta_\epsilon)}{16}, \frac{\delta}{n}\right)\right]\right\} \end{aligned}$$

Summing over all $i \in G_\omega$ achieves the desired result. □

6.B.1.3 Step 2: Controlling "crossing" events

Recall that we sample $i_1(t) \in \widehat{G}$ and $i_2(t) \in \widehat{G}^c$. In this section, we control the number of times that $i_1(t) \in G^c_{\epsilon+\frac{\gamma}{2}}$ and $i_2(t) \in G_{\epsilon+\frac{\gamma}{2}}$.

To do so, we first decompose the set $\left\{t : \neg\text{STOP and } i^* \in G_\omega \text{ and } C_{\delta/n}(T_{i^*}(t)) \leqslant \frac{\omega}{16}\right\}$ as

$$\begin{aligned} &\left\{t : \neg\text{STOP and } i^* \in G_\omega \text{ and } C_{\delta/n}(T_{i^*}(t)) \leqslant \frac{\omega}{16} \text{ and } i_1(t) \in G^c_{\epsilon+\frac{\gamma}{2}}\right\} \\ &\quad \cup \left\{t : \neg\text{STOP and } i^* \in G_\omega \text{ and } C_{\delta/n}(T_{i^*}(t)) \leqslant \frac{\omega}{16} \text{ and } i_1(t) \in G_{\epsilon+\frac{\gamma}{2}}\right\} \end{aligned}$$

Claim 0: $\left|\left\{t : \neg\text{STOP and } i^* \in G_\omega \text{ and } C_{\delta/n}(T_{i^*}(t)) \leqslant \frac{\omega}{16} \text{ and } i_1(t) \in G^c_{\epsilon+\frac{\gamma}{2}}\right\}\right| \leqslant$
$\sum_{i\in G^c_{\epsilon+\frac{\gamma}{2}}} \min\left[h\left(\frac{\Delta_i-\epsilon}{8}, \frac{\delta}{n}\right), h\left(\frac{\gamma}{8}, \frac{\delta}{n}\right)\right]$.

Proof. Recall that $\widehat{G}$ is the set of all arms whose empirical means exceed $\max_i \hat{\mu}_i(T_i(t)) - \epsilon$, and $i_1(t) \in \widehat{G}$ by definition. Note that $\max_i \hat{\mu}_i(T_i(t)) - \epsilon > \max_i \hat{\mu}_i(T_i(t)) - C_{\delta/n}(T_i(t)) - \epsilon = L_t$. Hence, if an arm's upper bound is below L_t, then the arm cannot be in $\widehat{G}$ and thus not be $i_1(t)$. By the above event, $C_{\delta/n}(T_{i^*}(t)) \leqslant \frac{\omega}{16}$. Hence,

$$\mu_i^* + \frac{\omega}{8} \geqslant \mu_i^* + 2C_{\delta/n}(T_{i^*}(t)) \overset{\mathcal{E}}{\geqslant} \hat{\mu}_i^*(T_{i^*}(t)) + C_{\delta/n}(T_{i^*}(t)) \geqslant \hat{\mu}_1(T_1(t)) + C_{\delta/n}(T_1(t)) \overset{\mathcal{E}}{\geqslant} \mu_1.$$

Therefore, $\mu_{i^*} \geqslant \mu_1 - \frac{\omega}{8}$ or equivalently, $i^* \in G_{\omega/8}$. Using this,

$$\begin{aligned} L_t = \max_i \hat{\mu}_i(T_i(t)) - C_{\delta/n}(T_i(t)) - \epsilon &\geqslant \hat{\mu}_{i^*}(T_{i^*}(t)) - C_{\delta/n}(T_{i^*}(t)) - \epsilon \\ &\overset{\mathcal{E}}{\geqslant} \mu_{i^*} - 2C_{\delta/n}(T_{i^*}(t)) - \epsilon \\ &\overset{\mathcal{E}}{\geqslant} \mu_{i^*} - \frac{\omega}{8} - \epsilon \\ &\geqslant \mu_1 - \frac{\omega}{4} - \epsilon \end{aligned}$$

Next, we bound the number of times an arm $i \in G^c_{\epsilon+\frac{\gamma}{2}}$ is sampled before its upper bound is below $\mu_1 - \frac{\omega}{4} - \epsilon$. Note that $C_{\delta/n}(T_i(t)) < \frac{1}{2}\left(\mu_1 - \frac{\omega}{4} - \epsilon - \mu_i\right)$, true when $T_i(t) > h\left(\frac{1}{2}\left(\mu_1 - \frac{\omega}{4} - \epsilon - \mu_i\right), \frac{\delta}{n}\right)$ implies that

$$\hat{\mu}_i(T_i(t)) + C_{\delta/n}(T_i(t)) \overset{\mathcal{E}}{\leqslant} \mu_i + 2C_{\delta/n}(T_i(t)) < \mu_1 - \frac{\omega}{4} - \epsilon \leqslant L_t.$$

Finally, we turn our attention to the difference $\mu_1 - \frac{\omega}{4} - \epsilon - \mu_i$. Recall that $\omega = \max(\gamma, \min(\alpha_\epsilon, \beta_\epsilon))$.

$$\begin{aligned} \mu_1 - \frac{\omega}{4} - \epsilon - \mu_i &= (\mu_1 - \epsilon) - \mu_i - \frac{1}{4}\omega \\ &= (\mu_1 - \epsilon) - \mu_i - \frac{1}{4}\max(\gamma, \min(\alpha_\epsilon, \beta_\epsilon)). \end{aligned}$$

By definition, $\beta_\epsilon = \min_{i \in G^c_\epsilon}(\mu_1 - \epsilon) - \mu_i$. Hence, $\min(\alpha_\epsilon, \beta_\epsilon) \leqslant (\mu_1 - \epsilon) - \mu_i$ for all $i \in G^c_{\epsilon+\frac{\gamma}{2}}$. Similarly, since $i \in G^c_{\epsilon+\frac{\gamma}{2}}$ by assumption, $(\mu_1 - \epsilon - \frac{\gamma}{2}) - \mu_i \geqslant 0$, which rearranges to $\frac{\gamma}{2} \leqslant (\mu_1 - \epsilon) - \mu_i$. Therefore,

$$(\mu_1 - \epsilon) - \mu_i - \frac{1}{4}\max(\gamma, \min(\alpha_\epsilon, \beta_\epsilon)) \geqslant \frac{1}{2}\left((\mu_1 - \epsilon) - \mu_i\right) = \frac{\Delta_i - \epsilon}{2}.$$

Hence, by monotonicity of $h(\cdot,\cdot)$,

$$h\left(\frac{1}{2}\left(\mu_1 - \frac{\omega}{4} - \epsilon - \mu_i\right), \frac{\delta}{n}\right) \leqslant h\left(\frac{\Delta_i - \epsilon}{4}, \frac{\delta}{n}\right).$$

Lastly, as above, since $i \in G^c_{\epsilon+\frac{\gamma}{2}}$, we have that $\Delta_i - \epsilon = (\mu_1 - \epsilon) - \mu_i \geqslant \frac{1}{2}\gamma$. Hence,

$$h\left(\frac{\Delta_i - \epsilon}{4}, \frac{\delta}{n}\right) \leqslant \min\left[h\left(\frac{\Delta_i - \epsilon}{8}, \frac{\delta}{n}\right), h\left(\frac{\gamma}{8}, \frac{\delta}{n}\right)\right].$$

Putting this together, if $T_i(t) \geqslant \min\left[h\left(\frac{\Delta_i-\epsilon}{8}, \frac{\delta}{n}\right), h\left(\frac{\gamma}{8}, \frac{\delta}{n}\right)\right]$, then $i \neq i_1(t)$ for all $i \in G^c_{\epsilon+\frac{\gamma}{2}}$. Summing over all such i bounds the size of set stated in the claim. □

We decompose the remaining event

$$\left\{t : \neg\text{STOP and } i^* \in G_\omega \text{ and } C_{\delta/n}(T_{i^*}(t)) \leqslant \frac{\omega}{16} \text{ and } i_1(t) \in G_{\epsilon+\frac{\gamma}{2}}\right\}$$

as

$$\left\{t : \neg\text{STOP and } i^* \in G_\omega \text{ and } C_{\delta/n}(T_{i^*}(t)) \leqslant \frac{\omega}{16} \text{ and } i_1(t) \in G_{\epsilon+\frac{\gamma}{2}} \text{ and } i_2(t) \in G_{\epsilon+\frac{\gamma}{2}}\right\}$$
$$\cup \left\{t : \neg\text{STOP and } i^* \in G_\omega \text{ and } C_{\delta/n}(T_{i^*}(t)) \leqslant \frac{\omega}{16} \text{ and } i_1(t) \in G_{\epsilon+\frac{\gamma}{2}} \text{ and } i_2(t) \in G^c_{\epsilon+\frac{\gamma}{2}}\right\}.$$

We proceed by bounding the size of the first set.

Claim 1:

$$\left|\left\{t : \neg\text{STOP and } i^* \in G_\omega \text{ and } C_{\delta/n}(T_{i^*}(t)) \leqslant \frac{\omega}{16} \text{ and } i_1(t) \in G_{\epsilon+\frac{\gamma}{2}} \text{ and } i_2(t) \in G_{\epsilon+\frac{\gamma}{2}}\right\}\right|$$
$$\leqslant \sum_{i \in G_{\epsilon+\frac{\gamma}{2}}} \min\left[h\left(\frac{\epsilon\Delta_i}{8}, \frac{\delta}{n}\right), h\left(\frac{\gamma}{8}, \frac{\delta}{n}\right)\right]$$

Proof. Recall that $K = \{i : \hat{\mu}(T_i(t)) + C_{\delta/n}(T_i(t)) < L_t \text{ or } \hat{\mu}(T_i(t)) - C_{\delta/n}(T_i(t)) > L_t\}$ and i_2 is sampled from the set $\widehat{G}^c \backslash K$, ie all arms in $\widehat{G}^c$ who have not been declared as above U_t or below L_t. Hence, if an arm's lower bound exceeds $U_t = \max_i \hat{\mu}(T_i(t)) + C_{\delta/n}(T_i(t)) - \epsilon - \gamma$, it must be in K an thus cannot be i_2. Recall

that $i^*(t) = \arg\max \hat{\mu}_i(T_i(t)) + C_{\delta/n}(T_i(t))$. By the above event, $i^*(t) \in G_\omega$ and $C_{\delta/n}(T_{i^*}(t)) \leqslant \frac{\omega}{16}$. Hence,

$$\begin{aligned} U_t = \max_i \hat{\mu}_i(T_i(t)) + C_{\delta/n}(T_i(t)) - \epsilon - \gamma &= \hat{\mu}_{i^*(t)}(T_{i^*(t)}(t)) + C_{\delta/n}(T_{i^*(t)}(t)) - \epsilon - \gamma \\ &\overset{\mathcal{E}}{\leqslant} \mu_{i^*(t)} + 2C_{\delta/n}(T_{i^*(t)}(t)) - \epsilon - \gamma \\ &\leqslant \mu_{i^*(t)} + \frac{\omega}{8} - \epsilon - \gamma \\ &\leqslant \mu_1 + \frac{\omega}{8} - \epsilon - \gamma \end{aligned}$$

Next, we bound the number of times an arm $i \in G_{\epsilon+\frac{\gamma}{2}}$ is sampled before its lower bound is above $\mu_1 + \frac{\omega}{8} - \epsilon - \gamma$. Note that $C_{\delta/n}(T_i(t)) < \frac{1}{2}\left(\mu_i - (\mu_1 + \frac{\omega}{8} - \epsilon - \gamma)\right)$, true when $T_i(t) > h\left(\frac{1}{2}\left(\mu_i - (\mu_1 + \frac{\omega}{8} - \epsilon - \gamma)\right), \frac{\delta}{n}\right)$ implies that

$$\hat{\mu}_i(T_i(t)) - C_{\delta/n}(T_i(t)) \overset{\mathcal{E}}{\geqslant} \mu_i - 2C_{\delta/n}(T_i(t)) > \mu_1 + \frac{\omega}{8} - \epsilon - \gamma.$$

Finally, we turn our attention to the difference $\mu_i - (\mu_1 + \frac{\omega}{8} - \epsilon - \gamma)$. Recall that $\omega = \max(\gamma, \min(\alpha_\epsilon, \beta_\epsilon))$.

$$\mu_i - \left(\mu_1 + \frac{\omega}{8} - \epsilon - \gamma\right) = \mu_i - (\mu_1 - \epsilon) + \gamma - \frac{1}{8}\omega$$

Case 1a, $\omega = \min(\alpha_\epsilon, \beta_\epsilon)$ and $i \in G_\epsilon$:.

By definition, $\alpha_\epsilon = \min_{i \in G_\epsilon} \mu_i - (\mu_1 - \epsilon)$. Hence, $\min(\alpha_\epsilon, \beta_\epsilon) \leqslant \mu_i - (\mu_1 - \epsilon)$ for all $i \in G_\epsilon$. Therefore,

$$\begin{aligned} \mu_i - (\mu_1 - \epsilon) + \gamma - \frac{1}{8}\omega &= \mu_i - (\mu_1 - \epsilon) + \gamma - \frac{1}{8}\min(\alpha_\epsilon, \beta_\epsilon) \\ &\geqslant \max\left(\mu_i - (\mu_1 - \epsilon) - \frac{1}{8}\min(\alpha_\epsilon, \beta_\epsilon), \gamma\right) \\ &\geqslant \max\left(\frac{7}{8}(\mu_i - (\mu_1 - \epsilon)), \gamma\right) \end{aligned}$$

Case 1b, $\omega = \min(\alpha_\epsilon, \beta_\epsilon)$ and $i \in G_\epsilon^c \cap G_{\epsilon+\frac{\gamma}{2}}$

Since $\omega = \max(\gamma, \min(\alpha_\epsilon, \beta_\epsilon))$, if $\omega = \min(\alpha_\epsilon, \beta_\epsilon)$, then $\frac{1}{2}\gamma < \min(\alpha_\epsilon, \beta_\epsilon)$.

Since $\min(\alpha_\epsilon, \beta_\epsilon) = \min|\mu_i - (\mu_1 - \epsilon)|$, the set $G_\epsilon^c \cap G_{\epsilon+\frac{\gamma}{2}}$ is empty and there is nothing to prove.

Case 2a, $\omega = \gamma$ and $i \in G_\epsilon$:

$$\mu_i - (\mu_1 - \epsilon) + \gamma - \frac{1}{8}\omega = \mu_i - (\mu_1 - \epsilon) + \frac{7}{8}\gamma \geqslant \max\left(\mu_i - (\mu_1 - \epsilon), \frac{7}{8}\gamma\right)$$

Case 2b, $\omega = \gamma$ and $i \in G_\epsilon^c \cap G_{\epsilon+\frac{\gamma}{2}}$:

For $i \in G_\epsilon^c \cap G_{\epsilon+\frac{\gamma}{2}}$, we have that $\mu_i - (\mu_1 - \epsilon - \gamma/2) \geqslant 0$. Hence $\mu_i - (\mu_1 - \epsilon) \geqslant \frac{-\gamma}{2}$. Therefore,

$$\mu_i - (\mu_1 - \epsilon) + \gamma - \frac{1}{8}\omega \geqslant \frac{3}{8}\gamma = \max\left(\frac{3}{8}((\mu_1 - \epsilon) - \mu_i), \frac{3}{8}\gamma\right).$$

Applying the above cases and using monotonicity of $h(\cdot,\cdot)$, we see that for $i \in G_{\epsilon+\frac{\gamma}{2}}$,

$$h\left(\frac{1}{2}\left(\mu_i - \left(\mu_1 + \frac{\omega}{8} - \epsilon\right)\right), \frac{\delta}{n}\right) \leqslant \min\left[h\left(\frac{\epsilon - \Delta_i}{8}, \frac{\delta}{n}\right), h\left(\frac{\gamma}{8}, \frac{\delta}{n}\right)\right].$$

Hence, if any $i \in G_{\epsilon+\frac{\gamma}{2}}$ has received this many samples, then its lower bound exceeds U_t and thus the arm must be in $\widehat{G}$. Putting this together, if $T_i(t) \geqslant \min\left[h\left(\frac{\epsilon-\Delta_i}{8}, \frac{\delta}{n}\right), h\left(\frac{\gamma}{8}, \frac{\delta}{n}\right)\right]$, then $i \neq i_2(t)$ for all $i \in G_{\epsilon+\frac{\gamma}{2}}$. Summing over all such i bounds the size of set stated in the claim. □

6.B.1.4 Step 3: Controlling the complexity until stopping occurs

In this step, we turn our attention to the final event to control:

$$\mathcal{S} := \left\{t : \neg\text{STOP and } i^* \in G_\omega \text{ and } C_{\delta/n}(T_{i^*}(t)) \leqslant \frac{\omega}{16} \text{ and } i_1(t) \in G_{\epsilon+\frac{\gamma}{2}} \text{ and } i_2(t) \in G_{\epsilon+\frac{\gamma}{2}}^c\right\}.$$

For brevity, we will refer to this set as $\mathcal{S}$ for this step. The objective will be to bound the time before each arms lower bound either clears U_t or its upper bound clears

L_t which implies the stopping condition. To do so, we introduce, two events:

$$E_1(t) := \{\hat{\mu}_{i_1(t)}(T_{i_1(t)}(t)) - C_{\delta/n}(T_{i_1(t)}(t)) > U_t\} \tag{6.5}$$

and

$$E_2(t) := \{\hat{\mu}_{i_2(t)}(T_{i_2(t)}(t)) + C_{\delta/n}(T_{i_2(t)}(t)) < L_t\}. \tag{6.6}$$

If $E_1(t)$ is true, then $\hat{\mu}_i(T_i) - C_{\delta/n}(T_i(t)) > L_t$ for all $i \in \widehat{G}$. If $E_2(t)$ is true, then $\hat{\mu}_i(T_i) + C_{\delta/n}(T_i(t)) < U_t$ for all $i \in \widehat{G}^c$. Hence, by line 6 of $(\mathtt{ST})^2$, if both $E_1(t)$ and $E_2(t)$ are true, then $(\mathtt{ST})^2$ terminates.

Claim 0: $|\mathcal{S} \cap \{t : \neg E_1(t)\}| \leqslant \sum_{i \in G_{\epsilon+\frac{\gamma}{2}}} \min\left[h\left(\frac{\epsilon-\Delta_i}{8}, \frac{\delta}{n}\right), h\left(\frac{\gamma}{8}, \frac{\delta}{n}\right)\right]$.

Proof. Recall that by the set $\mathcal{S}$, we have that $i_1(t) \in G_{\epsilon+\frac{\gamma}{2}}$. Furthermore, by the set $\mathcal{S}$, we have that $i^*(t) \in G_\omega$ and $C_{\delta/n}(T_{i^*}(t)) \leqslant \omega/16$. Hence,

$$\begin{aligned}
U_t &= \max_i \hat{\mu}_i(T_i(t)) + C_{\delta/n}(T_i(t)) - \epsilon - \gamma \\
&= \hat{\mu}_{i^*(t)}(T_{i^*(t)}(t)) + C_{\delta/n}(T_{i^*(t)}(t)) - \epsilon - \gamma \\
&\overset{\mathcal{E}}{\leqslant} \mu_{i^*(t)} + 2C_{\delta/n}(T_{i^*(t)}(t)) - \epsilon - \gamma \\
&\leqslant \mu_{i^*(t)} + \frac{\omega}{8} - \epsilon - \gamma \\
&\leqslant \mu_1 + \frac{\omega}{8} - \epsilon - \gamma
\end{aligned}$$

If $C_{\delta/n}(T_i) \leqslant \frac{1}{2}\left(\mu_i - \left(\mu_1 + \frac{\omega}{8} - \epsilon - \gamma\right)\right)$ which is true when
$T_i \geqslant h\left(\frac{1}{2}\left(\mu_i - \left(\mu_1 + \frac{\omega}{8} - \epsilon - \gamma\right)\right), \frac{\delta}{n}\right)$, then

$$\hat{\mu}_i(T_i) - C_{\delta/n}(T_i) \geqslant \mu_i - 2C_{\delta/n}(T_i) \geqslant \mu_1 + \frac{\omega}{8} - \epsilon - \gamma \geqslant U_t.$$

The remainder of the proof of this claim focuses on controlling the difference: $\mu_i - \left(\mu_1 + \frac{\omega}{8} - \epsilon - \gamma\right)$ in the case that $\omega = \min(\alpha_\epsilon, \beta_\epsilon)$ and $\omega = \gamma$. Recall that $\omega = \max(\gamma, \min(\alpha_\epsilon, \beta_\epsilon))$. Hence, if any possible $i \in G_{\epsilon+\frac{\gamma}{2}}$ has received sufficiently many samples, since $i_1(t) \in G_{\epsilon+\frac{\gamma}{2}}$, this implies $E_1(t)$.

Case 1a, $\omega = \min(\alpha_\epsilon, \beta_\epsilon)$ **and** $i \in G_\epsilon$

We focus on the difference $\mu_i - \left(\mu_1 + \frac{\omega}{8} - \epsilon - \gamma\right)$.

$$\begin{aligned}\mu_i - \left(\mu_1 + \frac{\omega}{8} - \epsilon - \gamma\right) &= \mu_i - \left(\mu_1 + \frac{\min(\alpha_\epsilon, \beta_\epsilon)}{8} - \epsilon - \gamma\right) \\ &= \mu_i - (\mu_1 - \epsilon) + \gamma - \frac{1}{8}\min(\alpha_\epsilon, \beta_\epsilon) \\ &\overset{(\gamma \geqslant 0)}{\geqslant} \frac{1}{2}(\mu_i - (\mu_1 - \epsilon)) = \frac{\epsilon - \Delta_i}{2}\end{aligned}$$

where the final step follows since $\min(\alpha_\epsilon, \beta_\epsilon) \leqslant \alpha_\epsilon \leqslant \mu_i - (\mu_1 - \epsilon)$ by definition for all $i \in G_\epsilon$. Then by monotonicity of $h(\cdot, \cdot)$,

$$h\left(\frac{1}{2}\left(\mu_i - \left(\mu_1 + \frac{\omega}{8} - \epsilon - \gamma\right)\right), \frac{\delta}{n}\right) \leqslant h\left(\frac{\epsilon - \Delta_i}{4}, \frac{\delta}{n}\right).$$

Lastly, in this setting, $\gamma \leqslant \min(\alpha_\epsilon, \beta_\epsilon) \leqslant \epsilon - \Delta_i$ since $\omega = \min(\alpha_\epsilon, \beta_\epsilon)$. Hence, it is trivially true that

$$h\left(\frac{\epsilon - \Delta_i}{4}, \frac{\delta}{n}\right) = \min\left[h\left(\frac{\epsilon - \Delta_i}{4}, \frac{\delta}{n}\right), h\left(\frac{\gamma}{4}, \frac{\delta}{n}\right)\right]$$

Case 1b, $\omega = \min(\alpha_\epsilon, \beta_\epsilon)$ **and** $i \in G_\epsilon^c \cap G_{\epsilon + \frac{\gamma}{2}}$

Since $\omega = \max(\gamma, \min(\alpha_\epsilon, \beta_\epsilon))$, if $\omega = \min(\alpha_\epsilon, \beta_\epsilon)$, then $\frac{1}{2}\gamma < \min(\alpha_\epsilon, \beta_\epsilon)$. Since $\min(\alpha_\epsilon, \beta_\epsilon) = \min|\mu_i - (\mu_1 - \epsilon)|$, the set $G_\epsilon^c \cap G_{\epsilon + \frac{\gamma}{2}}$ is empty and there is nothing to prove.

Case 2a, $\omega = \gamma$ **and** $i \in G_\epsilon$

Again, we bound the difference $\mu_i - \left(\mu_1 + \frac{\omega}{4} - \epsilon - \gamma\right)$.

$$\mu_i - \left(\mu_1 + \frac{\omega}{8} - \epsilon - \gamma\right) = \mu_i - (\mu_1 - \epsilon) + \frac{7}{8}\gamma$$

Since $i \in G_\epsilon$, $\mu_i - (\mu_1 - \epsilon) \geqslant 0$. Hence,

$$\mu_i - (\mu_1 - \epsilon) + \frac{7}{8}\gamma \geqslant \max\left(\mu_i - (\mu_1 - \epsilon), \frac{7}{8}\gamma\right)$$

$$\geqslant \frac{1}{2}\max\left(\epsilon - \Delta_i, \gamma\right)$$

Therefore, we have that

$$h\left(\frac{1}{2}\left(\mu_i - \left(\mu_1 + \frac{\omega}{8} - \epsilon - \gamma\right)\right), \frac{\delta}{n}\right) \leqslant h\left(\frac{\epsilon - \Delta_i}{4}, \frac{\delta}{n}\right)$$

and

$$h\left(\frac{1}{2}\left(\mu_i - \left(\mu_1 + \frac{\omega}{8} - \epsilon - \gamma\right)\right), \frac{\delta}{n}\right) \leqslant h\left(\frac{\gamma}{4}, \frac{\delta}{n}\right).$$

Hence,

$$h\left(\frac{1}{2}\left(\mu_i - \left(\mu_1 + \frac{\omega}{4} - \epsilon - \gamma\right)\right), \frac{\delta}{n}\right) \leqslant \min\left[h\left(\frac{\epsilon - \Delta_i}{4}, \frac{\delta}{n}\right), h\left(\frac{\gamma}{4}, \frac{\delta}{n}\right)\right].$$

Case 2b, $\omega = \gamma$ **and** $i \in G_\epsilon^c \cap G_{\epsilon + \frac{\gamma}{2}}$

As before,

$$\mu_i - \left(\mu_1 + \frac{\omega}{8} - \epsilon - \gamma\right) = \mu_i - (\mu_1 - \epsilon) + \frac{7}{8}\gamma$$

Since $i \in G_\epsilon^c \cap G_{\epsilon + \frac{\gamma}{2}}$, we have that $\mu_i - (\mu_1 - \epsilon - \frac{\gamma}{2}) \geqslant 0$. Rearranging implies that $\mu_i - (\mu_1 - \epsilon) \geqslant \frac{-1}{2}\gamma$. Hence,

$$\mu_i - (\mu_1 - \epsilon) + \frac{7}{8}\gamma \geqslant \frac{3}{8}\gamma.$$

Hence,

$$h\left(\frac{1}{2}\left(\mu_i - \left(\mu_1 + \frac{\omega}{8} - \epsilon - \gamma\right)\right), \frac{\delta}{n}\right) \leqslant h\left(\frac{\gamma}{8}, \frac{\delta}{n}\right).$$

Additionally, as above, if $i \in G_\epsilon^c \cap G_{\epsilon + \frac{\gamma}{2}}$, we have that $\mu_i - (\mu_1 - \epsilon - \frac{\gamma}{2}) \geqslant 0$ which implies that $(\mu_1 - \epsilon) - \mu_i \leqslant \gamma$. Hence

$$h\left(\frac{\gamma}{8}, \frac{\delta}{n}\right) = \min\left[h\left(\frac{\Delta_i - \epsilon}{8}, \frac{\delta}{n}\right), h\left(\frac{\gamma}{8}, \frac{\delta}{n}\right)\right].$$

Therefore, if T_i exceeds the above, then $E_1(t)$ is true for an $i_1 \in G_\epsilon^c \cap G_{\epsilon+\frac{\gamma}{2}}$. Combining all cases, and noting that $h(x,\delta) \geqslant h(x/2,\delta)\ \forall x$, we see that for $i_1 \in G_{\epsilon+\frac{\gamma}{2}}$, if

$$T_{i_1(t)}(t) > \min\left[h\left(\frac{\epsilon-\Delta_i}{8},\frac{\delta}{n}\right), h\left(\frac{\gamma}{8},\frac{\delta}{n}\right)\right],$$

Then $E_1(t)$ is true. Summing over all possible $i_1 \in G_{\epsilon+\frac{\gamma}{2}}$ proves the claim. □

Claim 1: $|\mathcal{S} \cap \{t : E_1(t)\} \cap \{t : \neg E_2(t)\}| \leqslant \sum_{i \in G^c_{\epsilon+\frac{\gamma}{2}}} \min\left[h\left(\frac{\epsilon-\Delta_i}{8},\frac{\delta}{n}\right), h\left(\frac{\gamma}{8},\frac{\delta}{n}\right)\right]$.

Proof. By the events in set $\mathcal{S}$, $C_{\delta/n}(T_{i^*}(t)) \leqslant \frac{\omega}{16}$. Hence,

$$\mu_i^* + \frac{\omega}{8} \geqslant \mu_i^* + 2C_{\delta/n}(T_{i^*}(t)) \overset{\mathcal{E}}{\geqslant} \hat{\mu}_i^*(T_{i^*}(t)) + C_{\delta/n}(T_{i^*}(t)) \geqslant \hat{\mu}_1(T_1(t)) + C_{\delta/n}(T_1(t)) \overset{\mathcal{E}}{\geqslant} \mu_1.$$

Therefore, $\mu_{i^*} \geqslant \mu_1 - \frac{\omega}{8}$ or equivalently, $i^* \in G_{\omega/8}$. Using this,

$$\begin{aligned} L_t = \max_i \hat{\mu}_i(T_i(t)) - C_{\delta/n}(T_i(t)) - \epsilon &\geqslant \hat{\mu}_{i^*}(T_{i^*}(t)) - C_{\delta/n}(T_{i^*}(t)) - \epsilon \\ &\overset{\mathcal{E}}{\geqslant} \mu_{i^*} - 2C_{\delta/n}(T_{i^*}(t)) - \epsilon \\ &\overset{\mathcal{E}}{\geqslant} \mu_{i^*} - \frac{\omega}{8} - \epsilon \\ &\geqslant \mu_1 - \frac{\omega}{4} - \epsilon \end{aligned}$$

For $i \in G^c_{\epsilon+\frac{\gamma}{2}}$, if $C_{\delta/n}(T_i) \leqslant \frac{1}{2}\left(\left(\mu_1 - \frac{\omega}{4} - \epsilon\right) - \mu_i\right)$, true when $T_i \geqslant h\left(\frac{1}{2}\left(\left(\mu_1 - \frac{\omega}{4} - \epsilon\right) - \mu_i\right), \frac{\delta}{n}\right)$, then

$$\hat{\mu}_i(T_i) + C_{\delta/n}(T_i) \leqslant \mu_i + 2C_{\delta/n}(T_i) \leqslant \mu_1 - \frac{\omega}{4} - \epsilon \leqslant L_t.$$

As before, we seek a lower bound for the difference $\left(\mu_1 - \frac{\omega}{4} - \epsilon\right) - \mu_i$.

Case 1: $\omega = \min(\alpha_\epsilon, \beta_\epsilon)$

$$\begin{aligned} \left(\mu_1 - \frac{\omega}{4} - \epsilon\right) - \mu_i &= (\mu_1 - \epsilon) - \mu_i - \frac{1}{4}\min(\alpha_\epsilon, \beta_\epsilon) \\ &\geqslant \frac{1}{2}\left((\mu_1 - \epsilon) - \mu_i\right) \end{aligned}$$

since $(\mu_1 - \epsilon) - \mu_i \geqslant \min(\alpha_\epsilon, \beta_\epsilon)$. Therefore, we have that

$$h\left(\frac{1}{2}\left(\left(\mu_1 - \frac{\omega}{4} - \epsilon\right) - \mu_i\right), \frac{\delta}{n}\right) \leqslant h\left(\frac{\Delta_i - \epsilon}{4}, \frac{\delta}{n}\right).$$

Lastly, in this setting, $\gamma \leqslant \min(\alpha_\epsilon, \beta_\epsilon) \leqslant \epsilon - \Delta_i$ since $\omega = \min(\alpha_\epsilon, \beta_\epsilon)$. Hence, it is trivially true that

$$h\left(\frac{\Delta_i - \epsilon}{4}, \frac{\delta}{n}\right) = \min\left[h\left(\frac{\Delta_i - \epsilon}{4}, \frac{\delta}{n}\right), h\left(\frac{\gamma}{4}, \frac{\delta}{n}\right)\right].$$

Case 2: $\omega = \gamma$

Assume that $\gamma > \min(\alpha_\epsilon, \beta_\epsilon)$, as equality is covered by the previous case. Hence,

$$\left(\mu_1 - \frac{\omega}{4} - \epsilon\right) - \mu_i = (\mu_1 - \epsilon) - \mu_i - \frac{1}{4}\gamma$$

Recall that we seek to control $i_2 \in G^c_{\epsilon+\frac{\gamma}{2}}$. For any $i \in G^c_{\epsilon+\frac{\gamma}{2}}$, we have that $\mu_1 - \epsilon - \frac{\gamma}{2} - \mu_i \geqslant 0$. Rearranging, we see that $(\mu_1 - \epsilon) - \mu_i \geqslant \frac{1}{2}\gamma$ which implies that

$$(\mu_1 - \epsilon) - \mu_i - \frac{1}{4}\gamma \geqslant \frac{1}{2}((\mu_1 - \epsilon) - \mu_i).$$

Therefore, we have that

$$h\left(\frac{1}{2}\left(\left(\mu_1 - \frac{\omega}{4} - \epsilon\right) - \mu_i\right), \frac{\delta}{n}\right) \leqslant h\left(\frac{\Delta_i - \epsilon}{4}, \frac{\delta}{n}\right)$$

is this setting as well. Similarly, since $\Delta_i - \epsilon \geqslant \frac{1}{2}\gamma$, we likewise have that

$$h\left(\frac{\Delta_i - \epsilon}{4}, \frac{\delta}{n}\right) \leqslant \min\left[h\left(\frac{\Delta_i - \epsilon}{8}, \frac{\delta}{n}\right), h\left(\frac{\gamma}{8}, \frac{\delta}{n}\right)\right].$$

Hence, if T_i exceeds the right-hand side of the preceding inequality, then for any $i \in G^c_{\epsilon+\frac{\gamma}{2}}$, its upper bound is below L_t. Hence for $i_2(t) \in G^c_{\epsilon+\frac{\gamma}{2}}$, this implies event $E_2(t)$. Summing over all possible values of $i_2(t) \in G^c_{\epsilon+\frac{\gamma}{2}}$ proves the claim. □

Claim 2: The cardinality of $\mathcal{S}$ is bounded as $|\mathcal{S}| \leqslant \sum_{i=1}^{n} \min\left[h\left(\frac{\Delta_i-\epsilon}{8}, \frac{\delta}{n}\right), h\left(\frac{\gamma}{8}, \frac{\delta}{n}\right)\right]$.

Proof. First, $\mathcal{S}$ may be decomposed as

$$|\mathcal{S}| = |\mathcal{S} \cap \{t : \neg E_1(t)\}| + |\mathcal{S} \cap \{t : E_1(t)\} \cap \{t : \neg E_2(t)\}| + |\mathcal{S} \cap \{t : E_1(t)\} \cap \{t : E_2(t)\}|$$

Note that $|\mathcal{S} \cap \{t : E_1(t)\} \cap \{t : E_2(t)\}| = 0$ because we have assumed in set $\mathcal{S}$ that $(ST)^2$ has not stopped, and $\{t : E_1(t)\} \cap \{t : E_2(t)\}$ implies termination. By Claim 0, $|\mathcal{S} \cap \{t : \neg E_1(t)\}| \leqslant \sum_{i \in G_{\epsilon+\frac{\gamma}{2}}} \min\left[h\left(\frac{\epsilon-\Delta_i}{4}, \frac{\delta}{n}\right), h\left(\frac{\gamma}{4}, \frac{\delta}{n}\right)\right]$. By Claim 1, $|\mathcal{S} \cap \{t : E_1(t)\} \cap \{t : \neg E_2(t)\}| \leqslant \sum_{i \in G^c_{\epsilon+\frac{\gamma}{2}}} \min\left[h\left(\frac{\epsilon-\Delta_i}{8}, \frac{\delta}{n}\right), h\left(\frac{\gamma}{8}, \frac{\delta}{n}\right)\right]$. Recalling that h is assumed to be symmetric in its first argument proves the claim. □

6.B.1.5 Step 4: Putting it all together

Recall that the total number of rounds T that $(ST)^2$ runs for is given by $T = |\{t : \neg\text{STOP}\}|$. To bound this quantity, we have decomposed the set $\{t : \neg\text{STOP}\}$ into many subsets. Below, we show this decomposition.

$$\begin{aligned}
&\{t : \neg\text{STOP}\} = \\
&\{t : \neg\text{STOP and } i^* \notin G_\omega\} \\
&\cup \left\{t : \neg\text{STOP and } i^* \in G_\omega \text{ and } C_{\delta/n}(T_{i^*}(t)) > \frac{\omega}{16}\right\} \\
&\cup \left\{t : \neg\text{STOP and } i^* \in G_\omega \text{ and } C_{\delta/n}(T_{i^*}(t)) \leqslant \frac{\omega}{16} \text{ and } i_1(t) \in G^c_{\epsilon+\frac{\gamma}{2}}\right\} \\
&\cup \left\{t : \neg\text{STOP and } i^* \in G_\omega \text{ and } C_{\delta/n}(T_{i^*}(t)) \leqslant \frac{\omega}{16} \text{ and } i_1(t) \in G_{\epsilon+\frac{\gamma}{2}} \text{ and } i_2(t) \in G_{\epsilon+\frac{\gamma}{2}}\right\} \\
&\cup \left\{t : \neg\text{STOP and } i^* \in G_\omega \text{ and } C_{\delta/n}(T_{i^*}(t)) \leqslant \frac{\omega}{16} \text{ and } i_1(t) \in G_{\epsilon+\frac{\gamma}{2}} \text{ and } i_2(t) \in G^c_{\epsilon+\frac{\gamma}{2}}\right\}.
\end{aligned}$$

Hence, by a union bound and plugging in the results of the above steps,

$$\begin{aligned}
&|\{t : \neg\text{STOP}\}| \leqslant \\
&|\{t : \neg\text{STOP and } i^* \notin G_\omega\}| \\
&+ \left|\left\{t : \neg\text{STOP and } i^* \in G_\omega \text{ and } \exists i \in G_\omega : C_{\delta/n}(T_{i^*}(t)) > \frac{\omega}{16}\right\}\right|
\end{aligned}$$

$$
\begin{aligned}
&+\left|\left\{t:\neg\text{STOP and } i^*\in G_\omega \text{ and } C_{\delta/n}(T_{i^*}(t))\leqslant\frac{\omega}{16} \text{ and } i_1(t)\in G^c_{\epsilon+\frac{\gamma}{2}}\right\}\right|\\
&+\left|\left\{t:\neg\text{STOP and } i^*\in G_\omega \text{ and } C_{\delta/n}(T_{i^*}(t))\leqslant\frac{\omega}{16} \text{ and } i_1(t)\in G_{\epsilon+\frac{\gamma}{2}} \text{ and } i_2(t)\in G_{\epsilon+\frac{\gamma}{2}}\right\}\right|\\
&+\left|\left\{t:\neg\text{STOP and } i^*\in G_\omega \text{ and } C_{\delta/n}(T_{i^*}(t))\leqslant\frac{\omega}{16} \text{ and } i_1(t)\in G_{\epsilon+\frac{\gamma}{2}} \text{ and } i_2(t)\in G^c_{\epsilon+\frac{\gamma}{2}}\right\}\right|\\
&\leqslant \sum_{i\in G^c_\omega}\min\left\{h\left(\frac{\gamma}{2},\frac{\delta}{n}\right),\min\left[h\left(\frac{\Delta_i}{2},\frac{\delta}{n}\right),h\left(\frac{\min(\alpha_\epsilon,\beta_\epsilon)}{2},\frac{\delta}{n}\right)\right]\right\}\\
&\quad+\sum_{i\in G_\omega}\min\left\{h\left(\frac{\gamma}{16},\frac{\delta}{n}\right),\min\left[h\left(\frac{\Delta_i}{16},\frac{\delta}{n}\right),h\left(\frac{\min(\alpha_\epsilon,\beta_\epsilon)}{16},\frac{\delta}{n}\right)\right]\right\}\\
&\quad+\sum_{i\in G^c_{\epsilon+\frac{\gamma}{2}}}\min\left[h\left(\frac{\Delta_i-\epsilon}{8},\frac{\delta}{n}\right),h\left(\frac{\gamma}{8},\frac{\delta}{n}\right)\right]\\
&\quad+\sum_{i\in G_{\epsilon+\frac{\gamma}{2}}}\min\left[h\left(\frac{\epsilon-\Delta_i}{8},\frac{\delta}{n}\right),h\left(\frac{\gamma}{8},\frac{\delta}{n}\right)\right]\\
&\quad+\sum_{i=1}^{n}\min\left[h\left(\frac{\Delta_i-\epsilon}{8},\frac{\delta}{n}\right),h\left(\frac{\gamma}{8},\frac{\delta}{n}\right)\right]\\
&\overset{(\epsilon\leqslant 1/2)}{\leqslant}\sum_{i=1}^{n}\min\left\{h\left(\frac{\gamma}{16},\frac{\delta}{n}\right),\min\left[h\left(\frac{\Delta_i}{16},\frac{\delta}{n}\right),h\left(\frac{\min(\alpha_\epsilon,\beta_\epsilon)}{16},\frac{\delta}{n}\right)\right]\right\}\\
&\quad+2\sum_{i=1}^{n}\min\left[h\left(\frac{\Delta_i-\epsilon}{8},\frac{\delta}{n}\right),h\left(\frac{\gamma}{8},\frac{\delta}{n}\right)\right]\\
&\leqslant 4\sum_{i=1}^{n}\min\left\{\max\left\{h\left(\frac{\Delta_i-\epsilon}{16},\frac{\delta}{n}\right),\min\left[h\left(\frac{\Delta_i}{16},\frac{\delta}{n}\right),h\left(\frac{\min(\alpha_\epsilon,\beta_\epsilon)}{16},\frac{\delta}{n}\right)\right]\right\},\right.\\
&\qquad\qquad\left.h\left(\frac{\gamma}{16},\frac{\delta}{n}\right)\right\}
\end{aligned}
$$

Next, by Lemma 6.33, we may bound the minimum of $h(\cdot,\cdot)$ functions.

$$
\begin{aligned}
&4\sum_{i=1}^{n}\min\left\{\max\left\{h\left(\frac{\Delta_i-\epsilon}{16},\frac{\delta}{n}\right),\min\left[h\left(\frac{\Delta_i}{16},\frac{\delta}{n}\right),h\left(\frac{\min(\alpha_\epsilon,\beta_\epsilon)}{16},\frac{\delta}{n}\right)\right]\right\},\right.\\
&\qquad\qquad\left.h\left(\frac{\gamma}{16},\frac{\delta}{n}\right)\right\}
\end{aligned}
$$

$$
\begin{aligned}
&= 4\sum_{i=1}^{n} \min\left\{\max\left\{h\left(\frac{\Delta_i-\epsilon}{16},\frac{\delta}{n}\right),\right.\right.\\
&\qquad\qquad \min\left[h\left(\frac{\Delta_i}{16},\frac{\delta}{n}\right),\max\left[h\left(\frac{\alpha_\epsilon}{16},\frac{\delta}{n}\right),h\left(\frac{\beta_\epsilon}{16},\frac{\delta}{n}\right)\right]\right]\Bigg\},\\
&\qquad\qquad \left.h\left(\frac{\gamma}{16},\frac{\delta}{n}\right)\right\}\\
&\leqslant 4\sum_{i=1}^{n} \min\left\{\max\left\{h\left(\frac{\Delta_i-\epsilon}{16},\frac{\delta}{n}\right),\right.\right.\\
&\qquad\qquad \max\left[h\left(\frac{\Delta_i+\alpha_\epsilon}{32},\frac{\delta}{n}\right),h\left(\frac{\Delta_i+\beta_\epsilon}{32},\frac{\delta}{n}\right)\right]\Bigg\},\\
&\qquad\qquad \left.h\left(\frac{\gamma}{16},\frac{\delta}{n}\right)\right\}\\
&= 4\sum_{i=1}^{n} \min\left\{\max\left\{h\left(\frac{\Delta_i-\epsilon}{16},\frac{\delta}{n}\right),h\left(\frac{\Delta_i+\alpha_\epsilon}{32},\frac{\delta}{n}\right),h\left(\frac{\Delta_i+\beta_\epsilon}{32},\frac{\delta}{n}\right)\right\},\right.\\
&\qquad\qquad \left.h\left(\frac{\gamma}{16},\frac{\delta}{n}\right)\right\}
\end{aligned}
$$

Finally, we use Lemma 6.32 to bound the function $h(\cdot,\cdot)$. Since $\delta \leqslant 1/2$, $\delta/n \leqslant 2e^{-e/2}$. Further, $|\epsilon-\Delta_i| \leqslant 8$ for all i and $\epsilon \leqslant 1/2$ implies that $\frac{1}{8}|\epsilon-\Delta_i| \leqslant 2$ and $\frac{1}{8}\min(\alpha_\epsilon,\beta_\epsilon) \leqslant 2$. $\Delta_i \leqslant 16$ for all i, gives $0.125\Delta_i \leqslant 2$. Lastly, $\gamma \leqslant 16$ implies that $\frac{\gamma}{8} \leqslant 2$. Therefore,

$$
\begin{aligned}
&4\sum_{i=1}^{n} \min\left\{\max\left\{h\left(\frac{\Delta_i-\epsilon}{16},\frac{\delta}{n}\right),h\left(\frac{\Delta_i+\alpha_\epsilon}{32},\frac{\delta}{n}\right),h\left(\frac{\Delta_i+\beta_\epsilon}{32},\frac{\delta}{n}\right)\right\},\right.\\
&\qquad\qquad \left.h\left(\frac{\gamma}{16},\frac{\delta}{n}\right)\right\}\\
&\leqslant 4\sum_{i=1}^{n} \min\left\{\max\left\{\frac{1024}{(\epsilon-\Delta_i)^2}\log\left(\frac{2n}{\delta}\log_2\left(\frac{3072n}{\delta(\epsilon-\Delta_i)^2}\right)\right),\right.\right.\\
&\qquad\qquad \frac{4096}{(\Delta_i+\alpha_\epsilon)^2}\log\left(\frac{2n}{\delta}\log_2\left(\frac{12288n}{\delta(\Delta_i+\alpha_\epsilon)^2}\right)\right),\\
&\qquad\qquad \left.\left.\frac{4096}{(\Delta_i+\beta_\epsilon)^2}\log\left(\frac{2n}{\delta}\log_2\left(\frac{12288n}{\delta(\Delta_i+\beta_\epsilon)^2}\right)\right)\right\},
\end{aligned}
$$

$$
\begin{aligned}
&\frac{1}{\gamma^2}\log\left(\frac{2n}{\delta}\log_2\left(\frac{3072n}{\delta\gamma^2}\right)\right)\Bigg\} \\
= 4\sum_{i=1}^{n}\min\Bigg\{&\max\left\{\frac{1024}{(\mu_1-\epsilon-\mu_i)^2}\log\left(\frac{2n}{\delta}\log_2\left(\frac{3072n}{\delta(\mu_1-\epsilon-\mu_i)^2}\right)\right),\right. \\
&\frac{4096}{(\mu_1+\alpha_\epsilon-\mu_i)^2}\log\left(\frac{2n}{\delta}\log_2\left(\frac{12288n}{\delta(\mu_1+\alpha_\epsilon-\mu_i)^2}\right)\right), \\
&\left.\frac{4096}{(\mu_1+\beta_\epsilon-\mu_i)^2}\log\left(\frac{2n}{\delta}\log_2\left(\frac{12288n}{\delta(\mu_1+\beta_\epsilon-\mu_i)^2}\right)\right)\right\}, \\
&\frac{1}{\gamma^2}\log\left(\frac{2n}{\delta}\log_2\left(\frac{3072n}{\delta\gamma^2}\right)\right)\Bigg\}.
\end{aligned}
$$

The above bounds the number of rounds T. Therefore, the total number of samples is at most $3T$. □

6.B.2 Optimism with multiplicative γ

Theorem 6.10. *Fix $\epsilon \in (0, 1/2]$, $0 < \delta \leqslant 1/2$, $\gamma \in [0, \min(16/\mu_1, 1/2)]$ and an instance ν such that $\max(\Delta_i, |\epsilon\mu_1 - \Delta_i|) \leqslant 8$ for all i. In the case that $M_\epsilon = [n]$, let $\tilde{\alpha}_\epsilon = \min(\tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon)$. With probability at least $1-\delta$, $(ST)^2$ correctly returns a set G such that $M_\epsilon \subset G \subset M_{\epsilon+\gamma}$ in at most*

$$
\begin{aligned}
12\sum_{i=1}^{n}\min\Bigg\{&\max\left\{\frac{1024}{((1-\epsilon)\mu_1-\mu_i)^2}\log\left(\frac{2n}{\delta}\log_2\left(\frac{3072n}{\delta((1-\epsilon)\mu_1-\mu_i)^2}\right)\right),\right. \\
&\frac{4096}{(\mu_1+\frac{\tilde{\alpha}_\epsilon}{1-\epsilon}-\mu_i)^2}\log\left(\frac{2n}{\delta}\log_2\left(\frac{12288n}{\delta(\mu_1+\frac{\tilde{\alpha}_\epsilon}{1-\epsilon})^2}\right)\right), \\
&\left.\frac{4096}{(\mu_1+\frac{\tilde{\beta}_\epsilon}{1-\epsilon}-\mu_i)^2}\log\left(\frac{2n}{\delta}\log_2\left(\frac{12288n}{\delta(\mu_1+\frac{\tilde{\beta}_\epsilon}{1-\epsilon}-\mu_i)^2}\right)\right)\right\}, \\
&\frac{1024}{\gamma^2\mu_1^2}\log\left(\frac{2n}{\delta}\log_2\left(\frac{3072n}{\delta\gamma^2\mu_1^2}\right)\right)\Bigg\}
\end{aligned}
$$

samples.

Proof. Throughout the proof, recall that $\Delta_i = \mu_1 - \mu_i$ for all i, $\tilde{\alpha}_\epsilon = \min_{i\in M_\epsilon}\mu_i -$

$(1-\epsilon)\mu_1$, and $\tilde{\beta}_\epsilon = \min_{i \in M_\epsilon^c}(1-\epsilon)\mu_1 - \mu_i$. Additionally, at any time t, we will take $T_j(t)$ to denote the number of samples of arm j up to time t.

Define the event

$$\mathcal{E} = \left\{ \bigcap_{i \in [n]} \bigcap_{t \in \mathbb{N}} |\hat{\mu}_i(t) - \mu_i| \leqslant C_{\delta/n}(t) \right\}.$$

Using standard anytime confidence bound results, and recalling that that $C_\delta(t) := \sqrt{\frac{4\log(\log_2(2t)/\delta)}{t}}$, we have

$$\begin{aligned}\mathbb{P}(\mathcal{E}^c) &= \mathbb{P}\left(\bigcup_{i \in [n]} \bigcup_{t \in \mathbb{N}} |\hat{\mu}_i - \mu_i| > C_{\delta/n}(t) \right) \\ &\leqslant \sum_{i=1}^{n} \mathbb{P}\left(\bigcup_{t \in \mathbb{N}} |\hat{\mu}_i - \mu_i| > C_{\delta/n}(t) \right) \leqslant \sum_{i=1}^{n} \frac{\delta}{n} = \delta\end{aligned}$$

Hence, $\mathbb{P}(\mathcal{E}) \geqslant 1-\delta$. Throughout, we will make use of a function $h(x,\delta)$ such that if $t \geqslant h(x,\delta)$, then $C_\delta(t) \leqslant |x|$. We bound $h(\cdot,\cdot)$ in Lemma 6.32. $h(\cdot,\cdot)$ is assumed to decrease monotonically in both arguments and is symmetric in its first argument.

6.B.2.1 Step 0: Correctness

We begin by showing that on $\mathcal{E}$, if $(\mathtt{ST})^2$ terminates, it returns a set G such that $M_\epsilon \subset G \subset M_{(\epsilon+\gamma)}$. Since $\mathbb{P}(\mathcal{E}) \geqslant 1-\delta$, this implies that $(\mathtt{ST})^2$ is correct with high probability.

Claim 0: On Event $\mathcal{E}$, at all times t, $U_t \geqslant (1-\epsilon-\gamma)\mu_1$.

Proof.

$$\begin{aligned}U_t = (1-\epsilon-\gamma)(\max_j \hat{\mu}_j(T_j(t)) + C_{\delta/n}(T_j(t))) &\geqslant (1-\epsilon-\gamma)(\hat{\mu}_1(T_1(t)) + C_{\delta/n}(T_1(t))) \\ &\overset{\mathcal{E}}{\geqslant} (1-\epsilon-\gamma)\mu_1\end{aligned}$$

□

Claim 1: On Event $\mathcal{E}$, at all times t, $L_t \leqslant (1-\epsilon)\mu_1$.

Proof.

$$L_t = (1-\epsilon)\left(\max_j \hat{\mu}_j(T_j(t)) - C_{\delta/n}(T_j(t))\right) \overset{\mathcal{E}}{\leqslant} (1-\epsilon)\max_j \mu_j = (1-\epsilon)\mu_1$$

□

Claim 2: On event $\mathcal{E}$, if there is a time t such that $\hat{\mu}_i(T_i(t)) - C_{\delta/n}(T_i(t)) > U_t$, then $i \in M_{\epsilon+\gamma}$.

Proof. Assume for some t, $\hat{\mu}_i(T_i(t)) - C_{\delta/n}(T_i(t)) > U_t$. Then

$$\mu_i \overset{\mathcal{E}}{\geqslant} \hat{\mu}_i(T_i(t)) - C_{\delta/n}(T_i(t)) \geqslant U_t \overset{\text{Claim 0}}{\geqslant} (1-\epsilon-\gamma)\mu_1$$

which implies $i \in M_{\epsilon+\gamma}$ □

Claim 3: On event $\mathcal{E}$, if there is a time t such that $\hat{\mu}_i(T_i(t)) + C_{\delta/n}(T_i(t)) < L_t$, then $i \in M_\epsilon^c$.

Proof. Assume that is a t for which $\hat{\mu}_i(T_i(t)) + C_{\delta/n}(T_i(t)) < L_t$. Then

$$\mu_i \overset{\mathcal{E}}{\leqslant} \hat{\mu}_i(T_i(t)) + C_{\delta/n}(T_i(t)) \leqslant L_t \overset{\text{Claim 1}}{\leqslant} (1-\epsilon)\mu_1$$

which implies $i \in M_\epsilon^c$. □

$(\texttt{ST})^2$ terminates at any time t such that simultaneously for all arms i, either $\hat{\mu}_i(T_i(t)) + C_{\delta/n}(T_i(t)) > U_t$ or $\hat{\mu}_i(T_i(t)) - C_{\delta/n}(T_i(t)) < L_t$. On $\mathcal{E}$, by Claim 3, $M_\epsilon \subset \{i : \hat{\mu}_i(T_i(t)) + C_{\delta/n}(T_i(t)) > U_t\}$. On $\mathcal{E}$, by Claim 2, $\{i : \hat{\mu}_i(T_i(t)) + C_{\delta/n}(T_i(t)) > U_t\} \subset M_{\epsilon+\gamma}$. Hence, on the event $\mathcal{E}$. $(\texttt{ST})^2$ returns a set G such that $M_\epsilon \subset G \subset M_{\epsilon+\gamma}$.

6.B.2.2 Step 1: Complexity of estimating the threshold, $(1-\epsilon)\mu_1$

Let STOP denote the termination event that for all arms i, either $\hat{\mu}_i(T_i(t)) + C_{\delta/n}(T_i(t)) > U_t$ or $\hat{\mu}_i(T_i(t)) - C_{\delta/n}(T_i(t)) < L_t$. Let ω denote the quantity

$$\omega := \max\{\gamma\mu_1, \min(\tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon)\}.$$

Let T denote the random variable of the total number of rounds before $(\texttt{ST})^2$ terminates. At most 3 samples are drawn in any round. Hence, the total sample complexity is bounded by $3T$. We may write T as

$$T \equiv |\{t : \neg\text{STOP}\}| = |\{t : \neg\text{STOP and } i^* \notin M_{\omega/\mu_1}\}| + |\{t : \neg\text{STOP and } i^* \in M_{\omega/\mu_1}\}|$$

Next, we bound the first event in this decomposition.

Claim 0: On $\mathcal{E}$, $|\{t : \neg\text{STOP and } i^* \notin M_{\omega/\mu_1}\}| \leqslant$
$\sum_{i \in M^c_{\omega/\mu_1}} \min\left\{h\left(\frac{\gamma\mu_1}{2}, \frac{\delta}{n}\right), \min\left[h\left(\frac{\Delta_i}{2}, \frac{\delta}{n}\right), h\left(\frac{\min(\tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon)}{2}, \frac{\delta}{n}\right)\right]\right\}$.

Proof. For each $i \in M^c_{\omega/\mu_1}$, $\mu_i + 2C_{\delta/n}(T_i(t)) < \mu_1$, true when $T_i(t) > h\left(\Delta_i/2, \frac{\delta}{n}\right)$ implies that

$$\hat{\mu}_i(T_i(t)) + C_{\delta/n}(T_i(t)) \overset{\mathcal{E}}{\leqslant} \mu_i + 2C_{\delta/n}(T_i(t)) < \mu_1 \overset{\mathcal{E}}{\leqslant} \hat{\mu}_1(T_1(t)) + C_{\delta/n}(T_1(t))$$

which implies that $i \neq i^*$. Additionally, since $i \in M^c_{\omega/\mu_1}$ by assumption, we have that $(1-\omega/\mu_1)\mu_1 - \mu_i \geqslant 0$, which reduces to $\Delta_i \geqslant \omega$. Since $\omega = \max(\gamma\mu_1, \min(\tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon))$, it is likewise true that

$$h\left(\frac{\Delta_i}{2}, \frac{\delta}{n}\right) = \min\left[h\left(\frac{\gamma\mu_1}{2}, \frac{\delta}{n}\right), \min\left\{h\left(\frac{\Delta_i}{2}, \frac{\delta}{n}\right), h\left(\frac{\min(\tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon)}{2}, \frac{\delta}{n}\right)\right]\right\}.$$

Summing over all $i \in M^c_{\omega/\mu_1}$ achieves the result. □

We may decompose the event $\{t : \neg\text{STOP and } i^* \in M_{\omega/\mu_1}\}$ as

$$\left\{t : \neg\text{STOP and } i^* \in M_{\omega/\mu_1} \text{ and } \exists i \in M_{\omega/\mu_1} : C_{\delta/n}(T_{i^*}(t)) > \frac{\omega}{16(1-\epsilon)}\right\}$$
$$\cup \left\{t : \neg\text{STOP and } i^* \in M_{\omega/\mu_1} \text{ and } C_{\delta/n}(T_{i^*}(t)) \leqslant \frac{\omega}{16(1-\epsilon)}\right\}$$

Claim 1: $\left|\left\{t : \neg\text{STOP and } i^* \in M_{\omega/\mu_1} \text{ and } C_{\delta/n}(T_{i^*}(t)) \geqslant \frac{\omega}{16(1-\epsilon)}\right\}\right|$
$\leqslant \sum_{i \in M_{\omega/\mu_1}} \min\left\{h\left(\frac{\gamma\mu_1}{16}, \frac{\delta}{n}\right), \min\left[h\left(\frac{\Delta_i}{16}, \frac{\delta}{n}\right), h\left(\frac{\min(\tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon)}{16(1-\epsilon)}, \frac{\delta}{n}\right)\right]\right\}$

Proof. $C_{\delta/n}(T_i(t)) \leqslant \frac{\omega}{16(1-\epsilon)}$ is true when $T_i(t) \geqslant h\left(\frac{\omega}{16(1-\epsilon)}, \frac{\delta}{n}\right)$. Since $i^* \in M_{\omega/\mu_1}$, $\mu_i - (1-\omega/\mu_1)\mu_1 \geqslant 0$, which implies $\Delta_i \leqslant \omega$. By definition, $\omega =$

$\min(\gamma\mu_1, \min(\tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon))$. Hence, by monotonicity of $h(\cdot,\cdot)$,

$$\begin{aligned} h\left(\frac{\omega}{16(1-\epsilon)}, \frac{\delta}{n}\right) &= \min\left[h\left(\frac{\Delta_i}{16(1-\epsilon)}, \frac{\delta}{n}\right), h\left(\frac{\omega}{16(1-\epsilon)}, \frac{\delta}{n}\right)\right] \\ &= \min\left\{h\left(\frac{\gamma\mu_1}{16(1-\epsilon)}, \frac{\delta}{n}\right), \min\left[h\left(\frac{\Delta_i}{16(1-\epsilon)}, \frac{\delta}{n}\right), h\left(\frac{\min(\tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon)}{16(1-\epsilon)}, \frac{\delta}{n}\right)\right]\right\} \\ &\leqslant \min\left\{h\left(\frac{\gamma\mu_1}{16}, \frac{\delta}{n}\right), \min\left[h\left(\frac{\Delta_i}{16}, \frac{\delta}{n}\right), h\left(\frac{\min(\tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon)}{16(1-\epsilon)}, \frac{\delta}{n}\right)\right]\right\}. \end{aligned}$$

Summing over all $i \in M_{\omega/\mu_1}$ achieves the desired result. □

6.B.2.3 Step 2: Controlling "crossing" events

Recall that we sample $i_1(t) \in \widehat{G}$ and $i_2(t) \in \widehat{G}^c$. In this section, we control the number of times that $i_1(t) \in M^c_{\epsilon+\frac{\gamma}{2}}$ and $i_2(t) \in M_{\epsilon+\frac{\gamma}{2}}$.

To do so, we first decompose the set $\left\{t : \neg\text{STOP and } i^* \in M_{\omega/\mu_1} \text{ and } C_{\delta/n}(T_{i^*}(t)) \leqslant \frac{\omega}{16(1-\epsilon)}\right\}$ as

$$\begin{aligned} &\left\{t : \neg\text{STOP and } i^* \in M_{\omega/\mu_1} \text{ and } C_{\delta/n}(T_{i^*}(t)) \leqslant \frac{\omega}{16(1-\epsilon)} \text{ and } i_1(t) \in M^c_{\epsilon+\frac{\gamma}{2}}\right\} \\ &\quad \cup \left\{t : \neg\text{STOP and } i^* \in M_{\omega/\mu_1} \text{ and } C_{\delta/n}(T_{i^*}(t)) \leqslant \frac{\omega}{16(1-\epsilon)} \text{ and } i_1(t) \in M_{\epsilon+\frac{\gamma}{2}}\right\} \end{aligned}$$

Claim 0: $\left|\left\{t : \neg\text{STOP and } i^* \in M_{\omega/\mu_1} \text{ and } C_{\delta/n}(T_{i^*}(t)) \leqslant \frac{\omega}{16(1-\epsilon)} \text{ and } i_1(t) \in M^c_{\epsilon+\frac{\gamma}{2}}\right\}\right| \leqslant \sum_{i \in M^c_{\epsilon+\frac{\gamma}{2}}} \min\left[h\left(\frac{\Delta_i - \epsilon\mu_1}{16}, \frac{\delta}{n}\right), h\left(\frac{\gamma\mu_1}{16}, \frac{\delta}{n}\right)\right]$.

Proof. Recall that $\widehat{G}$ is the set of all arms whose empirical means exceed $(1-\epsilon)\max_i \hat{\mu}_i(T_i(t))$, and $i_1(t) \in \widehat{G}$ by definition. Note that $(1-\epsilon)\max_i \hat{\mu}_i(T_i(t)) > (1-\epsilon)\left(\max_i \hat{\mu}_i(T_i(t)) - C_{\delta/n}(T_i(t))\right) = L_t$. Hence, if an arm's upper bound is below L_t, then the arm cannot be in $\widehat{G}$ and thus not be $i_1(t)$. By the above event, $C_{\delta/n}(T_{i^*}(t)) \leqslant \frac{\omega}{16(1-\epsilon)}$. Therefore,

$$\mu_{i^*} + \frac{\omega}{8(1-\epsilon)} \geqslant \mu_{i^*} + 2C_{\delta/n}(T_{i^*}(t)) \overset{\mathcal{E}}{\geqslant} \hat{\mu}_{i^*}(T_{i^*}(t)) + C_{\delta/n}(T_{i^*}(t)) \geqslant \hat{\mu}_1(T_1(t)) + C_{\delta/n}(T_1(t))$$

$$\overset{\mathcal{E}}{\geqslant} \mu_1.$$

Hence, $\mu_{i^*} \geqslant \mu_1 - \frac{\omega}{8(1-\epsilon)}$. Rearranging this, we see that $\mu_{i^*} - \left(1 - \frac{\omega}{8\mu_1(1-\epsilon)}\right)\mu_1 \geqslant 0$ which implies that $i^* \in M_{\frac{\omega}{8\mu_1(1-\epsilon)}}$. Hence,

$$\begin{aligned} L_t = (1-\epsilon)\left(\max_i \hat{\mu}_i(T_i(t)) - C_{\delta/n}(T_i(t))\right)(1-\epsilon)\left(\hat{\mu}_{i^*}(T_{i^*}(t)) - C_{\delta/n}(T_{i^*}(t))\right) \\ \overset{\mathcal{E}}{\geqslant} (1-\epsilon)\left(\mu_{i^*} - 2C_{\delta/n}(T_{i^*}(t))\right) \\ \geqslant (1-\epsilon)\left(\mu_{i^*} - \frac{\omega}{8(1-\epsilon)}\right) \\ \geqslant (1-\epsilon)\left(\mu_1 - \frac{\omega}{4(1-\epsilon)}\right) \end{aligned}$$

Next, we bound the number of times an arm $i \in M^c_{\epsilon+\frac{\gamma}{2}}$ is sampled before its upper bound is below $(1-\epsilon)\left(\mu_1 - \frac{\omega}{4(1-\epsilon)}\right)$. Note that $C_{\delta/n}(T_i(t)) < \frac{1}{2}\left((1-\epsilon)\left(\mu_1 - \frac{\omega}{4(1-\epsilon)}\right) - \mu_i\right)$, true when $T_i(t) > h\left(\frac{1}{2}\left((1-\epsilon)\left(\mu_1 - \frac{\omega}{4(1-\epsilon)}\right) - \mu_i\right), \frac{\delta}{n}\right)$ implies that

$$\hat{\mu}_i(T_i(t)) + C_{\delta/n}(T_i(t)) \overset{\mathcal{E}}{\leqslant} \mu_i + 2C_{\delta/n}(T_i(t)) < (1-\epsilon)\left(\mu_1 - \frac{\omega}{4(1-\epsilon)}\right) \leqslant L_t.$$

Finally, we turn our attention to the difference $(1-\epsilon)\left(\mu_1 - \frac{\omega}{4(1-\epsilon)}\right) - \mu_i$. Recall that $\omega = \max(\gamma\mu_1, \min(\tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon))$.

$$\begin{aligned} (1-\epsilon)\left(\mu_1 - \frac{\omega}{4(1-\epsilon)}\right) - \mu_i &= (1-\epsilon)\mu_1 - \mu_i - \frac{1}{4}\omega \\ &= (1-\epsilon)\mu_1 - \mu_i - \frac{1}{4}\max(\gamma\mu_1, \min(\tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon)). \end{aligned}$$

By definition, $\tilde{\beta}_\epsilon = \min_{i \in M^c_\epsilon}(1-\epsilon)\mu_1 - \mu_i$. Hence, $\min(\tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon) \leqslant (1-\epsilon)\mu_1 - \mu_i$ for all $i \in M^c_{\epsilon+\frac{\gamma}{2}}$. Similarly, since $i \in M^c_{\epsilon+\frac{\gamma}{2}}$ by assumption, $(1-\epsilon-\frac{\gamma}{2})\mu_1 - \mu_i \geqslant 0$, which rearranges to $\frac{\gamma\mu_1}{2} \leqslant (1-\epsilon)\mu_1 - \mu_i$. Therefore,

$$(1-\epsilon)\mu_1 - \mu_i - \frac{1}{4}\max(\gamma\mu_1, \min(\tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon)) \geqslant \frac{1}{2}\left((1-\epsilon)\mu_1 - \mu_i\right) = \frac{\Delta_i - \epsilon\mu_1}{2}.$$

Hence, by monotonicity of $h(\cdot,\cdot)$,

$$h\left(\frac{1}{2}\left((1-\epsilon)\left(\mu_1-\frac{\omega}{4(1-\epsilon)}\right)-\mu_i\right),\frac{\delta}{n}\right)\leqslant h\left(\frac{\Delta_i-\epsilon\mu_1}{4},\frac{\delta}{n}\right).$$

Lastly, as above, since $i\in M^c_{\epsilon+\frac{\gamma}{2}}$, we have that $\Delta_i-\epsilon\mu_1=(1-\epsilon)\mu_1-\mu_i\geqslant\frac{1}{2}\gamma\mu_1$. Hence,

$$h\left(\frac{\Delta_i-\epsilon\mu_1}{4},\frac{\delta}{n}\right)\leqslant\min\left[h\left(\frac{\Delta_i-\epsilon\mu_1}{8},\frac{\delta}{n}\right),h\left(\frac{\gamma\mu_1}{8},\frac{\delta}{n}\right)\right].$$

Putting this together, if $T_i(t)\geqslant\min\left[h\left(\frac{\Delta_i-\epsilon\mu_1}{8},\frac{\delta}{n}\right),h\left(\frac{\gamma\mu_1}{8},\frac{\delta}{n}\right)\right]$, then $i\neq i_1(t)$ for all $i\in M^c_{\epsilon+\frac{\gamma}{2}}$. Summing over all such i bounds the size of set stated in the claim. □

We decompose the remaining event

$$\left\{t:\neg\text{STOP and } i^*\in M_{\omega/\mu_1}\text{ and } C_{\delta/n}(T_{i^*}(t))\leqslant\frac{\omega}{16(1-\epsilon)}\text{ and } i_1(t)\in M_{\epsilon+\frac{\gamma}{2}}\right\}$$

as

$$\begin{aligned}&\left\{t:\neg\text{STOP and } i^*\in M_{\omega/\mu_1}\text{ and } C_{\delta/n}(T_{i^*}(t))\leqslant\frac{\omega}{16(1-\epsilon)}\text{ and } i_1(t)\in M_{\epsilon+\frac{\gamma}{2}}\right.\\&\qquad\left.\text{and } i_2(t)\in M_{\epsilon+\frac{\gamma}{2}}\right\}\\\cup&\left\{t:\neg\text{STOP and } i^*\in M_{\omega/\mu_1}\text{ and } C_{\delta/n}(T_{i^*}(t))\leqslant\frac{\omega}{16(1-\epsilon)}\text{ and } i_1(t)\in M_{\epsilon+\frac{\gamma}{2}}\right.\\&\qquad\left.\text{and } i_2(t)\in M^c_{\epsilon+\frac{\gamma}{2}}\right\}.\end{aligned}$$

We proceed by bounding the cardinality of the first set.

Claim 1:

$$\left|\left\{t:\neg\text{STOP and } i^*\in M_{\omega/\mu_1}\text{ and } C_{\delta/n}(T_{i^*}(t))\leqslant\frac{\omega}{16(1-\epsilon)}\text{ and } i_1(t)\in M_{\epsilon+\frac{\gamma}{2}}\right.\right.$$
$$\left.\left.\text{and } i_2(t)\in M_{\epsilon+\frac{\gamma}{2}}\right\}\right|$$

$$\leqslant \sum_{i \in M_{\epsilon+\frac{\gamma}{2}}} \min\left[h\left(\frac{\epsilon\mu_1 - \Delta_i}{8}, \frac{\delta}{n}\right), h\left(\frac{\gamma\mu_1}{8}, \frac{\delta}{n}\right)\right]$$

Proof. Recall that $K = \{i : \hat{\mu}_i(T_i(t)) + C_{\delta/n}(T_i(t)) < L_t \text{ or } \hat{\mu}_i(T_i(t)) - C_{\delta/n}(T_i(t)) > U_t\}$ is the set of known arms and i_2 is sampled from $\widehat{G}^c \backslash K$. Hence, if an arm's lower bound exceeds U_t, it must be in K and therefore cannot be i_2. Recall that $i^*(t) = \arg\max \hat{\mu}_i(T_i(t)) + C_{\delta/n}(T_i(t))$. By the above event, $i^*(t) \in M_{\omega/\mu_1}$ and $C_{\delta/n}(T_{i^*}(t)) \leqslant \frac{\omega}{16(1-\epsilon)}$. Hence,

$$\begin{aligned}
U_t &= (1-\epsilon-\gamma)\left(\max_i \hat{\mu}_i(T_i(t)) + C_{\delta/n}(T_i(t))\right) \\
&= (1-\epsilon-\gamma)\left(\hat{\mu}_{i^*(t)}(T_{i^*(t)}(t)) + C_{\delta/n}(T_{i^*(t)}(t))\right) \\
&\overset{\mathcal{E}}{\leqslant} (1-\epsilon-\gamma)\left(\mu_{i^*(t)} + 2C_{\delta/n}(T_{i^*(t)}(t))\right) \\
&\leqslant (1-\epsilon-\gamma)\left(\mu_{i^*(t)} + \frac{\omega}{8(1-\epsilon)}\right) \\
&\leqslant (1-\epsilon-\gamma)\left(\mu_1 + \frac{\omega}{8(1-\epsilon)}\right)
\end{aligned}$$

Next, we bound the number of times an arm $i \in M_{\epsilon+\frac{\gamma}{2}}$ is sampled before its lower bound is above $(1-\epsilon-\gamma)\left(\mu_1 + \frac{\omega}{8(1-\epsilon)}\right)$. Note that $C_{\delta/n}(T_i(t)) < \frac{1}{2}\left(\mu_i - (1-\epsilon-\gamma)\left(\mu_1 + \frac{\omega}{8(1-\epsilon)}\right)\right)$, true when $T_i(t) > h\left(\frac{1}{2}\left(\mu_i - (1-\epsilon-\gamma)\left(\mu_1 + \frac{\omega}{8(1-\epsilon)}\right)\right), \frac{\delta}{n}\right)$ implies that

$$\hat{\mu}_i(T_i(t)) - C_{\delta/n}(T_i(t)) \overset{\mathcal{E}}{\geqslant} \mu_i - 2C_{\delta/n}(T_i(t)) > (1-\epsilon-\gamma)\left(\mu_1 + \frac{\omega}{8(1-\epsilon)}\right) \geqslant U_t.$$

Finally, we turn our attention to the difference $\mu_i - (1-\epsilon)\left(\mu_1 + \frac{\omega}{8}\right)$. Recall that $\omega = \max(\gamma\mu_1, \min(\tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon))$. Additionally, recall $\epsilon + \gamma \leqslant 1$.

$$\begin{aligned}
\mu_i - (1-\epsilon-\gamma)\left(\mu_1 + \frac{\omega}{8(1-\epsilon)}\right) &= \mu_i - (1-\epsilon)\mu_1 + \gamma\mu_1 - \frac{1}{8}\left(\frac{1-\epsilon-\gamma}{1-\epsilon}\right)\omega \\
&\geqslant \mu_i - (1-\epsilon)\mu_1 + \gamma\mu_1 - \frac{1}{8}\omega
\end{aligned}$$

Case 1a, $\omega = \min(\tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon)$ **and** $i \in M_\epsilon$**:**

By definition, $\tilde{\alpha}_\epsilon = \min_{i \in M_\epsilon} \mu_i - (1-\epsilon)\mu_1$. Hence, $\min(\tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon) \leqslant \mu_i - (1-\epsilon)\mu_1$ for all $i \in M_\epsilon$. Therefore,

$$\begin{aligned}\mu_i - (1-\epsilon)\mu_1 + \gamma\mu_1 - \frac{1}{8}\omega &= \mu_i - (1-\epsilon)\mu_1 + \gamma\mu_1 - \frac{1}{8}\min(\tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon) \\ &\geqslant \max\left(\mu_i - (1-\epsilon)\mu_1 - \frac{1}{8}\min(\tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon), \gamma\mu_1\right) \\ &\geqslant \max\left(\frac{7}{8}(\mu_i - (1-\epsilon)\mu_1), \gamma\mu_1\right)\end{aligned}$$

Case 1b, $\omega = \min(\tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon)$ **and** $i \in M_\epsilon^c \cap M_{\epsilon+\frac{\gamma}{2}}$

Since $\omega = \max(\gamma\mu_1, \min(\tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon))$, if $\omega = \min(\tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon)$, then $\frac{1}{2}\gamma\mu_1 < \min(\tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon)$. Since $\min(\tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon) = \min|\mu_i - (1-\epsilon)\mu_1|$, the set $M_\epsilon^c \cap M_{\epsilon+\frac{\gamma}{2}}$ is empty and there is nothing to prove.

Case 2a, $\omega = \gamma\mu_1$ **and** $i \in M_\epsilon$

$$\mu_i - (1-\epsilon)\mu_1 + \gamma\mu_1 - \frac{1}{8}\omega = \mu_i - (1-\epsilon)\mu_1 + \frac{7}{8}\gamma\mu_1 \geqslant \max\left(\mu_i - (1-\epsilon)\mu_1, \frac{7}{8}\gamma\mu_1\right).$$

Case 2b, $\omega = \gamma\mu_1$ **and** $i \in M_\epsilon^c \cap M_{\epsilon+\frac{\gamma}{2}}$

For $i \in M_\epsilon^c \cap M_{\epsilon+\frac{\gamma}{2}}$, $\mu_i - (1-\epsilon-\frac{\gamma}{2})\mu_1 \geqslant 0$. Hence, $\mu_i - (1-\epsilon)\mu_1 \geqslant \frac{-\gamma\mu_1}{2}$. Therefore,

$$\mu_i - (1-\epsilon)\mu_1 + \gamma\mu_1 - \frac{1}{8}\omega = \mu_i - (1-\epsilon)\mu_1 + \frac{7}{8}\gamma\mu_1 \geqslant \frac{3}{8}\gamma\mu_1 \geqslant \max\left(\frac{1}{4}\gamma\mu_1, \frac{(1-\epsilon)\mu_1 - \mu_i}{4}\right).$$

Combining all cases, by monotonicity of $h(\cdot,\cdot)$ and symmetry in its first argument, we see that

$$h\left(\frac{1}{2}\left(\mu_i - (1-\epsilon-\gamma)\left(\mu_1 + \frac{\omega}{8(1-\epsilon)}\right)\right), \frac{\delta}{n}\right) \leqslant \min\left[h\left(\frac{\gamma\mu_1}{8}, \frac{\delta}{n}\right), h\left(\frac{\epsilon\mu_1 - \Delta_i}{8}, \frac{\delta}{n}\right)\right].$$

Putting this together, if $T_i(t) \geqslant \min\left[h\left(\frac{\epsilon\mu_1-\Delta_i}{8}, \frac{\delta}{n}\right), h\left(\frac{\gamma\mu_1}{8}, \frac{\delta}{n}\right)\right]$, then $i \neq i_2(t)$ for all $i \in M_{\epsilon+\frac{\gamma}{2}}$. Summing over all such i bounds the size of set stated in the claim. □

6.B.2.4 Step 3: Controlling the complexity until stopping occurs

In this step, we turn our attention to the final event to control:

$$\mathcal{S} := \Bigg\{ t : \neg\text{STOP and } i^* \in M_{\omega/\mu_1} \text{ and } C_{\delta/n}(T_{i^*}(t)) \leqslant \frac{\omega}{16(1-\epsilon)} \\ \text{and } i_1(t) \in M_{\epsilon+\frac{\gamma}{2}} \text{ and } i_2(t) \in M^c_{\epsilon+\frac{\gamma}{2}} \Bigg\}. \tag{6.7}$$

For brevity, we will refer to this set as $\mathcal{S}$ for this step. The objective will be to bound the time before each arms lower bound either clears U_t or its upper bound clears L_t which implies the stopping condition. To do so, we introduce, two events:

$$E_1(t) := \{\hat{\mu}_{i_1(t)}(T_{i_1(t)}(t)) - C_{\delta/n}(T_{i_1(t)}(t)) > U_t\} \tag{6.8}$$

and

$$E_2(t) := \{\hat{\mu}_{i_2(t)}(T_{i_2(t)}(t)) + C_{\delta/n}(T_{i_2(t)}(t)) < L_t\}. \tag{6.9}$$

If $E_1(t)$ is true, then $\hat{\mu}_i(T_i) - C_{\delta/n}(T_i(t)) > L_t$ for all $i \in \widehat{G}$. If $E_2(t)$ is true, then $\hat{\mu}_i(T_i) + C_{\delta/n}(T_i(t)) < U_t$ for all $i \in \widehat{G}^c$. Hence, by line 6 of $(\texttt{ST})^2$, if both $E_1(t)$ and $E_2(t)$ are true, then $(\texttt{ST})^2$ terminates.

Claim 0: $|\mathcal{S} \cap \{t : \neg E_1(t)\}| \leqslant \sum_{i \in M_{\epsilon+\frac{\gamma}{2}}} \min\left[h\left(\frac{\epsilon\mu_1 - \Delta_i}{4}, \frac{\delta}{n}\right), h\left(\frac{\gamma\mu_1}{4}, \frac{\delta}{n}\right)\right]$.

Proof. Recall that by the set $\mathcal{S}$, we have that $i_1(t) \in M_{\epsilon+\frac{\gamma}{2}}$. Furthermore, by the set $\mathcal{S}$, we have that $i^*(t) \in M_{\omega/\mu_1}$ and $C_{\delta/n}(T_{i^*}(t)) \leqslant \omega/16(1-\epsilon)$. Hence,

$$\begin{aligned}
U_t &= (1-\epsilon-\gamma)\left(\max_i \hat{\mu}_i(T_i(t)) + C_{\delta/n}(T_i(t))\right) \\
&= (1-\epsilon-\gamma)\left(\hat{\mu}_{i^*(t)}(T_{i^*(t)}(t)) + C_{\delta/n}(T_{i^*(t)}(t))\right) \\
&\overset{\mathcal{E}}{\leqslant} (1-\epsilon-\gamma)\left(\mu_{i^*(t)} + 2C_{\delta/n}(T_{i^*(t)}(t))\right) \\
&\leqslant (1-\epsilon-\gamma)\left(\mu_{i^*(t)} + \frac{\omega}{8(1-\epsilon)}\right) \\
&\leqslant (1-\epsilon-\gamma)\left(\mu_1 + \frac{\omega}{8(1-\epsilon)}\right)
\end{aligned}$$

If $C_{\delta/n}(T_i) \leqslant \frac{1}{2}\left(\mu_i - (1-\epsilon-\gamma)\left(\mu_1 + \frac{\omega}{8(1-\epsilon)}\right)\right)$,
true when $T_i \geqslant h\left(\frac{1}{2}\left(\mu_i - (1-\epsilon-\gamma)\left(\mu_1 + \frac{\omega}{8(1-\epsilon)}\right)\right), \frac{\delta}{n}\right)$, then

$$\hat{\mu}_i(T_i) - C_{\delta/n}(T_i) \geqslant \mu_i - 2C_{\delta/n}(T_i) \geqslant (1-\epsilon-\gamma)\left(\mu_1 + \frac{\omega}{8(1-\epsilon)}\right) \geqslant U_t.$$

The remainder of the proof of this claim focuses on controlling the difference: $\mu_i - (1-\epsilon-\gamma)\left(\mu_1 + \frac{\omega}{8(1-\epsilon)}\right)$ in the case that $\omega = \min(\tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon)$ and $\omega = \gamma\mu_1$. Recall that $\omega = \max(\gamma\mu_1, \min(\tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon))$. Hence, if any possible $i \in M_{\epsilon+\frac{\gamma}{2}}$ has received sufficiently many samples, since $i_1(t) \in M_{\epsilon+\frac{\gamma}{2}}$, this implies $E_1(t)$.

Case 1a, $\omega = \min(\tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon)$ **and** $i \in M_\epsilon$

We focus on the difference $\mu_i - (1-\epsilon-\gamma)\left(\mu_1 + \frac{\omega}{8(1-\epsilon)}\right)$. Recall that $\epsilon + \gamma \leqslant 1$.

$$\begin{aligned}
\mu_i - (1-\epsilon-\gamma)\left(\mu_1 + \frac{\omega}{8(1-\epsilon)}\right) &= \mu_i - (1-\epsilon-\gamma)\left(\mu_1 + \frac{\min(\tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon)}{8(1-\epsilon)}\right) \\
&= \mu_i - (1-\epsilon)\mu_1 + \gamma\mu_1 - \frac{1}{8}\left(\frac{1-\epsilon-\gamma}{1-\epsilon}\right)\min(\tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon) \\
&\overset{\gamma \geqslant 0 \text{ and } \epsilon+\gamma \leqslant 1}{\geqslant} \mu_i - (1-\epsilon)\mu_1 - \frac{1}{8}\min(\tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon) \\
&\geqslant \frac{1}{2}(\mu_i - (1-\epsilon)\mu_1) = \frac{\epsilon\mu_1 - \Delta_i}{2}
\end{aligned}$$

where the final step follows since $\min(\tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon) \leqslant \tilde{\alpha}_\epsilon \leqslant \mu_i - (1-\epsilon)\mu_1$ by definition for all $i \in M_\epsilon$. Then by monotonicity of $h(\cdot,\cdot)$,

$$h\left(\frac{1}{2}\left(\mu_i - (1-\epsilon-\gamma)\left(\mu_1 + \frac{\omega}{8(1-\epsilon)}\right)\right), \frac{\delta}{n}\right) \leqslant h\left(\frac{\epsilon\mu_1 - \Delta_i}{4}, \frac{\delta}{n}\right).$$

Lastly, in this setting, $\gamma\mu_1 \leqslant \min(\tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon) \leqslant \epsilon\mu_1 - \Delta_i$ since $\omega = \min(\tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon)$. Hence, it is trivially true that

$$h\left(\frac{\epsilon\mu_1 - \Delta_i}{4}, \frac{\delta}{n}\right) = \min\left[h\left(\frac{\epsilon\mu_1 - \Delta_i}{4}, \frac{\delta}{n}\right), h\left(\frac{\gamma\mu_1}{4}, \frac{\delta}{n}\right)\right]$$

Case 1b, $\omega = \min(\tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon)$ **and** $i \in M_\epsilon^c \cap M_{\epsilon+\frac{\gamma}{2}}$

Since $\omega = \max(\gamma\mu_1, \min(\tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon))$, if $\omega = \min(\tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon)$, then $\frac{1}{2}\gamma\mu_1 < \min(\tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon)$. Since $\min(\tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon) = \min|\mu_i - (1-\epsilon)\mu_1|$, the set $M_\epsilon^c \cap M_{\epsilon+\frac{\gamma}{2}}$ is empty and there is nothing to prove.

Case 2a, $\omega = \gamma\mu_1$ **and** $i \in M_\epsilon$

Next, we bound the difference $\mu_i - (1-\epsilon-\gamma)\left(\mu_1 + \frac{\omega}{4(1-\epsilon)}\right)$.

$$\begin{aligned}\mu_i - (1-\epsilon-\gamma)\left(\mu_1 + \frac{\omega}{8(1-\epsilon)}\right) &= \mu_i - (1-\epsilon)\mu_1 + \gamma\mu_1 - \frac{1}{8}\left(\frac{1-\epsilon-\gamma}{1-\epsilon}\right)\gamma\mu_1 \\ &\geqslant \mu_i - (1-\epsilon)\mu_1 + \gamma\mu_1\left(1 - \frac{1}{8}\left(\frac{1-\epsilon-\gamma}{1-\epsilon}\right)\right)\end{aligned}$$

Since $i \in M_\epsilon$, $\mu_i - (1-\epsilon)\mu_1 \geqslant 0$. Using this and the fact that $\epsilon, \gamma \geqslant 0$ and $\epsilon + \gamma \leqslant 1$,

$$\begin{aligned}\mu_i - (1-\epsilon)\mu_1 + \gamma\mu_1\left(1 - \frac{1}{8}\left(\frac{1-\epsilon-\gamma}{1-\epsilon}\right)\right) &\geqslant \mu_i - (1-\epsilon)\mu_1 + \frac{7}{8}\gamma\mu_1 \\ &\geqslant \max\left(\mu_i - (1-\epsilon)\mu_1, \frac{7}{8}\gamma\mu_1\right) \\ &\geqslant \frac{1}{2}\max\left(\epsilon\mu_1 - \Delta_i, \gamma\mu_1\right)\end{aligned}$$

Therefore, we have that

$$h\left(\frac{1}{2}\left(\mu_i - (1-\epsilon-\gamma)\left(\mu_1 + \frac{\omega}{8(1-\epsilon)}\right)\right), \frac{\delta}{n}\right) \leqslant h\left(\frac{\epsilon\mu_1 - \Delta_i}{4}, \frac{\delta}{n}\right)$$

and

$$h\left(\frac{1}{2}\left(\mu_i - (1-\epsilon-\gamma)\left(\mu_1 + \frac{\omega}{8(1-\epsilon)}\right)\right), \frac{\delta}{n}\right) \leqslant h\left(\frac{\gamma\mu_1}{4}, \frac{\delta}{n}\right).$$

Hence,

$$h\left(\frac{1}{2}\left(\mu_i - (1-\epsilon-\gamma)\left(\mu_1 + \frac{\omega}{8(1-\epsilon)}\right)\right), \frac{\delta}{n}\right) \leqslant \min\left[h\left(\frac{\epsilon\mu_1 - \Delta_i}{4}, \frac{\delta}{n}\right), h\left(\frac{\gamma\mu_1}{4}, \frac{\delta}{n}\right)\right].$$

Case 2b, $\omega = \gamma\mu_1$ **and** $i \in M_\epsilon^c \cap M_{\epsilon+\frac{\gamma}{2}}$

As before,

$$\mu_i - (1-\epsilon-\gamma)\left(\mu_1 + \frac{\omega}{8(1-\epsilon)}\right) = \mu_i - (1-\epsilon)\mu_1 + \gamma\mu_1 - \frac{1}{8}\left(\frac{1-\epsilon-\gamma}{1-\epsilon}\right)\gamma\mu_1$$

Since $i \in M_\epsilon^c \cap M_{\epsilon+\frac{\gamma}{2}}$, we have that $\mu_i - (1-\epsilon-\frac{\gamma}{2})\mu_1 \geqslant 0$. Rearranging implies that $\mu_i - (1-\epsilon)\mu_1 \geqslant \frac{-1}{2}\gamma\mu_1$. Hence,

$$\mu_i - (1-\epsilon)\mu_1 + \gamma\mu_1 - \frac{1}{8}\left(\frac{1-\epsilon-\gamma}{1-\epsilon}\right)\gamma\mu_1 \geqslant \frac{1}{2}\gamma\mu_1 - \frac{1}{8}\left(\frac{1-\epsilon-\gamma}{1-\epsilon}\right)\gamma\mu_1 \geqslant \frac{3}{8}\gamma\mu_1.$$

Hence,

$$h\left(\frac{1}{2}\left(\mu_i - (1-\epsilon-\gamma)\left(\mu_1 + \frac{\omega}{8(1-\epsilon)}\right)\right), \frac{\delta}{n}\right) \leqslant h\left(\frac{3\gamma\mu_1}{8}, \frac{\delta}{n}\right).$$

Additionally, as above, if $i \in M_\epsilon^c \cap M_{\epsilon+\frac{\gamma}{2}}$, we have that $\mu_i - (1-\epsilon-\frac{\gamma}{2})\mu_1 \geqslant 0$ which implies that $(1-\epsilon)\mu_1 - \mu_i \leqslant \frac{1}{2}\gamma\mu_1$. Hence

$$h\left(\frac{3\gamma\mu_1}{8}, \frac{\delta}{n}\right) \leqslant \min\left[h\left(\frac{\Delta_i - \epsilon\mu_1}{4}, \frac{\delta}{n}\right), h\left(\frac{\gamma\mu_1}{4}, \frac{\delta}{n}\right)\right].$$

Therefore, if T_i exceeds the above, then $E_1(t)$ is true for an $i_1 \in M_\epsilon^c \cap M_{\epsilon+\frac{\gamma}{2}}$. Combining all cases, we see that for $i_1 \in M_{\epsilon+\frac{\gamma}{2}}$, if

$$T_{i_1(t)}(t) > \min\left[h\left(\frac{\epsilon\mu_1 - \Delta_i}{4}, \frac{\delta}{n}\right), h\left(\frac{\gamma\mu_1}{4}, \frac{\delta}{n}\right)\right],$$

Then $E_1(t)$ is true. Summing over all possible $i_1 \in M_{\epsilon+\frac{\gamma}{2}}$ proves the claim. □

Claim 1: $|\mathcal{S} \cap \{t : E_1(t)\} \cap \{t : \neg E_2(t)\}| \leqslant \sum_{i \in M_{\epsilon+\frac{\gamma}{2}}^c} \min\left[h\left(\frac{\epsilon\mu_1 - \Delta_i}{8}, \frac{\delta}{n}\right), h\left(\frac{\gamma\mu_1}{8}, \frac{\delta}{n}\right)\right]$.

Proof. By the events in set $\mathcal{S}$, $C_{\delta/n}(T_{i^*}(t)) \leqslant \frac{\omega}{16(1-\epsilon)}$. Therefore,

$$\mu_{i^*} + \frac{\omega}{8(1-\epsilon)} \geqslant \mu_{i^*} + 2C_{\delta/n}(T_{i^*}(t)) \overset{\mathcal{E}}{\geqslant} \hat{\mu}_{i^*}(T_{i^*}(t)) + C_{\delta/n}(T_{i^*}(t)) \geqslant \hat{\mu}_1(T_1(t)) + C_{\delta/n}(T_1(t)) \overset{\mathcal{E}}{\geqslant} \mu_1.$$

Hence, $\mu_{i^*} \geqslant \mu_1 - \frac{\omega}{8(1-\epsilon)}$. Rearranging this, we see that $\mu_{i^*} - \left(1 - \frac{\omega}{8\mu_1(1-\epsilon)}\right)\mu_1 \geqslant 0$ which implies that $i^* \in M_{\frac{\omega}{8\mu_1(1-\epsilon)}}$. Hence,

$$\begin{aligned} L_t = (1-\epsilon)\left(\max_i \hat{\mu}_i(T_i(t)) - C_{\delta/n}(T_i(t))\right)(1-\epsilon)\left(\hat{\mu}_{i^*}(T_{i^*}(t)) - C_{\delta/n}(T_{i^*}(t))\right) \\ \overset{\mathcal{E}}{\geqslant} (1-\epsilon)\left(\mu_{i^*} - 2C_{\delta/n}(T_{i^*}(t))\right) \\ \geqslant (1-\epsilon)\left(\mu_{i^*} - \frac{\omega}{8(1-\epsilon)}\right) \\ \geqslant (1-\epsilon)\left(\mu_1 - \frac{\omega}{4(1-\epsilon)}\right) \end{aligned}$$

As before, we seek a lower bound for the difference $(1-\epsilon)\left(\mu_1 - \frac{\omega}{4(1-\epsilon)}\right) - \mu_i$.

Case 1: $\omega = \min(\tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon)$

$$\begin{aligned} (1-\epsilon)\left(\mu_1 - \frac{\omega}{4(1-\epsilon)}\right) - \mu_i &= (1-\epsilon)\mu_1 - \mu_i - \frac{1}{4}\min(\tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon) \\ &\geqslant \frac{1}{2}\left((1-\epsilon)\mu_1 - \mu_i\right) \end{aligned}$$

since $(1-\epsilon)\mu_1 - \mu_i \geqslant \min(\tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon)$. Therefore, we have that

$$h\left(\frac{1}{2}\left((1-\epsilon)\left(\mu_1 - \frac{\omega}{4(1-\epsilon)}\right) - \mu_i\right), \frac{\delta}{n}\right) \leqslant h\left(\frac{\Delta_i - \epsilon\mu_1}{4}, \frac{\delta}{n}\right).$$

Lastly, in this setting, $\gamma\mu_1 \leqslant \min(\tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon) \leqslant \epsilon\mu_1 - \Delta_i$ since $\omega = \min(\tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon)$. Hence, it is trivially true that

$$h\left(\frac{\Delta_i - \epsilon\mu_1}{4}, \frac{\delta}{n}\right) = \min\left[h\left(\frac{\Delta_i - \epsilon\mu_1}{4}, \frac{\delta}{n}\right), h\left(\frac{\gamma\mu_1}{4}, \frac{\delta}{n}\right)\right].$$

Case 2: $\omega = \gamma\mu_1$

Assume that $\gamma\mu_1 > \min(\tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon)$, as equality is covered by the previous case.

Hence,

$$(1-\epsilon)\left(\mu_1 - \frac{\omega}{4(1-\epsilon)}\right) - \mu_i = (1-\epsilon)\mu_1 - \mu_i - \frac{1}{4}\gamma\mu_1$$

Recall that we seek to control $i_2 \in M^c_{\epsilon+\frac{\gamma}{2}}$. For any $i \in M^c_{\epsilon+\frac{\gamma}{2}}$, we have that $(1-\epsilon-\frac{\gamma}{2})\mu_1 - \mu_i \geqslant 0$. Rearranging, we see that $(1-\epsilon)\mu_1 - \mu_i \geqslant \frac{1}{2}\gamma\mu_1$ which implies that

$$(1-\epsilon)\mu_1 - \mu_i - \frac{1}{4}\gamma\mu_1 \geqslant \frac{1}{2}((1-\epsilon)\mu_1 - \mu_i).$$

Therefore, we have that

$$h\left(\frac{1}{2}\left((1-\epsilon)\left(\mu_1 - \frac{\omega}{4(1-\epsilon)}\right) - \mu_i\right), \frac{\delta}{n}\right) \leqslant h\left(\frac{\Delta_i - \epsilon\mu_1}{4}, \frac{\delta}{n}\right)$$

is this setting as well. Similarly, since $\Delta_i - \epsilon\mu_1 \geqslant \frac{1}{2}\gamma\mu_1$, we likewise have that

$$h\left(\frac{\Delta_i - \epsilon\mu_1}{4}, \frac{\delta}{n}\right) \leqslant \min\left[h\left(\frac{\Delta_i - \epsilon\mu_1}{8}, \frac{\delta}{n}\right), h\left(\frac{\gamma\mu_1}{8}, \frac{\delta}{n}\right)\right].$$

Hence, if T_i exceeds the right-hand side of the preceding inequality, then for any $i \in M^c_{\epsilon+\frac{\gamma}{2}}$, its upper bound is below L_t. Hence, for $i_2(t) \in M^c_{\epsilon+\frac{\gamma}{2}}$, this implies event $E_2(t)$. Summing over all possible values of $i_2(t) \in M^c_{\epsilon+\frac{\gamma}{2}}$ proves the claim. □

Claim 2: The cardinality of $\mathcal{S}$ is bounded as $|\mathcal{S}| \leqslant \sum_{i=1}^{n} \min\left[h\left(\frac{\Delta_i - \epsilon\mu_1}{8}, \frac{\delta}{n}\right), h\left(\frac{\gamma\mu_1}{8}, \frac{\delta}{n}\right)\right]$.

Proof. First, $\mathcal{S}$ may be decomposed as

$$|\mathcal{S}| = |\mathcal{S} \cap \{t : \neg E_1(t)\}| + |\mathcal{S} \cap \{t : E_1(t)\} \cap \{t : \neg E_2(t)\}| + |\mathcal{S} \cap \{t : E_1(t)\} \cap \{t : E_2(t)\}|$$

Note that $|\mathcal{S} \cap \{t : E_1(t)\} \cap \{t : E_2(t)\}| = 0$ because we have assumed in set $\mathcal{S}$ that $(\mathtt{ST})^2$ has not stopped, and $\{t : E_1(t)\} \cap \{t : E_2(t)\}$ implies termination. By Claim 0, $|\mathcal{S} \cap \{t : \neg E_1(t)\}| \leqslant \sum_{i \in M_{\epsilon+\frac{\gamma}{2}}} \min\left[h\left(\frac{\epsilon\mu_1 - \Delta_i}{4}, \frac{\delta}{n}\right), h\left(\frac{\gamma\mu_1}{4}, \frac{\delta}{n}\right)\right]$. By Claim 1, $|\mathcal{S} \cap \{t : E_1(t)\} \cap \{t : \neg E_2(t)\}| \leqslant \sum_{i \in M^c_{\epsilon+\frac{\gamma}{2}}} \min\left[h\left(\frac{\epsilon\mu_1 - \Delta_i}{8}, \frac{\delta}{n}\right), h\left(\frac{\gamma\mu_1}{8}, \frac{\delta}{n}\right)\right]$. Recalling that h is assumed to be symmetric in its first argument and summing the two terms proves the claim. □

6.B.2.5 Step 4: Putting it all together

Recall that the total number of rounds T that $(\mathtt{ST})^2$ runs for is given by $T = |\{t : \neg\text{STOP}\}|$. To bound this quantity, we have decomposed the set $\{t : \neg\text{STOP}\}$ into many subsets. Below, we show this decomposition.

$$
\begin{aligned}
&\{t : \neg\text{STOP}\} = \\
&\{t : \neg\text{STOP and } i^* \notin M_{\omega/\mu_1}\} \\
&\cup \left\{t : \neg\text{STOP and } i^* \in M_{\omega/\mu_1} \text{ and } C_{\delta/n}(T_{i^*}(t)) > \frac{\omega}{16(1-\epsilon)}\right\} \\
&\cup \left\{t : \neg\text{STOP and } i^* \in M_{\omega/\mu_1} \text{ and } C_{\delta/n}(T_{i^*}(t)) \leqslant \frac{\omega}{16(1-\epsilon)} \text{ and } i_1(t) \in M^c_{\epsilon+\frac{\gamma}{2}}\right\} \\
&\cup \left\{t : \neg\text{STOP and } i^* \in M_{\omega/\mu_1} \text{ and } C_{\delta/n}(T_{i^*}(t)) \leqslant \frac{\omega}{16(1-\epsilon)} \text{ and } i_1(t) \in M_{\epsilon+\frac{\gamma}{2}}\right. \\
&\qquad\qquad \left.\text{and } i_2(t) \in M_{\epsilon+\frac{\gamma}{2}}\right\} \\
&\cup \left\{t : \neg\text{STOP and } i^* \in M_{\omega/\mu_1} \text{ and } C_{\delta/n}(T_{i^*}(t)) \leqslant \frac{\omega}{16(1-\epsilon)} \text{ and } i_1(t) \in M_{\epsilon+\frac{\gamma}{2}}\right. \\
&\qquad\qquad \left.\text{and } i_2(t) \in M^c_{\epsilon+\frac{\gamma}{2}}\right\}.
\end{aligned}
$$

Hence, by a union bound and plugging in the results of the above steps,

$$
\begin{aligned}
&|\{t : \neg\text{STOP}\}| \leqslant \\
&\left|\{t : \neg\text{STOP and } i^* \notin M_{\omega/\mu_1}\}\right| \\
&+ \left|\left\{t : \neg\text{STOP and } i^* \in M_{\omega/\mu_1} \text{ and } \exists i \in M_{\omega/\mu_1} : C_{\delta/n}(T_i(t)) > \frac{\omega}{8(1-\epsilon)}\right\}\right| \\
&+ \left|\left\{t : \neg\text{STOP and } i^* \in M_{\omega/\mu_1} \text{ and } C_{\delta/n}(T_{i^*}(t)) \leqslant \frac{\omega}{16(1-\epsilon)} \text{ and } i_1(t) \in M^c_{\epsilon+\frac{\gamma}{2}}\right\}\right| \\
&+ \left|\left\{t : \neg\text{STOP and } i^* \in M_{\omega/\mu_1} \text{ and } C_{\delta/n}(T_{i^*}(t)) \leqslant \frac{\omega}{16(1-\epsilon)} \text{ and } i_1(t) \in M_{\epsilon+\frac{\gamma}{2}}\right.\right. \\
&\qquad\qquad \left.\left.\text{and } i_2(t) \in M_{\epsilon+\frac{\gamma}{2}}\right\}\right| \\
&+ \left|\left\{t : \neg\text{STOP and } i^* \in M_{\omega/\mu_1} \text{ and } C_{\delta/n}(T_{i^*}(t)) \leqslant \frac{\omega}{16(1-\epsilon)} \text{ and } i_1(t) \in M_{\epsilon+\frac{\gamma}{2}}\right.\right.
\end{aligned}
$$

$$
\begin{aligned}
&\qquad \text{and } i_2(t) \in M^c_{\epsilon+\frac{\gamma}{2}} \Big\} \Big| \\
&\leqslant \sum_{i \in M^c_{\omega/\mu_1}} \min\left\{ h\left(\frac{\gamma\mu_1}{2}, \frac{\delta}{n}\right), \min\left[h\left(\frac{\Delta_i}{2}, \frac{\delta}{n}\right), h\left(\frac{\min(\tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon)}{2}, \frac{\delta}{n}\right)\right]\right\} \\
&\quad + \sum_{i \in M_{\omega/\mu_1}} \min\left\{ h\left(\frac{\gamma\mu_1}{16}, \frac{\delta}{n}\right), \min\left[h\left(\frac{\Delta_i}{16}, \frac{\delta}{n}\right), h\left(\frac{\min(\tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon)}{16(1-\epsilon)}, \frac{\delta}{n}\right)\right]\right\} \\
&\quad + \sum_{i \in M^c_{\epsilon+\frac{\gamma}{2}}} \min\left[h\left(\frac{\Delta_i - \epsilon\mu_1}{8}, \frac{\delta}{n}\right), h\left(\frac{\gamma\mu_1}{8}, \frac{\delta}{n}\right)\right] \\
&\quad + \sum_{i \in M_{\epsilon+\frac{\gamma}{2}}} \min\left[h\left(\frac{\epsilon\mu_1 - \Delta_i}{8}, \frac{\delta}{n}\right), h\left(\frac{\gamma\mu_1}{8}, \frac{\delta}{n}\right)\right] \\
&\quad + \sum_{i=1}^{n} \min\left[h\left(\frac{\Delta_i - \epsilon\mu_1}{8}, \frac{\delta}{n}\right), h\left(\frac{\gamma\mu_1}{8}, \frac{\delta}{n}\right)\right] \\
&\overset{(\epsilon \leqslant 1/2)}{\leqslant} \sum_{i=1}^{n} \min\left\{ h\left(\frac{\gamma\mu_1}{16}, \frac{\delta}{n}\right), \min\left[h\left(\frac{\Delta_i}{16}, \frac{\delta}{n}\right), h\left(\frac{\min(\tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon)}{16(1-\epsilon)}, \frac{\delta}{n}\right)\right]\right\} \\
&\quad + 2\sum_{i=1}^{n} \min\left[h\left(\frac{\Delta_i - \epsilon\mu_1}{8}, \frac{\delta}{n}\right), h\left(\frac{\gamma\mu_1}{8}, \frac{\delta}{n}\right)\right] \\
&\leqslant 4\sum_{i=1}^{n} \min\left\{ \max\left\{ h\left(\frac{\Delta_i - \epsilon\mu_1}{16}, \frac{\delta}{n}\right), \min\left[h\left(\frac{\Delta_i}{16}, \frac{\delta}{n}\right), h\left(\frac{\min(\tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon))}{16(1-\epsilon)}, \frac{\delta}{n}\right)\right]\right\}, \right. \\
&\qquad\qquad \left. h\left(\frac{\gamma\mu_1}{16}, \frac{\delta}{n}\right)\right\}
\end{aligned}
$$

Next, by Lemma 6.33, we may bound the minimum of $h(\cdot,\cdot)$ functions.

$$
\begin{aligned}
&4\sum_{i=1}^{n} \min\left\{ \max\left\{ h\left(\frac{\Delta_i - \epsilon\mu_1}{16}, \frac{\delta}{n}\right), \min\left[h\left(\frac{\Delta_i}{16}, \frac{\delta}{n}\right), h\left(\frac{\min(\tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon)}{16(1-\epsilon)}, \frac{\delta}{n}\right)\right]\right\}, \right. \\
&\qquad\qquad \left. h\left(\frac{\gamma\mu_1}{16}, \frac{\delta}{n}\right)\right\} \\
&= 4\sum_{i=1}^{n} \min\left\{ \max\left\{ h\left(\frac{\Delta_i - \epsilon\mu_i}{16}, \frac{\delta}{n}\right),\right.\right.
\end{aligned}
$$

$$
\begin{aligned}
&\min\left[h\left(\frac{\Delta_i}{16},\frac{\delta}{n}\right),\max\left[h\left(\frac{\tilde{\alpha}_\epsilon}{16(1-\epsilon)},\frac{\delta}{n}\right),h\left(\frac{\tilde{\beta}_\epsilon}{16(1-\epsilon)},\frac{\delta}{n}\right)\right]\right]\Bigg\},\\
&h\left(\frac{\gamma\mu_i}{16},\frac{\delta}{n}\right)\Bigg\}\\
\leqslant 4\sum_{i=1}^{n}\min\Bigg\{&\max\left\{h\left(\frac{\Delta_i-\epsilon\mu_i}{16},\frac{\delta}{n}\right),\right.\\
&\left.\max\left[h\left(\frac{\Delta_i+\frac{\tilde{\alpha}_\epsilon}{1-\epsilon}}{32},\frac{\delta}{n}\right),h\left(\frac{\Delta_i+\frac{\tilde{\beta}_\epsilon}{1-\epsilon}}{32},\frac{\delta}{n}\right)\right]\right\},\\
&h\left(\frac{\gamma\mu_i}{16},\frac{\delta}{n}\right)\Bigg\}\\
=4\sum_{i=1}^{n}\min\Bigg\{&\max\left\{h\left(\frac{\Delta_i-\epsilon\mu_i}{16},\frac{\delta}{n}\right),h\left(\frac{\Delta_i+\frac{\tilde{\alpha}_\epsilon}{1-\epsilon}}{32},\frac{\delta}{n}\right),h\left(\frac{\Delta_i+\frac{\tilde{\beta}_\epsilon}{1-\epsilon}}{32},\frac{\delta}{n}\right)\right\},\\
&h\left(\frac{\gamma\mu_i}{16},\frac{\delta}{n}\right)\Bigg\}
\end{aligned}
$$

Finally, we use Lemma 6.32 to bound the function $h(\cdot,\cdot)$. Since $\delta \leqslant 1/2$, $\delta/n \leqslant 2e^{-e/2}$. Further, $|\epsilon\mu_1-\Delta_i| \leqslant 8$ for all i and $\epsilon \leqslant 1/2$ implies that $\frac{1}{8(1-\epsilon)}|\epsilon\mu_1-\Delta_i| \leqslant 2$ and $\frac{1}{8(1-\epsilon)}\min(\tilde{\alpha}_\epsilon,\tilde{\beta}_\epsilon) \leqslant 2$. $\Delta_i \leqslant 16$ for all i, gives $0.125\Delta_i \leqslant 2$. Lastly, $\gamma \leqslant 16/\mu_1$ implies that $\frac{\gamma\mu_1}{8} \leqslant 2$. Therefore,

$$
\begin{aligned}
&4\sum_{i=1}^{n}\min\left\{\max\left\{h\left(\frac{\Delta_i-\epsilon\mu_i}{16},\frac{\delta}{n}\right),h\left(\frac{\Delta_i+\frac{\tilde{\alpha}_\epsilon}{1-\epsilon}}{32},\frac{\delta}{n}\right),h\left(\frac{\Delta_i+\frac{\tilde{\beta}_\epsilon}{1-\epsilon}}{32},\frac{\delta}{n}\right)\right\},\right.\\
&\left.h\left(\frac{\gamma\mu_i}{16},\frac{\delta}{n}\right)\right\}\\
\leqslant 4\sum_{i=1}^{n}\min\Bigg\{&\max\left\{\frac{1024}{(\epsilon\mu_1-\Delta_i)^2}\log\left(\frac{2n}{\delta}\log_2\left(\frac{3072n}{\delta(\epsilon\mu_1-\Delta_i)^2}\right)\right),\right.\\
&\frac{4096}{(\Delta_i+\frac{\tilde{\alpha}_\epsilon}{1-\epsilon})^2}\log\left(\frac{2n}{\delta}\log_2\left(\frac{12288n}{\delta(\Delta_i+\frac{\tilde{\alpha}_\epsilon}{1-\epsilon})^2}\right)\right),\\
&\left.\frac{4096}{(\Delta_i+\frac{\tilde{\beta}_\epsilon}{1-\epsilon})^2}\log\left(\frac{2n}{\delta}\log_2\left(\frac{12288n}{\delta(\Delta_i+\frac{\tilde{\beta}_\epsilon}{1-\epsilon})^2}\right)\right)\right\}
\end{aligned}
$$

$$\frac{1024}{\gamma^2\mu_1^2}\log\left(\frac{2n}{\delta}\log_2\left(\frac{3072n}{\delta\gamma^2\mu_1^2}\right)\right)\Bigg\}$$

$$= 4\sum_{i=1}^{n}\min\Bigg\{\max\Bigg\{\frac{1024}{((1-\epsilon)\mu_1-\mu_i)^2}\log\left(\frac{2n}{\delta}\log_2\left(\frac{3072n}{\delta((1-\epsilon)\mu_1-\mu_i)^2}\right)\right),$$

$$\frac{4096}{(\mu_1+\frac{\tilde{\alpha}_\epsilon}{1-\epsilon}-\mu_i)^2}\log\left(\frac{2n}{\delta}\log_2\left(\frac{12288n}{\delta(\mu_1+\frac{\tilde{\alpha}_\epsilon}{1-\epsilon})^2}\right)\right),$$

$$\frac{4096}{(\mu_1+\frac{\tilde{\beta}_\epsilon}{1-\epsilon}-\mu_i)^2}\log\left(\frac{2n}{\delta}\log_2\left(\frac{12288n}{\delta(\mu_1+\frac{\tilde{\beta}_\epsilon}{1-\epsilon}-\mu_i)^2}\right)\right)\Bigg\},$$

$$\frac{1024}{\gamma^2\mu_1^2}\log\left(\frac{2n}{\delta}\log_2\left(\frac{3072n}{\delta\gamma^2\mu_1^2}\right)\right)\Bigg\}.$$

The above bounds the number of rounds T. Therefore, the total number of samples is at most $3T$. □

6.C Proof of instance dependent lower bounds, Theorem 6.4

First we restate and prove the lower bound.

Theorem 6.11. *(additive and multiplicative lower bound) Fix $\delta, \epsilon > 0$. Consider n arms, such that the i^{th} is distributed according to $\mathcal{N}(\mu_i, 1)$. Any δ-PAC algorithm for the additive setting satisfies*

$$\mathbb{E}[\tau] \geqslant 2\sum_{i=1}^{n}\max\left\{\frac{1}{(\mu_1-\epsilon-\mu_i)^2}, \frac{1}{(\mu_1+\alpha_\epsilon-\mu_i)^2}\right\}\log\left(\frac{1}{2.4\delta}\right)$$

and if $\mu_1 > 0$ any δ-PAC algorithm for the multiplicative algorithm satisfies,

$$\mathbb{E}[\tau] \geqslant 2\sum_{i=1}^{n}\max\left\{\frac{1}{((1-\epsilon)\mu_1-\mu_i)^2}, \frac{1}{(\mu_1+\frac{\tilde{\alpha}_\epsilon}{1-\epsilon}-\mu_i)^2}\right\}\log\left(\frac{1}{2.4\delta}\right)$$

Proof of Theorem 6.4 in the additive case. Recall that ν denotes the given instance, and

without loss of generality we have assumed that $\mu_1 \geqslant \mu_2 \geqslant \cdots \geqslant \mu_n$. Then $G_\epsilon(\nu) = \{1, \cdots, k\}$. Consider the event E that an algorithm returns $\{1, \cdots, k\}$. For any δ-PAC algorithm, E occurs with probability at least $1 - \delta$. For each arm $i \in [n]$ we consider two alternative instances

$$\nu_i' = \{\mu_1, \cdots, \mu_i', \cdots, \mu_n\}$$

and

$$\nu_i'' = \{\mu_1, \cdots, \mu_i'', \cdots, \mu_n\}$$

such that only the mean of arm i differs compared to ν but $G_\epsilon(\nu) \neq G_\epsilon(\nu_i')$ and $G_\epsilon(\nu) \neq G_\epsilon(\nu_i'')$. Therefore, on these alternate instances, E occurs with probability at most δ.

For ν_i', if $i \leqslant k$, let $\mu_i' = \mu_1 - \epsilon - \eta$. Then $i \in G_\epsilon(\nu)$ but $i \notin G_\epsilon(\nu_i')$. If $k < n$ and $i \geqslant k+1$, let $\mu_i' = \mu_1 - \epsilon + \eta$. Then $i \notin G_\epsilon(\nu)$ but $i \in G_\epsilon(\nu_i')$.

More subtly, for ν_i'', for any $i \in [n] \backslash \{k\}$, let $\mu_i'' = \mu_k + \epsilon + \eta$. In particular, arm i is now the best arm. Under this definition, $\mu_i'' - \epsilon > \mu_k$. Therefore, $k \notin G_\epsilon(\nu_i'')$ but $k \in G_\epsilon(\nu)$.

The above holds for all $\eta > 0$. Let N_i denote the random variable of the number of samples of arm i and $\mathbb{E}_\nu$ denote expectation with respect to instance ν. Using the fact that we have assumed the distributions are Gaussian, considering ν_i', by Lemma 1 of Kaufmann et al. (2016), taking $\eta \to 0$ we have that for any δ-PAC algorithm,

$$\mathbb{E}_\nu[N_i] \geqslant \frac{2\log(1/2.4\delta)}{(\mu_i - (\mu_1 - \epsilon))^2}.$$

Furthermore, considering ν_i'', and again taking $\eta \to 0$, we have by the same lemma that for $i \neq k$

$$\mathbb{E}_\nu[N_i] \geqslant \frac{2\log(1/2.4\delta)}{(\mu_k + \epsilon - \mu_i)^2} = \frac{2\log(1/2.4\delta)}{(\mu_1 + \alpha_\epsilon - \mu_i)^2},$$

where the later equality holds since $\mu_k + \epsilon = \mu_1 + \alpha_\epsilon$ by definition of α_ϵ. For $i = k$, note that $\frac{1}{(\mu_k - (\mu_1 - \epsilon))} = \frac{1}{\alpha_\epsilon^2} \geqslant \frac{1}{\epsilon^2} = \frac{1}{(\mu_k - \mu_k - \epsilon)^2}$ since $\alpha_\epsilon = \min_{i \in G_\epsilon} \mu_i - (\mu_1 - \epsilon) =$

$\min_{i \in G_\epsilon} \epsilon - \Delta_i$. Putting these pieces together, we see that for any i,

$$\mathbb{E}_\nu[N_i] \geqslant \max\left(\frac{1}{(\mu_i - (\mu_1 - \epsilon))^2}, \frac{1}{(\mu_k + \epsilon - \mu_i)^2}\right) 2\log(1/2.4\delta).$$

Summing over all i establishes a lower bound in the additive case. □

Proof of Theorem 6.4 in the multiplicative case. Recall that ν denotes the given instance, and without loss of generality we have assumed that $\mu_1 \geqslant \mu_2 \geqslant \cdots \geqslant \mu_n$. Let $M_\epsilon(\nu) = \{1, \cdots, k\}$. Consider the event E that an algorithm returns $\{1, \cdots, k\}$. For any δ-PAC algorithm, E occurs with probability at least $1 - \delta$. For each arm $i \in [n]$ we consider two alternative instances

$$\nu_i' = \{\mu_1, \cdots, \mu_i', \cdots, \mu_n\}$$

and

$$\nu_i'' = \{\mu_1, \cdots, \mu_i'', \cdots, \mu_n\}$$

such that only the mean of arm i differs compared to ν but $M_\epsilon(\nu) \neq M_\epsilon(\nu_i')$ and $M_\epsilon(\nu) \neq M_\epsilon(\nu_i'')$. Therefore, E occurs with probability at most δ on these alternate instances.

For ν_i', if $i \leqslant k$, let $\mu_i' = (1 - \epsilon - \eta)\mu_1$. Then $i \in M_\epsilon(\nu)$ but $i \notin M_\epsilon(\nu_i')$. If $k < n$ and $i \geqslant k+1$, let $\mu_i' = (1 - \epsilon + \eta)\mu_1$. Then $i \notin M_\epsilon(\nu)$ but $i \in M_\epsilon(\nu_i')$.

More subtly, for ν_i'', for any $i \in [n]\backslash\{k\}$, let $\mu_i'' = \frac{\mu_k}{1-\epsilon-\eta}$. In particular, arm i is now the best arm. Under this definition, $\mu_i'' - \epsilon > \mu_k$. Therefore, $k \notin M_\epsilon(\nu_i'')$ but $k \in M_\epsilon(\nu)$.

The above holds for all $\eta > 0$. Let N_i denote the random variable of the number of samples of arm i and $\mathbb{E}_\nu$ denote expectation with respect to instance ν. Using the fact that we have assumed the distributions are Gaussian, considering ν_i', by Lemma 1 of Kaufmann et al. (2016), taking $\eta \to 0$, we have that for any δ-PAC algorithm,

$$\mathbb{E}_\nu[N_i] \geqslant \frac{2\log(1/2.4\delta)}{(\mu_i - (1-\epsilon)\mu_1)^2} = \frac{2\log(1/2.4\delta)}{(\epsilon\mu_1 - \Delta_i)^2}.$$

Additionally, by the same Lemma, considering ν_i'' and again taking $\eta \to 0$ we have that for $i \neq k$

$$\mathbb{E}_\nu[N_i] \geqslant \frac{2\log(1/2.4\delta)}{\left(\mu_i - \frac{\mu_k}{1-\epsilon}\right)^2} = \frac{2\log(1/2.4\delta)}{\left(\mu_1 + \frac{\tilde{\alpha}_\epsilon}{1-\epsilon} - \mu_i\right)^2},$$

where the later equality holds since $\frac{\mu_k}{1-\epsilon} = \mu_1 + \frac{\tilde{\alpha}_\epsilon}{1-\epsilon}$ by definition of $\tilde{\alpha}_\epsilon$. Next recall that $\tilde{\alpha}_\epsilon := \min_{i \in M_\epsilon} \mu_i - (1-\epsilon)\mu_1 = \mu_k - (1-\epsilon)\mu_1$, we have that $\mu_k = \tilde{\alpha}_\epsilon + (1-\epsilon)\mu_1$. Hence, $\frac{\mu_k}{1-\epsilon} = \mu_1 + \frac{\tilde{\alpha}_\epsilon}{1-\epsilon}$. Then, for $i = k$

$$\begin{aligned}
&\frac{1}{\left(\frac{\mu_k}{1-\epsilon} - \mu_k\right)^2} \leqslant \frac{1}{(\mu_k - (1-\epsilon)\mu_1)^2} = \frac{1}{\tilde{\alpha}_\epsilon^2} \\
&\iff \tilde{\alpha}_\epsilon \leqslant \frac{\mu_k}{1-\epsilon} - \mu_k = \frac{\tilde{\alpha}_\epsilon}{1-\epsilon} + \mu_1 - \mu_k = \frac{\tilde{\alpha}_\epsilon}{1-\epsilon} + \Delta_k \\
&\overset{(\Delta_k \geqslant 0)}{\impliedby} \tilde{\alpha}_\epsilon \leqslant \frac{\tilde{\alpha}_\epsilon}{1-\epsilon}
\end{aligned}$$

which is always true since $\epsilon > 0$. Therefore,

$$\frac{1}{(\mu_k - (1-\epsilon)\mu_1)^2} = \max\left(\frac{1}{(\mu_k - (1-\epsilon)\mu_1)^2}, \frac{1}{\left(\frac{\mu_k}{1-\epsilon} - \mu_k\right)^2}\right).$$

Hence, for all arms i,

$$\mathbb{E}_\nu[N_i] \geqslant 2\max\left(\frac{1}{(\mu_i - (1-\epsilon)\mu_1)^2}, \frac{1}{\left(\mu_1 + \frac{\tilde{\alpha}_\epsilon}{1-\epsilon} - \mu_i\right)^2}\right)\log(1/2.4\delta).$$

Summing over all i gives a lower bound for this problem in the multiplicative case. □

6.D Theorem 6.7: Lower bounds in the moderate confidence regime

In this section, we prove a tighter lower bound that includes *moderate confidence* terms independent of the value of δ similar to those that appear in the upper bound

on the sample complexity of FAREAST, Theorem 6.8.

Outline. To give a tight lower bound in the isolated setting, we break our argument into pieces performing a series of reductions that link the all-ϵ problem to a hypothesis test, and then the hypothesis test to the problem of identifying the best-arm.

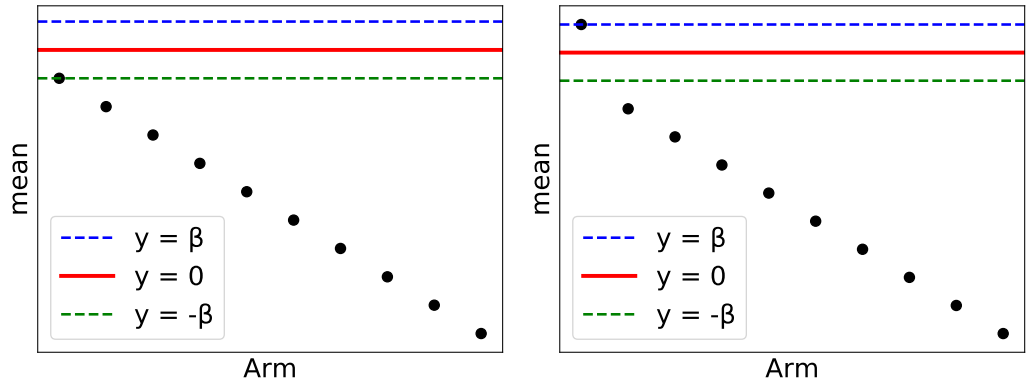
Step 1. Finding an isolated arm. We first consider the following problem. Imagine that you are given an *isolated* instance, depicted in Figure 6.D.1b where there are n distributions, with one of them at mean β and the rest with mean $-\beta$. Theorem 6.15, captures the sample complexity of any algorithm that can return i^* with probability greater than $1-\delta$.

Step 2. Deciding if an instance is isolated. We then consider a composite hypothesis test on n distributions where the null hypothesis, H_0, is that the mean of each distribution is less that $-\beta$ and the alternate hypothesis, H_1, is that there exists *single* distribution i^* with mean β and the remainder have mean less than $-\beta$ (i.e. the instance is isolated). In Figure 6.D.1, we show a picture of an instance where the null is true and where the alternate is true. In Theorem 6.19 we lower bound the complexity of performing this test. To link this to Step 1, we show that if you can solve this composite hypothesis test then you can find i^*, hence the lower bound of step 1 is a lower bound for the hypothesis test.

Step 3: Reducing ALL-ϵ to Step 2 Finally in step 3 we link this to the all-ϵ problem. Using the above, we lower bound the complexity of ALL-ϵ in Theorem 6.7 when $|G_{2\beta_\epsilon}| = 1$. The key insight of our proof is that any algorithm that can solve the ALL-ϵ problem can be used to solve the hypothesis test in Step 2.

6.D.1 Step 1: Finding an Isolated Arm

Fix $n \in \mathbb{N}$, $0 < \beta$, and $\delta > 0$. We refer to a β-*isolated instance* $\nu = \{\rho_1, \cdots, \rho_n\}$, as a collection of n, Gaussian distributions with variance one satisfying two properties. Firstly, there exists a single arm $i^* \in [n]$ with $\rho_{i^*} = \mathcal{N}(\beta, 1)$. We refer to this as the *isolated arm*. Secondly, for $i \neq i^*$, $\rho_i = \mathcal{N}(\mu_i, 1)\ \forall\ i \in [n]\backslash\{i^*\}$ have means $\mu_i \leqslant -\beta$. We introduce the additional notation $\Delta_{i,j} = \mu_i - \mu_j$.



(a) A non-isolated instance (H_0 is true) (b) An isolated instance (H_0 is true)

Figure 6.D.1: Example of an isolated and non-isolated instance

Lemma 6.12. *Fix* n, $0 < \beta$ *and consider a set* ν *of* n *Gaussian random variables such that for a uniformly random chosen* $i^* \in [n]$, $\rho_{i^*} = \mathcal{N}(\beta, 1)$ *and* $\rho_i = \mathcal{N}(\mu_i, 1)$ *for* $\mu_i \leqslant -\beta$ *for all* $i \neq i^*$. *Any algorithm that correctly returns* i^* *with probability at least* $1 - \delta$, *pulls arm* i^* *at least*

$$\frac{1}{2\beta^2} \log(1/2.4\delta)$$

times in expectation.

Proof. Consider the oracle setting where the value of i^* is known and the algorithm only seeks to confirm that $\mu_{i^*} > -\beta$. Lemma 1 of Kaufmann et al. (2016) implies that any δ-PAC algorithm requires at least $\frac{1}{2\beta^2} \log(1/2.4\delta)$ samples in expectation. □

The above bound controls the number of samples that any algorithm must gather from i^*, and is independent of n. The proof considered an oracle setting where the value of i^* is known, and one only wishes to confirm that $\mu_{i^*} > -\beta$ with probability at least $1 - \delta$. To lower bound the number of samples drawn from $[n]\backslash\{i^*\}$, we need significantly more powerful tools. In particular, to rule out trivial algorithms that always output a fixed index, we consider a permutation model, as in Mannor and Tsitsiklis (2004); Simchowitz et al. (2017); Katz-Samuels and Jamieson (2020); Chen et al. (2017). Informally, we consider an additional expectation in the lower bound over a random permutation π of the arms where π

is sampled uniformly from the set of all permutations. In particular, we with use a *Simulator* argument, as in Simchowitz et al. (2017); Katz-Samuels and Jamieson (2020). In what follows, we will let $\pi : [n] \to [n]$ denote a permutation selected uniformly at random from the set of $n!$ permutations. For instance ν, let $\pi(\nu)$ denote the permuted instance such that the i^{th} distribution is mapped to $\pi(i)$, by a slight overloading of the definition of $\pi(\cdot)$. In what follows, we proceed similarly to the proof of Theorem 1 in Katz-Samuels and Jamieson (2020).

Theorem 6.13. *Fix n, $0 < \beta$, and $\delta < 1/16$ and consider a set ν of n Gaussian random variables with variance 1 such that for $i^* \in [n]$, $\rho_{i^*} = \mathcal{N}(\beta, 1)$ and $\rho_i = \mathcal{N}(\mu_i, 1)$ for $\mu_i \leqslant -\beta$ for all $i \neq i^*$. Let π be a uniformly chosen permutation of $[n]$ and $\pi(\nu)$ be the permutation applied to instance ν. Let T be the random variable denoting the total number of samples at termination by an algorithm. Any δ-PAC algorithm to detect $\pi(i^*)$ on $\pi(\nu)$ requires*

$$\mathbb{E}_\pi \mathbb{E}_{\pi(\nu)}[T] \geqslant \frac{1}{16} \sum_{k \neq i^*} \frac{1}{\Delta_{i^*,k}^2}$$

samples in expectation from arms in $[n] \backslash \{i^*\}$.

Proof. Fix a permutation π. Let $\pi(\nu)$ be the permutation applied to ν and $\pi(i)$ be the index of i under the permuted instance, $\pi(\nu)$. Let $\mathcal{A}$ be any algorithm that detects and returns $\pi(i^*)$ on $\pi(\nu)$ with probability at least $1 - \delta$. We will take $\mathbb{P}_\mathcal{A}$ and $\mathbb{E}_\mathcal{A}$ to denote probability and expectation with respect any internal randomness in $\mathcal{A}$. Throughout, we will take $\rho_i = \mathcal{N}(\mu_i, 1)$ to denote the i^{th} distribution of ν. $\mu_{i^*} > 0$ and $\mu_i < 0$ for all $i \neq i^*$. Additionally, let $\Delta_{ij} = \mu_i - \mu_j$

Fix $k \neq i^*$. To bound the necessary number of samples for arm k, we turn to the Simulator Simchowitz et al. (2017). We begin by defining an alternate instance $\nu'_k = \{\rho'_1, \cdots, \rho'_n\}$ as

$$\rho'_j = \begin{cases} \rho_j, & j \neq i^* \\ \rho_k, & j = i^* \\ \rho_{i^*}, & j = k \end{cases}$$

Note that ν'_k is identical to ν except that the distributions of i^* and k are swapped.

Let E be the event that $\mathcal{A}$ returns $\pi(i^*)$. We may bound the total variation distance between the joint distribution on $\mathcal{A} \times \pi(\nu)$ and $\mathcal{A} \times \pi(\nu'_k)$ as

$$\begin{aligned} TV(\mathbb{P}_{\mathcal{A}\times\pi(\nu)}, \mathbb{P}_{\mathcal{A}\times\pi(\nu'_k)}) &= \sup_A \left|\mathbb{P}_{\mathcal{A}\times\pi(\nu)}(A) - \mathbb{P}_{\mathcal{A}\times\pi(\nu'_k)}(A)\right| \\ &\geqslant \left|\mathbb{P}_{\mathcal{A}\times\pi(\nu)}(E) - \mathbb{P}_{\mathcal{A}\times\pi(\nu'_k)}(E)\right| \\ &\geqslant 1 - 2\delta. \end{aligned}$$

Let Ω_t denote the multiset of the transcript of samples up to time t.

$$\Omega_t = \{i_s \in [n] \text{ for } 1 \leqslant s \leqslant t\}$$

and define the events

$$W_j(\Omega_t) := \left\{ \sum_{i_t \in \Omega_t} \mathbb{1}(i_t = j) \leqslant \tau \right\}$$

for a τ to be defined later. With the definitions of $W_j(\Omega_t)$, we define a simulator $\mathrm{Sim}(\nu, \Omega_t)$ with respect to ν. Let $\mathrm{Sim}(\nu, \Omega_t)_i$ denote the distribution of arm i on $\mathrm{Sim}(\nu, \Omega_t)$.

$$\mathrm{Sim}(\nu, \Omega_t)_j = \begin{cases} \rho_j, & \text{if } j \notin \{i^*, k\} \\ \rho_j, & \text{if } j \in \{i^*, k\} \text{ and } W_{i^*}(\Omega_t) \cap W_k(\Omega_t) \\ \rho_{i^*}, & \text{if } j \in \{i^*, k\} \text{ and } (W_{i^*}(\Omega_t) \cap W_k(\Omega_t))^c \end{cases}$$

Furthermore, we define $\mathrm{Sim}(\nu'_k, \Omega_t)$ with respect to ν'_k as

$$\mathrm{Sim}(\nu'_k, \Omega_t)_j = \begin{cases} \rho'_j, & \text{if } j \notin \{i^*, k\} \\ \rho'_j, & \text{if } j \in \{i^*, k\} \text{ and } W_{i^*}(\Omega_t) \cap W_k(\Omega_t) \\ \rho_{i^*}, & \text{if } j \in \{i^*, k\} \text{ and } (W_{i^*}(\Omega_t) \cap W_k(\Omega_t))^c \end{cases}$$

For ease of notation, let $\mathrm{Sim}(\pi(\nu), \Omega_t)$ be the same simulator defined on $\pi(\nu)$ and

with respect to events $W_{\pi(i^*)}(\Omega_t)$ and $W_{\pi(k)}(\Omega_t)$. Note that in the simulator of ν'_k, if $(W_{i^*}(\Omega_t) \cap W_k(\Omega_t))^c$ is true, then i^* and k draw samples according to instance ν not ν'_k.

Definition 6.14. *(Truthfulness of an event W, Katz-Samuels and Jamieson (2020)) For an algorithm $\mathcal{A}$, we say that an event W is truthful on a simulator $Sim(\eta)$ with respect to an instance η if for all events E in the filtration $\mathcal{F}_T$ generated by playing algorithm $\mathcal{A}$ on instance η*

$$\mathbb{P}_\eta(E \cap W) = \mathbb{P}_{Sim(\eta)}(E \cap W)$$

By our definition of both simulators, if $(W_{i^*}(\Omega_t) \cap W_k(\Omega_t))^c$ is true, then $\text{Sim}(\nu, \Omega_t)_i = \text{Sim}(\nu'_k, \Omega_t)_i \ \forall i \in [n]$. Contrarily, if $W_{i^*}(\Omega_t) \cap W_k(\Omega_t)$ is true, then $\text{Sim}(\nu, \Omega_t) = \nu$ and $\text{Sim}(\nu'_k, \Omega_t) = \nu'_k$. Similarly, on $W_{\pi(i^*)}(\Omega_t) \cap W_{\pi(k)}(\Omega_t)$, $\text{Sim}(\pi(\nu), \Omega_t) = \pi(\nu)$ and $\text{Sim}(\pi(\nu'_k), \Omega_t) = \pi(\nu'_k)$. Therefore, by the proof of Theorem 1 of Katz-Samuels and Jamieson (2020), $W_{\pi(k)}(\Omega_t)$ is truthful on $\text{Sim}(\pi(\nu), \Omega_t)$ and $W_{\pi(i^*)}(\Omega_t)$ is truthful on $\text{Sim}(\pi(\nu'_k), \Omega_t)$.

Let i_t be the arm queried at time $t \in \mathbb{N}$ by $\mathcal{A}$. Following the proof of Theorem 1 of Katz-Samuels and Jamieson (2020), we may bound the KL-Divergence between $\text{Sim}(\pi(\nu), \Omega_t)$ and $\text{Sim}(\pi(\nu'_k), \Omega_t)$ as

$$\begin{aligned}
\max_{i_1,\cdots,i_T} \sum_{t=1}^{T} & \quad \text{KL}\left(\text{Sim}(\pi(\nu), \{i_s\}_{s=1}^t), \text{Sim}(\pi(\nu'_k), \{i_s\}_{s=1}^t\right) \\
& \leqslant \tau \text{KL}(\pi(\nu)_{\pi(i^*)}, \pi(\nu'_k)_{\pi(i^*)}) + \tau \text{KL}(\pi(\nu)_{\pi(k)}, \pi(\nu'_k)_{\pi(k)}) \\
& = \tau \frac{\Delta^2_{i^*,k}}{2} + \tau \frac{\Delta^2_{i^*,k}}{2} \\
& = \tau \Delta^2_{i^*,k}.
\end{aligned}$$

For any instance η, an algorithm $\mathcal{A}$ is defined to be symmetric if

$$\mathbb{P}_{\mathcal{A},\eta}((i_1, \cdots, i_T) = (I_1, \cdots, I_T)) = \mathbb{P}_{\mathcal{A},\pi(\eta)}((\pi(i_1), \cdots, \pi(i_T)) = (\pi(I_1), \cdots, \pi(I_T))).$$

Semantically, this implies that the proportion of times $\mathcal{A}$ pulls any arm i on the non-permuted instance η is the same as the proportion of times it pulls $\pi(i)$ on the

permuted instance, $\pi(\eta)$.

In particular, the expected complexity of a symmetric algorithm is independent of the permutation π. By Lemma 1 of Simchowitz et al. (2017), if any algorithm $\mathcal{B}$ (not necessarily symmetric) achieves an expected stopping time τ where the expectation is taken over all the randomness in the permutation and in the instance, then there is a symmetric algorithm that achieves the same expected stopping time. Hence, we may assume that $\mathcal{A}$ is symmetric and capture the same set of possible stopping times. If $\mathcal{A}$ is not symmetric, we may form an algorithm $\mathcal{A}'$ by permuting the input, passing it to $\mathcal{A}$, getting the output of $\mathcal{A}$ on the permuted input, and then undoing the permutation before return an answer.

Since $W_{\pi(k)}(\Omega_t)$ and $W_{\pi(i^*)}(\Omega_t)$ are truthful on $\mathrm{Sim}(\pi(\nu), \Omega_t)$ and $\mathrm{Sim}(\pi(\nu'_k), \Omega_t)$ respectively, by Lemma 2 of Simchowitz et al. (2017), we have that

$$\begin{aligned}&\mathbb{P}_{\mathcal{A},\pi(\nu)}(W_{\pi(k)}(\Omega_t)) + \mathbb{P}_{\mathcal{A},\pi(\nu'_k)}(W_{\pi(i^*)}(\Omega_t)) \\ &\qquad \geqslant \mathrm{TV}(\mathbb{P}_{\mathcal{A},\pi(\nu)}, \mathbb{P}_{\mathcal{A},\pi(\nu'_k)}) - Q\left(\mathrm{KL}(\mathrm{Sim}(\pi(\nu), \Omega_t), \mathrm{Sim}(\pi(\nu'_k), \Omega_t))\right)\end{aligned}$$

for $Q(x) = \min\{1 - 1/2e^{-x}, \sqrt{x/2}\}$. Since $\mathcal{A}$ is symmetric, for any permutation π, we have that

$$\begin{aligned}&\mathbb{P}_{\mathcal{A},\pi(\nu)}(W_{\pi(k)}(\Omega_t)) + \mathbb{P}_{\mathcal{A},\pi(\nu'_k)}(W_{\pi(i^*)}(\Omega_t)) \\ &= \mathbb{P}_{\mathcal{A},\nu}(W_k(\Omega_t)) + \mathbb{P}_{\mathcal{A},\nu'_k}(W_{i^*}(\Omega_t)) = 2\mathbb{P}_{\mathcal{A},\nu}(W_k(\Omega_t)).\end{aligned}$$

The first equality holds since event W_i depend only on the number of times that arm i is pulled. Since $\mathcal{A}$ is symmetric, the probability that $\mathcal{A}$ pulls arm i at most τ times on instance ν is equal to the probability that $\mathcal{A}$ pulls $\pi(i)$ at most τ times on instance $\pi(\nu)$. The second equality is true using symmetry as well since instances ν and ν'_k are themselves equal up to a permutation.

Combining the above with the previous bounds on the total variation and KL divergence, we have that

$$\mathbb{P}_{\mathcal{A}\times\nu}(N_k > \tau) = \mathbb{P}_{\mathcal{A}\times\nu}(W_k(\Omega_t)) \geqslant \frac{1}{2}\left(1 - 2\delta - \sqrt{\frac{\tau\Delta_{i^*,k}^2}{2}}\right)$$

Plugging in $\tau = 1/(2\Delta_{i^*,k}^2)$, we see that $\mathbb{P}_{\mathcal{A}\times\nu}(N_k > 1/(2\Delta_{i^*,k}^2)) \geqslant 1/2(1/2 - 2\delta)$. Since k was arbitrary, we may repeat this argument for each k in $[n]\backslash\{i^*\}$. Combining this with Markov's inequality, we see that

$$\begin{aligned}\mathbb{E}_{\mathcal{A}\times\nu}\left[\sum_{k\neq i^*} N_k\right] &\geqslant \frac{1}{4}(1/2 - 2\delta)\sum_{k\neq i^*}\frac{1}{\Delta_{i^*,k}^2} \\ &> \frac{1}{16}\sum_{k\neq i^*}\frac{1}{\Delta_{i^*,k}^2}\end{aligned}$$

where the final inequality follows from $\delta < 1/16$. The above holds for any δ-PAC algorithm $\mathcal{A}$. □

We now state our strong lower bound on the expected number of samples for any algorithm that can find an isolated arm.

Theorem 6.15. *Fix n, $0 < \beta$, and $\delta < 1/16$ and consider a set ν of n Gaussian random variables with variance 1 such that for a uniformly random chosen $i^* \in [n]$, $\rho_{i^*} = \mathbb{N}(\beta, 1)$ and $\rho_i = \mathcal{N}(\mu_i, 1)$ for $\mu_i \leqslant -\beta$ for all $i \neq i^*$. Let π be a uniformly chosen permutation of $[n]$ and $\pi(\nu)$ be the permutation applied to instance ν. Any δ-PAC algorithm to detect $\pi(i^*)$ on $\pi(\nu)$ requires*

$$\frac{1}{16}\sum_{k\neq i^*}\frac{1}{\Delta_{i^*,k}^2} + \frac{1}{2\beta^2}\log(1/2.4\delta)$$

samples in expectation, where the expectation is taken both over the randomness in the permutation, the randomness in $\pi(\nu)$, and any internal randomness to the algorithm.

Proof. By Lemma 6.12, arm i^* must be sampled $\frac{1}{2\beta^2}\log(1/2.4\delta)$ times. By Theorem 6.13, arms in $[n]\backslash\{i^*\}$ must collectively be sampled $\frac{1}{16}\sum_{k\neq i^*}\frac{1}{\Delta_{i^*,k}^2}$ times. Joining

these two results gives the stated result. □

6.D.2 Step 2. Deciding if an instance is isolated

Next, we consider a composite hypothesis test that is related to the question of finding an isolated arm. As we will show, this test has the interesting property that the alternate hypothesis may be declared in significantly fewer samples than the null.

Definition 6.16 (β-Isolated Hypothesis Test)**.** *Fix* $0 < \epsilon$ *and* $0 < \beta$*. Consider an instance* $\nu = \{\rho_1, \cdots, \rho_n\}$ *where* $\rho_i = \mathcal{N}(\mu_i, 1)$*. By sampling individual distributions* ρ_i*, one wishes to perform the following composite hypothesis test:*

Null Hypothesis H_0***:*** $\mu_i < -\beta$ *for all* $i \in [n]$*.*

Alternate Composite Hypothesis H_1***:*** $\exists i^* : \mu_{i^*} = \beta > 0$ *and* $\mu_i \leqslant -\beta$ *for all* $i \neq i^*$*.*

For any instance ν, we say "H_1 is true on ν" if $\exists i^* : \mu_{i^*} = \beta > 0$ and otherwise we say "H_0 is true on ν." Next, we bound the sample complexity of any algorithm to perform the β-isolated hypothesis test with probability at least $1 - \delta$ in the case that H_0 is true.

Figure 6.2 shows an two example instance. One where H_0 is true and one where H_1 is true.

Lemma 6.17. *Fix* n*,* β*, and* δ *and consider a set* ν *of* n *standard normal random variables where* H_0 *is true. Any algorithm to correctly declare* H_0 *in the* β*-isolated hypothesis test problem with probability at least* $1 - \delta$ *requires*

$$\sum_{i=1}^{n} \frac{2}{(\beta - \mu_i)^2} \log\left(\frac{1}{2.4\delta}\right)$$

samples in expectation.

Proof. Notice that for each $i \in [n]$, we may construct an alternate instance ν_i by changing the distribution of ρ_i to be $\mathcal{N}(\beta, 1)$ and leaving others unchanged. On

ν_i, H_1 is instead true. To distinguish between ν and ν_i, necessary to declare H_0 versus H_1, by Lemma 1 of Kaufmann et al. (2016), any δ-PAC algorithm requires $\mathbb{E}_\nu[N_i] \geqslant \frac{2}{(\beta-\mu_i)^2}\log(1/2.4\delta)$ where $\mathbb{E}_\nu$ denotes expectation with respect to the instance ν and N_i denotes the number of samples of arm i. Repeating this argument for each $i \in [n]$ gives the desired result. □

To lower bound the expected sample complexity of any algorithm to perform the β-isolated hypothesis test in the setting where H_1 is true, we consider a reduction to the problem studied in Step 1, Section 6.D.1. For the reduction to an algorithm that can find an isolated arm, we show that if there is an algorithm to declare H_1 in fewer than $O\left(\sum_{i=1}^n \frac{1}{\Delta_{i^*,k}^2}\right)$ samples, then one can design an algorithm akin to binary search that returns i^* in fewer than $O\left(\sum_{i=1}^n \frac{1}{\Delta_{i^*,k}^2}\right)$ samples, contradicting Lemma 6.13.

Lemma 6.18. *Fix n, β, and $\delta < 1/16$. Let π be a random permutation. Consider an instance ν where H_1 is true. In this setting, any algorithm to correctly declare H_1 in the β-Isolated Hypothesis Testing problem on $\pi(\nu)$ with probability at least $1-\delta$ requires $\frac{1}{32}\sum_{j\neq i^*}\Delta_{i^*,k}^{-2}$ samples in expectation.*

Proof. Fix $\delta > 0$ and let i^* denote the single distribution such that $\rho_{i^*} = \mathcal{N}(\beta, 1)$ where $\beta > 0$. In particular, only i^* has a positive mean. Assume for contradiction that there is an algorithm $\mathcal{A}(\pi(\nu), \delta, \beta)$ that correctly declares H_1 on $\pi(\nu)$ in at most $\frac{1}{32}\sum_{k\neq i^*}\Delta_{i^*,k}^{-2}$ samples in expectation with probability at least $1-\delta$ on any instance $\pi(\nu)$ of n distributions if H_1 is true. Otherwise, if H_0 is true, assume that $\mathcal{A}$ correctly declares H_0 in an arbitrary number of samples in expectation, $N_{H_0}(\nu)$ lower bounded by Lemma 6.17. As in the proof of Theorem 6.13, if any algorithm $\mathcal{B}$ (not necessarily symmetric) achieves an expected stopping time τ where the expectation is taken over all the randomness in the permutation and in the instance, by Lemma 1 of Simchowitz et al. (2017), there is a symmetric algorithm that achieves the same expected stopping time. Hence, we may assume that $\mathcal{A}$ is symmetric and capture the same set of possible stopping times. For the remainder of this proof, we assume $\mathcal{A}$ is symmetric. Therefore, its expected complexity is independent of

the permutation π. Without loss of generality, assume that $n = 2^k$ for some $k \in \mathbb{N}$. Otherwise, we may hallucinate $(2^{\lceil \log_2(n) \rceil} - n)$ normal distributions, $\mathcal{N}(-\beta, 1)$, and form an instance ν' comprised of these additional distribution and those in ν. If so, anytime $\mathcal{A}$ requests a sample from a distribution in $\nu' \backslash \nu$, draw a sample from $\mathcal{N}(-\beta, 1)$ and pass it to $\mathcal{A}$, only tracking the number of samples drawn from ν.

Step a). In what follows, we use $\mathcal{A}$ to develop a method for isolated-arm identification. To do so, we show that one may use $\mathcal{A}$ to perform binary search for the distribution i^* such that $\rho_{i^*} = \mathcal{N}(\beta, 1)$ and this leads to a contradiction of Theorem 6.13. For ease of exposition, for a set $\mathcal{S} \subset [n]$, let $\nu(\mathcal{S}) := \{i \in \mathcal{S} : \rho_i\}$, the subset of instance ν of distributions whose indices are in $\mathcal{S}$.

If H_1 is true on $\nu(S)$, by assumption, with probability at least $1 - \delta$, $\mathcal{A}$ correctly declares H_1 on $\nu(\mathcal{S})$ in at most $\frac{1}{32} \sum_{i \in \mathcal{S} \setminus \{i^*\}} \Delta_{i^*,k}^{-2}$ samples in expectation. Similarly, if H_0 is true on $\nu(\mathcal{S})$, the sample complexity is $N_{H_0}(\nu(\mathcal{S}))$ in expectation.

Algorithm 7 Binary search for Isolated Arm Identification

Require: $\delta > 0$, $\beta > 0$, instance ν such that H_1 is true, algorithm $\mathcal{A}$

Let Low $= 1$ and High $= n$
for $i = 1, \cdots, \log_2(n)$ **do**
    1) Choose sets $\mathcal{S}_1, \mathcal{S}_2$ uniformly at random such that $\mathcal{S}_1 \cup \mathcal{S}_2 = \mathcal{S}$, $\mathcal{S}_1 \cap \mathcal{S}_2 = \emptyset$, and $\mathbb{P}(i \in \mathcal{S}_1) = \mathbb{P}(i \in \mathcal{S}_2)$ for all $i \in \mathcal{S}$
    2) In parallel, run $\mathcal{A}_1 = \mathcal{A}(\nu(\mathcal{S}_1), \beta, \delta/2\log_2(n))$ and $\mathcal{A}_2 = \mathcal{A}(\nu(\mathcal{S}_2), \beta, \delta/2\log_2(n))$
    3) If either terminates, terminate the other
    if $\mathcal{A}_1$ declares H_1 or $\mathcal{A}_2$ declares H_0 **then**
        $\mathcal{S} = \mathcal{S}_1$
    else
        $\mathcal{S} = \mathcal{S}_2$
    end if
end forreturn $i^* \in \mathcal{S}$ (note: $|\mathcal{S}| = 1$ at this point)

In step 1, we choose 2 random subsets of $\mathcal{S}$, $\mathcal{S}_1$ and $\mathcal{S}_2$ that partition $\mathcal{S}$ such that each arm is assigned with equal probability to either $\mathcal{S}_1$ or $\mathcal{S}_2$ independently.

In step 2) if the loop, we separately run $\mathcal{A}$ in parallel on $\nu(\mathcal{S}_1)$ and $\nu(\mathcal{S}_2)$, each

with failure probability $\delta/2\log(n)$. We alternate between passing a sample to $\mathcal{A}_1$ and to $\mathcal{A}_2$.

In Step 3), we terminate $\mathcal{A}_1$ if $\mathcal{A}_2$ terminates and vice versa. If, for instance, $\mathcal{A}_1$ terminates and declares H_0, we may infer H_1 on $\mathcal{S}_2$. Alternately, if $\mathcal{A}_1$ declares H_1 on S_1, we may infer H_0 on S_2 as there is a single positive mean, μ_{i^*}. This process continues until $|\mathcal{S}_1| = |\mathcal{S}_2| = 1$, when there is a single distribution remaining in each. At this point, if $\mathcal{A}_1$ declares H_1, then the single arm $i \in \mathcal{S}_1$ is the positive mean i^*. Otherwise, the single arm $j \in \mathcal{S}_2$ is.

First, we show that this algorithm is correct with probability at least $1 - \delta$. The algorithm errs if and only if in any round i, either $\mathcal{A}_1$ or $\mathcal{A}_2$ errs, each with occurs with probability at most $\delta/2\log_2(n)$. Union bounding over the $\log_2(n)$ rounds, we see that the algorithm errs with probability at most δ. For the remainder of the proof, we will assume that in no round does either $\mathcal{A}_1$ or $\mathcal{A}_2$ incorrectly declare H_0 or H_1 if the reverse is true for the given instances $\nu(\mathcal{S}_1)$ and $\nu(S_2)$.

Now we introduce some notation for the remainder of this proof. As the set $\mathcal{S}$, $\mathcal{S}_1$, and $\mathcal{S}_2$ change in each round, let $\mathcal{S}(r)$, $\mathcal{S}_1(r)$, and $\mathcal{S}_2(r)$ denote their values in round r for $r = 1, \cdots, \log_2(n)$. Define $\mathcal{A}_1(r)$ and $\mathcal{A}_2(r)$ similarly. We stop $\mathcal{A}_1(r)$ if $\mathcal{A}_2(r)$ terminates and vice versa.

Let T_r denote the random variable of the total number of samples of drawn in round r. Let $T_{r,1}$ be the number of samples drawn by $\mathcal{A}_1(r)$, and $T_{r,2}$ be the number of samples drawn by $\mathcal{A}_2(r)$.

Next, define $\mathcal{S}^*(r)$ be the set in $\{\mathcal{S}_1(r), \mathcal{S}_2(r)\}$ that contains i^*, i.e. let $\mathcal{S}^*(r)$ denote $\mathcal{S}_1(r)$ if $i^* \in \mathcal{S}_1(r)$ and $S_2(r)$ otherwise for all r. Similarly, let $\mathcal{A}^*(r)$ denote $\mathcal{A}_1(r)$ if $i^* \in \mathcal{S}_1(r)$ and $\mathcal{A}_2(r)$ otherwise. Define $T_{r,\mathcal{A}^*}$ to be the random number of samples given to $\mathcal{A}^*(r)$. Hence, $T_{r,\mathcal{A}^*} = T_{r,1}$ or $T_{r,\mathcal{A}^*} = T_{r,2}$.

By Step 2, $\mathcal{A}_1(r)$ and $\mathcal{A}_2(r)$ are run in parallel. Hence, $T_{r,1} = T_{r,2}$ deterministically. Furthermore, $T_r = T_{r,1} + T_{r,2}$ deterministically. Therefore,

$$T_{r,\mathcal{A}^*} = \frac{T_{r,1} + T_{r,2}}{2} = \frac{T_r}{2}.$$

Therefore, the expected number of samples in round r, taken over the random-

ness in the set $\mathcal{S}^*(r)$, the randomness in the instance $\nu(\mathcal{S}^*(r))$, and any randomness in $\mathcal{A}^*(r)$ is

$$\begin{aligned}
\mathbb{E}_{\mathcal{S}^*(1),\cdots,\mathcal{S}^*(r),\nu(\mathcal{S}^*(r))}[T_r] &= 2\mathbb{E}_{\mathcal{S}^*(1),\cdots,\mathcal{S}^*(r),\nu(\mathcal{S}^*(r))}\left[T_{r,\mathcal{A}^*}\right] \\
&= 2\mathbb{E}_{\mathcal{S}^*(1),\cdots,\mathcal{S}^*(r)}\left[\mathbb{E}_{\nu(\mathcal{S}^*(r))}\left[T_{r,\mathcal{A}^*}|S^*(r)\right]\right] \\
&= 2\mathbb{E}_{\mathcal{S}^*(1),\cdots,\mathcal{S}^*(r)}\left[\min\left(\frac{1}{32}\sum_{j\in\mathcal{S}^*(r)\setminus\{i^*\}}\frac{1}{\Delta_{i^*,j}^2},\ N_{H_0}(\nu(\mathcal{S}^*(r)^c))\right)\right] \\
&\leqslant 2\mathbb{E}_{\mathcal{S}^*(1),\cdots,\mathcal{S}^*(r)}\left[\frac{1}{32}\sum_{j\in\mathcal{S}^*(r)\setminus\{i^*\}}\frac{1}{\Delta_{i^*,j}^2}\right] \\
&= 2\mathbb{E}_{\mathcal{S}^*(1),\cdots,\mathcal{S}^*(r-1)}\left[\mathbb{E}_{\mathcal{S}^*(r)}\left[\frac{1}{32}\sum_{j\neq i^*}\mathbb{1}[j\in\mathcal{S}^*(r)]\frac{1}{\Delta_{i^*,j}^2}\middle|\mathcal{S}^*(r-1)\right]\right] \\
&= 2\mathbb{E}_{\mathcal{S}^*(1),\cdots,\mathcal{S}^*(r-1)}\left[\frac{1}{32}\cdot\left(\frac{1}{2}\right)\sum_{j\neq i^*}\mathbb{1}[j\in\mathcal{S}^*(r-1)]\frac{1}{\Delta_{i^*,j}^2}\right] \\
&\ \ \vdots \\
&= 2\mathbb{E}_{\mathcal{S}^*(1)}\left[\frac{1}{32}\cdot\left(\frac{1}{2}\right)^{r-1}\sum_{j\neq i^*}\mathbb{1}[j\in\mathcal{S}^*(1)]\frac{1}{\Delta_{i^*,j}^2}\right] \\
&= \frac{1}{16}\cdot\left(\frac{1}{2}\right)^{r}\sum_{j\neq i^*}\frac{1}{\Delta_{i^*,j}^2}.
\end{aligned}$$

Therefore, we may bound the expected total number of samples for the above binary search algorithm to return i^* as

$$\mathbb{E}\left[\sum_{r=1}^{\log_2(n)}T_r\right] = \sum_{r=1}^{\log_2(n)}\mathbb{E}\left[T_r\right] \leqslant \frac{1}{16}\sum_{j\neq i^*}\frac{1}{\Delta_{i^*,j}^2}\sum_{r=1}^{\log_2(n)}\left(\frac{1}{2}\right)^r \leqslant \frac{1}{16}\sum_{j\neq i^*}\frac{1}{\Delta_{i^*,j}^2}.$$

However, this contradicts Theorem 6.13 for $\delta < 1/16$. Hence no such algorithm $\mathcal{A}$ exists and any algorithm to declare H_1 on instance ν requires at least $\frac{1}{32}\sum_{j\neq i^*}\frac{1}{\Delta_{i^*,j}^2}$ samples in expectation. □

Theorem 6.19. *Fix* n, β, *and* $\delta < 1/16$ *and consider an instance* ν. *If* H_0 *is true on* ν, *any algorithm requires at least*

$$\sum_{j=1}^{n} \frac{2}{(\beta - \mu_j)^2} \log\left(\frac{1}{2.4\delta}\right)$$

samples in expectation to perform the β*-isolated Hypothesis Test. If* H_1 *is true on* ν, *any algorithm requires at least*

$$\frac{1}{4\beta^2} \log\left(\frac{1}{2.4\delta}\right) + \frac{1}{64} \sum_{j \neq i^*} \frac{1}{\Delta_{i^*,j}^2}$$

samples in expectation to perform the β*-isolated Hypothesis Test.*

Proof. If H_0 is true for ν, the result follows immediately from Lemma 6.17. Otherwise, assume H_1 is true for ν and let i^* be the single distribution such that $\rho_{i^*} = \mathcal{N}(\beta, 1)$. Similar to the proof of Lemma 6.12, one may consider an alternate instance ν' where $\rho_{i^*} = \mathcal{N}(-\beta, 1)$ and all other distributions are unchanged. Therefore, on ν', H_0 is true and any algorithm that is correct with probability at least $1 - \delta$ must be able to distinguish between these two instances. By Lemma 1 of Kaufmann et al. (2016), any algorithm that is correct with probability at least $1 - \delta$ must therefore sample i^* $\frac{1}{2\beta^2} \log\left(\frac{1}{2.4\delta}\right)$ times in expectation. Combining this with the result of Lemma 6.18, any algorithm that is correct with probability at least $1 - \delta$ must collect at least

$$\max\left\{\frac{1}{32} \sum_{j \neq i^*} \frac{1}{\Delta_{i^*,j}^2}, \frac{1}{2\beta^2} \log\left(\frac{1}{2.4\delta}\right)\right\} \geqslant \frac{1}{4\beta^2} \log\left(\frac{1}{2.4\delta}\right) + \frac{1}{64} \sum_{j \neq i^*} \frac{1}{\Delta_{i^*,j}^2}$$

samples in expectation. □

6.D.3 Step 3: Reducing ALL-ϵ to isolated instance detection

In this section, we prove that for any instance ν for ALL-ϵ such that $|G_{\beta_\epsilon}(\nu)| = 1$ requires at least $O\left(\sum_{2=1}^{n} \frac{1}{\Delta_i^2}\right)$ samples in expectation. To do so, we prove a reduction

from finding all ϵ-good arms to the β-Isolated Hypothesis Testing. In particular, we show that if one has a generic method to find all ϵ-good arms (with slack $\gamma = 0$), then one may use this to develop a method to perform the β-Isolated Hypothesis Test. Therefore, lower bounds on the this test apply to the problem of finding all ϵ-good arms as well.

Lemma 6.20. *Fix $\delta \leqslant 1/16$, $n \geqslant 2/\delta$, $\epsilon > 0$, $\beta \in (0, \epsilon/2)$. Let ν be an instance of n arms such that the i^{th} is distributed as $\mathcal{N}(\mu_i, 1)$, $|G_{2\beta_\epsilon}| = 1$, and there exists an arm in G_ϵ^c such that $\mu_1 - \epsilon - \mu_i = \beta$. Select a permutation $\pi : [n] \to [n]$ uniformly from the set of $n!$ permutations, and consider the permuted instance $\pi(\nu)$. Any algorithm that returns $G_\epsilon(\pi(\nu))$ on $\pi(\nu)$ with correctly probability at least $1 - \delta$ requires at least*

$$\frac{1}{64}\sum_{i=2}^{n}\frac{1}{\Delta_i^2} + \frac{1}{4\beta_\epsilon^2}\log\left(\frac{1}{2.4\delta}\right)$$

samples in expectation taken jointly over the randomness in ν and π.

Proof. Fix $0 < \delta < 1/16$, $n > 2/\delta$, $\epsilon > 0$, $0 < \beta < \epsilon/2$, and an arbitrary constant $c \in \mathbb{R}$. Consider a given instance $\nu = \{\rho_1, \cdots, \rho_n\}$ such that $\mu_1 \in \{-\beta, \beta\}$, and $\mu_2, \cdots, \mu_n < -\beta$. We wish to perform the β-isolated hypothesis test on $\pi(\nu)$. Assume for contradiction that there exists a generic algorithm $\mathcal{A}(\nu', \epsilon, \delta)$ such that if given a generic instance ν' where $|G_{2\beta_\epsilon}(\nu')| = 1$, it returns $G_\epsilon(\nu')$ with probability at least $1 - \delta$ in at most $\frac{1}{64}\sum_{i=2}^{n}\frac{1}{\Delta_i^2}$ samples where μ_1' is the largest mean in ν'. Algorithm 8 uses $\mathcal{A}$ to perform the hypothesis test.

Note that as $n \geqslant 2/\delta$, $\mathbb{P}(\hat{\imath} = \pi(1)) \leqslant \delta/2$. The method replaces $\rho_{\hat{\imath}}$ with $\mathcal{N}(c-\epsilon, 1)$. All other means μ_i are shifted up by c. The test then runs $\mathcal{A}$ on this new instance ν' with failure probability $\delta/2$. If H_0 is true on $\pi(\nu)$, all distributions have means less than $-\beta$, and $\hat{\imath}$ therefore is ϵ-good on instance ν'. If H_1 is true on $\pi(\nu)$, then $\rho_{\pi(1)} = \mathcal{N}(\beta, 1)$ and $\hat{\imath}$ is not ϵ-good on instance ν'. This method correctly performs the test if $\hat{\imath} \neq \pi(1)$ and $\mathcal{A}$ does not fail, the joint event of which occurs with probability at most 2δ. Therefore, this test is correct with probability at least $1 - \delta$.

Let $\mathcal{T}_\mathcal{A}(\nu')$ denote the random variable of the number of samples drawn by $\mathcal{A}$ on instance ν' and let T denote the random variable of the total number of samples

Algorithm 8 Using All-ϵ for β-isolated hypothesis test

Require: $\delta > 0$, $\epsilon > 0$, $0 < \beta$, instance $\pi(\nu)$, constant c, and algorithm $\mathcal{A}$

1: **Step 1:** Choose an index $\hat{i} \in [n]$ uniformly
2: **Step 2:** Let ν' be the instance

$$\nu' = \begin{cases} \rho_{\pi(i)} + c & \text{if } i \neq \hat{i} \\ \mathcal{N}(c - \epsilon, 1) & \text{if } i = \hat{i} \end{cases}$$

3: **Step 3:** $G = \mathcal{A}(\nu', \epsilon, \delta/2)$
4: **if** $\hat{i} \in G$ **then**:
5: Declare H_0 and terminate
6: **else**
7: Declare H_1 and terminate
8: **end if**

drawn by this procedure before it terminates and declares H_0 or H_1 on ν'. Therefore, $\mathbb{E}_{\pi,\nu}[T] = \mathbb{E}_{\pi,\nu}[\mathcal{T}_{\mathcal{A}}(\nu')]$.

By Lemma 1 of Simchowitz et al. (2017), averaging over all permutations is equivalent to first permuting the instance ν and then passing it to $\mathcal{A}$ and undoing the permutation when returning the answer. We therefore assume that $\mathcal{A}$ is *symmetric* in that its expected sample complexity of $\mathcal{A}$ is invariant to the permutation π. Otherwise, we may use $\mathcal{A}$ to form a symmetric algorithm. Therefore, $\mathbb{E}_{\pi,\nu}[T] = \mathbb{E}_{\pi,\nu}[\mathcal{T}_{\mathcal{A}}(\nu')] = \mathbb{E}_{\nu}[\mathcal{T}_{\mathcal{A}}(\nu')]$. By Theorem 6.19, if H_1 is true,

$$\mathbb{E}_{\pi,\nu}[T] \geqslant \frac{1}{64} \sum_{i=2}^{n} \frac{1}{\Delta_i^2} + \frac{1}{4\beta^2} \log\left(\frac{1}{2.4\delta}\right).$$

Hence,

$$\mathbb{E}_{\nu}[\mathcal{T}_{\mathcal{A}}(\nu')] \geqslant \frac{1}{64} \sum_{i=2}^{n} \frac{1}{\Delta_i^2} + \frac{1}{4\beta^2} \log\left(\frac{1}{2.4\delta}\right).$$

Lastly, as the constant c was chosen arbitrarily, and β is an number in $(0, \epsilon/2)$ this argument applies to any ALL-ϵ instance ν' such that $\beta_\epsilon \in (0, \epsilon/2)$ and $|G_{2\beta_\epsilon}| = 1$ for n appropriate choice of c. □

With the above proof, we restate the following moderate confidence lower bound on the sample complexity of returning all ϵ-good stated in Section 6.4. In particular, this bound highlights *moderate confidence* terms that are independent of δ. Moderate confidence terms have been studied in works such as Simchowitz et al. (2017); Chen et al. (2017). Despite being independent of δ, these terms can have important effects in real world scenarios. The following bound demonstrates that there are instances for which moderate confidence terms are necessary for finding all ϵ-good arms. Moderate confidence terms likewise appear in the upper bound of the complexity of `FAREAST`, Theorem 6.8.

Theorem 6.21. *Fix $\delta \leqslant 1/16$, $n \geqslant 2/\delta$, and $\epsilon > 0$. Let ν be an instance of n arms such that the i^{th} is distributed as $\mathcal{N}(\mu_i, 1)$, $|G_{2\beta_\epsilon}| = 1$, and $\beta_\epsilon < \epsilon/2$. Select a permutation $\pi : [n] \to [n]$ uniformly from the set of $n!$ permutations, and consider the permuted instance $\pi(\nu)$. Any algorithm that returns $G_\epsilon(\pi(\nu))$ on $\pi(\nu)$ with correctly probability at least $1 - \delta$ requires at least*

$$c_2 \sum_{i=1}^{n} \max\left(\frac{1}{(\mu_1 - \epsilon - \mu_i)^2}, \frac{1}{(\mu_1 + \alpha_\epsilon - \mu_i)^2}\right) \log\left(\frac{1}{2.4\delta}\right) + c_2 \sum_{i=1}^{n} \frac{1}{(\mu_1 + \beta_\epsilon - \mu_i)^2}$$

samples in expectation over the randomness in ν and π for a universal constant c_2.

Proof. We may equivalently consider the same instance with all means shifted down by $\epsilon - 2\beta$ since a method for that instance could be used to return all ϵ good arms in the stated instance. By Lemma 6.20, $c_2 \frac{n}{\beta^2}$ samples are necessary in expectation. By Theorem 6.4,

$$2 \sum_{i=1}^{n} \max\left(\frac{1}{(\mu_1 - \epsilon - \mu_i)^2}, \frac{1}{(\mu_1 + \alpha_\epsilon - \mu_i)^2}\right) \log\left(\frac{1}{2.4\delta}\right)$$

samples are necessary in expectation. By Lemma 6.20,

$$\frac{1}{64} \sum_{i=2}^{n} \frac{1}{\Delta_i^2} + \frac{1}{4\beta_\epsilon^2} \log\left(\frac{1}{2.4\delta}\right) \geqslant \frac{1}{64} \sum_{i=2}^{n} \frac{1}{(\mu_1 + \beta_\epsilon - \mu_i)^2} + \frac{1}{4\beta_\epsilon^2} \log\left(\frac{1}{2.4\delta}\right)$$

$$\geqslant \frac{1}{64}\sum_{i=1}^{n}\frac{1}{(\mu_1+\beta_\epsilon-\mu_i)^2}$$

samples are necessary in expectation taken over the randomness in the permutation and in the instance. In particular, the maximum and therefore the average is a valid bound. Therefore, any algorithm requires

$$\sum_{i=1}^{n}\max\left(\frac{1}{(\mu_1-\epsilon-\mu_i)^2},\frac{1}{(\mu_1+\alpha_\epsilon-\mu_i)^2}\right)\log\left(\frac{1}{2.4\delta}\right)+\frac{1}{128}\sum_{i=1}^{n}\frac{1}{(\mu_1+\beta_\epsilon-\mu_i)^2}$$

samples in expectation. □

6.E Sample Complexity of finding positive means

Algorithm 6.F.1 builds on a more general idea of finding arms with positive means quickly. This is the core idea of `FAREAST`. To show that any arm $j \notin G_\epsilon$, it is sufficient to find any arm i such that $\mu_i - \mu_j > \epsilon$ in the additive case and similarly $(1-\epsilon)\mu_i > \mu_j$ in the multiplicative case. Focussing on the additive case, this is equivalent to finding any i such that $\mu_i - \mu_j - \epsilon > 0$. Not that neither of these conditions require i to be in G_ϵ. If we fix j, we can consider all difference distributions with respect to any arm i shifted down by ϵ: $\rho_i - \rho_j - \epsilon$ and try to quickly find one with a positive mean. As we show in this section, this can be done in relative few samples in expectation since we have an easy way to verify if any guess that we make does in fact have a positive mean. Below we state an algorithm, `FindPos`, 9, and a theorem bounding its complexity. The algorithm proceeds in rounds, each consisting of and *explore* step and a *verify* step similar to Karnin (2016); Mannor and Tsitsiklis (2004). In the *explore* step, we make a guess that is correct with constant probability at a β-good distribution. In the *verify* step, sufficiently many samples are drawn from the this distribution to form a $\beta/2$ confidence width with high probability. If the lower bound exceeds 0, the algorithm terminates. Otherwise, the process repeats. To account for the fact that β is unknown, we make dyadically decreasing guesses at β. In particular, in the k^{th} round, we guess that $\beta \geqslant 2^{-k}$. Similar tools have been

employed in for pure-exploration the infinite armed bandit literature Jamieson et al. (2016); Li et al. (2017). It relies on a generic subroutine called ϵ-GOOD which given n subGaussian arms, and $\epsilon, \kappa > 0$, returns a single ϵ-good arm with probability at least $1 - \kappa$. Examples of such a routine include LUCB and Median-Elimination Kalyanakrishnan et al. (2012); Even-Dar et al. (2002).

Algorithm 9 The FindPos algorithm

Require: $0 < \delta < 1/8, 0 < \kappa < 1/16$

1: **for** $k = 1, 2 \cdots$ **do**
2: Guess: $\beta_k = 2^{-k}$
3: $i_k \leftarrow \epsilon$-GOOD(β_k, κ) ▷ Explore step
4: Sample i_k $\lceil 4\beta_k^{-2} \log(4k^2/\delta) \rceil$ times and compute $\hat{\mu}_{i_k}$ ▷ Verify step
5: **if** $\hat{\mu}_{i_k} > \frac{1}{2}\beta_k$ **then return** i_k
6: **end if**
7: **end for**

Theorem 6.22. *Fix $0 < \delta < 1/8$ and $0 < \kappa < 1/16$ and consider an instance of n 1-subGaussian arms such that the largest mean is $\beta > 0$, unknown to the algorithm. Let ϵ-GOOD be performed by Median-Elimination. With probability at least $1 - \delta$, FindPos returns a distribution with positive mean in at most.*

$$O\left(\frac{n}{\beta^2} + \frac{1}{\beta^2} \log\left(\frac{\log_2(2/\beta)}{\delta}\right)\right)$$

samples in expectation.

To appreciate, the above result, consider an oracle that *knows* all means, and merely wishes to prove that any arm has a positive mean with probability at least $1-\delta$. If this arm has mean β, the oracle can simply sample that arm $O(1/\beta^2 \log(1/\delta))$ times to do so. Naturally such a method is unrealistic, since a practical algorithm needs to also find an arm with positive mean as well as verify this fact. The above result states that roughly $O(n/\beta^2)$ samples are necessary to find an arm with positive mean. In particular, this term exists in *moderate confidence*, as it vanishes as $\delta \to 0$. Hence, asymptotically, FindPos achieves the same complexity as an

oracle that knows all means! Furthermore, we may also recover the spike detection problem studied in Appendix 6.D.1. In that setting, exactly 1 arm has mean β and all others have mean 0. A lower bound for that problem, given in Theorem 6.13 indicates that the result of Theorem 6.22 is optimal up to a doubly logarithmic factor for that problem.

6.E.1 Proof of Theorem 6.22

Now we bound the expected sample complexity of `FindPos` sample complexity of. Let $\mu_1, \cdots, \mu_n$ denote the means of the n arms and $\mu_1 = \max\{\mu_1, \cdots, \mu_n\} = \beta$. Throughout, for any $k \in \mathbb{N}$ consider the indicator random variable Y_k that `FindPos` does not terminate in round k or before it. Namely,

$$Y_k = \bigcap_{r=1}^{k} (\hat{\mu}_{i_r}(\tau_r) \leqslant \beta_r)$$

Additionally, we let $\tau_k = \left\lceil 4\beta_k^{-2} \log(4k^2/\delta) \right\rceil$ for $\beta_k = 2^{-k}$. Let $\mu_i(t)$ denote the empirical mean of arm i after t iid pulls. Let $H_{ME}(n, \epsilon, \kappa) = \lceil c' \frac{n}{\epsilon^2} \log(1/\kappa) \rceil$ be the complexity of `Median-Elimination`.

6.E.1.1 Step 0: Correctness

First, we show that with probability $1 - \delta$, `FindPos` returns an arm with a positive mean. Since i_k is sampled τ_k times, by Hoedffding's inequality

$$\mathbb{P}(|\hat{\mu}_j(\tau_k) - \mu_j| > 2^{-k+1} | i_k = j) \leqslant \frac{\delta}{2k^2}$$

Next,

$$\begin{aligned}
\mathbb{P}(|\hat{\mu}_j(\tau_k) - \mu_j| > 2^{-k+1}) &= \sum_{i=1}^{n} \mathbb{P}(|\hat{\mu}_j(\tau_k) - \mu_j| > 2^{-k+1} | i_k = j) \mathbb{P}(i_k = j) \\
&\leqslant \frac{\delta}{2k^2} \sum_{i=1}^{n} \mathbb{P}(i_k = j)
\end{aligned}$$

$$= \frac{\delta}{2k^2}$$

Therefore, union bounding over rounds $k \in \mathbb{N}$,

$$\mathbb{P}\left(\bigcup_{t=1}^{\infty} |\hat{\mu}_{i_k}(\tau_k) - \mu_{i_k}| > 2^{-(k+1)}\right) \leqslant \sum_{k=1}^{\infty} \frac{\delta}{2k^2} = \delta$$

`FindPos` errs if and only if in any round k $\mu_{i_k} < 0$ but $\hat{\mu}_{i_k}(\tau_k) \geqslant \frac{1}{2}\beta_k = 2^{-(k+1)}$. By the above, this occurs with probability δ.

6.E.1.2 Step 1: Bounding $\mathbb{E}[Y_k]$

Next, we bound the probability that `FindPos` proceeds to round k.

Claim 1: For $k \geqslant \left\lceil \log_2\left(\frac{2}{\beta}\right)\right\rceil$, $\mathbb{E}[Y_k] \leqslant \left(\frac{1}{8}\right)^{k-\left\lceil \log_2\left(\frac{2}{\beta}\right)\right\rceil}$

Proof. Condition on $Y_{k-1} = 1$, the event that `FindPos` does not terminate through round $k-1$. By Hoeffding's inequality an a union bound over possible values of i_k, with probability at least $1 - \frac{\delta}{2k^2}$, $|\hat{\mu}_{i_k} - \mu_{i_k}| \leqslant 2^{-(k+1)}$ and therefore, $\hat{\mu}_{i_k} \geqslant \mu_{i_k} - 2^{-(k+1)}$. If `Median-Elimination` also succeeds, the joint event of which occurs with probability $\frac{15}{16}(1 - \frac{\delta}{2k^2})$, $\mu_{i_k} \geqslant \beta - 2^{-k}$. Hence, $\hat{\mu}_{i_k} \geqslant \beta - 2^{-k} - 2^{-(k+1)}$. Then for $k \geqslant \left\lceil \log_2\left(\frac{2}{\beta}\right)\right\rceil$, $2^{-k} \leqslant \beta/2$. Hence, $\hat{\mu}_{i_k} \geqslant \beta/4 > 0$ and `FindPos` terminates and returns i_k. In particular, then $\mathbb{E}[Y_k | Y_{k-1} = 1] \geqslant \frac{15}{16}(1 - \frac{\delta}{2k^2})$. Note that if `FindPos` terminates in round k, and $Y_k = 0$ for all $k' \geqslant k$. Since Y_k is an indicator random variable and $\mathbb{E}[Y_k | Y_{k-1} = 0] = 0$ deterministically by definition of Y_k,

$$\begin{aligned}
\mathbb{E}[Y_k] &= \mathbb{E}[Y_k | Y_{k-1} = 0]\mathbb{P}(Y_{k-1} = 0) + \mathbb{E}[Y_k | Y_{k-1} = 1]\mathbb{P}(Y_{k-1} = 1) \\
&= \mathbb{E}[Y_k | Y_{k-1} = 1]\mathbb{P}(Y_{k-1} = 1) \\
&= \mathbb{E}[Y_k | Y_{k-1} = 1]\mathbb{P}(Y_{k-1}) \\
&= \mathbb{E}[Y_k | Y_{k-1} = 1]\mathbb{E}[Y_{k-1}] \\
&\leqslant \left(\frac{1}{16} + \frac{\delta}{4k^2}\right)\mathbb{E}\left[\mathbb{1}(Y_{k-1} = 0)\right].
\end{aligned}$$

For $k < \left\lceil \log_2\left(\frac{2}{\beta}\right)\right\rceil$, trivially, $\mathbb{E}[Y_k] \leqslant 1$. Recall $\delta \leqslant 1/8$. For $k \geqslant \left\lceil \log_2\left(\frac{2}{\beta}\right)\right\rceil$,

$$\mathbb{E}[Y_k] \leqslant \prod_{s=\left\lceil \log_2\left(\frac{2}{\beta}\right)\right\rceil}^{k} \left(\frac{1}{16} + \frac{\delta}{2s^2}\right) \leqslant \left(\frac{1}{8}\right)^{k-\left\lceil \log_2\left(\frac{2}{\beta}\right)\right\rceil}.$$

□

6.E.1.3 Bounding the total number of samples drawn by `FindPos`

Let T denote the random variable of the total number of samples drawn by `FindPos`.

Claim: $\mathbb{E}[T] \leqslant c\frac{n}{\beta^2} + \frac{c}{\beta^2}\log\left(\log_2\left(\frac{2}{\beta}\right)/\delta\right)$ for a constant c.

Proof. We may write T as

$$T = \sum_{k=1}^{\infty} Y_{k-1}\left(H_{ME}\left(n, 2^{-k}, \frac{1}{16}\right) + \tau_k\right)$$

and $\mathbb{E}[T]$ as

$$\mathbb{E}[T] = \sum_{k=1}^{\infty} \mathbb{E}[Y_{k-1}]\left(H_{ME}\left(n, 2^{-k}, \frac{1}{16}\right) + \tau_k\right)$$

since H_{ME} and τ_k are deterministic quantities. We will show that the right side of the above equation is finite. Hence, the equality is true by the Fubini-Tonelli Theorem.

This sum decomposes into two terms.

$$\begin{aligned}
&\sum_{k=1}^{\infty} \mathbb{E}[Y_k]\left(\tau_k + H_{ME}(n, 2^{-k}, 1/16)\right) \\
&\quad = \sum_{k=1}^{\left\lfloor \log_2\left(\frac{2}{\beta}\right)\right\rfloor} \mathbb{E}[Y_{k-1}]\left(H_{ME}(n, 2^{-k}, 1/16) + \left\lceil 2^{2k+2}\log\left(\frac{4k^2}{\delta}\right)\right\rceil\right) \\
&\quad + \sum_{k=\left\lceil \log_2\left(\frac{2}{\beta}\right)\right\rceil}^{\infty} \mathbb{E}[Y_{k-1}]\left(H_{ME}(n, 2^{-k}, 1/16) + \left\lceil 2^{2k+2}\log\left(\frac{4k^2}{\delta}\right)\right\rceil\right)
\end{aligned}$$

We begin by bounding the first term.

$$\begin{aligned}
&\sum_{k=1}^{\lfloor \log_2(\frac{2}{\beta}) \rfloor} \mathbb{E}\left[Y_{k-1}\right] \left(H_{ME}(n, 2^{-k}, 1/16) + \left\lceil 2^{2k+2} \log\left(\frac{4k^2}{\delta}\right) \right\rceil \right) \\
&\qquad \leqslant \sum_{k=1}^{\lfloor \log_2(\frac{2}{\beta}) \rfloor} \left(H_{ME}(n, 2^{-k}, 1/16) + \left\lceil 2^{2k+2} \log\left(\frac{4k^2}{\delta}\right) \right\rceil \right) \\
&\qquad \leqslant \sum_{k=1}^{\lfloor \log_2(\frac{2}{\beta}) \rfloor} \left(c' n 2^{2k} \log(16) + 1 + 2^{2k+2} \log\left(\frac{4k^2}{\delta}\right) \right) \\
&\qquad \leqslant \log_2\left(\frac{2}{\beta}\right) + \left(c' n \log(16) + 4 \log\left(\frac{4}{\delta}\right) \right) \sum_{k=1}^{\lfloor \log_2(\frac{2}{\beta}) \rfloor} 2^{2k} \\
&\qquad\qquad + 8 \sum_{k=1}^{\lfloor \log_2(\frac{2}{\beta}) \rfloor} 2^{2k} \log(k) \\
&\qquad \leqslant \log_2\left(\frac{2}{\beta}\right) + \left(c' n \log(16) + 4 \log\left(\frac{4}{\delta}\right) + \log\log_2\left(\frac{2}{\beta}\right) \right) \sum_{k=1}^{\lfloor \log_2(\frac{2}{\beta}) \rfloor} 2^{2k} \\
&\qquad \leqslant \log_2\left(\frac{2}{\beta}\right) + \left(c' n \log(16) + 4 \log\left(\frac{4}{\delta} \log_2\left(\frac{2}{\beta}\right)\right) \right) 2 \cdot 2^{2\log_2(\frac{2}{\beta})} \\
&\qquad \leqslant \log_2\left(\frac{2}{\beta}\right) + \frac{8}{\beta^2} \left(c' n \log(16) + 4 \log\left(\frac{4}{\delta} \log_2\left(\frac{2}{\beta}\right)\right) \right)
\end{aligned}$$

Next, we plug in the bound from claim 1 controlling $\mathbb{E}[Y_k]$.

Using Claim 1, we bound the second sum as follows:

$$\begin{aligned}
&\sum_{r=\lceil \log_2(\frac{2}{\beta}) \rceil}^{\infty} \mathbb{E}\left[Y_{k-1}\right] \left(H_{ME}(n, 2^{-k}, 1/16) + \left\lceil 2^{2k+2} \log\left(\frac{4k^2}{\delta}\right) \right\rceil \right) \\
&\qquad \leqslant \sum_{k=\lceil \log_2(\frac{2}{\beta}) \rceil}^{\infty} \left(\frac{1}{8}\right)^{k - \lceil \log_2(\frac{2}{\beta}) \rceil - 1} \left(c' n 2^{2k} \log(16) + 1 + 2^{2k+2} \log\left(\frac{4k^2}{\delta}\right) \right)
\end{aligned}$$

$$
\begin{aligned}
&= c'n\log(16)\sum_{i=1}^{\infty}\left(\frac{1}{8}\right)^{i-1}2^{2\left(i+\left\lceil\log_2\left(\frac{2}{\beta}\right)\right\rceil\right)}+\sum_{i=1}^{\infty}\left(\frac{1}{8}\right)^{i-1}\\
&\qquad+4\sum_{i=1}^{\infty}\left(\frac{1}{8}\right)^{i-1}2^{2\left(i+\left\lceil\log_2\left(\frac{2}{\beta}\right)\right\rceil\right)}\log\left(\frac{4\left(i+\left\lceil\log_2\left(\frac{2}{\beta}\right)\right\rceil\right)^2}{\delta}\right)\\
&\leqslant 2+c'n\log(16)\sum_{i=1}^{\infty}2^{-3i+3}2^{2\left(i+\log_2\left(\frac{2}{\beta}\right)+1\right)}\\
&\qquad+4\sum_{i=1}^{\infty}2^{-3i+3}2^{2\left(i+\log_2\left(\frac{2}{\beta}\right)+1\right)}\log\left(\frac{4\left(i+\left\lceil\log_2\left(\frac{2}{\beta}\right)\right\rceil\right)^2}{\delta}\right)\\
&= 2+\left(\frac{2^7c'n\log(16)}{\beta^2}+\frac{2^{11}}{\beta^2}\log\left(\frac{4}{\delta}\right)\right)\sum_{i=1}^{\infty}2^{-i}\\
&\qquad+\frac{2^{11}}{\beta^2}\sum_{i=1}^{\infty}2^{-i}\log\left(\left(i+\left\lceil\log_2\left(\frac{2}{\beta}\right)\right\rceil\right)^2\right)\\
&\leqslant 2+\frac{2^7c'n\log(16)}{\beta^2}+\frac{2^{11}}{\beta^2}\log\left(\frac{4}{\delta}\right)\\
&\qquad+\frac{2^{12}}{\beta^2}\sum_{i=1}^{\infty}2^{-i}\log\left(i+\left\lceil\log_2\left(\frac{2}{\beta}\right)\right\rceil\right)\\
&= (**)
\end{aligned}
$$

We may bound the final summand, $\sum_{i=1}^{\infty}2^{-i}\log\left(i+\left\lceil\log_2\left(\frac{2}{\beta}\right)\right\rceil\right)$ as follows:

$$
\sum_{i=1}^{\infty}2^{-i}\log\left(i+\left\lceil\log_2\left(\frac{2}{\beta}\right)\right\rceil\right)\leqslant\log\left(\frac{e}{2}\log_2\left(\frac{128}{\beta^2}\right)\right)
$$

Plugging this back into $(**)$, we have that

$$
(**)\leqslant 3+\frac{2^7cn\log(16)}{\beta^2}+\frac{2^{12}}{\beta^2}\log\left(\frac{4}{\delta}\right)+\frac{2^{12}}{\beta^2}\log\left(\frac{e}{2}\log_2\left(\frac{128}{\beta^2}\right)\right)
$$

Combining the above with the bound on the first sum, we have that

$$\sum_{k=1}^{\infty} \mathbb{E}\left[Y_{k-1}\right]\left(\tau_k + H_{ME}(n, 2^{-k}, 1/16)\right) \leqslant \frac{c''n}{\beta^2} + c''\frac{1}{\beta^2}\log\left(\frac{1}{\delta}\log_2\left(\frac{2}{\beta}\right)\right)$$

for a sufficiently large, universal constant c''. □

6.F An optimal method for finding all additive and multiplicative ϵ-good arms

6.F.1 The FAREAST Algorithm

Below, we present an algorithm called FAREAST (**F**ast **A**rm **R**emoval **E**limination **A**lgorithm for a **S**ampled **T**hreshold) that achieves the lower bound when $\gamma = 0$. Similar to $(\mathtt{ST})^2$, it relies on anytime-correct confidence widths, $C_\delta(t) := \sqrt{\frac{4\log(\log_2(2t)/\delta)}{t}}$. The algorithm proceeds in rounds, and creates a filter for *good* arms and a filter for *bad* arms. The good filter detects arms in G_ϵ of M_ϵ and adds them to a set G_k. Similarly, the bad filter detects arms in G_ϵ^c or M_ϵ^c and adds them to a set B_k. At any given time, we may represent the set of arms that have *not* been declared as either in G_ϵ/M_ϵ or $G_\epsilon^c/M_\epsilon^c$ as $(G_k \cup B_k)^c$. In either the additive or multiplicative case, the algorithm terminates when it can certify that $G_\epsilon \subset G_k$ and $G_k \cap G_{\epsilon+\gamma}^c = \emptyset$ or $M_\epsilon \subset G_k$ and $G_k \cap M_{\epsilon+\gamma}^c = \emptyset$, respectively– i.e., when G_k contains all additive or multiplicative ϵ-good arms and none worse than $(\epsilon+\gamma)$-good.

In each round, the bad filter uses MedianElimination Even-Dar et al. (2002) which given an instance ν, a value of ϵ, and a failure probability κ, returns an ϵ-good arm with probability at least $1-\kappa$. In the k^{th} round, for an arm i in $(G_k \cup B_k)^c$, the bad filter uses MedianElimination to find a 2^{-k} good arm i_k with failure probability $\kappa = O(1)$ and then samples both arms i and i_k $\tilde{O}(2^{2k}\log(1/\delta))$ times. Let $\hat{\mu}_i$ and $\hat{\mu}_{i_k}$ denote the empirical means. For instance, in the additive case, if $\hat{\mu}_{i_k} - \hat{\mu}_i \geqslant \epsilon + 2^{-k+1}$, we may declare that $i \in G_\epsilon^c$, and the bad filter adds i to the set B_k. This allows the bad filter to commit to a single arm and sample it sufficiently to remove arms in G_ϵ^c.

The good filter is a simple elimination scheme. It maintains an upper bound U_t and lower bound L_t on $\mu_1 - \epsilon$. If an arm's upper bound drops below L_t (line 20), the good filter eliminates that arm, otherwise, if an arm's lower bound rises above U_t (19), the good filter adds the arm to G_k, but only eliminates this arm if its upper bound falls below the highest lower bound. This ensures that μ_1 is never eliminated and U_t and L_t are always valid bounds This scheme works as an independent algorithm and achieves the sample complexity as $(\mathtt{ST})^2$, though worse empirical performance. We analyze this method in Appendix 6.F.5. Indeed, this gives an additional high probability guarantee on the number of samples drawn by FAREAST in both the additive and multiplicative regimes. As the sampling is split across rounds, the good filter always samples the least sampled arm, breaking ties arbitrarily. The number of samples given to the good filter in each round is such that both filters receive identically many samples. Note that this is a random quantity since the number of arms in $(G_k \cup B_k)^c$ in round k is random. Despite this, we prove a lower bound on the number of samples drawn per round which ensures the Good Filter always receives a positive number of samples in each round. Note that by design elimination only occurs when all arms in the active set have received equal numbers of samples. This is crucial as it prevents the good filter from over-sampling bad arms and vice versa. In our proof, we show that in some round, unknown to the algorithm, $G_k = G_\epsilon$, ie all good arms have been found, and this takes no more than $O\left(\sum_{i=1}^{n} \max\left\{(\mu_1 - \epsilon - \mu_i)^{-2}, (\mu_1 + \alpha_\epsilon - \mu_i)^{-2}\right\} \log(n/\delta)\right)$ samples, matching the lower bound.

FAREAST

Input: ϵ, δ, Instance ν, slack $\gamma \geqslant 0$. If multiplicative, $\epsilon \in (0, 1/2]$

Let $G_0 = \emptyset$ be the set of arms declared as good and $B_0 = \emptyset$ the set of arms declared as bad.

Let $\mathcal{A} = [n]$ be the active set, $N_i = 0$ track the total number of samples of arm i by the Good Filter.

Let $t = 0$ denote the total number of times that line 19 is true in the Good Filter.

Let $C_{\delta/2n}(t)$ be an anytime $\delta/2n$-correct confidence width on t samples.

Let $H_{ME}(n, \epsilon, \kappa) = \lceil c' \frac{n}{\epsilon^2} \log(1/\kappa) \rceil$ be the complexity of `MedianElimination`.

for $k = 1, 2, \cdots$

  Let $\delta_k = \delta/2k^2$, $\tau_k = \left\lceil 2^{2k+3} \log\left(\frac{8n}{\delta_k}\right) \right\rceil$, Initialize $G_k = G_{k-1}$ and $B_k = B_{k-1}$

  // Bad Filter: find bad arms in G_ϵ^c or M_ϵ^c

  Let $i_k = \texttt{MedianElimination}(\nu, 2^{-k}, 1/16)$, sample i_k τ_k times, and compute $\hat{\mu}_{i_k}$

  for $i \notin G_{k-1} \cup B_{k-1}$**:**

    Sample μ_i τ_k times and compute $\hat{\mu}_i$

    If $\hat{\mu}_{i_k} - \hat{\mu}_i \geqslant \epsilon + 2^{-k+1}$ or $(1-\epsilon)\hat{\mu}_{i_k} - \hat{\mu}_i > 2^{-(k+1)}(2-\epsilon)$**:**

      Add i to B_k

  // Good Filter: find good arms in G_ϵ or M_ϵ

  for $s = 1, \cdots, H_{\mathbf{ME}}(n, 2^{-k}, 1/16) + \tau_k \cdot (|(G_{k-1} \cup B_{k-1})^c| + 1)$**:**

    Pull arm $I_s \in \arg\min_{j \in \mathcal{A}}\{N_j\}$ and set $N_{I_s} \leftarrow N_{I_s} + 1$.

    if $\min_{j \in \mathcal{A}}\{N_j\} = \max_{j \in \mathcal{A}}\{N_j\}$**:**

      $t = t + 1$

      For $i \in \mathcal{A}$ denote $\hat{\mu}_i(t)$ the average of the first t samples of arm i.

      Let $U_t = \max_{j \in \mathcal{A}} \hat{\mu}_i(t) + C_{\delta/2n}(t) - \epsilon$ or $U_t = (1-\epsilon)\left(\max_{j \in \mathcal{A}} \hat{\mu}_i(t) + C_{\delta/2n}(t)\right)$

      Let $L_t = \max_{j \in \mathcal{A}} \hat{\mu}_i(t) - C_{\delta/2n}(t) - \epsilon$ or $L_t = (1-\epsilon)\left(\max_{j \in \mathcal{A}} \hat{\mu}_i(t) - C_{\delta/2n}(t)\right)$

      for $i \in \mathcal{A}$**:**

        if $\hat{\mu}_i(t) - C_{\delta/2n}(t) \geqslant U_t$**:**

          Add i to G_k

        if $\hat{\mu}_i(t) + C_{\delta/2n}(t) \leqslant L_t$**:** // Bad arms are removed from $\mathcal{A}$

          Remove i from $\mathcal{A}$

        if $i \in G_k$ **and** $\hat{\mu}_i(t) + C_{\delta/2n}(t) \leqslant \max_{j \in \mathcal{A}} \hat{\mu}(t) - C_{\delta/2n}(t)$**:** // Good arms removed

          Remove i from $\mathcal{A}$

        If $\mathcal{A} \subset G_k$ or $G_k \cup B_k = [n]$**:**

          Output: the set G_k // Stopping condition for returning G_ϵ exactly.

        If $U_t - L_t < \frac{1}{2}\gamma$ or $U_t - L_t < \frac{\gamma}{2-\epsilon} L_t$**:**

          Output: the set $\mathcal{A} \cup G_k$ // Stopping condition for $\gamma > 0$.

The algorithm stops on either of three conditions. First, if $G_k \cup B_k = [n]$, every arm has been declared as either in G_ϵ or G_ϵ^c (or M_ϵ or M_ϵ^c). Second, if $\mathcal{A} \subset G_k$, the

Good Filter has found every arm in G_ϵ and FAREAST can terminate. This is the same stopping condition as EAST itself. In either case, FAREAST returns the set $G_k = G_\epsilon$ exactly. The third condition allows for γ slack. The good filter maintains upper and lower bounds U_t and L_t on the threshold in both the additive an multiplicative cases. In the additive case, if $U_t - L_t < \gamma/2$, then all arms in $G^c_{\epsilon+\gamma}$ have been added to B_k, and FAREAST may return $G_k \cup \mathcal{A}$. The condition for the multiplicative case is similar, though slightly more complicated. Throughout, we will use red text to denote pieces specific to the additive case and blue text to denote pieces specific to the multiplicative case.

Remark 6.23. *Note that the active set $\mathcal{A}$ defined in line 4 of FAREAST is only used and updated internally by the Good Filter. In particular, it is not necessarily true that $(G_k \cup B_k)^c = \mathcal{A}$. Furthermore, a bad arm $i \in G^c_\epsilon$ maybe removed from $\mathcal{A}$ even though it is not in B_k and vice versa as the Good Filter only seeks to detect good arms in G_ϵ and the Bad Filter only seeks to detect arms in G^c_ϵ. The same is true in the multiplicative case.*

Remark 6.24. *It is possible that when the loop in line 17 finishes in any given round, some arms in $\mathcal{A}$ have received more samples than others. Because $I_s \in \arg\min_{j\in\mathcal{A}}\{N_j\}$ in line 18, this difference is no more than 1, and the arms with fewer samples are the first to be sampled in the next round. The condition on line 19 ensures that all arms have equal numbers of samples* by the Good Filter *(e.g., the N_i's) when the Good Filter identifies good arms or eliminates arms from $\mathcal{A}$.*

Now, we restate Theorem 6.8 for reference.

Theorem 6.25. *Fix $0 < \epsilon$, $0 < \delta < 1/8$, slack $\gamma \in [0, 8]$ and an instance ν of n arms such that $\max(\Delta_i, |\epsilon - \Delta_i|) \leqslant 8$ for all i. There exists an event E such that $\mathbb{P}(E) \geqslant 1 - \delta$, and on E, FAREAST terminates and returns G such that $G_\epsilon \subset G \subset G_{\epsilon+\gamma}$ in at most*

$$c_4 \sum_{i=1}^{n} \min\left\{ \max\left\{ \frac{1}{(\mu_1 - \epsilon - \mu_i)^2} \log\left(\frac{n}{\delta}\log_2\left(\frac{n}{\delta(\mu_1 - \epsilon - \mu_i)^2}\right)\right), \right.\right.$$
$$\left.\left. \frac{1}{(\mu_1 + \alpha_\epsilon - \mu_i)^2} \log\left(\frac{n}{\delta}\log_2\left(\frac{n}{\delta(\mu_1 + \alpha_\epsilon - \mu_i)^2}\right)\right)\right.\right.,$$

$$\frac{1}{(\mu_1+\beta_\epsilon-\mu_i)^2}\log\left(\frac{n}{\delta}\log_2\left(\frac{n}{\delta(\mu_1+\beta_\epsilon-\mu_i)^2}\right)\right)\Bigg\},$$
$$\frac{1}{\gamma^2}\log\left(\frac{n}{\delta}\log_2\left(\frac{n}{\delta\gamma^2}\right)\right)\Bigg\}$$

samples for a constant c_4*. Furthermore*

$$\mathbb{E}[\mathbb{1}_E T] \leqslant c_3 \sum_{i\in G_\epsilon} \max\left\{\frac{1}{(\mu_1-\epsilon-\mu_i)^2}\log\left(\frac{n}{\delta}\log_2\left(\frac{n}{\delta(\mu_1-\epsilon-\mu_i)^2}\right)\right),\right.$$
$$\left.\frac{1}{(\mu_1+\alpha_\epsilon-\mu_i)^2}\log\left(\frac{n}{\delta}\log_2\left(\frac{n}{\delta(\mu_1+\alpha_\epsilon-\mu_i)^2}\right)\right)\right\}$$
$$+ c_3 \sum_{i\in G_\epsilon^c} \frac{n}{(\mu_1-\epsilon-\mu_i)^2} + \frac{1}{(\mu_1-\epsilon-\mu_i)^2}\log\left(\frac{n}{\delta}\log_2\left(\frac{n}{\delta(\mu_1-\epsilon-\mu_i)^2}\right)\right)$$

for a sufficiently large constant c_3 *where* T *denotes the number of samples.*

Next, we present a theorem bounding the sample complexity of `FAREAST` for returning multiplicative ϵ-good arms. Recall that $\tilde{\alpha}_\epsilon := \min_{i\in M_\epsilon}\mu_i - (1-\epsilon)\mu_1$ and $\tilde{\beta}_\epsilon := \min_{i\in M_\epsilon^c}(1-\epsilon)\mu_1 - \mu_i$, the distance for the smallest good arm and best arm that is not good to the threshold $(1-\epsilon)\mu_1$.

Theorem 6.26. *Fix* $\epsilon \in (0, 1/2]$, $\gamma \in [0, \min(1, 6/\mu_1))$, $0 < \delta < 1/8$ *and an instance* ν *of* n *arms such that* $\max(\Delta_i, |\epsilon\mu_1 - \Delta_i|) \leqslant 6$*. Assume that the highest mean is non-negative, i.e.,* $\mu_1 \geqslant 0$*. There exists an event* E *such that* $\mathbb{P}(E) \geqslant 1-\delta$*, and on* E*,* ***`FAREAST`*** *terminates and returns* G *such that* $M_\epsilon \subset G \subset M_{\epsilon+\gamma}$ *in at most*

$$c_5 \sum_{i=1}^{n} \min\left\{\max\left\{\frac{1}{((1-\epsilon)\mu_1-\mu_i)^2}\log\left(\frac{n}{\delta}\log_2\left(\frac{n}{\delta((1-\epsilon)\mu_1-\mu_i)^2}\right)\right),\right.\right.$$
$$\frac{}{(\mu_1+\frac{\tilde{\alpha}_\epsilon}{1-\epsilon}-\mu_i)^2}\log\left(\frac{n}{\delta}\log_2\left(\frac{n}{\delta(\mu_1+\frac{\tilde{\alpha}_\epsilon}{1-\epsilon})^2}\right)\right),$$
$$\left.\frac{1}{(\mu_1+\frac{\tilde{\beta}_\epsilon}{1-\epsilon}-\mu_i)^2}\log\left(\frac{n}{\delta}\log_2\left(\frac{n}{\delta(\mu_1+\frac{\tilde{\beta}_\epsilon}{1-\epsilon}-\mu_i)^2}\right)\right)\right\},$$
$$\left.\frac{(1-\epsilon+\gamma)^2}{\gamma^2\mu_1^2}\log\left(\frac{n}{\delta}\log_2\left(\frac{(1-\epsilon+\gamma)^2 n}{\delta\gamma^2\mu_1^2}\right)\right)\right\}$$

samples for a sufficiently large constant c_5*. Furthermore*

$$\mathbb{E}[\mathbb{1}_E T] \leqslant c_6 \sum_{i=1}^{n} \max\left\{ \frac{1}{((1-\epsilon)\mu_1 - \mu_i)^2} \log\left(\frac{n}{\delta}\log_2\left(\frac{n}{\delta((1-\epsilon)\mu_1-\mu_i)^2}\right)\right), \right.$$
$$\left. \frac{1}{\left(\mu_1 + \frac{\tilde{\alpha}_\epsilon}{1-\epsilon} - \mu_i\right)^2} \log\left(\frac{n}{\delta}\log_2\left(\frac{n}{\delta\left(\mu_1 + \frac{\tilde{\alpha}_\epsilon}{1-\epsilon} - \mu_i\right)^2}\right)\right)\right\}$$
$$+ c_6 \sum_{i \in M_\epsilon^c} \frac{n}{((1-\epsilon)\mu_1 - \mu_i)^2}$$

for a sufficiently large constant c_6*, where* T *denotes the number of samples.*

6.F.2 Key ideas of the proof

The proof revolves around a central idea: there is an event in unknown round K_{Good} in which the final arm from G_ϵ or M_ϵ is added to G_k. We may split the total number of samples drawn as the number taken through round K_{Good} and the number taken from $K_{\text{Good}} + 1$ until termination if the algorithm does not terminate in round K_{Good}. Note that the Good filter and Bad filter are given the same number of samples in each round. The proof of FAREAST in the multiplicative regime is similar and deferred to Appendix 6.F.4.

We begin by bounding the number of samples given to the Good filter when this event occurs that $G_k = G_\epsilon$. Next, since this happens at a random time within round K_{Good}, we bound the total number of additional samples in this round. Collectively, this gives us control over the number of samples drawn through round K_{Good}.

Next, we bound the number of samples from $K_{\text{Good}} + 1$ until termination. To do so, we analyze the expected number of samples drawn by the Bad filter before all arms in G_ϵ^c have been added to B_k. The total number of samples from $K_{\text{Good}} + 1$ until termination is no worse than twice this value. The proof is split into 12 steps and logically are organized as follows:

1. Step 0: We show that $G_k \subset G_\epsilon$ and $B_k \subset G_\epsilon^c$. In particular, this is implies that $G_k \cup B_k = [n] \implies G_k = G_\epsilon$ so FAREAST terminates correctly.

2. Step 1: We split the total number of samples drawn by FAREAST into two sums that we will control individually.

3. Steps 2-4: We control the number of samples given to the Good filter before $G_k = G_\epsilon$.

4. Steps 5-6: Using the result of steps 2-4, we bound the total number of samples through round K_{Good}

5. Steps 7-8: We use the result of step 6 to bound the total *expected* number of samples drawn by FAREAST, simplifying slightly in the process.

6. Step 9: We bound the number of samples that the Bad filter draws in adding a single bad arm to B_k.

7. Step 10: Repeating the argument in step 9, for every $i \in G_\epsilon^c$, we bound the total number of samples from round $K_{\text{Good}} + 1$ until termination. We finish by combining the bound on the number of samples drawn through K_{Good} with the bound from $K_{\text{Good}} + 1$ until termination. This controls the expected sample complexity of FAREAST.

8. Step 11: We provide a high probability bound on the sample complexity of FAREAST.

6.F.3 Proof of Theorem 6.8, FAREAST in the additive regime

Proof. **Notation for the proof:** Throughout, recall $\Delta_i = \mu_1 - \mu_i$. Recall that t counts the number of times the conditional in line 19 is true. By Line 19 of FAREAST, all arms in $\mathcal{A}$ have received t samples when the loop in line 23 is executed for the t^{th} time. Within any round k, let $\mathcal{A}(t)$ and $G_k(t)$ denote the sets $\mathcal{A}$ and G_k at this time since both sets can change in lines 27 and 29 and 25 respectively. Let t_k denote the maximum value of t in round k. By Lines 18 and 19 of FAREAST, the total number of samples given to the good filter when the conditional in line 19 is true for the t^{th} time is $\sum_{s=1}^{t} |\mathcal{A}(s)|$.

For $i \in G_\epsilon$, let T_i denote the random variable of the number of times arm i is sampled by the good filter before it is added to G_k in Line 25. For $i \in G_\epsilon^c$, let T_i denote the random variable of the number of times arm i is sampled by the good filter before it is removed from $\mathcal{A}$ in Line 27. For any arm i, let T_i' denote the random variable of the of the number of times i is sampled by the good filter before $\hat{\mu}_i(t) + C_{\delta/2n}(t) \leqslant \max_{j \in \mathcal{A}} \hat{\mu}_j(t) - C_{\delta/2n}(t)$. Lastly, let T_γ denote the random variable of the number of times any arm is sampled by the good filter before $U_t - L_t < \gamma/2$.

Define the event

$$\mathcal{E}_1 = \left\{ \bigcap_{i \in [n]} \bigcap_{t \in \mathbb{N}} |\hat{\mu}_i(t) - \mu_i| \leqslant C_{\delta/2n}(t) \right\}.$$

Using standard anytime confidence bound results, and recalling that that $C_\delta(t) := \sqrt{\frac{4\log(\log_2(2t)/\delta)}{t}}$, we have

$$\begin{aligned}
\mathbb{P}(\mathcal{E}_1^c) &= \mathbb{P}\left(\bigcup_{i \in [n]} \bigcup_{t \in \mathbb{N}} |\hat{\mu}_i - \mu_i| > C_{\delta/2n}(t) \right) \\
&\leqslant \sum_{i=1}^{n} \mathbb{P}\left(\bigcup_{t \in \mathbb{N}} |\hat{\mu}_i - \mu_i| > C_{\delta/2n}(t) \right) \leqslant \sum_{i=1}^{n} \frac{\delta}{2n} = \frac{\delta}{2}
\end{aligned}$$

Next, recall that $\hat{\mu}_i(t)$ denotes the empirical average of t samples of ρ_i. Consider the event,

$$\mathcal{E}_2 = \bigcap_{i \in G_\epsilon} \bigcap_{k \in \mathbb{N}} |(\hat{\mu}_{i_k}(\tau_k) - \hat{\mu}_i(\tau_k)) - (\mu_{i_k} - \mu_i)| \leqslant 2^{-k}$$

By Hoeffding's inequality,

$$\mathbb{P}\left(|(\hat{\mu}_j(\tau_k) - \hat{\mu}_i(\tau_k)) - (\mu_j - \mu_i)| > 2^{-k} \middle| i_k = j \right) \leqslant \frac{\delta}{4nk^2}.$$

Then

$$\begin{aligned}
&\mathbb{P}\big(\,|(\hat{\mu}_j(\tau_k) - \hat{\mu}_i(\tau_k)) - (\mu_j - \mu_i)| > 2^{-k}\big) \\
&\quad = \sum_{j=1}^{n} \mathbb{P}\left(|(\hat{\mu}_j(\tau_k) - \hat{\mu}_i(\tau_k)) - (\mu_j - \mu_i)| > 2^{-k} \middle| i_k = j\right) \mathbb{P}(i_k = j) \\
&\quad \leqslant \frac{\delta}{4nk^2} \sum_{j=1}^{n} \mathbb{P}(i_k = j) \\
&\quad = \frac{\delta}{4nk^2}
\end{aligned}$$

Therefore, union bounding over the rounds $k \in \mathbb{N}$, $\mathbb{P}(\mathcal{E}_2^c) \leqslant \sum_{i \in G_\epsilon} \sum_{k=1}^{\infty} \frac{\delta}{4nk^2} \leqslant \frac{\delta}{2}$. Hence, $\mathbb{P}(\mathcal{E}_1 \cap \mathcal{E}_2) \geqslant 1 - \delta$.

6.F.3.1 Step 0: Correctness.

On $\mathcal{E}_1 \cap \mathcal{E}_2$, first we prove that if there exists a random round k at which $G_k \cup B_k = [n]$ then $G_k = G_\epsilon$. Additionally, we prove that on $\mathcal{E}_1 \cap \mathcal{E}_2$, if $\mathcal{A} \subset G_k$, then $G_k = G_\epsilon$. Therefore, for either stopping condition for `FAREAST` in line 31, on the event $\mathcal{E}_1 \cap \mathcal{E}_2$, `FAREAST` correctly returns the set G_ϵ.

Claim 0: On $\mathcal{E}_1 \cap \mathcal{E}_2$, for all $k \in \mathbb{N}$, $G_k \subset G_\epsilon$.

Proof. Firstly we show $1 \in \mathcal{A}$ for all $t \in \mathbb{N}$, namely the best arm is never removed from $\mathcal{A}$. Note for any i

$$\hat{\mu}_1 + C_{\delta/2n}(t) \geqslant \mu_1 \geqslant \mu_i \geqslant \hat{\mu}_i(t) - C_{\delta/2n}(t) > \hat{\mu}_i(t) - C_{\delta/2n}(t) - \epsilon.$$

In particular this shows, $\hat{\mu}_1 + C_{\delta/2n}(t) > \max_{i \in \mathcal{A}} \hat{\mu}_i(t) - C_{\delta/2n}(t) - \epsilon = L_t$ and $\hat{\mu}_1 + C_{\delta/2n}(t) \geqslant \max_{i \in \mathcal{A}} \hat{\mu}_i(t) - C_{\delta/2n}(t)$ showing that 1 will never exit $\mathcal{A}$ in line 28.

Secondly, we show that at all times t, $\mu_1 - \epsilon \in [L_t, U_t]$. By the above, since μ_1 never leaves $\mathcal{A}$,

$$U_t = \max_{i \in \mathcal{A}} \hat{\mu}_i(t) + C_{\delta/2n}(t) - \epsilon \geqslant \hat{\mu}_1(t) + C_{\delta/2n}(t) - \epsilon \geqslant \mu_1 - \epsilon$$

and for any i,

$$\mu_1 - \epsilon \geqslant \mu_i - \epsilon \geqslant \hat{\mu}_i(t) - C_{\delta/2n}(t) - \epsilon$$

Hence $\mu_1 - \epsilon \geqslant \max_i \hat{\mu}_i(t) - C_{\delta/2n}(t) - \epsilon = L_t$.

Next, we show that $G_k \subset G_\epsilon$ for all $k \geqslant 1, t \geqslant 1$. Suppose not. Then $\exists, k, t \in N$ and $\exists i \in G_\epsilon^c \cap G_k(t)$ such that,

$$\mu_i \geqslant \hat{\mu}_i(t) - C_{\delta/2n}(t) \geqslant U_t \geqslant \mu_1 - \epsilon > \mu_i,$$

with the last inequality following from the previous assertion, giving a contradiction. □

Claim 1: On $\mathcal{E}_1 \cap \mathcal{E}_2$, for all $k \in \mathbb{N}$, $B_k \subset G_\epsilon^c$.

Proof. Next, we show $B_k \subset G_\epsilon^c$. Suppose not. Either a good arm was added to the bad set by the bad filter or by the good filter. First, consider the case, that the bad filter added an arm in G_ϵ to B_k for some k. By definition, $B_0 = \emptyset$ and $B_{k-1} \subset B_k$ for all k. Then there must exist $k \in \mathbb{N}$ and an $i \in G_\epsilon$ such that $i \in B_k$ and $i \notin B_{k-1}$. Following line 14 of the algorithm, this occurs if and only if

$$\hat{\mu}_{i_k} - \hat{\mu}_i \geqslant \epsilon + 2^{-k+1}.$$

On the event $\mathcal{E}_2$, the above implies

$$\mu_{i_k} - \mu_i + 2^{-k} \geqslant \epsilon + 2^{-k+1},$$

and simplifying, we see that $\epsilon + 2^{-k} \leqslant \mu_{i_k} - \mu_i \leqslant \mu_1 - \mu_i$ which contradicts the assertion that $i \in G_\epsilon$.

Next, consider the case that the good filter incorrectly adds a good arm $i \in G_\epsilon$ to B_k in some round k. Then there must be a $t \in \mathbb{N}$ such that.

$$\mu_i \overset{\mathcal{E}_1}{\leqslant} \hat{\mu}_i + C_{\delta/2n}(t) < L_t \overset{\mathcal{E}_1}{\leqslant} \mu_1 - \epsilon$$

which contradicts $i \in G_\epsilon$. Hence, in both cases $B_k \subset G_\epsilon^c$ for all k. □

Combining the above claims, we see that $\mathcal{E}_1 \cap \mathcal{E}_2$ implies $(G_k \cup B_k = [n])$ and $G_k \cap$

$B_k = \emptyset \implies G_k = G_\epsilon$. Since $\mathbb{P}(\mathcal{E}_1 \cap \mathcal{E}_2) \geqslant 1 - \delta$, if FAREAST terminates, with probability at least $1 - \delta$, it correctly returns the set G_ϵ.

Claim 2: Next, we show that on $\mathcal{E}_1$, $G_\epsilon \subset \mathcal{A}(t) \cup G(t)$ for all $t \in \mathbb{N}$.

In particular this implies that if $\mathcal{A} \subset G$, then $G_\epsilon \subset G$. Combining this with the previous claim gives $G \subset G_\epsilon \subset G$, hence $G = G_\epsilon$. On this condition, FAREAST terminates by line 33 and returns the set $\mathcal{A} \cup G = G$. Note that by definition, $G_\epsilon \subset G_{(\epsilon+\gamma)}$ for all $\gamma \geqslant 0$. Therefore FAREAST terminates correctly on this condition.

Proof. Suppose for contradiction that there exists $i \in G_\epsilon$ such that $i \notin \mathcal{A}(t) \cup G(t)$. This occurs only if i is eliminated in line 28. Hence, there exists a $t' \leqslant t$ such that $\hat{\mu}_i(t') + C_{\delta/n}(t') < L_{t'}$. Therefore, on the event $\mathcal{E}_1$,

$$\mu_1 - \epsilon \overset{\mathcal{E}_1}{\geqslant} L_{t'} = \max_{j \in \mathcal{A}} \hat{\mu}_j(t') - C_{\delta/n}(t') - \epsilon > \hat{\mu}_i(t') + C_{\delta/n}(t') \overset{\mathcal{E}_1}{\geqslant} \mu_i$$

which contradicts $i \in G_\epsilon$. □

Claim 3: Finally, we show that on $\mathcal{E}_1$, if $U_t - L_t \leqslant \gamma/2$, then $\mathcal{A} \cup G \subset G_{(\epsilon+\gamma)}$.

Combining with Claim 3 that $G_\epsilon \subset \mathcal{A} \cup G$, if FAREAST terminates on this condition by line 33, it does so correctly and returns all arms in G_ϵ.

Proof. Assume $U_t - L_t \leqslant \gamma/2$. Since all arms in $\mathcal{A}(t)$ have received exactly t samples, this implies that

$$(\max_{i \in \mathcal{A}(t)} \hat{\mu}_i(t) + C_{\delta/n}(t) - \epsilon) - (\max_{i \in \mathcal{A}(t)} \hat{\mu}_i(t) - C_{\delta/n}(t) - \epsilon) = 2C_{\delta/n}(t) \leqslant \gamma/2.$$

Suppose for contradiction that there exists $i \in G^c_{(\epsilon+\gamma)}$ such that $i \in \mathcal{A} \cup G$. Since $G_\epsilon \cap G^c_{(\epsilon+\gamma)} = \emptyset$ and we have previously shown than $G(t) \subset G_\epsilon$ for all t, we have that $i \in A \backslash G$. Therefore, by the condition in line 27, $\hat{\mu}_i(t) + C_{\delta/n}(t) \geqslant L_t$. Hence, $\mu_i + 2C_{\delta/n}(t) \overset{\mathcal{E}_1}{\geqslant} \hat{\mu}_i(t) + C_{\delta/n}(t) \geqslant L_t$. By assumption, we have that $U_t - \gamma/2 \leqslant L_t$, and the event $\mathcal{E}_1$ implies that $U_t \geqslant \mu_1 - \epsilon$. Therefore, $\mu_i + 2C_{\delta/n}(t) \geqslant U_t - \gamma/2 \geqslant \mu_1 - \epsilon - \gamma/2$. Combining this with the inequality $2C_{\delta/n} \leqslant \gamma/2$, we have that

$$\gamma \geqslant 2C_{\delta/n}(t) + \gamma/2 \geqslant \mu_1 - \epsilon - \mu_i \overset{i \in G^c_{(\epsilon+\gamma)}}{>} \gamma$$

which is a contradiction. □

6.F.3.2 Step 1: An expression for the total number of samples drawn and introducing several helper random variables

Next, we write an expression for the total number of samples drawn by FAREAST. In particular, we introduce two sums that we will spend the remainder of the proof controlling. Additionally, we show that the conditional in line 19 in the good filter is true at least once in each round. Based on this, we more precisely define the random variables T_i and T_i' introduces in the notation section in subsection 6.F.3. Additionally, we introduce the time T_γ at which $U_t - L_t < \frac{1}{2}\gamma$.

Recall that the largest value of t in round k is denoted t_k. Let E_k^γ be the event that $U_t - L_t \geqslant \gamma/2$ for all t in round k:

$$E_k^\gamma := \{U_t - L_t \geqslant \gamma/2 : t \in (t_{k-1}, t_k]\}.$$

Note that if E_{k-1}^γ is false, then FAREAST terminates in round $k-1$ by line 33. We may write the total number of samples drawn by the algorithm as

$$T = \sum_{k=1}^{\infty} 2\mathbb{1}\left[\mathcal{A} \not\subset G_{k-1} \text{ and } G_{k-1} \cup B_{k-1} \neq [n] \text{ and } E_{k-1}^\gamma\right] \left(H_{ME}(n, 2^{-k}, 1/16) + \tau_k + \tau_k|(G_{k-1} \cup B_{k-1})^c|\right)$$

Deterministically, $\mathbb{1}\left[\mathcal{A} \not\subset G_{k-1} \text{ and } G_{k-1} \cup B_{k-1} \neq [n] \text{ and } E_{k-1}^\gamma\right]$
$\leqslant \mathbb{1}\left[G_{k-1} \cup B_{k-1} \neq [n]\right]$
Applying this,

$$\begin{aligned} T &\leqslant \sum_{k=1}^{\infty} 2\mathbb{1}\left[G_{k-1} \cup B_{k-1} \neq [n]\right] \left(H_{ME}(n, 2^{-k}, 1/16) + \tau_k + \tau_k|(G_{k-1} \cup B_{k-1})^c|\right) \\ &= \sum_{k=1}^{\infty} 2\mathbb{1}\left[G_{k-1} \neq G_\epsilon\right] \mathbb{1}\left[G_{k-1} \cup B_{k-1} \neq [n]\right] \\ &\qquad \left(H_{ME}(n, 2^{-k}, 1/16) + \tau_k + \tau_k|(G_{k-1} \cup B_{k-1})^c|\right) \end{aligned} \tag{6.10}$$

$$+\sum_{k=1}^{\infty} 2\mathbb{1}\left[G_{k-1}=G_\epsilon\right]\mathbb{1}\left[G_{k-1}\cup B_{k-1}\neq[n]\right]$$
$$\left(H_{ME}(n,2^{-k},1/16)+\tau_k+\tau_k|(G_{k-1}\cup B_{k-1})^c|\right) \tag{6.11}$$

In round k, line 18 of the Good Filter, whereby an arm is sampled, is evaluated

$$\left(H_{ME}(n,2^{-k},1/16)+\tau_k+\tau_k|(G_{k-1}\cup B_{k-1})^c|\right)\geqslant\left(H_{ME}(n,2^{-k},1/16)+2\tau_k\right)\geqslant n$$

times since $H_{ME}(n,2^{-k},1/16))\geqslant n$ for all k and $|(G_{k-1}\cup B_{k-1})^c|\geqslant 1$ unless $G_{k-1}\cup B_{k-1}=[n]$ which implies termination in round $k-1$. Each time line 18 is called, $N_{I_s}\leftarrow N_{I_s}+1$. Since $|\arg\min_{j\in\mathcal{A}}\{N_j\}|\leqslant|\mathcal{A}|\leqslant n$, line 18 is called at most n times before $\min_{j\in\mathcal{A}}\{N_j\}=\max_{j\in\mathcal{A}}\{N_j\}$. When this occurs, the conditional in line 19 is true and $t\leftarrow t+1$.

If $\min_{i\in\mathcal{A}(t)}\{N_i\}=\max_{i\in\mathcal{A}(t)}\{N_i\}$, then $N_i=t$ for any $i\in\mathcal{A}(t)$. By Step 0, only arms in G_ϵ are added to G_k. Therefore, T_i is defined as

$$T_i=\min\left\{t:\begin{array}{ll} i\in G_k(t+1) & \text{if } i\in G_\epsilon\\ i\notin\mathcal{A}(t+1) & \text{if } i\in G_\epsilon^c\end{array}\right\}\overset{\mathcal{E}_1}{=}\min\left\{t:\begin{array}{ll}\hat{\mu}_i-C_{\delta/2n}(t)\geqslant U_t & \text{if } i\in G_\epsilon\\ \hat{\mu}_i+C_{\delta/2n}(t)\leqslant L_t & \text{if } i\in G_\epsilon^c\end{array}\right\} \tag{6.12}$$

Define $T_i=\infty$ if this never occurs. Note that this may happen if `FAREAST` terminates due to the condititon in line 32 that $U_t-L_t<\gamma/2$. Similarly, recall T_i' denotes the random variable of the of the number of times i is sampled before $\hat{\mu}_i(t)+C_{\delta/2n}(t)\leqslant\max_{j\in\mathcal{A}}\hat{\mu}_j(t)-C_{\delta/2n}(t)$. Hence,

$$T_i'=\min\left\{t:\hat{\mu}_i(t)+C_{\delta/2n}(t)\leqslant\max_{j\in\mathcal{A}(t)}\hat{\mu}_j(t)-C_{\delta/2n}(t)\right\} \tag{6.13}$$

Define $T_i'=\infty$ if this never occurs. Note that this may happen if `FAREAST` terminates due to the condititon in line 32 that $U_t-L_t<\gamma/2$. Finally, we define the time T_γ such that $U_t-L_t<\frac{1}{2}\gamma$.

$$T_\gamma=\min\left\{t:U_t-L_t<\frac{1}{2}\gamma\right\} \tag{6.14}$$

By design, no arm is sampled more that T_γ times by the good filter, controlling the cases that T_i or T_i' are infinite.

6.F.3.3 Step 2: Bounding T_i and T_i' for $i \in G_\epsilon$

Step 2a: For $i \in G_\epsilon$, we have that $T_i \leqslant h(0.25(\epsilon - \Delta_i), \delta/2n)$.
Proof. Note that, $4C_{\delta/2n}(t) \leqslant \mu_i - (\mu_1 - \epsilon)$, true when $t > h\left(0.25(\epsilon - \Delta_i), \frac{\delta}{2n}\right)$, implies that for all j,

$$\begin{aligned}
\hat{\mu}_i(t) - C_{\delta/2n}(t) &\overset{\mathcal{E}_1}{\geqslant} \mu_i - 2C_{\delta/2n}(t) \\
&\geqslant \mu_1 + 2C_{\delta/2n}(t) - \epsilon \\
&\geqslant \mu_j + 2C_{\delta/2n}(t) - \epsilon \\
&\overset{\mathcal{E}_1}{\geqslant} \hat{\mu}_j(t) + C_{\delta/2n}(t) - \epsilon
\end{aligned}$$

so in particular, $\hat{\mu}_i(t) - C_{\delta/2n}(t) \geqslant \max_{j \in \mathcal{A}} \hat{\mu}_j(t) + C_{\delta/2n}(t) - \epsilon = U_t$. □

Additionally, we define a time T_{max} when all good arms have entered G_k.
Step 2b: Defining $T_{max} := \min\{t : G_k(t) = G_\epsilon\} = \max_{i \in G_\epsilon} T_i$, we also have that $T_{max} \leqslant h(0.25\alpha_\epsilon, \delta/2n)$ (in other words, if $t > h(0.25\alpha_\epsilon, \delta/2n)$ (i.e. line 23 has been run t times), then we have that $G_k(t) = G_\epsilon$).
Proof. Recall that $\alpha_\epsilon = \min_{i \in G_\epsilon} \mu_i - \mu_1 + \epsilon = \min_{i \in G_\epsilon} \epsilon - \Delta_i$. By Step 1a, $T_i \leqslant h\left(0.25(\epsilon - \Delta_i), \frac{\delta}{2n}\right)$. Furthermore, $h(\cdot, \cdot)$ is monotonic in its first argument, such that if $0 < x' < x$, then $h(x', \delta) > h(x, \delta)$ for any $\delta > 0$. Therefore $T_{max} = \max_{i \in G_\epsilon} T_i \leqslant \max_{i \in G_\epsilon} h\left(0.25(\epsilon - \Delta_i), \frac{\delta}{2n}\right) = h\left(0.25\alpha_\epsilon, \frac{\delta}{2n}\right)$. □

Step 2c: For $i \in G_\epsilon$, we have that $T_i' \leqslant h(0.25\Delta_i, \delta/2n)$.
Proof. Note that $4C_{\delta/2n}(t) \leqslant \mu_1 - \mu_i$, true when $t > h\left(0.25\Delta_i, \frac{\delta}{2n}\right)$, implies that

$$\begin{aligned}
\hat{\mu}_i(t) + C_{\delta/2n}(t) &\overset{\mathcal{E}_1}{\leqslant} \mu_i + 2C_{\delta/2n}(t) \\
&\leqslant \mu_1 - 2C_{\delta/2n}(t) \\
&\overset{\mathcal{E}_1}{\leqslant} \hat{\mu}_1(t) - C_{\delta/2n}(t).
\end{aligned}$$

As shown in Step 0, $1 \in \mathcal{A}(t)$ for all $t \in \mathbb{N}$, and in particular $\hat{\mu}_1(t) \leqslant \max_{i \in \mathcal{A}(t)} \hat{\mu}_i(t)$. Hence, $\hat{\mu}_i(t) + C_{\delta/2n}(t) \leqslant \max_{j \in \mathcal{A}(t)} \hat{\mu}_j(t) - C_{\delta/2n}(t)$. □

6.F.3.4 Step 3: Bounding T_i for $i \in G_\epsilon^c$

Next, we bound T_i for $i \in G_\epsilon^c$. $i \in G_\epsilon^c$ is eliminated from $\mathcal{A}$ if it has received at least T_i samples.

Claim: $T_i \leqslant h\left(0.25(\epsilon - \Delta_i), \frac{\delta}{2n}\right)$ for $i \in G_\epsilon^c$

Proof. Note that, $4C_{\delta/2n}(t) \leqslant \mu_1 - \epsilon - \mu_i$, true when $t > h\left(0.25(\epsilon - \Delta_i), \frac{\delta}{2n}\right)$, implies that

$$\begin{aligned}
\hat{\mu}_i(t) + C_{\delta/2n}(t) &\overset{\mathcal{E}_1}{\leqslant} \mu_i + 2C_{\delta/2n}(t) \\
&\leqslant \mu_1 - 2C_{\delta/2n}(t) - \epsilon \\
&\overset{\mathcal{E}_1}{\leqslant} \hat{\mu}_1(t) - C_{\delta/2n}(t) - \epsilon
\end{aligned}$$

As shown in Step 0, $1 \in \mathcal{A}(t)$ for all $t \in \mathbb{N}$, and in particular $\hat{\mu}_1(t) \leqslant \max_{i \in \mathcal{A}(t)} \hat{\mu}_i(t)$. Therefore $\hat{\mu}_i(t) + C_{\delta/2n}(t) \leqslant \max_{j \in \mathcal{A}} \hat{\mu}_j(t) - C_{\delta/2n}(t) - \epsilon = L_t$. □

6.F.3.5 Step 4: bounding the total number of samples given to the good filter at time $t = T_{\max}$

Note that for a time $t = T$, the total number of samples given to the good filter is $\sum_{s=1}^{T} |\mathcal{A}(s)|$. Therefore, the total number of samples up to time $T_{\max}$ is $\sum_{t=1}^{T_{\max}} |\mathcal{A}(t)|$.

Let $S_i = \min\{t : i \notin A(t+1)\}$. Hence,

$$\sum_{t=1}^{T_{\max}} |\mathcal{A}(t)| = \sum_{t=1}^{T_{\max}} \sum_{i=1}^{n} \mathbb{1}[i \in \mathcal{A}(t)] = \sum_{i=1}^{n} \sum_{t=1}^{T_{\max}} \mathbb{1}[i \in \mathcal{A}(t)] = \sum_{i=1}^{n} \min\{T_{\max}, S_i\}$$

For arms $i \in G_\epsilon^c$, $S_i = T_i$ by definition. For $i \in G_\epsilon$, $S_i = \max(T_i, T_i')$ by line 28 of the algorithm. Then

$$\begin{aligned}
\sum_{i=1}^{n} \min\{T_{max}, S_i\} &= \sum_{i \in G_\epsilon} \min\{\mathcal{T}_{max}, \max(T_i, T_i')\} + \sum_{i \in G_\epsilon^c} \min\{T_{max}, T_i\} \\
&\leqslant \sum_{i \in G_\epsilon} \min\{T_{max}, \max(T_i, T_i')\} + |G_\epsilon^c \cap G_{\epsilon+\alpha_\epsilon}| T_{max} + \sum_{i \in G_{\epsilon+\alpha_\epsilon}^c} T_i \\
&= \sum_{i \in G_\epsilon} \max\{T_i, \min(T_i', T_{max})\} + |G_\epsilon^c \cap G_{\epsilon+\alpha_\epsilon}| T_{max} + \sum_{i \in G_{\epsilon+\alpha_\epsilon}^c} T_i \\
&\overset{(a)}{\leqslant} \sum_{i \in G_\epsilon} \max\left\{ h\left(0.25(\epsilon - \Delta_i), \frac{\delta}{2n}\right), \min\left[h\left(0.25\Delta_i, \frac{\delta}{2n}\right), h\left(0.25\alpha_\epsilon, \frac{\delta}{2n}\right)\right]\right\} \\
&\quad + \sum_{i \in G_{\epsilon+\alpha_\epsilon}^c} h\left(0.25(\epsilon - \Delta_i), \frac{\delta}{2n}\right) + |G_\epsilon^c \cap G_{\epsilon+\alpha_\epsilon}| h\left(0.25\alpha_\epsilon, \frac{\delta}{2n}\right).
\end{aligned}$$

Equality (a) follows from $T_{max} \leqslant h\left(0.25\alpha_\epsilon, \frac{\delta}{2n}\right)$ by Step 1b, $T_i \leqslant h\left(0.25(\epsilon - \Delta_i), \frac{\delta}{2n}\right)$ in Steps 2a and 3, and $T_i' \leqslant h\left(0.25\Delta_i, \frac{\delta}{2n}\right)$ in Step 2c.

6.F.3.6 Step 5: Bounding the number of samples in round k versus $k-1$

Now we show that the total number of samples taken in round k is no more than 9 times the number taken in the previous round.

Claim: For $k > 1$

$$\begin{aligned}
&\left(H_{ME}(n, 2^{-k}, 1/16) + \tau_k + \tau_k |(G_{k-1} \cup B_{k-1})^c|\right) \\
&\leqslant 9\left(H_{ME}(n, 2^{-k+1}, 1/16) + \tau_{k-1} + \tau_{k-1} |(G_{k-2} \cup B_{k-2})^c|\right)
\end{aligned}$$

Proof. In round k, $\left(H_{ME}(n, 2^{-k}, 1/16) + \tau_k + \tau_k |(G_{k-1} \cup B_{k-1})^c|\right)$ samples are drawn. Since $G_{k-1} \subset G_k$ and $B_{k-1} \subset B_k$ $\forall k$ deterministically, we see that $|(G_{k-1} \cup B_{k-1})^c| \geqslant |(G_k \cup B_k)^c|$ $\forall k$. By definition,
$H_{ME}(n, 2^{-k-1}, 1/16) = 4H_{ME}(n, 2^{-k}, 1/16)$.

Next, recall $\tau_k = \left\lceil 2^{2k+3} \log\left(\frac{8}{\delta_k}\right)\right\rceil$. We bound τ_k/τ_{k-1} as

$$\begin{aligned}
\frac{\tau_k}{\tau_{k-1}} &= \frac{\left\lceil 2^{2k+3} \log\left(\frac{8}{\delta_k}\right)\right\rceil}{\left\lceil 2^{2k+1} \log\left(\frac{8}{\delta_{k-1}}\right)\right\rceil} = \frac{\left\lceil 2^{2k+3} \log\left(\frac{16nk^2}{\delta}\right)\right\rceil}{\left\lceil 2^{2k+1} \log\left(\frac{16n(k-1)^2}{\delta}\right)\right\rceil} \\
&\leqslant \frac{2^{2k+3} \log\left(\frac{16nk^2}{\delta}\right) + 1}{2^{2k+1} \log\left(\frac{16n(k-1)^2}{\delta}\right)} \leqslant \frac{4 \log\left(\frac{16nk^2}{\delta}\right)}{\log\left(\frac{16n(k-1)^2}{\delta}\right)} + 1 \\
&\leqslant 4 \frac{\log\left(\frac{16n}{\delta}\right) + 2\log(k)}{\log\left(\frac{16n}{\delta}\right) + 2\log(k-1)} + 1 = (*)
\end{aligned}$$

If $k = 2$, $(*) \leqslant 1 + 4 * \log(32)/\log(8) \leqslant 9$. Otherwise,

$$\begin{aligned}
(*) &= \frac{4(\log\left(\frac{16n}{\delta}\right) + 2\log(k))}{\log\left(\frac{16n}{\delta}\right) + 2\log(k-1)} + 1 \\
&\leqslant \frac{4\log(k)}{\log(k-1)} + 1 \\
&\leqslant 4 \cdot 2 + 1 = 9
\end{aligned}$$

Putting these pieces together,

$$\begin{aligned}
&\left(H_{ME}(n, 2^{-k}, 1/16) + \tau_k + \tau_k |(G_{k-1} \cup B_{k-1})^c|\right) \\
&\leqslant \left(4H_{ME}(n, 2^{-k+1}, 1/16) + 9\tau_{k-1} + 9\tau_{k-1} |(G_{k-2} \cup B_{k-2})^c|\right) \\
&\leqslant 9\left(H_{ME}(n, 2^{-k+1}, 1/16) + \tau_{k-1} + \tau_{k-1} |(G_{k-2} \cup B_{k-2})^c|\right)
\end{aligned}$$

□

6.F.3.7 Step 6: Bounding Equation (6.10)

Here, we introduce the round K_{Good}, when $G_{K_{Good}} = G_\epsilon$ at some point within the round. Using the result of the previous step, we may bound the total number of samples taken though this round, controlling Equation (6.10).

With the result of Step 5, we prove the following inequality.

Claim:

$$\sum_{k=1}^{\infty} 2\mathbb{1}\left[G_{k-1} \neq G_\epsilon\right] \mathbb{1}\left[G_{k-1} \cup B_{k-1} \neq [n]\right] \left(H_{ME}(n, 2^{-k}, 1/16) + \tau_k + \tau_k|(G_{k-1} \cup B_{k-1})^c|\right) \tag{6.15}$$

$$\leqslant c \sum_{i \in G_\epsilon} \max\left\{h\left(0.25(\epsilon - \Delta_i), \frac{\delta}{2n}\right), \min\left[h\left(0.25\Delta_i, \frac{\delta}{2n}\right), h\left(0.25\alpha_\epsilon, \frac{\delta}{2n}\right)\right]\right\}$$
$$+ c|G_\epsilon^c \cap G_{\epsilon+\alpha_\epsilon}| h\left(0.25\alpha_\epsilon, \frac{\delta}{2n}\right) + c \sum_{i \in G_{\epsilon+\alpha_\epsilon}^c} h\left(0.25(\epsilon - \Delta_i), \frac{\delta}{2n}\right)$$

for a constant c.

Proof. Recall $t_k = \max\{t : t \in k\}$ denotes the maximum value of t in round k and $T_{max} = \max_{i \in G_\epsilon} T_i$ denotes the minimum t such that $G_k(t) = G_\epsilon$. Define the random round

$$K_{Good} := \min\{k : G_k = G_\epsilon\} = \min\{k : t_k \geqslant T_{max}\}$$

By definition of K_{Good},

$$\sum_{k=1}^{\infty} 2\mathbb{1}[G_{k-1} \neq G_\epsilon]\mathbb{1}\left[G_{k-1} \cup B_{k-1} \neq [n]\right] \left(H_{ME}(n, 2^{-k}, 1/16) + \tau_k + \tau_k|(G_{k-1} \cup B_{k-1})^c|\right)$$
$$= \sum_{k=1}^{K_{Good}} 2\mathbb{1}\left[G_{k-1} \cup B_{k-1} \neq [n]\right] \left(H_{ME}(n, 2^{-k}, 1/16) + \tau_k + \tau_k|(G_{k-1} \cup B_{k-1})^c|\right).$$

Next, applying Step 5, if $K_{Good} > 1$,

$$\sum_{k=1}^{K_{Good}} 2\mathbb{1}\left[G_{k-1} \cup B_{k-1} \neq [n]\right] \left(H_{ME}(n, 2^{-k}, 1/16) + \tau_k + \tau_k|(G_{k-1} \cup B_{k-1})^c|\right)$$
$$\leqslant 18 \sum_{k=1}^{K_{Good}-1} \mathbb{1}\left[G_{k-1} \cup B_{k-1} \neq [n]\right] \left(H_{ME}(n, 2^{-k}, 1/16) + \tau_k + \tau_k|(G_{k-1} \cup B_{k-1})^c|\right).$$

Observe that by lines 17 and 20 of FAREAST, for any round r and for any $t > t_{r-1}$,

$$\sum_{k=1}^{r-1} \mathbb{1}\left[G_{k-1} \cup B_{k-1} \neq [n]\right] \left(H_{ME}(n, 2^{-k}, 1/16) + \tau_k + \tau_k |(G_{k-1} \cup B_{k-1})^c|\right) \leqslant \sum_{s=1}^{t} |\mathcal{A}(s)|.$$

By definition, for the round $K_{Good} - 1$, we see that $t_{(K_{Good}-1)} < T_{max}$. Applying the above inequality with the inequality proven in Step 4,

$$\begin{aligned} 18 \sum_{k=1}^{K_{Good}-1} \mathbb{1}\left[G_{k-1} \cup B_{k-1} \neq [n]\right] \left(H_{ME}(n, 2^{-k}, 1/16) + \tau_k + \tau_k |(G_{k-1} \cup B_{k-1})^c|\right) &\leqslant 18 \sum_{s=1}^{T_{max}} |\mathcal{A}(s)| \\ \leqslant 18 \sum_{i \in G_\epsilon} \max\left\{h\left(0.25(\epsilon - \Delta_i), \frac{\delta}{2n}\right), \min\left[h\left(0.25\Delta_i, \frac{\delta}{2n}\right), h\left(0.25\alpha_\epsilon, \frac{\delta}{2n}\right)\right]\right\} \\ + 18 \sum_{i \in G^c_{\epsilon+\alpha_\epsilon}} h\left(0.25(\epsilon - \Delta_i), \frac{\delta}{2n}\right) + 18|G^c_\epsilon \cap G_{\epsilon+\alpha_\epsilon}| h\left(0.25\alpha_\epsilon, \frac{\delta}{2n}\right). \end{aligned}$$

Otherwise, if $K_{Good} = 1$, exactly $4c'n\log(16) + 32n\log(16n/\delta)$ samples are given to the good filter in round 1. One may use Lemma 6.32 to invert $h(\cdot, \cdot)$ and show that the summation on the right had side of the above inequality is within a constant of this and the claim holds in this case as well for a different constant, potentially larger than 18. □

6.F.3.8 Step 7: Bounding Equation (6.11)

Next, we bound

$$\sum_{k=1}^{\infty} 2\mathbb{1}\left[G_{k-1} = G_\epsilon\right] \mathbb{1}\left[G_{k-1} \cup B_{k-1} \neq [n]\right] \left(H_{ME}(n, 2^{-k}, 1/16) + \tau_k + \tau_k |(G_{k-1} \cup B_{k-1})^c|\right).$$

$$\sum_{k=1}^{\infty} 2\mathbb{1}\left[G_{k-1} = G_\epsilon\right] \mathbb{1}\left[G_{k-1} \cup B_{k-1} \neq [n]\right] \left(H_{ME}(n, 2^{-k}, 1/16) + \tau_k + \tau_k |(G_{k-1} \cup B_{k-1})^c|\right)$$

$$
\begin{aligned}
&= \sum_{k=1}^{\infty} 2\mathbb{1}\left[G_{k-1} = G_\epsilon\right] \mathbb{1}\left[G_{k-1} \cup B_{k-1} \neq [n]\right] \left(H_{ME}(n, 2^{-k}, 1/16) + \tau_k + \tau_k |(G_\epsilon \cup B_{k-1})^c|\right) \\
&= \sum_{k=1}^{\infty} 2\mathbb{1}\left[G_{k-1} = G_\epsilon\right] \mathbb{1}\left[G_{k-1} \cup B_{k-1} \neq [n]\right] \left(H_{ME}(n, 2^{-k}, 1/16) + \tau_k + \tau_k |G_\epsilon^c \backslash B_{k-1}|\right) \\
&= \sum_{k=K_{Good}+1}^{\infty} 2\mathbb{1}\left[G_{k-1} \cup B_{k-1} \neq [n]\right] \left(H_{ME}(n, 2^{-k}, 1/16) + \tau_k + \tau_k |G_\epsilon^c \backslash B_{k-1}|\right) \\
&\overset{\mathcal{E}_1, \mathcal{E}_2}{=} \sum_{k=K_{Good}+1}^{\infty} 2\mathbb{1}\left[B_{k-1} \neq G_\epsilon^c\right] \left(H_{ME}(n, 2^{-k}, 1/16) + \tau_k + \tau_k |G_\epsilon^c \backslash B_{k-1}|\right) \\
&= \sum_{k=K_{Good}+1}^{\infty} 2\mathbb{1}\left[B_{k-1} \neq G_\epsilon^c\right] \left(H_{ME}(n, 2^{-k}, 1/16) + \tau_k\right) \\
&\qquad + \sum_{k=K_{Good}+1}^{\infty} 2\mathbb{1}\left[B_{k-1} \neq G_\epsilon^c\right] \left(\tau_k |G_\epsilon^c \backslash B_{k-1}|\right) \\
&= \sum_{k=K_{Good}+1}^{\infty} 2\mathbb{1}\left[B_{k-1} \neq G_\epsilon^c\right] \left(H_{ME}(n, 2^{-k}, 1/16) + \tau_k\right) + \sum_{k=K_{Good}+1}^{\infty} 2\tau_k |G_\epsilon^c \backslash B_{k-1}| \\
&= \sum_{k=K_{Good}+1}^{\infty} 2\mathbb{1}\left[B_{k-1} \neq G_\epsilon^c\right] \left(H_{ME}(n, 2^{-k}, 1/16) + \tau_k\right) + \sum_{k=K_{Good}+1}^{\infty} \sum_{i \in G_\epsilon^c} 2\tau_k \mathbb{1}[i \notin B_{k-1}] \\
&\leqslant \sum_{k=K_{Good}+1}^{\infty} 2|G_\epsilon^c \backslash B_{k-1}| \left(H_{ME}(n, 2^{-k}, 1/16) + \tau_k\right) + \sum_{k=K_{Good}+1}^{\infty} \sum_{i \in G_\epsilon^c} 2\tau_k \mathbb{1}[i \notin B_{k-1}] \\
&= \sum_{k=K_{Good}+1}^{\infty} \sum_{i \in G_\epsilon^c} 2\mathbb{1}[i \notin B_{k-1}] \left(H_{ME}(n, 2^{-k}, 1/16) + \tau_k\right) + \sum_{k=K_{Good}+1}^{\infty} \sum_{i \in G_\epsilon^c} 2\tau_k \mathbb{1}[i \notin B_{k-1}] \\
&= \sum_{k=K_{Good}+1}^{\infty} \sum_{i \in G_\epsilon^c} 2\mathbb{1}[i \notin B_{k-1}] \left(2\tau_k + H_{ME}(n, 2^{-k}, 1/16)\right) \\
&= \sum_{i \in G_\epsilon^c} \sum_{k=K_{Good}+1}^{\infty} 2\mathbb{1}[i \notin B_{k-1}] \left(2\tau_k + H_{ME}(n, 2^{-k}, 1/16)\right) \\
&\leqslant \sum_{i \in G_\epsilon^c} \sum_{k=1}^{\infty} 2\mathbb{1}[i \notin B_{k-1}] \left(2\tau_k + H_{ME}(n, 2^{-k}, 1/16)\right) \qquad (6.16)
\end{aligned}
$$

6.F.3.9 Step 8: Bounding the expected total number of samples drawn by FAREAST

Now we take expectations over the number of samples drawn. These expectations are conditional on the high probability event $\mathcal{E}_1 \cap \mathcal{E}_2$. The bound in step 5 holds deterministically conditioned on this event.

Note τ_k and $H_{ME}(n, 2^{-k}, 1/16)$ are deterministic constants for any k. Let all expectations are be jointly over the random instance ν and the randomness in FAREAST.

$$
\begin{aligned}
&\mathbb{E}[T|\mathbb{1}[\mathcal{E}_1 \cap \mathcal{E}_2] = 1] \leqslant \\
&\quad \sum_{k=1}^{\infty} 2\mathbb{E}\left[\mathbb{1}[G_k \cup B_k \neq [n]] \middle| \mathbb{1}[\mathcal{E}_1 \cap \mathcal{E}_2] = 1\right] \left(\tau_k + H_{ME}(n, 2^{-k}, 1/16) + \tau_k \middle| (G_{k-1} \cup B_{k-1})^c \middle|\right) \\
&\qquad = \sum_{k=1}^{\infty} 2\mathbb{E}\left[\mathbb{1}\left[G_{k-1} \neq G_\epsilon\right] \mathbb{1}[G_{k-1} \cup B_{k-1} \neq [n]] \middle| \mathbb{1}[\mathcal{E}_1 \cap \mathcal{E}_2] = 1\right] \\
&\qquad\qquad\qquad \left(\tau_k + H_{ME}(n, 2^{-k}, 1/16) + \tau_k \middle| (G_{k-1} \cup B_{k-1})^c \middle|\right) \\
&\qquad\quad + \sum_{k=1}^{\infty} 2\mathbb{E}\left[\mathbb{1}\left[G_{k-1} = G_\epsilon\right] \mathbb{1}[G_{k-1} \cup B_{k-1} \neq [n]] \middle| \mathbb{1}[\mathcal{E}_1 \cap \mathcal{E}_2] = 1\right] \\
&\qquad\qquad\qquad \left(\tau_k + H_{ME}(n, 2^{-k}, 1/16) + \tau_k \middle| (G_{k-1} \cup B_{k-1})^c \middle|\right) \\
&\qquad \overset{\text{Step 6}}{\leqslant} c \sum_{i \in G_\epsilon} \max\left\{h\left(0.25(\epsilon - \Delta_i), \frac{\delta}{2n}\right), \min\left[h\left(0.25\Delta_i, \frac{\delta}{2n}\right), h\left(0.25\alpha_\epsilon, \frac{\delta}{2n}\right)\right]\right\} \\
&\qquad\quad + c \sum_{i \in G^c_{\epsilon+\alpha_\epsilon}} h\left(0.25(\epsilon - \Delta_i), \frac{\delta}{2n}\right) + c|G^c_\epsilon \cap G_{\epsilon+\alpha_\epsilon}| h\left(0.25\alpha_\epsilon, \frac{\delta}{2n}\right) \\
&\qquad\quad + \sum_{k=1}^{\infty} 2\mathbb{E}\left[\mathbb{1}\left[G_{k-1} = G_\epsilon\right] \mathbb{1}[G_{k-1} \cup B_{k-1} \neq [n]] \middle| \mathbb{1}[\mathcal{E}_1 \cap \mathcal{E}_2] = 1\right] \\
&\qquad\qquad\qquad \left(\tau_k + H_{ME}(n, 2^{-k}, 1/16) + \tau_k \middle| (G_{k-1} \cup B_{k-1})^c \middle|\right) \\
&\qquad \overset{\text{Step 7}}{\leqslant} c \sum_{i \in G_\epsilon} \max\left\{h\left(0.25(\epsilon - \Delta_i), \frac{\delta}{2n}\right), \min\left[h\left(0.25\Delta_i, \frac{\delta}{2n}\right), h\left(0.25\alpha_\epsilon, \frac{\delta}{2n}\right)\right]\right\}
\end{aligned}
$$

$$
\begin{aligned}
&+ c \sum_{i \in G_{\epsilon+\alpha_\epsilon}^c} h\left(0.25(\epsilon - \Delta_i), \frac{\delta}{2n}\right) + c|G_\epsilon^c \cap G_{\epsilon+\alpha_\epsilon}| h\left(0.25\alpha_\epsilon, \frac{\delta}{2n}\right) \\
&+ \sum_{i \in G_\epsilon^c} \sum_{k=1}^{\infty} 2\mathbb{E}_\nu\left[\mathbb{1}[i \notin B_{k-1}] \middle| \mathbb{1}[\mathcal{E}_1 \cap \mathcal{E}_2] = 1\right] \left(2\tau_k + H_{ME}(n, 2^{-k}, 1/16)\right) \\
&\overset{(a)}{=} c \sum_{i \in G_\epsilon} \max\left\{h\left(0.25(\epsilon - \Delta_i), \frac{\delta}{2n}\right), \min\left[h\left(0.25\Delta_i, \frac{\delta}{2n}\right), h\left(0.25\alpha_\epsilon, \frac{\delta}{2n}\right)\right]\right\} \\
&+ c \sum_{i \in G_{\epsilon+\alpha_\epsilon}^c} h\left(0.25(\epsilon - \Delta_i), \frac{\delta}{2n}\right) + c|G_\epsilon^c \cap G_{\epsilon+\alpha_\epsilon}| h\left(0.25\alpha_\epsilon, \frac{\delta}{2n}\right) \\
&+ \sum_{i \in G_\epsilon^c} \sum_{k=1}^{\infty} 2\mathbb{E}_\nu\left[\mathbb{1}[i \notin B_{k-1}] \middle| \mathcal{E}_1\right] \left(2\tau_k + H_{ME}(n, 2^{-k}, 1/16)\right)
\end{aligned}
$$

where (a) follows from $\mathbb{E}_\nu\left[\mathbb{1}[i \notin B_{k-1}] \middle| \mathcal{E}_1 \cap \mathcal{E}_2\right] = \mathbb{E}_\nu\left[\mathbb{1}[i \notin B_{k-1}] \middle| \mathcal{E}_1\right]$ for $i \in G_\epsilon^c$, since the event $\{i \in B_{k-1}\}$ is independent of $\mathcal{E}_2$ for all $i \in G_\epsilon^c$. This can be observed since $\mathcal{E}_2$ deals only with independent samples taken of arms in G_ϵ.

6.F.3.10 Step 9: Bounding the expectation remaining from step 8.

Next, we bound $\sum_{k=1}^{\infty} \mathbb{E}_\nu\left[\mathbb{1}[i \notin B_{k-1}] \middle| \mathcal{E}_1\right] \left(2\tau_k + H_{ME}(n, 2^{-k}, 1/16)\right)$ for $i \in G_\epsilon^c$, the expectation remaining from step 8. In particular, this is the number of samples drawn by the bad filter to add arm $i \in G_\epsilon^c$ to B_k.

First, we bound the probability that for a given $i \in G_\epsilon^c$ and a given k $i \notin B_k$. Note that by Borel-Cantelli, this implies that the probability that i is never added to any B_k is 0.

Claim 1: For $i \in G_\epsilon^c$, $k \geqslant \left\lceil \log_2\left(\frac{4}{\Delta_i - \epsilon}\right)\right\rceil \implies \mathbb{E}_\nu\left[\mathbb{1}[i \notin B_k] \middle| \mathcal{E}_1\right] \leqslant \left(\frac{1}{8}\right)^{k - \left\lceil \log_2\left(\frac{4}{\Delta_i - \epsilon}\right)\right\rceil}$

Proof. $i \in B_k$ if either the good filter or the bad filter added it. Note that the behavior of the bad filter is independent of the event $\mathcal{E}_1$. Hence,

$$
\begin{aligned}
\mathbb{E}_\nu\left[\mathbb{1}[i \notin B_k] \middle| \mathcal{E}_1\right] &= \mathbb{E}_\nu\left[\mathbb{1}[\hat{\mu}_i + C_{\delta/2n}(t_k) \geqslant L_{t_k}] \mathbb{1}[\hat{\mu}_{i_k} - \hat{\mu}_i < \epsilon + 2^{-k+1}] \middle| \mathcal{E}_1\right] \\
&\leqslant \mathbb{E}_\nu\left[\mathbb{1}[\hat{\mu}_{i_k} - \hat{\mu}_i < \epsilon + 2^{-k+1}] \middle| \mathcal{E}_1\right] \\
&= \mathbb{E}_\nu\left[\mathbb{1}[\hat{\mu}_{i_k} - \hat{\mu}_i < \epsilon + 2^{-k+1}]\right].
\end{aligned}
$$

Intuitively, the time at which an arm in G_ϵ^c enters B_k, which occurs if either the good filter adds it or the bad filter does, in expectation is at most the time at which the bad filter does on its own in expectation.

If $i \in B_{k-1}$ then $i \in B_k$ by definition. Otherwise, if $i \notin B_{k-1}$, by Hoeffding's Inequality conditional on the value of i_k and a sum over conditional probabilities as in step 0, with probability at least $1 - \frac{\delta}{4nk^2}$

$$|(\hat{\mu}_{i_k} - \hat{\mu}_i) - (\mu_{i_k} - \mu_i)| \leqslant 2^{-k}$$

If `MedianElimination` also succeeds, the joint event of which occurs with probability $\frac{15}{16}\left(1 - \frac{\delta}{4nk^2}\right)$ by independence[6],

$$\hat{\mu}_{i_k} - \hat{\mu}_i \geqslant \mu_{i_k} - \mu_i - 2^{-k} \geqslant \mu_1 - \mu_i - 2^{-k+1} = \Delta_i - 2^{-k+1}.$$

Then for $k \geqslant \left\lceil \log_2\left(\frac{4}{\Delta_i - \epsilon}\right)\right\rceil$,

$$\hat{\mu}_{i_k} - \hat{\mu}_i \geqslant \Delta_i - 2^{-k+1} \geqslant \frac{1}{2}(\Delta_i + \epsilon) \geqslant \epsilon + 2^{-k+1},$$

which implies that $i \in B_k$ by line 15 of `FAREAST`. In particular,

$$\mathbb{E}\left[\mathbb{1}[\hat{\mu}_{i_k} - \hat{\mu}_i \geqslant \epsilon + 2^{-k+1}] \middle| i \notin B_{k-1}\right] \geqslant \frac{15}{16}\left(1 - \frac{\delta}{4nk^2}\right).$$

Furthermore, $i \notin B_0$ by definition. Additionally, recall that $\mathbb{1}[\hat{\mu}_{i_k} - \hat{\mu}_i < \epsilon + 2^{-k+1}]$ is independent of $\mathcal{E}_1$. Then for $k \geqslant \left\lceil \log_2\left(\frac{4}{\Delta_i - \epsilon}\right)\right\rceil$,

$$\begin{aligned}
\mathbb{E}\left[\mathbb{1}[\hat{\mu}_{i_k} - \hat{\mu}_i < \epsilon + 2^{-k+1}]\right] &= \mathbb{E}\left[\mathbb{1}[\hat{\mu}_{i_k} - \hat{\mu}_i < \epsilon + 2^{-k+1}](\mathbb{1}[i \notin B_{k-1}] + \mathbb{1}[i \in B_{k-1}]) \middle| \mathcal{E}_1\right] \\
&= \mathbb{E}\left[\mathbb{1}[\hat{\mu}_{i_k} - \hat{\mu}_i < \epsilon + 2^{-k+1}]\mathbb{1}[i \notin B_{k-1}] \middle| \mathcal{E}_1\right] \\
&\quad + \mathbb{E}\left[\mathbb{1}[\hat{\mu}_{i_k} - \hat{\mu}_i < \epsilon + 2^{-k+1}]\mathbb{1}[i \in B_{k-1}] \middle| \mathcal{E}_1\right]
\end{aligned}$$

[6]Note that the success of `MedianElimination` and the concentration of $(\hat{\mu}_{i_k} - \hat{\mu}_i)$ around $(\mu_{i_k} - \mu_i)$ are independent of the events $\mathcal{E}_1$ and $\mathcal{E}_2$ conditioned on in Step 8.

Deterministically, $\mathbb{1}[i \notin B_k]\mathbb{1}[i \in B_{k-1}] = 0$. Therefore,

$$\begin{aligned}
&\mathbb{E}\left[\mathbb{1}[\hat{\mu}_{i_k} - \hat{\mu}_i < \epsilon + 2^{-k+1}]\mathbb{1}[i \notin B_{k-1}]\middle|\mathcal{E}_1\right] \\
&\qquad + \mathbb{E}\left[\mathbb{1}[\hat{\mu}_{i_k} - \hat{\mu}_i < \epsilon + 2^{-k+1}]\mathbb{1}[i \in B_{k-1}]\middle|\mathcal{E}_1\right] \\
&= \mathbb{E}\left[\mathbb{1}[\hat{\mu}_{i_k} - \hat{\mu}_i < \epsilon + 2^{-k+1}]\mathbb{1}[i \notin B_{k-1}]\middle|\mathcal{E}_1\right] \\
&= \mathbb{E}\left[\mathbb{1}[\hat{\mu}_{i_k} - \hat{\mu}_i < \epsilon + 2^{-k+1}]\mathbb{1}[i \notin B_{k-1}]|i \notin B_{k-1}\mathcal{E}_1\right]\mathbb{P}(i \notin B_{k-1}\middle|\mathcal{E}_1) \\
&\qquad + \mathbb{E}\left[\mathbb{1}[\hat{\mu}_{i_k} - \hat{\mu}_i < \epsilon + 2^{-k+1}]\mathbb{1}[i \notin B_{k-1}]\middle|i \in B_{k-1}, \mathcal{E}_1\right]\mathbb{P}(i \in B_{k-1}\middle|\mathcal{E}_1) \\
&= \mathbb{E}\left[\mathbb{1}[\hat{\mu}_{i_k} - \hat{\mu}_i < \epsilon + 2^{-k+1}]\mathbb{1}[i \notin B_{k-1}]|i \notin B_{k-1}, \mathcal{E}_1\right]\mathbb{P}(i \notin B_{k-1}\middle|\mathcal{E}_1) \\
&= \mathbb{E}\left[\mathbb{1}[\hat{\mu}_{i_k} - \hat{\mu}_i < \epsilon + 2^{-k+1}]\middle|i \notin B_{k-1}, \mathcal{E}_1\right]\mathbb{E}\left[\mathbb{1}[i \notin B_{k-1}]\middle|\mathcal{E}_1\right] \\
&= \mathbb{E}\left[\mathbb{1}[\hat{\mu}_{i_k} - \hat{\mu}_i < \epsilon + 2^{-k+1}]\middle|i \notin B_{k-1}\right]\mathbb{E}\left[\mathbb{1}[i \notin B_{k-1}]\middle|\mathcal{E}_1\right] \\
&\leqslant \left(\frac{1}{16} + \frac{\delta}{4nk^2}\right)\mathbb{E}\left[\mathbb{1}[i \notin B_{k-1}]\middle|\mathcal{E}_1\right] \\
&\leqslant \left(\frac{1}{16} + \frac{\delta}{4nk^2}\right)\mathbb{E}\left[\mathbb{1}[\hat{\mu}_{i_k} - \hat{\mu}_i < \epsilon + 2^{-k+2}]\right]
\end{aligned}$$

where the final inequality follows by the same argument upper bounding $\mathbb{E}\left[\mathbb{1}[i \notin B_k]\middle|\mathcal{E}_1\right]$. For $k < \left\lceil \log_2\left(\frac{4}{\Delta_i - \epsilon}\right)\right\rceil$, trivially, $\mathbb{E}\left[\mathbb{1}[i \notin B_k]\right] \leqslant 1$. Recall $\delta \leqslant 1/8$. For $k \geqslant \left\lceil \log_2\left(\frac{4}{\Delta_i - \epsilon}\right)\right\rceil$,

$$\mathbb{E}\left[\mathbb{1}[i \notin B_k]\middle|\mathcal{E}_1\right] \leqslant \prod_{s=\left\lceil \log_2\left(\frac{4}{\Delta_i - \epsilon}\right)\right\rceil}^{k}\left(\frac{1}{16} + \frac{\delta}{2ns^2}\right) \leqslant \left(\frac{1}{8}\right)^{k - \left\lceil \log_2\left(\frac{4}{\Delta_i - \epsilon}\right)\right\rceil}.$$

□

Claim 2: For $j \in G_\epsilon^c$, $\sum_{k=1}^{\infty} 2\mathbb{E}_\nu\left[\mathbb{1}[i \notin B_{k-1}]\middle|\mathcal{E}_1\right]\left(2\tau_k + H_{ME}(n, 2^{-k}, 1/16)\right) \leqslant c''\frac{n}{(\Delta_i - \epsilon)^2} + c''h\left(0.25(\Delta_i - \epsilon), \frac{\delta}{2n}\right)$

Proof. This sum decomposes into two terms.

$$\sum_{k=1}^{\infty}\mathbb{E}_\nu\left[\mathbb{1}[i \notin B_{k-1}]\middle|\mathcal{E}_1\right]\left(2\tau_k + H_{ME}(n, 2^{-k}, 1/16)\right)$$

$$
\begin{aligned}
&= \sum_{k=1}^{\left\lfloor \log_2\left(\frac{4}{\Delta_i-\epsilon}\right)\right\rfloor} \mathbb{E}_\nu\left[\mathbb{1}[i \notin B_{k-1}] \middle| \mathcal{E}_1\right]\left(H_{ME}(n, 2^{-k}, 1/16) + 2\left\lceil 2^{2k+3}\log\left(\frac{16nk^2}{\delta}\right)\right\rceil\right) \\
&+ \sum_{k=\left\lceil \log_2\left(\frac{4}{\Delta_i-\epsilon}\right)\right\rceil}^{\infty} \mathbb{E}_\nu\left[\mathbb{1}[i \notin B_{k-1}] \middle| \mathcal{E}_1\right]\left(H_{ME}(n, 2^{-k}, 1/16) + 2\left\lceil 2^{2k+3}\log\left(\frac{16nk^2}{\delta}\right)\right\rceil\right)
\end{aligned}
$$

We begin by bounding the first term.

$$
\begin{aligned}
&\sum_{k=1}^{\left\lfloor \log_2\left(\frac{4}{\Delta_i-\epsilon}\right)\right\rfloor} \mathbb{E}_\nu\left[\mathbb{1}[i \notin B_{k-1}]\right]\left(H_{ME}(n, 2^{-k}, 1/16) + 2\left\lceil 2^{2k+3}\log\left(\frac{16nk^2}{\delta}\right)\right\rceil\right) \\
&\qquad \leqslant \sum_{k=1}^{\left\lfloor \log_2\left(\frac{4}{\Delta_i-\epsilon}\right)\right\rfloor}\left(H_{ME}(n, 2^{-k}, 1/16) + 2\left\lceil 2^{2k+3}\log\left(\frac{16nk^2}{\delta}\right)\right\rceil\right) \\
&\qquad \leqslant \sum_{k=1}^{\left\lfloor \log_2\left(\frac{4}{\Delta_i-\epsilon}\right)\right\rfloor}\left(c' n 2^{2k}\log(16) + 2 + 2^{2k+4}\log\left(\frac{16nk^2}{\delta}\right)\right) \\
&\qquad \leqslant 2\log_2\left(\frac{4}{\Delta_i-\epsilon}\right) + \left(c' n\log(16) + 16\log\left(\frac{16n}{\delta}\right)\right)\sum_{k=1}^{\left\lfloor \log_2\left(\frac{4}{\Delta_i-\epsilon}\right)\right\rfloor} 2^{2k} \\
&\qquad\qquad + 32\sum_{k=1}^{\left\lfloor \log_2\left(\frac{4}{\Delta_i-\epsilon}\right)\right\rfloor} 2^{2k}\log(k) \\
&\qquad \leqslant 2\log_2\left(\frac{4}{\Delta_i-\epsilon}\right) \\
&\qquad\qquad + \left(c' n\log(16) + 16\log\left(\frac{16n}{\delta}\right) + 32\log\log_2\left(\frac{4}{\Delta_i-\epsilon}\right)\right)\sum_{k=1}^{\left\lfloor \log_2\left(\frac{4}{\Delta_i-\epsilon}\right)\right\rfloor} 2^{2k} \\
&\qquad \leqslant 2\log_2\left(\frac{4}{\Delta_i-\epsilon}\right) + \frac{16}{(\Delta_i-\epsilon)^2}\left(c' n\log(16) + 32\log\left(\frac{16n}{\delta}\log_2\left(\frac{4}{\Delta_i-\epsilon}\right)\right)\right)
\end{aligned}
$$

Next, we plug in the bound from claim 1 controlling the probability that $i \notin B_k$.

Using Claim 1, we bound the second sum as follows:

$$
\begin{aligned}
&\sum_{r=\left\lceil \log_2\left(\frac{4}{\Delta_i-\epsilon}\right)\right\rceil}^{\infty} \mathbb{E}_\nu\left[\mathbb{1}[i \notin B_{k-1}] \middle| \mathcal{E}_1\right]\left(H_{ME}(n, 2^{-k}, 1/16) + 2\left\lceil 2^{2k+3}\log\left(\frac{16nk^2}{\delta}\right)\right\rceil\right) \\
&\leqslant \sum_{k=\left\lceil \log_2\left(\frac{4}{\Delta_i-\epsilon}\right)\right\rceil}^{\infty} \left(\frac{1}{8}\right)^{k-\left\lceil \log_2\left(\frac{4}{\Delta_i-\epsilon}\right)\right\rceil-1}\left(c'n2^{2k}\log(16) + 2 + 2^{2k+4}\log\left(\frac{16nk^2}{\delta}\right)\right) \\
&= c'n\log(16)\sum_{k=1}^{\infty}\left(\frac{1}{8}\right)^{k-1} 2^{2\left(k+\left\lceil \log_2\left(\frac{4}{\Delta_i-\epsilon}\right)\right\rceil\right)} + 2\sum_{k=1}^{\infty}\left(\frac{1}{8}\right)^{k-1} \\
&\qquad + 16\sum_{k=1}^{\infty}\left(\frac{1}{8}\right)^{k-1} 2^{2\left(k+\left\lceil \log_2\left(\frac{4}{\Delta_i-\epsilon}\right)\right\rceil\right)}\log\left(\frac{16n\left(k+\left\lceil \log_2\left(\frac{4}{\Delta_i-\epsilon}\right)\right\rceil\right)^2}{\delta}\right) \\
&\leqslant 3 + c'n\log(16)\sum_{k=1}^{\infty} 2^{-3k+3}2^{2\left(k+\log_2\left(\frac{4}{\Delta_i-\epsilon}\right)+1\right)} \\
&\qquad + 16\sum_{k=1}^{\infty} 2^{-3k+3}2^{2\left(k+\log_2\left(\frac{4}{\Delta_i-\epsilon}\right)+1\right)}\log\left(\frac{16n\left(k+\left\lceil \log_2\left(\frac{4}{\Delta_i-\epsilon}\right)\right\rceil\right)^2}{\delta}\right) \\
&= 3 + \left(\frac{2^9c'n\log(16)}{(\Delta_i-\epsilon)^2} + \frac{2^{13}}{(\Delta_i-\epsilon)^2}\log\left(\frac{16n}{\delta}\right)\right)\sum_{k=1}^{\infty} 2^{-k} \\
&\qquad + \frac{2^{13}}{(\Delta_i-\epsilon)^2}\sum_{k=1}^{\infty} 2^{-k}\log\left(\left(k+\left\lceil \log_2\left(\frac{4}{\Delta_i-\epsilon}\right)\right\rceil\right)^2\right) \\
&\leqslant 3 + \frac{2^9c'n\log(16)}{(\Delta_i-\epsilon)^2} + \frac{2^{13}}{(\Delta_i-\epsilon)^2}\log\left(\frac{16n}{\delta}\right) \\
&\qquad + \frac{2^{14}}{(\Delta_i-\epsilon)^2}\sum_{k=1}^{\infty} 2^{-k}\log\left(k+\left\lceil \log_2\left(\frac{4}{\Delta_i-\epsilon}\right)\right\rceil\right) \\
&= (**)
\end{aligned}
$$

We may bound the final summand, $\sum_{k=1}^{\infty} 2^{-k} \log\left(k + \left\lceil \log_2\left(\frac{4}{\Delta_i - \epsilon}\right)\right\rceil\right)$ as follows:

$$\sum_{k=1}^{\infty} 2^{-k} \log\left(k + \left\lceil \log_2\left(\frac{4}{\Delta_i - \epsilon}\right)\right\rceil\right) \leqslant \log\left(\frac{e}{2}\log_2\left(\frac{256}{(\Delta_i - \epsilon)^2}\right)\right)$$

Plugging this back into $(**)$, we have that

$$(**) \leqslant 3 + \frac{2^9 c n \log(16)}{(\Delta_i - \epsilon)^2} + \frac{2^{13}}{(\Delta_i - \epsilon)^2}\log\left(\frac{16n}{\delta}\right) + \frac{2^{14}}{(\Delta_i - \epsilon)^2}\log\left(\frac{e}{2}\log_2\left(\frac{256}{(\Delta_i - \epsilon)^2}\right)\right)$$

Combining the above with the bound on the first sum, we have that

$$\begin{aligned}
&\sum_{k=1}^{\infty} \mathbb{E}_\nu\left[\mathbb{1}[i \notin B_{k-1}] \middle| \mathcal{E}_1\right]\left(2\tau_k + H_{ME}(n, 2^{-k}, 1/16)\right) \\
&\quad \leqslant c''\left(\frac{n}{(\Delta_i - \epsilon)^2} + \frac{c}{(\Delta_i - \epsilon)^2}\log\left(\frac{2n}{\delta}\log_2\left(\frac{4}{(\Delta_i - \epsilon)^2}\right)\right)\right) \\
&\quad = \frac{c'' n}{(\Delta_i - \epsilon)^2} + c'' h\left(0.25(\Delta_i - \epsilon), \frac{\delta}{2n}\right)
\end{aligned}$$

for a sufficiently large, universal constant c'' and c from the definition of $h(\cdot, \cdot)$. $\square$

6.F.3.11 Step 10: Applying the result of Step 9 to the result of Step 8

We may repeat the result of step 9 for every $i \in G_\epsilon^c$ and plug this into the result of Step 8. From this point, we simplify to return the final result.

By Step 8, the total number of samples T drawn by `FAREAST` is bounded in expectation by

$$\begin{aligned}
\mathbb{E}[T|\mathcal{E}_1 \cap \mathcal{E}_2] \leqslant\ & c \sum_{i \in G_\epsilon} \max\left\{h\left(0.25(\epsilon - \Delta_i), \frac{\delta}{2n}\right), \min\left[h\left(0.25\Delta_i, \frac{\delta}{2n}\right), h\left(0.25\alpha_\epsilon, \frac{\delta}{2n}\right)\right]\right\} \\
& + c \sum_{i \in G_{\epsilon+\alpha_\epsilon}^c} h\left(0.25(\epsilon - \Delta_i), \frac{\delta}{2n}\right) + c|G_\epsilon^c \cap G_{\epsilon+\alpha_\epsilon}| h\left(0.25\alpha_\epsilon, \frac{\delta}{2n}\right)
\end{aligned}$$

$$+ 2 \sum_{i \in G_\epsilon^c} \sum_{k=1}^{\infty} \mathbb{E}_\nu \left[\mathbb{1}[i \notin B_{k-1}] \middle| \mathcal{E}_1 \right] \left(2\tau_k + H_{ME}(n, 2^{-k}, 1/16) \right).$$

Applying the bound from Step 9 to each $i \in G_\epsilon^c$, we have that

$$\begin{aligned}
\mathbb{E}[T|\mathcal{E}_1 \cap \mathcal{E}_2] \leqslant c \sum_{i \in G_\epsilon} \max \left\{ h\left(0.25(\epsilon - \Delta_i), \frac{\delta}{2n}\right), \min \left[h\left(0.25\Delta_i, \frac{\delta}{2n}\right), h\left(0.25\alpha_\epsilon, \frac{\delta}{2n}\right) \right] \right\} \\
+ c \sum_{i \in G_{\epsilon+\alpha_\epsilon}^c} h\left(0.25(\epsilon - \Delta_i), \frac{\delta}{2n}\right) + c|G_\epsilon^c \cap G_{\epsilon+\alpha_\epsilon}| h\left(0.25\alpha_\epsilon, \frac{\delta}{2n}\right) \\
+ 2c'' \sum_{i \in G_\epsilon^c} \frac{n}{(\Delta_i - \epsilon)^2} + h\left(0.25(\Delta_i - \epsilon), \frac{\delta}{2n}\right).
\end{aligned}$$

For $i \in G_\epsilon^c \cap G_{\epsilon+\alpha_\epsilon}$, $\alpha_\epsilon = \min_{j \in G_\epsilon} \epsilon - \Delta_j \geqslant \Delta_i - \epsilon$. By monotonicity of $h(\cdot,\cdot)$, $h\left(0.25\alpha_\epsilon, \frac{\delta}{2n}\right) \leqslant \frac{c''n}{(\Delta_i - \epsilon)^2} + c'' h\left(\Delta_i - \epsilon, \frac{\delta}{2n}\right)$. Therefore,

$$\begin{aligned}
\mathbb{E}[T|\mathcal{E}_1 \cap \mathcal{E}_2] \leqslant c \sum_{i \in G_\epsilon} \max \left\{ h\left(0.25(\epsilon - \Delta_i), \frac{\delta}{2n}\right), \min \left[h\left(0.25\Delta_i, \frac{\delta}{2n}\right), h\left(0.25\alpha_\epsilon, \frac{\delta}{2n}\right) \right] \right\} \\
+ (2c'' + c) \sum_{i \in G_\epsilon^c} \frac{n}{(\Delta_i - \epsilon)^2} + h\left(0.25(\Delta_i - \epsilon), \frac{\delta}{2n}\right).
\end{aligned}$$

Next, we use Lemma 6.33 to bound the minimum of $h(\cdot, \cdots)$ functions.

$$\begin{aligned}
& c \sum_{i \in G_\epsilon} \max \left\{ h\left(0.25(\epsilon - \Delta_i), \frac{\delta}{2n}\right), \min \left[h\left(0.25\Delta_i, \frac{\delta}{2n}\right), h\left(0.25\alpha_\epsilon, \frac{\delta}{2n}\right) \right] \right\} \\
& \qquad + (2c'' + c) \sum_{i \in G_\epsilon^c} \frac{n}{(\Delta_i - \epsilon)^2} + h\left(0.25(\Delta_i - \epsilon), \frac{\delta}{2n}\right) \\
& \leqslant c \sum_{i \in G_\epsilon} \max \left\{ h\left(0.25(\epsilon - \Delta_i), \frac{\delta}{2n}\right), h\left(\frac{\Delta_i + \alpha_\epsilon}{8}, \frac{\delta}{2n}\right) \right\} \\
& \qquad + (2c'' + c) \sum_{i \in G_\epsilon^c} \frac{n}{(\Delta_i - \epsilon)^2} + h\left(0.25(\Delta_i - \epsilon), \frac{\delta}{2n}\right)
\end{aligned}$$

Finally, we use Lemma 6.32 to bound the function $h(\cdot,\cdot)$. Since $\delta \leqslant 1/2$, $\delta/n \leqslant$

$2e^{-\epsilon/2}$. Further, $\max(\Delta_i, |\epsilon - \Delta_i|) \leqslant 8$ for all i, we have that $0.25\Delta_i \leqslant 2$, $0.25|\epsilon - \Delta_i| \leqslant 2$, and $0.25 \min(\alpha_\epsilon, \beta_\epsilon) \leqslant 2$. Therefore,

$$
\begin{aligned}
\mathbb{E}[T|\mathcal{E}_1 \cap \mathcal{E}_2] &\leqslant c \sum_{i \in G_\epsilon} \max\left\{ h\left(0.25(\epsilon - \Delta_i), \frac{\delta}{2n}\right), h\left(\frac{\Delta_i + \alpha_\epsilon}{8}, \frac{\delta}{2n}\right)\right\} \\
&\quad + (2c'' + c) \sum_{i \in G_\epsilon^c} \frac{n}{(\Delta_i - \epsilon)^2} + h\left(0.25(\Delta_i - \epsilon), \frac{\delta}{2n}\right) \\
&\leqslant c \sum_{i \in G_\epsilon} \max\left\{ \frac{64}{(\epsilon - \Delta_i)^2} \log\left(\frac{4n}{\delta} \log_2\left(\frac{384n}{\delta(\epsilon - \Delta_i)^2}\right)\right), \right. \\
&\qquad \left. \frac{256}{(\Delta_i + \alpha_\epsilon)^2} \log\left(\frac{4n}{\delta} \log_2\left(\frac{768n}{\delta(\Delta_i + \alpha_\epsilon)^2}\right)\right)\right\} \\
&\quad + (2c'' + c) \sum_{i \in G_\epsilon^c} \frac{n}{(\Delta_i - \epsilon)^2} + \frac{64}{(\epsilon - \Delta_i)^2} \log\left(\frac{4n}{\delta} \log_2\left(\frac{384n}{\delta(\epsilon - \Delta_i)^2}\right)\right) \\
&\leqslant c_3 \sum_{i \in G_\epsilon} \max\left\{ \frac{1}{(\epsilon - \Delta_i)^2} \log\left(\frac{n}{\delta} \log_2\left(\frac{n}{\delta(\epsilon - \Delta_i)^2}\right)\right), \right. \\
&\qquad \left. \frac{1}{(\Delta_i + \alpha_\epsilon)^2} \log\left(\frac{n}{\delta} \log_2\left(\frac{n}{\delta(\Delta_i + \alpha_\epsilon)^2}\right)\right)\right\} \\
&\quad + c_3 \sum_{i \in G_\epsilon^c} \frac{n}{(\Delta_i - \epsilon)^2} + \frac{1}{(\epsilon - \Delta_i)^2} \log\left(\frac{n}{\delta} \log_2\left(\frac{n}{\delta(\epsilon - \Delta_i)^2}\right)\right) \\
&= c_3 \sum_{i \in G_\epsilon} \max\left\{ \frac{1}{(\mu_1 - \epsilon - \mu_i)^2} \log\left(\frac{n}{\delta} \log_2\left(\frac{n}{\delta(\mu_1 - \epsilon - \mu_i)^2}\right)\right), \right. \\
&\qquad \left. \frac{1}{(\mu_1 + \alpha_\epsilon - \mu_i)^2} \log\left(\frac{n}{\delta} \log_2\left(\frac{n}{\delta(\mu_1 + \alpha_\epsilon - \mu_i)^2}\right)\right)\right\} \\
&\quad + c_3 \sum_{i \in G_\epsilon^c} \frac{n}{(\mu_1 - \epsilon - \mu_i)^2} + \frac{1}{(\mu_1 - \epsilon - \mu_i)^2} \log\left(\frac{n}{\delta} \log_2\left(\frac{n}{\delta(\mu_1 - \epsilon - \mu_i)^2}\right)\right)
\end{aligned}
$$

for a sufficiently large constant c_4.

6.F.3.12 Step 11: High probability sample complexity bound

Finally, the Good Filter is equivalent to `EAST`, Algorithm 10, except split across rounds. Note that the Good Filter is union bounded over $2n$ events whereas the

bounds in EAST are union bounded over n events. The Good Filter and Bad Filter are given the same number of samples in each round, and the Good Filter can terminate within a round, conditioned on $\mathcal{E}_1 \cap \mathcal{E}_2$. Therefore, we can bound the complexity of FAREAST in terms of that of EAST run at failure probability $\delta/2$. If FAREAST terminates in the second round or later, the arguments in Steps 4 and 5 can be used to show that FAREAST draws no more than a factor of 18 more samples than EAST, though this estimate is highly pessimistic. If FAREAST terminates in round 1 (when gaps are large), we may still show that this is within a constant factor of the complexity of EAST, but the story is more complicated. In the first round, the bad filter draws at most $c'n\log(16) + 32n\log(8n/\delta)$ samples where c' is the constant from Median Elimination. Since we have assumed that $\max(\Delta_i, |\epsilon - \Delta_i|) \leqslant 8$, this sum is likewise within a constant factor of the complexity of EAST. Hence, by Theorem 6.27,

$$T \leqslant c_4 \sum_{i=1}^{n} \min \left\{ \max \left\{ \frac{1}{(\mu_1 - \epsilon - \mu_i)^2} \log\left(\frac{n}{\delta}\log_2\left(\frac{n}{\delta(\mu_1 - \epsilon - \mu_i)^2}\right)\right), \right.\right.$$
$$\frac{1}{(\mu_1 + \alpha_\epsilon - \mu_i)^2} \log\left(\frac{n}{\delta}\log_2\left(\frac{n}{\delta(\mu_1 + \alpha_\epsilon - \mu_i)^2}\right)\right),$$
$$\left.\left.\frac{1}{(\mu_1 + \beta_\epsilon - \mu_i)^2} \log\left(\frac{n}{\delta}\log_2\left(\frac{n}{\delta(\mu_1 + \beta_\epsilon - \mu_i)^2}\right)\right)\right\}$$
$$\left.\frac{1}{\gamma^2}\log\left(\frac{n}{\delta}\log_2\left(\frac{n}{\delta\gamma^2}\right)\right)\right\}$$

samples. □

6.F.4 Proof of Theorem 6.26, FAREAST in the multiplicative regime

Proof. **Notation for the proof:** Throughout, recall $\Delta_i = \mu_1 - \mu_i$. Recall that t counts the number of times the conditional in line 19 is true. By Line 19 of FAREAST, all arms in $\mathcal{A}$ have received t samples when the loop in line 23 is executed for the t^{th} time. Within any round k, let $\mathcal{A}(t)$ and $G_k(t)$ denote the sets $\mathcal{A}$ and G_k at this time

since both sets can change in lines 27 and 29 and 25 respectively. Let t_k denote the maximum value of t in round k. By Lines 18 and 19 of FAREAST, the total number of samples given to the good filter when the conditional in line 19 is true for the t^{th} time is $\sum_{s=1}^{t} |\mathcal{A}(s)|$.

For $i \in M_\epsilon$, let T_i denote the random variable of the number of times arm i is sampled before it is added to G_k in Line 25. For $i \in M_\epsilon^c$, let T_i denote the random variable of the number of times arm i is sampled before it is removed from $\mathcal{A}$ in Line 27. For any arm i, let T_i' denote the random variable of the of the number of times i is sampled before $\hat{\mu}_i(t) + C_{\delta/2n}(t) \leqslant \max_{j \in \mathcal{A}} \hat{\mu}_j(t) - C_{\delta/2n}(t)$.

Define the event

$$\mathcal{E}_1 = \left\{ \bigcap_{i \in [n]} \bigcap_{t \in \mathbb{N}} |\hat{\mu}_i(t) - \mu_i| \leqslant C_{\delta/2n}(t) \right\}.$$

Using standard anytime confidence bound results, and recalling that that $C_\delta(t) := \sqrt{\frac{4\log(\log_2(2t)/\delta)}{t}}$, we have

$$\begin{aligned}
\mathbb{P}(\mathcal{E}_1^c) &= \mathbb{P}\left(\bigcup_{i \in [n]} \bigcup_{t \in \mathbb{N}} |\hat{\mu}_i - \mu_i| > C_{\delta/2n}(t) \right) \\
&\leqslant \sum_{i=1}^{n} \mathbb{P}\left(\bigcup_{t \in \mathbb{N}} |\hat{\mu}_i - \mu_i| > C_{\delta/2n}(t) \right) \leqslant \sum_{i=1}^{n} \frac{\delta}{2n} = \frac{\delta}{2}
\end{aligned}$$

Next, recall that $\hat{\mu}_i(t)$ denotes the empirical average of t samples of ρ_i. Consider the event,

$$\mathcal{E}_2 = \bigcap_{i \in M_\epsilon} \bigcap_{k \in \mathbb{N}} |((1-\epsilon)\hat{\mu}_{i_k}(\tau_k) - \hat{\mu}_i(\tau_k)) - ((1-\epsilon)\mu_{i_k} - \mu_i)| \leqslant 2^{-(k+1)}(2-\epsilon)$$

By Hoeffding's inequality,

$$\mathbb{P}\left(|((1-\epsilon)\hat{\mu}_{i_k}(\tau_k) - \hat{\mu}_i(\tau_k)) - ((1-\epsilon)\mu_{i_k} - \mu_i)| \leqslant 2^{-(k+1)}(2-\epsilon)\middle| i_k = j\right) \leqslant \frac{\delta}{4nk^2}.$$

Then

$$\begin{aligned}
&\mathbb{P}\big(|((1-\epsilon)\hat{\mu}_{i_k}(\tau_k) - \hat{\mu}_i(\tau_k)) - ((1-\epsilon)\mu_{i_k} - \mu_i)| \leqslant 2^{-(k+1)}(2-\epsilon)\big) \\
&\quad = \sum_{j=1}^{n} \mathbb{P}\left(|((1-\epsilon)\hat{\mu}_{i_k}(\tau_k) - \hat{\mu}_i(\tau_k)) - ((1-\epsilon)\mu_{i_k} - \mu_i)| \leqslant 2^{-(k+1)}(2-\epsilon)\middle| i_k = j\right)\mathbb{P}(i_k = j) \\
&\quad \leqslant \frac{\delta}{4nk^2}\sum_{j=1}^{n}\mathbb{P}(i_k = j) \\
&\quad = \frac{\delta}{4nk^2}
\end{aligned}$$

Therefore, union bounding over the rounds $k \in \mathbb{N}$, $\mathbb{P}(\mathcal{E}_2^c) \leqslant \sum_{i\in M_\epsilon}\sum_{k=1}^{\infty}\frac{\delta}{4nk^2} \leqslant \frac{\delta}{2}$. Hence, $\mathbb{P}(\mathcal{E}_1 \cap \mathcal{E}_2) \geqslant 1-\delta$.

6.F.4.1 Step 0: Correctness.

On $\mathcal{E}_1 \cap \mathcal{E}_2$, first we prove that if there exists a random round k at which $G_k \cup B_k = [n]$ then $G_k = M_\epsilon$. Additionally, we prove that on $\mathcal{E}_1 \cap \mathcal{E}_2$, if $\mathcal{A} \subset G_k$, then $G_k = M_\epsilon$. Therefore, for either stopping condition for `FAREAST` in line 31, on the event $\mathcal{E}_1 \cap \mathcal{E}_2$, `FAREAST` correctly returns the set M_ϵ.

Claim 0: On $\mathcal{E}_1 \cap \mathcal{E}_2$, for all $k \in \mathbb{N}$, $G_k \subset M_\epsilon$.

Proof. Firstly we show $1 \in \mathcal{A}$ for all $t \in \mathbb{N}$, namely the best arm is never removed from $\mathcal{A}$. Note for any i such that $\hat{\mu}_i(t) - C_{\delta/2n}(t) \geqslant 0$,

$$\hat{\mu}_1 + C_{\delta/2n}(t) \geqslant \mu_1 \geqslant \mu_i \geqslant \hat{\mu}_i(t) - C_{\delta/2n}(t) > (1-\epsilon)(\hat{\mu}_i(t) - C_{\delta/2n}(t)).$$

For i such that $\hat{\mu}_i(t) - C_{\delta/2n}(t) < 0$, if $\hat{\mu}_1 + C_{\delta/2n}(t) \geqslant 0$, then

$$\hat{\mu}_1 + C_{\delta/2n}(t) \geqslant 0 > (1-\epsilon)(\hat{\mu}_i(t) - C_{\delta/2n}(t)).$$

Note that $\hat{\mu}_1 + C_{\delta/2n}(t) < 0$ implies on the event $\mathcal{E}_1$ that $\mu_1 < 0$, which contradicts the assumption that $\mu_1 \geqslant 0$ made in the theorem. In particular this shows, $\hat{\mu}_1 + C_{\delta/2n}(t) > (1-\epsilon)(\max_{i\in\mathcal{A}} \hat{\mu}_i(t) - C_{\delta/2n}(t)) = L_t$ and $\hat{\mu}_1 + C_{\delta/2n}(t) \geqslant \max_{i\in\mathcal{A}} \hat{\mu}_i(t) - C_{\delta/2n}(t)$ showing that 1 will never exit $\mathcal{A}$ in line 28.

Secondly, we show that at all times t, $(1-\epsilon)\mu_1 \in [L_t, U_t]$. By the above, since μ_1 never leaves $\mathcal{A}$,

$$U_t = (1-\epsilon)(\max_{i\in\mathcal{A}} \hat{\mu}_i(t) + C_{\delta/2n}(t)) \geqslant (1-\epsilon)(\hat{\mu}_1(t) + C_{\delta/2n}(t)) \geqslant (1-\epsilon)\mu_1$$

and for any i,

$$(1-\epsilon)\mu_1 \geqslant (1-\epsilon)\mu_i \geqslant (1-\epsilon)(\hat{\mu}_i(t) - C_{\delta/2n}(t))$$

Hence $(1-\epsilon)\mu_1 \geqslant (1-\epsilon)(\max_i \hat{\mu}_i(t) - C_{\delta/2n}(t)) = L_t$.

Next, we show that $G_k \subset M_\epsilon$ for all $k \geqslant 1, t \geqslant 1$. Suppose not. Then $\exists, k, t \in N$ and $\exists i \in M_\epsilon^c \cap G_k(t)$ such that,

$$\mu_i \geqslant \hat{\mu}_i(t) - C_{\delta/2n}(t) \geqslant U_t \geqslant (1-\epsilon)\mu_1 > \mu_i,$$

with the last inequality following from the previous assertion, giving a contradiction. □

Claim 1: On $\mathcal{E}_1 \cap \mathcal{E}_2$, for all $k \in \mathbb{N}$, $B_k \subset M_\epsilon^c$.

Proof. Next, we show $B_k \subset M_\epsilon^c$. Suppose not. Then either the good filter or the bad filter added an arm in M_ϵ to B_k. Take $i \in M_\epsilon$. In the former, this implies that

$$\mu_i \overset{\mathcal{E}_1}{\leqslant} \hat{\mu}_i(t) + C_{\delta/2n}(t) < L_t \overset{\mathcal{E}_1}{\leqslant} (1-\epsilon)\mu_1$$

which contradicts $i \in M_\epsilon$. Consider the alternate case that the bad filter adds i to B_k for some k. By definition, $B_0 = \emptyset$ and $B_{k-1} \subset B_k$ for all k. Then there must exist $k \in \mathbb{N}$ and an $i \in M_\epsilon$ such that $i \in B_k$ and $i \notin B_{k-1}$. Following line 14 of the

algorithm, this occurs if and only if

$$(1-\epsilon)\hat{\mu}_{i_k} - \hat{\mu}_i > 2^{-(k+1)}(2-\epsilon).$$

On the event $\mathcal{E}_2$, the above implies

$$(1-\epsilon)\mu_{i_k} - \mu_i + 2^{-(k+1)}(2-\epsilon) > 2^{-(k+1)}(2-\epsilon),$$

and simplifying, we see that $0 < (1-\epsilon)\mu_{i_k} - \mu_i \leqslant (1-\epsilon)\mu_1 - \mu_i$ which contradicts the assertion that $i \in M_\epsilon$. Combining the above claims, we see that $\mathcal{E}_1 \cap \mathcal{E}_2$ implies $(G_k \cup B_k = [n])$ and $G_k \cap B_k = \emptyset \implies G_k = M_\epsilon$. Since $\mathbb{P}(\mathcal{E}_1 \cap \mathcal{E}_2) \geqslant 1-\delta$, if FAREAST terminates, with probability at least $1-\delta$, it correctly returns the set M_ϵ. □

Claim 2: Next, we show that on $\mathcal{E}_1$, $M_\epsilon \subset \mathcal{A}(t) \cup G(t)$ for all $t \in \mathbb{N}$.

In particular this implies that if $\mathcal{A} \subset G$, then $M_\epsilon \subset G$. Combining this with the previous claim gives $G \subset M_\epsilon \subset G$, hence $G = M_\epsilon$. On this condition, FAREAST terminates by line 33 and returns the set $\mathcal{A} \cup G = G$. Note that by definition, $M_\epsilon \subset M_{(\epsilon+\gamma)}$ for all $\gamma \geqslant 0$. Therefore FAREAST terminates correctly on this condition.

Proof. Suppose for contradiction that there exists $i \in M_\epsilon$ such that $i \notin \mathcal{A}(t) \cup G(t)$. This occurs only if i is eliminated in line 28. Hence, there exists a $t' \leqslant t$ such that $\hat{\mu_i}(t') + C_{\delta/n}(t') < L_{t'}$. Therefore, on the event $\mathcal{E}_1$,

$$(1-\epsilon)\mu_1 \overset{\mathcal{E}_1}{\geqslant} L_{t'} = (1-\epsilon)\left(\max_{j\in\mathcal{A}} \hat{\mu}_j(t') - C_{\delta/n}(t')\right) > \hat{\mu_i}(t') + C_{\delta/n}(t') \overset{\mathcal{E}_1}{\geqslant} \mu_i$$

which contradicts $i \in M_\epsilon$. □

Claim 3: Finally, we show that on $\mathcal{E}_1$, if $U_t - L_t \leqslant \frac{\gamma}{2-\epsilon} L_t$, then $\mathcal{A} \cup G \subset M_{(\epsilon+\gamma)}$.

Combining with Claim 3 that $M_\epsilon \subset \mathcal{A} \cup G$, if FAREAST terminates on this condition by line 33, it does so correctly and returns all arms in M_ϵ and none in $M^c_{(\epsilon+\gamma)}$.

Proof. By Claim 0, $G \subset M_\epsilon \subset M_{\epsilon+\gamma}$. Hence, $G \cap M^c_{(\epsilon+\gamma)} = \emptyset$. Therefore, we wish to show that $\mathcal{A} \cap M^c_{(\epsilon+\gamma)} = \emptyset$ which implies that $G \cap \mathcal{A} \subset M_{\epsilon+\gamma}$. Assume

$U_t - L_t < \frac{\gamma}{2-\epsilon} L_t$. Recall that

$$U_t = (1-\epsilon)\left(\max_{i\in\mathcal{A}} \hat{\mu}_i(t) + C_{\delta/2n}(t)\right)$$

and

$$L_t = (1-\epsilon)\left(\max_{i\in\mathcal{A}} \hat{\mu}_i(t) - C_{\delta/2n}(t)\right)$$

All arms in $\mathcal{A}(t)$ have received exactly t samples. Hence, $U_t - L_t = 2(1-\epsilon)C_{\delta/2n}(t)$. On $\mathcal{E}_1$, $L_t \leqslant (1-\epsilon)\mu_1$ This implies that

$$2(1-\epsilon)C_{\delta/2n}(t) < \frac{\gamma}{2-\epsilon} L_t \leqslant \frac{1-\epsilon}{2-\epsilon}\gamma\mu_1,$$

and in particular,

$$2C_{\delta/2n}(t) < \frac{\gamma\mu_1}{2-\epsilon}.$$

Therefore, we wish to show that when the above is true, then for any $i \in M^c_{\epsilon+\gamma}$, $L_t - (\hat{\mu}_i(t) + C_{\delta/n}(t)) > 0$, implying that $i \notin \mathcal{A}$.

$$\begin{aligned}
L_t - (\hat{\mu}_i(t) + C_{\delta/n}(t)) &= (1-\epsilon)\left(\max_{j\in\mathcal{A}} \hat{\mu}_j - C_{\delta/n}(t)\right) - (\hat{\mu}_i(t) + C_{\delta/n}(t)) \\
&\geqslant (1-\epsilon)\left(\max_{j\in\mathcal{A}} \mu_j - 2C_{\delta/n}(t)\right) - (\mu_i + 2C_{\delta/n}(t)) \\
&\overset{(a)}{\geqslant} (1-\epsilon)\left(\mu_1 - 2C_{\delta/n}(t)\right) - ((1-\epsilon-\gamma)\mu_1 + 2C_{\delta/n}(t)) \\
&= \gamma\mu_1 - 2(2-\epsilon)C_{\delta/n}(t) \\
&> \gamma\mu_1 - (2-\epsilon)\frac{\gamma\mu_1}{2-\epsilon} = 0
\end{aligned}$$

which implies that $i \notin \mathcal{A}$. Inequality (a) follows jointly from the fact that $1 \in \mathcal{A}$ and the fact that all arms in $\mathcal{A}$ have received t samples implies $\max_{j\in\mathcal{A}} \mu_j - 2C_{\delta/n}(t) = \mu_1 - 2C_{\delta/n}(t)$. Additionally, inequality (a) follows from $\mu_i \leqslant (1-\epsilon-\gamma)\mu_1$ since $i \in M^c_{\epsilon+\gamma}$. □

6.F.4.2 Step 1: An expression for the total number of samples drawn and introducing several helper random variables

Next, we write an expression for the total number of samples drawn by FAREAST. In particular, we introduce two sums that we will spend the remainder of the proof controlling. Additionally, we show that the conditional in line 19 in the good filter is true at least once in each round. Based on this, we more precisely define the random variables T_i and T_i' introduced in the notation section in section 6.F.4. Additionally, we introduce the time T_γ at which $U_t - L_t < \frac{\gamma}{2-\epsilon} L_t$.

Recall that the largest value of t in round k is denoted t_k. Let E_k^γ be the event that $U_t - L_t \geqslant \frac{\gamma}{2-\epsilon} L_t$ for all t in round k:

$$E_k^\gamma := \left\{ U_t - L_t \geqslant \frac{\gamma}{2-\epsilon} L_t : t \in (t_{k-1}, t_k] \right\}.$$

Note that if E_{k-1}^γ is false, then FAREAST terminates in round $k-1$ by line 33. We may write the total number of samples drawn by the algorithm as

$$T = \sum_{k=1}^{\infty} 2\mathbb{1}\left[\mathcal{A} \not\subset G_{k-1} \text{ and } G_{k-1} \cup B_{k-1} \neq [n] \text{ and } E_{k-1}^\gamma\right] \left(H_{ME}(n, 2^{-k}, 1/16) + \tau_k + \tau_k | (G_{k-1} \cup B_{k-1})^c|\right)$$

Deterministically,

$$\mathbb{1}\left[\mathcal{A} \not\subset G_{k-1} \text{ and } G_{k-1} \cup B_{k-1} \neq [n] \text{ and } E_{k-1}^\gamma\right] \leqslant \mathbb{1}\left[G_{k-1} \cup B_{k-1} \neq [n]\right].$$

Applying this,

$$\begin{aligned} T &\leqslant \sum_{k=1}^{\infty} 2\mathbb{1}\left[G_{k-1} \cup B_{k-1} \neq [n]\right] \left(H_{ME}(n, 2^{-k}, 1/16) + \tau_k + \tau_k | (G_{k-1} \cup B_{k-1})^c|\right) \\ &= \sum_{k=1}^{\infty} 2\mathbb{1}\left[G_{k-1} \neq M_\epsilon\right] \mathbb{1}\left[G_{k-1} \cup B_{k-1} \neq [n]\right] \\ &\qquad \left(H_{ME}(n, 2^{-k}, 1/16) + \tau_k + \tau_k | (G_{k-1} \cup B_{k-1})^c|\right) \end{aligned} \tag{6.17}$$

$$+\sum_{k=1}^{\infty} 2\mathbb{1}\left[G_{k-1}=M_\epsilon\right]\mathbb{1}\left[G_{k-1}\cup B_{k-1}\neq[n]\right]$$
$$\left(H_{ME}(n,2^{-k},1/16)+\tau_k+\tau_k|(G_{k-1}\cup B_{k-1})^c|\right) \tag{6.18}$$

In round k, line 18 of the Good Filter, whereby an arm is sampled, is evaluated

$$\left(H_{ME}(n,2^{-k},1/16)+\tau_k+\tau_k|(G_{k-1}\cup B_{k-1})^c|\right)\geqslant(2\tau_k+H_{ME}(n,2^{-k},1/16))\geqslant n$$

times since $H_{ME}(n,2^{-k},1/16))\geqslant n$ for all k and $|(G_{k-1}\cup B_{k-1})^c|\geqslant 1$ unless $G_{k-1}\cup B_{k-1}=[n]$ which implies termination in round $k-1$. Each time line 18 is called, $N_{I_s}\leftarrow N_{I_s}+1$. Since $|\arg\min_{j\in\mathcal{A}}\{N_j\}|\leqslant|\mathcal{A}|\leqslant n$, line 18 is called at most n times before $\min_{j\in\mathcal{A}}\{N_j\}=\max_{j\in\mathcal{A}}\{N_j\}$. When this occurs, the conditional in line 19 is true and $t\leftarrow t+1$.

If $\min_{i\in\mathcal{A}(t)}\{N_i\}=\max_{i\in\mathcal{A}(t)}\{N_i\}$, then $N_i=t$ for any $i\in\mathcal{A}(t)$. By Step 0, only arms in M_ϵ are added to G_k. Therefore, T_i is defined as

$$T_i=\min\left\{t:\begin{array}{ll} i\in G_k(t+1) & \text{if } i\in M_\epsilon\\ i\notin\mathcal{A}(t+1) & \text{if } i\in M_\epsilon^c\end{array}\right\}\overset{\mathcal{E}_1}{=}\min\left\{t:\begin{array}{ll} \hat{\mu}_i-C_{\delta/2n}(t)\geqslant U_t & \text{if } i\in M_\epsilon\\ \hat{\mu}_i+C_{\delta/2n}(t)\leqslant L_t & \text{if } i\in M_\epsilon^c\end{array}\right\} \tag{6.19}$$

Define $T_i=\infty$ if this never occurs. Note that this may happen if `FAREAST` terminates due to the condititon in line 32 that $U_t-L_t<\frac{\gamma}{2-\epsilon}L_t$. Similarly, recall T_i' denotes the random variable of the of the number of times i is sampled before $\hat{\mu}_i(t)+C_{\delta/2n}(t)\leqslant\max_{j\in\mathcal{A}}\hat{\mu}_j(t)-C_{\delta/2n}(t)$. Hence,

$$T_i'=\min\left\{t:\hat{\mu}_i(t)+C_{\delta/2n}(t)\leqslant\max_{j\in\mathcal{A}(t)}\hat{\mu}_j(t)-C_{\delta/2n}(t)\right\} \tag{6.20}$$

Define $T_i'=\infty$ if this never occurs. Note that this may happen if `FAREAST` terminates due to the condititon in line 32 that $U_t-L_t<\frac{\gamma}{2-\epsilon}L_t$. Finally, we define the time

T_γ such that $U_t - L_t < \frac{\gamma}{2-\epsilon} L_t$.

$$T_\gamma = \min\left\{ t : U_t - L_t < \frac{\gamma}{2-\epsilon} L_t \right\} \tag{6.21}$$

By design, no arm is sampled more that T_γ times by the good filter, controlling the cases that T_i or T_i' are infinite.

6.F.4.3 Step 2: Bounding T_i and T_i' for $i \in M_\epsilon$

Step 2a: For $i \in M_\epsilon$, we have that $T_i \leqslant h\left(\frac{\epsilon\mu_1 - \Delta_i}{4-2\epsilon}, \frac{\delta}{2n}\right)$.

Proof. Note that $\mu_i - 2C_{\delta/2n}(t) \geqslant (1-\epsilon)(\mu_1 + 2C_{\delta/2n}(t))$ may be rearranged as $(4-2\epsilon)C_{\delta/2n}(t) \leqslant \epsilon\mu_1 - \Delta_i$, and this is true when $t > h\left(\frac{\epsilon\mu_1 - \Delta_i}{4-2\epsilon}, \frac{\delta}{2n}\right)$. This condition implies that for all j,

$$\begin{aligned}
\hat{\mu}_i(t) - C_{\delta/2n}(t) &\overset{\mathcal{E}_1}{\geqslant} \mu_i - 2C_{\delta/2n}(t) \\
&\geqslant (1-\epsilon)(\mu_1 + 2C_{\delta/2n}(t)) \\
&\geqslant (1-\epsilon)(\mu_j + 2C_{\delta/2n}(t)) \\
&\overset{\mathcal{E}_1}{\geqslant} (1-\epsilon)(\hat{\mu}_j(t) + C_{\delta/2n}(t))
\end{aligned}$$

so in particular, $\hat{\mu}_i(t) - C_{\delta/2n}(t) \geqslant (1-\epsilon)(\max_{j\in\mathcal{A}} \hat{\mu}_j(t) + C_{\delta/2n}(t)) = U_t$. □

Additionally, we define a time T_{max} when all good arms have entered G_k.

Step 2b: Defining $T_{max} := \min\{t : G_k(t) = M_\epsilon\} = \max_{i\in M_\epsilon} T_i$, we also have that $T_{max} \leqslant h(\tilde{\alpha}_\epsilon/(4-2\epsilon), \delta/2n)$ (in other words, if $t > h(\tilde{\alpha}_\epsilon/(4-2\epsilon), \delta/2n)$ (i.e. line 23 has been run t times, then we have that $G_k(t) = M_\epsilon$).

Proof. Recall that $\tilde{\alpha}_\epsilon = \min_{i\in M_\epsilon} \mu_i - \mu_1 + \epsilon = \min_{i\in M_\epsilon} \epsilon\mu_1 - \Delta_i$. By Step 1a, $T_i \leqslant h\left(\frac{\epsilon\mu_1 - \Delta_i}{4-2\epsilon}, \frac{\delta}{2n}\right)$. Furthermore, $h(\cdot,\cdot)$ is monotonic in its first argument, such that if $0 < x' < x$, then $h(x',\delta) > h(x,\delta)$ for any $\delta > 0$. Therefore $T_{max} = \max_{i\in M_\epsilon} T_i \leqslant \max_{i\in M_\epsilon} h\left(\frac{\epsilon\mu_1 - \Delta_i}{4-2\epsilon}, \frac{\delta}{2n}\right) = h\left(\tilde{\alpha}_\epsilon/(4-2\epsilon), \frac{\delta}{2n}\right)$. □

Step 2c: For $i \in M_\epsilon$, we have that $T_i' \leqslant h(0.25\Delta_i, \delta/2n)$.

Proof. Note that $4C_{\delta/2n}(t) \leqslant \mu_1 - \mu_i$, true when $t > h\left(0.25\Delta_i, \frac{\delta}{2n}\right)$, implies that

$$\begin{aligned}\hat{\mu}_i(t) + C_{\delta/2n}(t) &\overset{\mathcal{E}_1}{\leqslant} \mu_i + 2C_{\delta/2n}(t) \\ &\leqslant \mu_1 - 2C_{\delta/2n}(t) \\ &\overset{\mathcal{E}_1}{\leqslant} \hat{\mu}_1(t) - C_{\delta/2n}(t).\end{aligned}$$

As shown in Step 0, $1 \in \mathcal{A}(t)$ for all $t \in \mathbb{N}$, and in particular $\hat{\mu}_1(t) \leqslant \max_{i\in\mathcal{A}(t)} \hat{\mu}_i(t)$. Hence, $\hat{\mu}_i(t) + C_{\delta/2n}(t) \leqslant \max_{j\in\mathcal{A}(t)} \hat{\mu}_j(t) - C_{\delta/2n}(t)$. □

6.F.4.4 Step 3: Bounding T_i for $i \in M_\epsilon^c$

Next, we bound T_i for $i \in M_\epsilon^c$. $i \in M_\epsilon^c$ is eliminated from $\mathcal{A}$ if it has received at least T_i samples.

Claim: $T_i \leqslant h\left(\frac{\Delta_i - \epsilon\mu_1}{4-2\epsilon}, \frac{\delta}{2n}\right)$ for $i \in M_\epsilon^c$

Proof. Note that $\mu_i + 2C_{\delta/2n}(t) \leqslant (1-\epsilon)(\mu_1 - 2C_{\delta/2n}(t))$ may be rearranged as $(4-2\epsilon)C_{\delta/2n}(t) \leqslant \Delta_i - \epsilon\mu_1$, and this is true when $t > h\left(\frac{\epsilon\mu_1 - \Delta_i}{4-2\epsilon}, \frac{\delta}{2n}\right)$. This condition implies that

$$\begin{aligned}\hat{\mu}_i(t) + C_{\delta/2n}(t) &\overset{\mathcal{E}_1}{\leqslant} \mu_i + 2C_{\delta/2n}(t) \\ &\leqslant (1-\epsilon)(\mu_1 - 2C_{\delta/2n}(t)) \\ &\overset{\mathcal{E}_1}{\leqslant} (1-\epsilon)(\hat{\mu}_1(t) - C_{\delta/2n}(t))\end{aligned}$$

As shown in Step 0, $1 \in \mathcal{A}(t)$ for all $t \in \mathbb{N}$, and in particular $\hat{\mu}_1(t) \leqslant \max_{i\in\mathcal{A}(t)} \hat{\mu}_i(t)$. Therefore $\hat{\mu}_i(t) + C_{\delta/2n}(t) \leqslant (1-\epsilon)(\max_{j\in\mathcal{A}} \hat{\mu}_j(t) - C_{\delta/2n}(t)) = L_t$. □

6.F.4.5 Step 4: bounding the total number of samples given to the good filter at time $t = T_{max}$

Note that for a time $t = T$, the total number of samples given to the good filter is $\sum_{s=1}^{T} |\mathcal{A}(s)|$. Therefore, the total number of samples up to time T_{max} is $\sum_{t=1}^{T_{max}} |\mathcal{A}(t)|$.

Let $S_i = \min\{t : i \notin A(t+1)\}$. Hence,

$$\sum_{t=1}^{T_{max}} |\mathcal{A}(t)| = \sum_{t=1}^{T_{max}} \sum_{i=1}^{n} \mathbb{1}[i \in \mathcal{A}(t)] = \sum_{i=1}^{n} \sum_{t=1}^{T_{max}} \mathbb{1}[i \in \mathcal{A}(t)] = \sum_{i=1}^{n} \min\{T_{max}, S_i\}$$

For arms $i \in M_\epsilon^c$, $S_i = T_i$ by definition. For $i \in M_\epsilon$, $S_i = \max(T_i, T_i')$ by line 28 of the algorithm. Then

$$\begin{aligned}
\sum_{i=1}^{n} \min\{T_{max}, S_i\} &= \sum_{i \in M_\epsilon} \min\{\mathcal{T}_{max}, \max(T_i, T_i')\} + \sum_{i \in M_\epsilon^c} \min\{T_{max}, T_i\} \\
&\leqslant \sum_{i \in M_\epsilon} \min\{T_{max}, \max(T_i, T_i')\} + |M_\epsilon^c \cap M_{\epsilon+\tilde{\alpha}_\epsilon}| T_{max} + \sum_{i \in M_{\epsilon+\tilde{\alpha}_\epsilon}^c} T_i \\
&= \sum_{i \in M_\epsilon} \max\{T_i, \min(T_i', T_{max})\} + |M_\epsilon^c \cap M_{\epsilon+\tilde{\alpha}_\epsilon/\mu_1}| T_{max} + \sum_{i \in M_{\epsilon+\tilde{\alpha}_\epsilon/\mu_1}^c} T_i \\
&\overset{(a)}{\leqslant} \sum_{i \in M_\epsilon} \max\left\{ h\left(\frac{\epsilon\mu_1 - \Delta_i}{4-2\epsilon}, \frac{\delta}{2n}\right), \min\left[h\left(0.25\Delta_i, \frac{\delta}{2n}\right), h\left(\frac{\tilde{\alpha}_\epsilon}{4-2\epsilon}, \frac{\delta}{2n}\right)\right]\right\} \\
&\qquad + \sum_{i \in M_{\epsilon+\tilde{\alpha}_\epsilon/\mu_1}^c} h\left(\frac{\epsilon\mu_1 - \Delta_i}{4-2\epsilon}, \frac{\delta}{2n}\right) + |M_\epsilon^c \cap M_{\epsilon+\tilde{\alpha}_\epsilon/\mu_1}| h\left(\frac{\tilde{\alpha}_\epsilon}{4-2\epsilon}, \frac{\delta}{2n}\right).
\end{aligned}$$

Equality (a) follows from $T_{max} \leqslant h\left(\frac{\tilde{\alpha}_\epsilon}{4-2\epsilon}, \frac{\delta}{2n}\right)$ by Step 1b, $T_i \leqslant h\left(\frac{\epsilon\mu_1 - \Delta_i}{4-2\epsilon}, \frac{\delta}{2n}\right)$ in Steps 2a and 3, and $T_i' \leqslant h\left(0.25\Delta_i, \frac{\delta}{2n}\right)$ in Step 2c.

6.F.4.6 Step 5: Bounding the number of samples in round k versus $k-1$

Now we show that the total number of samples taken in round k is no more than 9 times the number taken in the previous round.

Claim: For $k > 1$

$$\begin{aligned}&\left(H_{ME}(n, 2^{-k}, 1/16) + \tau_k + \tau_k|(G_{k-1} \cup B_{k-1})^c|\right)\\&\leqslant 9\left(H_{ME}(n, 2^{-k+1}, 1/16) + \tau_{k-1} + \tau_{k-1}|(G_{k-2} \cup B_{k-2})^c|\right)\end{aligned}$$

Proof. In round k, $\left(H_{ME}(n, 2^{-k}, 1/16) + \tau_k + \tau_k|(G_{k-1} \cup B_{k-1})^c|\right)$ samples are drawn. Since $G_{k-1} \subset G_k$ and $B_{k-1} \subset B_k$ $\forall k$ deterministically, we see that $|(G_{k-1} \cup B_{k-1})^c| \geqslant |(G_k \cup B_k)^c|$ $\forall k$. By definition,
$H_{ME}(n, 2^{-k-1}, 1/16) = 4H_{ME}(n, 2^{-k}, 1/16)$.

Next, recall $\tau_k = \left\lceil 2^{2k+3} \log\left(\frac{8}{\delta_k}\right)\right\rceil$. We bound τ_k/τ_{k-1} as

$$\begin{aligned}\frac{\tau_k}{\tau_{k-1}} &= \frac{\left\lceil 2^{2k+3} \log\left(\frac{8}{\delta_k}\right)\right\rceil}{\left\lceil 2^{2k+1} \log\left(\frac{8}{\delta_{k-1}}\right)\right\rceil} = \frac{\left\lceil 2^{2k+3} \log\left(\frac{16nk^2}{\delta}\right)\right\rceil}{\left\lceil 2^{2k+1} \log\left(\frac{16n(k-1)^2}{\delta}\right)\right\rceil}\\&\leqslant \frac{2^{2k+3} \log\left(\frac{16nk^2}{\delta}\right) + 1}{2^{2k+1} \log\left(\frac{16n(k-1)^2}{\delta}\right)} \leqslant \frac{4 \log\left(\frac{16nk^2}{\delta}\right)}{\log\left(\frac{16n(k-1)^2}{\delta}\right)} + 1\\&\leqslant 4\frac{\log\left(\frac{16n}{\delta}\right) + 2\log(k)}{\log\left(\frac{16n}{\delta}\right) + 2\log(k-1)} + 1 = (*)\end{aligned}$$

If $k = 2$, $(*) \leqslant 1 + 4 * \log(32)/\log(8) \leqslant 9$. Otherwise,

$$\begin{aligned}(*) &= \frac{4(\log\left(\frac{16n}{\delta}\right) + 2\log(k))}{\log\left(\frac{16n}{\delta}\right) + 2\log(k-1)} + 1\\&\leqslant \frac{4\log(k)}{\log(k-1)} + 1\\&\leqslant 4 \cdot 2 + 1 = 9\end{aligned}$$

Putting these pieces together,

$$\begin{aligned}&\left(H_{ME}(n, 2^{-k}, 1/16) + \tau_k + \tau_k|(G_{k-1} \cup B_{k-1})^c|\right)\\&\leqslant \left(4H_{ME}(n, 2^{-k+1}, 1/16) + 9\tau_{k-1} + 9\tau_{k-1}|(G_{k-2} \cup B_{k-2})^c|\right)\end{aligned}$$

$$\leqslant 9\left(H_{ME}(n, 2^{-k+1}, 1/16) + \tau_{k-1} + \tau_{k-1}|(G_{k-2} \cup B_{k-2})^c|\right)$$

□

6.F.4.7 Step 6: Bounding Equation (6.17)

Here, we introduce the round K_{Good}, when $G_{K_{Good}} = M_\epsilon$ at some point within the round. Using the result of the previous step, we may bound the total number of samples taken though this round, controlling Equation (6.17). With the result of Step 5, we prove the following inequality.

Claim:

$$\begin{aligned}
&\sum_{k=1}^{\infty} 2\mathbb{1}\left[G_{k-1} \neq M_\epsilon\right] \mathbb{1}[G_{k-1} \cup B_{k-1} \neq [n]] \\
&\qquad \left(H_{ME}(n, 2^{-k}, 1/16) + \tau_k + \tau_k|(G_{k-1} \cup B_{k-1})^c|\right) \qquad (6.22)\\
&\leqslant c \sum_{i \in M_\epsilon} \max\left\{h\left(\frac{\epsilon\mu_1 - \Delta_i}{4-2\epsilon}, \frac{\delta}{2n}\right), \min\left[h\left(0.25\Delta_i, \frac{\delta}{2n}\right), h\left(0.25\frac{\tilde{\alpha}_\epsilon}{4-2\epsilon}, \frac{\delta}{2n}\right)\right]\right\} \\
&\quad + c|M_\epsilon^c \cap M_{\epsilon+\tilde{\alpha}_\epsilon/\mu_1}| h\left(\frac{\tilde{\alpha}_\epsilon}{4-2\epsilon}, \frac{\delta}{2n}\right) + c \sum_{i \in M^c_{\epsilon+\tilde{\alpha}_\epsilon/\mu_1}} h\left(\frac{\epsilon\mu_1 - \Delta_i}{4-2\epsilon}, \frac{\delta}{2n}\right)
\end{aligned}$$

Proof. Recall $t_k = \max\{t : t \in k\}$ denotes the maximum value of t in round k and $T_{max} = \max_{\in M_\epsilon} T_i$ denotes the minimum t such that $G_k(t) = M_\epsilon$. Define the random round

$$K_{Good} := \min\{k : G_k = M_\epsilon\} = \min\{k : t_k \geqslant T_{max}\}$$

By definition of K_{Good},

$$\begin{aligned}
&\sum_{k=1}^{\infty} 2\mathbb{1}[G_{k-1} \neq M_\epsilon]\mathbb{1}[G_{k-1} \cup B_{k-1} \neq [n]] \left(H_{ME}(n, 2^{-k}, 1/16) + \tau_k + \tau_k|(G_{k-1} \cup B_{k-1})^c|\right) \\
&\qquad = \sum_{k=1}^{K_{Good}} 2\mathbb{1}[G_{k-1} \cup B_{k-1} \neq [n]] \left(H_{ME}(n, 2^{-k}, 1/16) + \tau_k + \tau_k|(G_{k-1} \cup B_{k-1})^c|\right).
\end{aligned}$$

Next, applying Step 5, if $K_{\text{Good}} > 1$

$$\sum_{k=1}^{K_{\text{Good}}} 2\mathbb{1}[G_{k-1} \cup B_{k-1} \neq [n]] \left(H_{\text{ME}}(n, 2^{-k}, 1/16) + \tau_k + \tau_k |(G_{k-1} \cup B_{k-1})^c|\right)$$
$$\leqslant 18 \sum_{k=1}^{K_{\text{Good}}-1} \mathbb{1}[G_{k-2} \cup B_{k-2} \neq [n]]$$
$$\left(H_{\text{ME}}(n, 2^{-k+1}, 1/16) + \tau_{k-1} + \tau_{k-1} |(G_{k-2} \cup B_{k-2})^c|\right).$$

Observe that by lines 17 and 20 of `FAREAST`, for any round r and for any $t > t_{r-1}$,

$$\sum_{k=1}^{r-1} \mathbb{1}[G_{k-1} \cup B_{k-1} \neq [n]] \left(H_{\text{ME}}(n, 2^{-k}, 1/16) + \tau_k + \tau_k |(G_{k-1} \cup B_{k-1})^c|\right) \leqslant \sum_{s=1}^{t} |\mathcal{A}(s)|.$$

By definition, for the round $K_{\text{Good}} - 1$, we see that $t_{(K_{\text{Good}}-1)} < T_{\max}$. Applying the above inequality with the inequality proven in Step 4,

$$18 \sum_{k=1}^{K_{\text{Good}}-1} |(G_{k-1} \cup B_{k-1})^c| \left(2\tau_k + H_{\text{ME}}(n, 2^{-k}, 1/16)\right) \leqslant 18 \sum_{s=1}^{T_{\max}} |\mathcal{A}(s)|$$
$$\leqslant 18 \sum_{i \in M_\epsilon} \max\left\{h\left(\frac{\epsilon\mu_1 - \Delta_i}{4 - 2\epsilon}, \frac{\delta}{2n}\right), \min\left[h\left(0.25\Delta_i, \frac{\delta}{2n}\right), h\left(\frac{\tilde{\alpha}_\epsilon}{4 - 2\epsilon}, \frac{\delta}{2n}\right)\right]\right\}$$
$$+ 18 \sum_{i \in M^c_{\epsilon + \tilde{\alpha}_\epsilon/\mu_1}} h\left(\frac{\epsilon\mu_1 - \Delta_i}{4 - 2\epsilon}, \frac{\delta}{2n}\right) + 18|M^c_\epsilon \cap M_{\epsilon + \tilde{\alpha}_\epsilon/\mu_1}| h\left(\frac{\tilde{\alpha}_\epsilon}{4 - 2\epsilon}, \frac{\delta}{2n}\right).$$

Otherwise, if $K_{\text{Good}} = 1$, exactly $4c'n\log(16) + 32n\log(16n/\delta)$ samples are given to the good filter in round 1. One may use Lemma 6.32 to invert $h(\cdot, \cdot)$ and show that the summation on the right had side of the above inequality is within a constant of this and the claim holds in this case as well for a different constant, potentially larger than 18. □

6.F.4.8 Step 7: Bounding Equation (6.18)

Next, we bound

$$\sum_{k=1}^{\infty} 2\mathbb{1}\left[G_{k-1} = M_\epsilon\right] \mathbb{1}\left[G_{k-1} \cup B_{k-1} \neq [n]\right] \left(H_{ME}(n, 2^{-k}, 1/16) + \tau_k + \tau_k |(G_{k-1} \cup B_{k-1})^c|\right).$$

$$\begin{aligned}
&\sum_{k=1}^{\infty} 2\mathbb{1}\left[G_{k-1} = M_\epsilon\right] \mathbb{1}\left[G_{k-1} \cup B_{k-1} \neq [n]\right] \left(H_{ME}(n, 2^{-k}, 1/16) + \tau_k + \tau_k |(G_{k-1} \cup B_{k-1})^c|\right) \\
&= \sum_{k=1}^{\infty} 2\mathbb{1}\left[G_{k-1} = M_\epsilon\right] \mathbb{1}\left[G_{k-1} \cup B_{k-1} \neq [n]\right] \left(H_{ME}(n, 2^{-k}, 1/16) + \tau_k + \tau_k |(M_\epsilon \cup B_{k-1})^c|\right) \\
&= \sum_{k=1}^{\infty} 2\mathbb{1}\left[G_{k-1} = M_\epsilon\right] \mathbb{1}\left[G_{k-1} \cup B_{k-1} \neq [n]\right] \left(H_{ME}(n, 2^{-k}, 1/16) + \tau_k + \tau_k |M_\epsilon^c \backslash B_{k-1}|\right) \\
&= \sum_{k=K_{Good}+1}^{\infty} 2\mathbb{1}\left[G_{k-1} \cup B_{k-1} \neq [n]\right] \left(H_{ME}(n, 2^{-k}, 1/16) + \tau_k + \tau_k |M_\epsilon^c \backslash B_{k-1}|\right) \\
&\overset{\mathcal{E}_1, \mathcal{E}_2}{=} \sum_{k=K_{Good}+1}^{\infty} 2\mathbb{1}\left[B_{k-1} \neq M_\epsilon^c\right] \left(H_{ME}(n, 2^{-k}, 1/16) + \tau_k + \tau_k |M_\epsilon^c \backslash B_{k-1}|\right) \\
&= \sum_{k=K_{Good}+1}^{\infty} 2\mathbb{1}\left[B_{k-1} \neq M_\epsilon^c\right] \left(H_{ME}(n, 2^{-k}, 1/16) + \tau_k\right) \\
&\qquad + \sum_{k=K_{Good}+1}^{\infty} 2\mathbb{1}\left[B_{k-1} \neq M_\epsilon^c\right] \left(\tau_k |M_\epsilon^c \backslash B_{k-1}|\right) \\
&= \sum_{k=K_{Good}+1}^{\infty} 2\mathbb{1}\left[B_{k-1} \neq M_\epsilon^c\right] \left(H_{ME}(n, 2^{-k}, 1/16) + \tau_k\right) + \sum_{k=K_{Good}+1}^{\infty} 2\tau_k |M_\epsilon^c \backslash B_{k-1}| \\
&= \sum_{k=K_{Good}+1}^{\infty} 2\mathbb{1}\left[B_{k-1} \neq M_\epsilon^c\right] \left(H_{ME}(n, 2^{-k}, 1/16) + \tau_k\right) + \sum_{k=K_{Good}+1}^{\infty} \sum_{i \in M_\epsilon^c} 2\tau_k \mathbb{1}[i \notin B_{k-1}] \\
&\leqslant \sum_{k=K_{Good}+1}^{\infty} 2|M_\epsilon^c \backslash B_{k-1}| \left(H_{ME}(n, 2^{-k}, 1/16) + \tau_k\right) + \sum_{k=K_{Good}+1}^{\infty} \sum_{i \in M_\epsilon^c} 2\tau_k \mathbb{1}[i \notin B_{k-1}]
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{k=K_{\text{Good}}+1}^{\infty} \sum_{i \in M_\epsilon^c} 2\mathbb{1}[i \notin B_{k-1}] \left(H_{\text{ME}}(n, 2^{-k}, 1/16) + \tau_k\right) \\
&\qquad + \sum_{k=K_{\text{Good}}+1}^{\infty} \sum_{i \in M_\epsilon^c} 2\tau_k \mathbb{1}[i \notin B_{k-1}] \\
&= \sum_{k=K_{\text{Good}}+1}^{\infty} \sum_{i \in M_\epsilon^c} 2\mathbb{1}[i \notin B_{k-1}] \left(2\tau_k + H_{\text{ME}}(n, 2^{-k}, 1/16)\right) \\
&= \sum_{i \in M_\epsilon^c} \sum_{k=K_{\text{Good}}+1}^{\infty} 2\mathbb{1}[i \notin B_{k-1}] \left(2\tau_k + H_{\text{ME}}(n, 2^{-k}, 1/16)\right) \\
&\leqslant \sum_{i \in M_\epsilon^c} \sum_{k=1}^{\infty} 2\mathbb{1}[i \notin B_{k-1}] \left(2\tau_k + H_{\text{ME}}(n, 2^{-k}, 1/16)\right) \qquad (6.23)
\end{aligned}
$$

6.F.4.9 Step 8: Bounding the expected total number of samples drawn by FAREAST

Now we take expectations over the number of samples drawn. These expectations are conditional on the high probability event $\mathcal{E}_1 \cap \mathcal{E}_2$. The bound in step 5 holds deterministically conditioned on this event.

Note τ_k and $H_{\text{ME}}(n, 2^{-k}, 1/16)$ are deterministic constants for any k. Let all expectations are be jointly over the random instance ν and the randomness in FAREAST.

$$
\begin{aligned}
&\mathbb{E}[T|\mathbb{1}[\mathcal{E}_1 \cap \mathcal{E}_2] = 1] = \\
&\quad \sum_{k=1}^{\infty} 2\mathbb{E}\left[\mathbb{1}[G_k \cup B_k \neq [n]] \middle| \mathbb{1}[\mathcal{E}_1 \cap \mathcal{E}_2] = 1\right] \\
&\qquad\qquad \left(H_{\text{ME}}(n, 2^{-k}, 1/16) + \tau_k + \tau_k \middle| (G_{k-1} \cup B_{k-1})^c\right) \\
&\qquad = \sum_{k=1}^{\infty} 2\mathbb{E}\left[\mathbb{1}\left[G_{k-1} \neq M_\epsilon\right] \mathbb{1}[G_k \cup B_k \neq [n]] \middle| \mathbb{1}[\mathcal{E}_1 \cap \mathcal{E}_2] = 1\right] \\
&\qquad\qquad \left(H_{\text{ME}}(n, 2^{-k}, 1/16) + \tau_k + \tau_k \middle| (G_{k-1} \cup B_{k-1})^c\right) \\
&\qquad\quad + \sum_{k=1}^{\infty} 2\mathbb{E}\left[\mathbb{1}\left[G_{k-1} = M_\epsilon\right] \mathbb{1}[G_k \cup B_k \neq [n]] \middle| \mathbb{1}[\mathcal{E}_1 \cap \mathcal{E}_2] = 1\right]
\end{aligned}
$$

$$
\begin{aligned}
&\left(H_{ME}(n, 2^{-k}, 1/16) + \tau_k + \tau_k \middle| (G_{k-1} \cup B_{k-1})^c \right) \\
&\overset{\text{Step 6}}{\leqslant} c \sum_{i \in M_\epsilon} \max\left\{ h\left(\frac{\epsilon\mu_1 - \Delta_i}{4-2\epsilon}, \frac{\delta}{2n}\right), \min\left[h\left(0.25\Delta_i, \frac{\delta}{2n}\right), h\left(\frac{\tilde{\alpha}_\epsilon}{4-2\epsilon}, \frac{\delta}{2n}\right)\right]\right\} \\
&\quad + c \sum_{i \in M^c_{\epsilon + \tilde{\alpha}_\epsilon/\mu_1}} h\left(\frac{\epsilon\mu_1 - \Delta_i}{4-2\epsilon}, \frac{\delta}{2n}\right) + c|M^c_\epsilon \cap M_{\epsilon+\tilde{\alpha}_\epsilon/\mu_1}| h\left(\frac{\tilde{\alpha}_\epsilon}{4-2\epsilon}, \frac{\delta}{2n}\right) \\
&\quad + \sum_{k=1}^{\infty} 2\mathbb{E}\left[\mathbb{1}\left[G_{k-1} = M_\epsilon\right] \mathbb{1}[G_k \cup B_k \neq [n]] \middle| \mathbb{1}[\mathcal{E}_1 \cap \mathcal{E}_2] = 1\right] \\
&\left(H_{ME}(n, 2^{-k}, 1/16) + \tau_k + \tau_k \middle| (G_{k-1} \cup B_{k-1})^c \right) \\
&\overset{\text{Step 7}}{\leqslant} c \sum_{i \in M_\epsilon} \max\left\{ h\left(\frac{\epsilon\mu_1 - \Delta_i}{4-2\epsilon}, \frac{\delta}{2n}\right), \min\left[h\left(0.25\Delta_i, \frac{\delta}{2n}\right), h\left(\frac{\tilde{\alpha}_\epsilon}{4-2\epsilon}, \frac{\delta}{2n}\right)\right]\right\} \\
&\quad + c \sum_{i \in M^c_{\epsilon + \tilde{\alpha}_\epsilon/\mu_1}} h\left(\frac{\epsilon\mu_1 - \Delta_i}{4-2\epsilon}, \frac{\delta}{2n}\right) + c|M^c_\epsilon \cap M_{\epsilon+\tilde{\alpha}_\epsilon/\mu_1}| h\left(\frac{\tilde{\alpha}_\epsilon}{4-2\epsilon}, \frac{\delta}{2n}\right) \\
&\quad + \sum_{i \in M^c_\epsilon} \sum_{k=1}^{\infty} 2\mathbb{E}_\nu\left[\mathbb{1}[i \notin B_{k-1}] \middle| \mathbb{1}[\mathcal{E}_1 \cap \mathcal{E}_2] = 1\right] \left(2\tau_k + H_{ME}(n, 2^{-k}, 1/16)\right) \\
&\overset{(a)}{=} c \sum_{i \in M_\epsilon} \max\left\{ h\left(\frac{\epsilon\mu_1 - \Delta_i}{4-2\epsilon}, \frac{\delta}{2n}\right), \min\left[h\left(0.25\Delta_i, \frac{\delta}{2n}\right), h\left(\frac{\tilde{\alpha}_\epsilon}{4-2\epsilon}, \frac{\delta}{2n}\right)\right]\right\} \\
&\quad + c \sum_{i \in M^c_{\epsilon + \tilde{\alpha}_\epsilon/\mu_1}} h\left(\frac{\epsilon\mu_1 - \Delta_i}{4-2\epsilon}, \frac{\delta}{2n}\right) + c|M^c_\epsilon \cap M_{\epsilon+\tilde{\alpha}_\epsilon}| h\left(\frac{\tilde{\alpha}_\epsilon/\mu_1}{4-2\epsilon}, \frac{\delta}{2n}\right) \\
&\quad + \sum_{i \in M^c_\epsilon} \sum_{k=1}^{\infty} 2\mathbb{E}_\nu\left[\mathbb{1}[i \notin B_{k-1}] | \mathbb{1}[\mathcal{E}_1]\right] \left(2\tau_k + H_{ME}(n, 2^{-k}, 1/16)\right)
\end{aligned}
$$

where (a) follows from $\mathbb{E}_\nu\left[\mathbb{1}[i \notin B_{k-1}] \middle| \mathbb{1}[\mathcal{E}_1 \cap \mathcal{E}_2]\right] = \mathbb{E}_\nu\left[\mathbb{1}[i \notin B_{k-1}] \middle| \mathbb{1}[\mathcal{E}_1]\right]$ for $i \in M^c_\epsilon$, since the event $\{i \in B_{k-1}\}$ is independent of $\mathcal{E}_2$ for all $i \in M^c_\epsilon$. This can be observed since $\mathcal{E}_2$ deals only with independent samples taken of arms in M_ϵ.

6.F.4.10 Step 9: Bounding the expectation remaining from step 8

Next, we bound $\sum_{k=1}^{\infty} \mathbb{E}_\nu\left[\mathbb{1}[i \notin B_{k-1}] | \mathbb{1}[\mathcal{E}_1]\right] \left(2\tau_k + H_{ME}(n, 2^{-k}, 1/16)\right)$ for $i \in M^c_\epsilon$, the expectation remaining from step 8. In particular, this is the number of samples

drawn by the bad filter to add arm $i \in M_\epsilon^c$ to B_k.

First, we bound the probability that for a given $i \in M_\epsilon^c$ and a given k $i \notin B_k$. Note that by Borel-Cantelli, this implies that the probability that i is never added to any B_k is 0.

Claim 1: For $i \in M_\epsilon^c, k \geqslant \left\lceil \log_2 \left(\frac{2-\epsilon}{\Delta_i - \epsilon\mu_1} \right) \right\rceil \implies \mathbb{E}_\nu \left[\mathbb{1}[i \notin B_k] | \mathbb{1}[\mathcal{E}_1] \right] \leqslant \left(\frac{1}{8}\right)^{k - \left\lceil \log_2 \left(\frac{2-\epsilon}{\Delta_i - \epsilon\mu_1} \right) \right\rceil}$

Proof. If $i \in B_k$, either the good or the bad filter may have added it. The behavior of the bad filter on arms inn M_ϵ^c is independent of $\mathcal{E}_1$. Hence.

$$\begin{aligned}
\mathbb{E}_\nu \left[\mathbb{1}[i \notin B_k] | \mathbb{1}[\mathcal{E}_1] \right] &= \mathbb{E}_\nu \left[\mathbb{1}[\hat{\mu}_i + C_{\delta/2n}(t) \geqslant L_{t_k}] \mathbb{1}[\hat{\mu}_{i_k} - \hat{\mu}_i \leqslant 2^{-(k+1)}(2-\epsilon)] | \mathbb{1}[\mathcal{E}_1] \right] \\
&\leqslant \mathbb{E}_\nu \left[\mathbb{1}[\hat{\mu}_{i_k} - \hat{\mu}_i \leqslant 2^{-(k+1)}(2-\epsilon)] | \mathbb{1}[\mathcal{E}_1] \right] \\
&= \mathbb{E}_\nu \left[\mathbb{1}[\hat{\mu}_{i_k} - \hat{\mu}_i \leqslant 2^{-(k+1)}(2-\epsilon)] \right]
\end{aligned}$$

If $i \in B_{k-1}$ then $i \in B_k$ by definition. Otherwise, if $i \notin B_{k-1}$, by Hoeffding's Inequality conditional on the value of i_k and a sum over conditional probabilities as in step 0, with probability at least $1 - \frac{\delta}{4nk^2}$

$$|((1-\epsilon)\hat{\mu}_{i_k} - \hat{\mu}_i) - ((1-\epsilon)\mu_{i_k} - \mu_i)| \leqslant 2^{-(k+1)}$$

If `MedianElimination` also succeeds, the joint event of which occurs with probability $\frac{15}{16}\left(1 - \frac{\delta}{4nk^2}\right)$ by independence[7],

$$\begin{aligned}
(1-\epsilon)\hat{\mu}_{i_k} - \hat{\mu}_i &\geqslant (1-\epsilon)\mu_{i_k} - \mu_i - 2^{-(k+1)} \\
&\geqslant (1-\epsilon)\mu_1 - \mu_i - 2^{-(k+1)}(2-\epsilon) \\
&= \Delta_i - \epsilon\mu_1 - 2^{-(k+1)}(2-\epsilon).
\end{aligned}$$

Then for $k \geqslant \left\lceil \log_2 \left(\frac{2-\epsilon}{\Delta_i - \epsilon\mu_1} \right) \right\rceil$,

$$(1-\epsilon)\hat{\mu}_{i_k} - \hat{\mu}_i \geqslant \Delta_i - \epsilon\mu_1 - 2^{-(k+1)}(2-\epsilon) \geqslant 2^{-(k+1)}(2-\epsilon),$$

[7] Note that the success of `MedianElimination` and the concentration of $(\hat{\mu}_{i_k} - \hat{\mu}_i)$ around $(\mu_{i_k} - \mu_i)$ are independent of the events $\mathcal{E}_1$ and $\mathcal{E}_2$ conditioned on in Step 8.

which implies that $i \in B_k$ by line 15 of FAREAST. In particular,

$$\begin{aligned}\mathbb{E}\left[\mathbb{1}[i \in B_k] \middle| i \notin B_{k-1}\mathbb{1}[\mathcal{E}_1]\right] &\geqslant \mathbb{E}\left[\hat{\mu}_{i_k} - \hat{\mu}_i > 2^{-(k+1)}(2-\epsilon) \middle| i \notin B_{k-1}, \mathbb{1}[\mathcal{E}_1]\right] \\ &\geqslant \frac{15}{16}\left(1 - \frac{\delta}{4nk^2}\right).\end{aligned}$$

Furthermore, $i \notin B_0$ by definition. Then for $k \geqslant \left\lceil \log_2\left(\frac{2-\epsilon}{\Delta_i - \epsilon\mu_1}\right)\right\rceil$,

$$\begin{aligned}\mathbb{E}\left[\mathbb{1}[i \notin B_k] | \mathbb{1}[\mathcal{E}_1]\right] &= \mathbb{E}\left[\mathbb{1}[i \notin B_k](\mathbb{1}[i \notin B_{k-1}] + \mathbb{1}[i \in B_{k-1}]) | \mathbb{1}[\mathcal{E}_1]\right] \\ &= \mathbb{E}\left[\mathbb{1}[i \notin B_k]\mathbb{1}[i \notin B_{k-1}] | \mathbb{1}[\mathcal{E}_1]\right] + \mathbb{E}\left[\mathbb{1}[i \notin B_k]\mathbb{1}[i \in B_{k-1}] | \mathbb{1}[\mathcal{E}_1]\right]\end{aligned}$$

Deterministically, $\mathbb{1}[i \notin B_k]\mathbb{1}[i \in B_{k-1}] = 0$. Therefore,

$$\begin{aligned}&\mathbb{E}\left[\mathbb{1}[i \notin B_k]\mathbb{1}[i \notin B_{k-1}] | \mathbb{1}[\mathcal{E}_1]\right] + \mathbb{E}\left[\mathbb{1}[i \notin B_k]\mathbb{1}[i \in B_{k-1}] | \mathbb{1}[\mathcal{E}_1]\right] \\ &= \mathbb{E}\left[\mathbb{1}[i \notin B_k]\mathbb{1}[i \notin B_{k-1}] | \mathbb{1}[\mathcal{E}_1]\right] \\ &= \mathbb{E}\left[\mathbb{1}[i \notin B_k]\mathbb{1}[i \notin B_{k-1}] | i \notin B_{k-1}, \mathbb{1}[\mathcal{E}_1]\right] \mathbb{P}(i \notin B_{k-1} | \mathbb{1}[\mathcal{E}_1]) \\ &\qquad + \mathbb{E}\left[\mathbb{1}[i \notin B_k]\mathbb{1}[i \notin B_{k-1}] \middle| i \in B_{k-1}, \mathbb{1}[\mathcal{E}_1]\right] \mathbb{P}(i \in B_{k-1} | \mathbb{1}[\mathcal{E}_1]) \\ &= \mathbb{E}\left[\mathbb{1}[i \notin B_k]\mathbb{1}[i \notin B_{k-1}] | i \notin B_{k-1}, \mathbb{1}[\mathcal{E}_1]\right] \mathbb{P}(i \notin B_{k-1} | \mathbb{1}[\mathcal{E}_1]) \\ &= \mathbb{E}\left[\mathbb{1}[i \notin B_k] \middle| i \notin B_{k-1}, \mathbb{1}[\mathcal{E}_1]\right] \mathbb{E}\left[\mathbb{1}[i \notin B_{k-1}] | \mathbb{1}[\mathcal{E}_1]\right] \\ &\leqslant \left(\frac{1}{16} + \frac{\delta}{4nk^2}\right) \mathbb{E}\left[\mathbb{1}[i \notin B_{k-1}] | \mathbb{1}[\mathcal{E}_1]\right].\end{aligned}$$

For $k < \left\lceil \log_2\left(\frac{2-\epsilon}{\Delta_i - \epsilon\mu_1}\right)\right\rceil$, trivially, $\mathbb{E}\left[\mathbb{1}[i \notin B_k] | \mathbb{1}[\mathcal{E}_1]\right] \leqslant 1$. Recall $\delta \leqslant 1/8$. For $k \geqslant \left\lceil \log_2\left(\frac{2-\epsilon}{\Delta_i - \epsilon\mu_1}\right)\right\rceil$,

$$\mathbb{E}\left[\mathbb{1}[i \notin B_k] | \mathbb{1}[\mathcal{E}_1]\right] \leqslant \prod_{s=\left\lceil \log_2\left(\frac{2-\epsilon}{\Delta_i - \epsilon\mu_1}\right)\right\rceil}^{k} \left(\frac{1}{16} + \frac{\delta}{2ns^2}\right) \leqslant \left(\frac{1}{8}\right)^{k - \left\lceil \log_2\left(\frac{2-\epsilon}{\Delta_i - \epsilon\mu_1}\right)\right\rceil}.$$

□

Claim 2: For $j \in M_\epsilon^c$, $\sum_{k=1}^{\infty} 2\mathbb{E}_\nu\left[\mathbb{1}[i \notin B_{k-1}] | \mathbb{1}[\mathcal{E}_1]\right]\left(2\tau_k + H_{ME}(n, 2^{-k}, 1/16)\right) \leqslant$

$c'' \frac{4n(2-\epsilon)^2}{(\Delta_i - \epsilon\mu_1)^2} + c'' h\left(\frac{\Delta_i - \epsilon\mu_1}{4-2\epsilon}, \frac{\delta}{2n}\right)$

Proof. This sum decomposes into two terms.

$$\sum_{k=1}^{\infty} \mathbb{E}_\nu \left[\mathbb{1}[i \notin B_{k-1}] | \mathbb{1}[\mathcal{E}_1]\right] \left(2\tau_k + H_{ME}(n, 2^{-k}, 1/16)\right)$$

$$= \sum_{k=1}^{\left\lfloor \log_2\left(\frac{2-\epsilon}{\Delta_i - \epsilon\mu_1}\right)\right\rfloor} \mathbb{E}_\nu \left[\mathbb{1}[i \notin B_{k-1}] | \mathbb{1}[\mathcal{E}_1]\right] \left(H_{ME}(n, 2^{-k}, 1/16) + 2\left\lceil 2^{2k+3} \log\left(\frac{16nk^2}{\delta}\right)\right\rceil\right)$$

$$+ \sum_{k=\left\lceil \log_2\left(\frac{2-\epsilon}{\Delta_i - \epsilon\mu_1}\right)\right\rceil}^{\infty} \mathbb{E}_\nu \left[\mathbb{1}[i \notin B_{k-1}] | \mathbb{1}[\mathcal{E}_1]\right] \left(H_{ME}(n, 2^{-k}, 1/16) + 2\left\lceil 2^{2k+3} \log\left(\frac{16nk^2}{\delta}\right)\right\rceil\right)$$

We begin by bounding the first term.

$$\sum_{k=1}^{\left\lfloor \log_2\left(\frac{2-\epsilon}{\Delta_i - \epsilon\mu_1}\right)\right\rfloor} \mathbb{E}_\nu \left[\mathbb{1}[i \notin B_{k-1}] | \mathbb{1}[\mathcal{E}_1]\right] \left(H_{ME}(n, 2^{-k}, 1/16) + 2\left\lceil 2^{2k+3} \log\left(\frac{16nk^2}{\delta}\right)\right\rceil\right)$$

$$\leqslant \sum_{k=1}^{\left\lfloor \log_2\left(\frac{2-\epsilon}{\Delta_i - \epsilon\mu_1}\right)\right\rfloor} \left(H_{ME}(n, 2^{-k}, 1/16) + 2\left\lceil 2^{2k+3} \log\left(\frac{16nk^2}{\delta}\right)\right\rceil\right)$$

$$\leqslant \sum_{k=1}^{\left\lfloor \log_2\left(\frac{2-\epsilon}{\Delta_i - \epsilon\mu_1}\right)\right\rfloor} \left(c' n 2^{2k} \log(16) + 2 + 2^{2k+4} \log\left(\frac{16nk^2}{\delta}\right)\right)$$

$$\leqslant 2\log_2\left(\frac{2-\epsilon}{\Delta_i - \epsilon\mu_1}\right) + \left(c' n \log(16) + 16 \log\left(\frac{16n}{\delta}\right)\right) \sum_{k=1}^{\left\lfloor \log_2\left(\frac{2-\epsilon}{\Delta_i - \epsilon\mu_1}\right)\right\rfloor} 2^{2k}$$

$$+ 32 \sum_{k=1}^{\left\lfloor \log_2\left(\frac{2-\epsilon}{\Delta_i - \epsilon\mu_1}\right)\right\rfloor} 2^{2k} \log(k)$$

$$\leqslant 2\log_2\left(\frac{2-\epsilon}{\Delta_i - \epsilon\mu_1}\right)$$

$$+ \left(c' n \log(16) + 16 \log\left(\frac{16n}{\delta}\right) + 32 \log\log_2\left(\frac{2-\epsilon}{\Delta_i - \epsilon\mu_1}\right)\right) \sum_{k=1}^{\left\lfloor \log_2\left(\frac{2-\epsilon}{\Delta_i - \epsilon\mu_1}\right)\right\rfloor} 2^{2k}$$

$$\leqslant 2\log_2\left(\frac{2-\epsilon}{\Delta_i-\epsilon\mu_1}\right)+ \frac{(2-\epsilon)^2}{(\Delta_i-\epsilon\mu_1)^2}\left(c'n\log(16)+32\log\left(\frac{16n}{\delta}\log_2\left(\frac{2-\epsilon}{\Delta_i-\epsilon\mu_1}\right)\right)\right)$$

Next, we plug in the bound from claim 1 controlling the probability that $i \notin B_k$. Using Claim 1, we bound the second sum as follows:

$$\sum_{r=\left\lceil\log_2\left(\frac{2-\epsilon}{\Delta_i-\epsilon\mu_1}\right)\right\rceil}^{\infty}\mathbb{E}_\nu\left[\mathbb{1}[i\notin B_{k-1}]|\mathbb{1}[\mathcal{E}_1]\right]\left(H_{ME}(n,2^{-k},1/16)+2\left\lceil 2^{2k+3}\log\left(\frac{16nk^2}{\delta}\right)\right\rceil\right)$$

$$\leqslant \sum_{k=\left\lceil\log_2\left(\frac{2-\epsilon}{\Delta_i-\epsilon\mu_1}\right)\right\rceil}^{\infty}\left(\frac{1}{8}\right)^{k-\left\lceil\log_2\left(\frac{2-\epsilon}{\Delta_i-\epsilon\mu_1}\right)\right\rceil-1}\left(c'n2^{2k}\log(16)+2+2^{2k+4}\log\left(\frac{16nk^2}{\delta}\right)\right)$$

$$= c'n\log(16)\sum_{k=1}^{\infty}\left(\frac{1}{8}\right)^{k-1}2^{2\left(k+\left\lceil\log_2\left(\frac{2-\epsilon}{\Delta_i-\epsilon\mu_1}\right)\right\rceil\right)}+2\sum_{k=1}^{\infty}\left(\frac{1}{8}\right)^{k-1}$$
$$+16\sum_{k=1}^{\infty}\left(\frac{1}{8}\right)^{k-1}2^{2\left(k+\left\lceil\log_2\left(\frac{2-\epsilon}{\Delta_i-\epsilon\mu_1}\right)\right\rceil\right)}\log\left(\frac{16n\left(k+\left\lceil\log_2\left(\frac{2-\epsilon}{\Delta_i-\epsilon\mu_1}\right)\right\rceil\right)^2}{\delta}\right)$$
$$+16\sum_{k=1}^{\infty}2^{-3k+3}2^{2\left(k+\log_2\left(\frac{2-\epsilon}{\Delta_i-\epsilon\mu_1}\right)+1\right)}\log\left(\frac{16n\left(k+\left\lceil\log_2\left(\frac{2-\epsilon}{\Delta_i-\epsilon\mu_1}\right)\right\rceil\right)^2}{\delta}\right)$$

$$= 3+\left(c'n\log(16)\frac{2^5(2-\epsilon)^2}{(\Delta_i-\epsilon\mu_1)^2}+\frac{2^9(2-\epsilon)^2}{(\Delta_i-\epsilon\mu_1)^2}\log\left(\frac{16n}{\delta}\right)\right)\sum_{k=1}^{\infty}2^{-k}$$
$$+\frac{2^9(2-\epsilon)^2}{(\Delta_i-\epsilon\mu_1)^2}\sum_{k=1}^{\infty}2^{-k}\log\left(\left(k+\left\lceil\log_2\left(\frac{2-\epsilon}{\Delta_i-\epsilon\mu_1}\right)\right\rceil\right)^2\right)$$

$$\leqslant 3+c'n\log(16)\frac{2^5(2-\epsilon)^2}{(\Delta_i-\epsilon\mu_1)^2}+\frac{2^9(2-\epsilon)^2}{(\Delta_i-\epsilon\mu_1)^2}\log\left(\frac{16n}{\delta}\right)$$
$$+\frac{2^{10}(2-\epsilon)^2}{(\Delta_i-\epsilon\mu_1)^2}\sum_{k=1}^{\infty}2^{-k}\log\left(k+\left\lceil\log_2\left(\frac{2-\epsilon}{\Delta_i-\epsilon\mu_1}\right)\right\rceil\right)$$

$$= (**)$$

We may bound the final summand, $\sum_{k=1}^{\infty} 2^{-k} \log\left(k + \left\lceil \log_2\left(\frac{2-\epsilon}{\Delta_i - \epsilon\mu_1}\right)\right\rceil\right)$ as follows:

$$\sum_{k=1}^{\infty} 2^{-k} \log\left(k + \left\lceil \log_2\left(\frac{2-\epsilon}{\Delta_i - \epsilon\mu_1}\right)\right\rceil\right) \leqslant \log\left(\frac{e}{2}\log_2\left(\frac{16(2-\epsilon)^2}{(\Delta_i - \epsilon\mu_1)^2}\right)\right)$$

Plugging this back into $(**)$, we have that

$$\begin{aligned}(**) \leqslant 3 + c'n\log(16)\frac{2^5(2-\epsilon)^2}{(\Delta_i - \epsilon\mu_1)^2} + \frac{2^9(2-\epsilon)^2}{(\Delta_i - \epsilon\mu_1)^2}\log\left(\frac{16n}{\delta}\right) \\ + \frac{2^{10}(2-\epsilon)^2}{(\Delta_i - \epsilon\mu_1)^2}\log\left(\frac{e}{2}\log_2\left(\frac{16(2-\epsilon)^2}{(\Delta_i - \epsilon\mu_1)^2}\right)\right)\end{aligned}$$

Combining the above with the bound on the first sum, we have that

$$\begin{aligned}&\sum_{k=1}^{\infty}\mathbb{E}_\nu\left[\mathbb{1}[i \notin B_{k-1}]\mathbb{1}[\mathcal{E}_1]\right]\left(2\tau_k + H_{ME}(n, 2^{-k}, 1/16)\right) \\ &\quad\leqslant c''\left(\frac{4n(2-\epsilon)^2}{(\Delta_i - \epsilon\mu_1)^2} + \frac{4c(2-\epsilon)^2}{(\Delta_i - \epsilon\mu_1)^2}\log\left(\frac{2n}{\delta}\log_2\left(\frac{4-2\epsilon}{(\Delta_i - \epsilon\mu_1)^2}\right)\right)\right) \\ &\quad= \frac{4c''n(2-\epsilon)^2}{(\Delta_i - \epsilon\mu_1)^2} + c''h\left(\frac{\Delta_i - \epsilon\mu_1}{4-2\epsilon}, \frac{\delta}{2n}\right)\end{aligned}$$

for a sufficiently large, universal constant c'' and c from the definition of $h(\cdot,\cdot)$. □

6.F.4.11 Step 10: Applying the result of Step 9 to the result of Step 8

We may repeat the result of step 9 for every $i \in M_\epsilon^c$ and plug this into the result of Step 8. From this point, we simplify to return the final result.

By Step 8, the total number of samples T drawn by `FAREAST` is bounded in expectation by

$$\begin{aligned}\mathbb{E}[T|\mathcal{E}_1 \cap \mathcal{E}_2] \leqslant c\sum_{i\in M_\epsilon}\max\left\{h\left(\frac{\epsilon\mu_1 - \Delta_i}{4-2\epsilon}, \frac{\delta}{2n}\right), \min\left[h\left(0.25\Delta_i, \frac{\delta}{2n}\right), h\left(\frac{\tilde{\alpha}_\epsilon/\mu_1}{4-2\epsilon}, \frac{\delta}{2n}\right)\right]\right\} \\ + c\sum_{i\in M^c_{\epsilon+\tilde{\alpha}_\epsilon/\mu_1}} h\left(\frac{\epsilon\mu_1 - \Delta_i}{4-2\epsilon}, \frac{\delta}{2n}\right) + c|M_\epsilon^c \cap M_{\epsilon+\tilde{\alpha}_\epsilon}|h\left(\frac{\tilde{\alpha}_\epsilon}{4-2\epsilon}, \frac{\delta}{2n}\right)\end{aligned}$$

$$+ 2 \sum_{i \in M_\epsilon^c} \sum_{k=1}^{\infty} \mathbb{E}_\nu \left[\mathbb{1}[i \notin B_{k-1}] | \mathbb{1}[\mathcal{E}_1] \right] \left(2\tau_k + H_{ME}(n, 2^{-k}, 1/16) \right).$$

Applying the bound from Step 9 to each $i \in M_\epsilon^c$, we have that

$$\begin{aligned}
\mathbb{E}[T|\mathcal{E}_1 \cap \mathcal{E}_2] \leqslant c \sum_{i \in M_\epsilon} \max \left\{ h\left(0.25(\epsilon - \Delta_i), \frac{\delta}{2n}\right), \min \left[h\left(0.25\Delta_i, \frac{\delta}{2n}\right), h\left(\frac{\tilde{\alpha}_\epsilon}{4-2\epsilon}, \frac{\delta}{2n}\right) \right] \right\} \\
+ c \sum_{i \in M_{\epsilon+\tilde{\alpha}_\epsilon/\mu_1}^c} h\left(0.25(\epsilon - \Delta_i), \frac{\delta}{2n}\right) + c|M_\epsilon^c \cap M_{\epsilon+\tilde{\alpha}_\epsilon/\mu_1}| h\left(\frac{\tilde{\alpha}_\epsilon}{4-2\epsilon}, \frac{\delta}{2n}\right) \\
+ 2c'' \sum_{i \in M_\epsilon^c} \frac{4n(2-\epsilon)^2}{(\Delta_i - \epsilon\mu_1)^2} + h\left(\frac{\Delta_i - \epsilon\mu_1}{4-2\epsilon}, \frac{\delta}{2n}\right).
\end{aligned}$$

For $i \in M_\epsilon^c \cap M_{\epsilon+\tilde{\alpha}_\epsilon/\mu_1}$, $\tilde{\alpha}_\epsilon = \min_{j \in M_\epsilon} \epsilon\mu_1 - \Delta_j \geqslant \Delta_i - \epsilon\mu_1$. By monotonicity of $h(\cdot,\cdot)$, $h\left(\frac{\tilde{\alpha}_\epsilon}{4-2\epsilon}, \frac{\delta}{2n}\right) \leqslant \frac{c''n(4-2\epsilon)}{(\Delta_i - \epsilon\mu_1)^2} + c'' h\left(\frac{\Delta_i - \epsilon\mu_1}{4-2\epsilon}, \frac{\delta}{2n}\right)$. Therefore,

$$\begin{aligned}
\mathbb{E}[T|\mathcal{E}_1 \cap \mathcal{E}_2] \leqslant c \sum_{i \in M_\epsilon} \max \left\{ h\left(\frac{\Delta_i - \epsilon\mu_1}{4-2\epsilon}, \frac{\delta}{2n}\right), \min \left[h\left(0.25\Delta_i, \frac{\delta}{2n}\right), h\left(\frac{\tilde{\alpha}_\epsilon}{4-2\epsilon}, \frac{\delta}{2n}\right) \right] \right\} \\
+ (2c'' + c) \sum_{i \in M_\epsilon^c} \frac{n(4-2\epsilon)}{(\Delta_i - \epsilon\mu_1)^2} + h\left(\frac{\Delta_i - \epsilon\mu_1}{4-2\epsilon}, \frac{\delta}{2n}\right).
\end{aligned}$$

Lastly, note that $\frac{1}{3(1-x)} \leqslant \frac{1}{2-x}$ for $x \leqslant 1/2$. By monotonicity of h, we may lower bound the denominators $\frac{1}{4-2\epsilon}$ and $\frac{1}{2(2-\epsilon+\gamma)}$ as $\frac{1}{6(1-\epsilon)}$ and $\frac{1}{6(1-\epsilon+\gamma)}$ respectively. Since $\epsilon \in (0, 1/2]$, $\frac{1}{4-2\epsilon} \leqslant 1/4$. Plugging this in, we see that

$$\begin{aligned}
\mathbb{E}[T|\mathcal{E}_1 \cap \mathcal{E}_2] \leqslant c \sum_{i \in M_\epsilon} \max \left\{ h\left(\frac{\Delta_i - \epsilon\mu_1}{4}, \frac{\delta}{2n}\right), \min \left[h\left(0.25\Delta_i, \frac{\delta}{2n}\right), h\left(\frac{\tilde{\alpha}_\epsilon}{6(1-\epsilon)}, \frac{\delta}{2n}\right) \right] \right\} \\
+ (2c'' + c) \sum_{i \in M_\epsilon^c} \frac{4n}{(\Delta_i - \epsilon\mu_1)^2} + h\left(\frac{\Delta_i - \epsilon\mu_1}{4}, \frac{\delta}{2n}\right).
\end{aligned}$$

Next, we use Lemma 6.33 to bound the minimum of $h(\cdot,\cdot)$ functions.

$$
\begin{aligned}
& c \sum_{i \in M_\epsilon} \max \left\{ h\left(\frac{\Delta_i - \epsilon\mu_1}{4}, \frac{\delta}{2n}\right), \min\left[h\left(0.25\Delta_i, \frac{\delta}{2n}\right), h\left(\frac{\tilde{\alpha}_\epsilon}{6(1-\epsilon)}, \frac{\delta}{2n}\right)\right]\right\} \\
& \qquad + (2c'' + c) \sum_{i \in M_\epsilon^c} \frac{4n}{(\Delta_i - \epsilon\mu_1)^2} + h\left(\frac{\Delta_i - \epsilon\mu_1}{4}, \frac{\delta}{2n}\right) \\
& = c \sum_{i \in M_\epsilon} \max \left\{ h\left(\frac{\Delta_i - \epsilon\mu_1}{4}, \frac{\delta}{2n}\right), h\left(\frac{\Delta_i + \frac{\tilde{\alpha}_\epsilon}{1-\epsilon}}{12}, \frac{\delta}{2n}\right)\right\} \\
& \qquad + (2c'' + c) \sum_{i \in M_\epsilon^c} \frac{4n}{(\Delta_i - \epsilon\mu_1)^2} + h\left(\frac{\Delta_i - \epsilon\mu_1}{4}, \frac{\delta}{2n}\right)
\end{aligned}
$$

Finally, we use Lemma 6.32 to bound the function $h(\cdot,\cdot)$. Since $\delta \leqslant 1/2$, $\delta/n \leqslant 2e^{-e/2}$. Further, $|\epsilon\mu_1 - \Delta_i| \leqslant 6$ for all i and $\epsilon \leqslant 1/2$ implies that $\frac{1}{6(1-\epsilon)}|\epsilon\mu_1 - \Delta_i| \leqslant 2$ and $\frac{1}{6(1-\epsilon)} \min(\tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon) \leqslant 2$. $\Delta_i \leqslant 8$ for all i, gives $0.25\Delta_i \leqslant 2$. Lastly, $\gamma \leqslant 6/\mu_1$ implies that $\frac{\gamma\mu_1}{6(1-\epsilon+\gamma)} \leqslant 2$. Therefore,

$$
\begin{aligned}
\mathbb{E}[T|\mathcal{E}_1 \cap \mathcal{E}_2] & \leqslant c \sum_{i \in M_\epsilon} \max \left\{ h\left(\frac{\Delta_i - \epsilon\mu_1}{4}, \frac{\delta}{2n}\right), h\left(\frac{\Delta_i + \frac{\tilde{\alpha}_\epsilon}{1-\epsilon}}{12}, \frac{\delta}{2n}\right)\right\} \\
& \qquad + (2c'' + c) \sum_{i \in M_\epsilon^c} \frac{4n}{(\Delta_i - \epsilon\mu_1)^2} + h\left(\frac{\Delta_i - \epsilon\mu_1}{4}, \frac{\delta}{2n}\right) \\
& \leqslant c \sum_{i \in M_\epsilon} \max \left\{ \frac{64}{(\epsilon\mu_1 - \Delta_i)^2} \log\left(\frac{4n}{\delta}\log_2\left(\frac{384n}{\delta(\epsilon\mu_1 - \Delta_i)^2}\right)\right), \right. \\
& \qquad \left. \frac{576}{\left(\Delta_i + \frac{\tilde{\alpha}_\epsilon}{1-\epsilon}\right)^2} \log\left(\frac{4n}{\delta}\log_2\left(\frac{1728n}{\delta\left(\Delta_i + \frac{\tilde{\alpha}_\epsilon}{1-\epsilon}\right)^2}\right)\right)\right\} \\
& \qquad + (2c'' + c) \sum_{i \in M_\epsilon^c} \frac{4n}{(\Delta_i - \epsilon\mu_1)^2} + \frac{64}{(\epsilon\mu_1 - \Delta_i)^2} \log\left(\frac{4n}{\delta}\log_2\left(\frac{384n}{\delta(\epsilon\mu_1 - \Delta_i)^2}\right)\right) \\
& \leqslant c_6 \sum_{i=1}^{n} \max \left\{ \frac{1}{(\epsilon\mu_1 - \Delta_i)^2} \log\left(\frac{n}{\delta}\log_2\left(\frac{n}{\delta(\epsilon\mu_1 - \Delta_i)^2}\right)\right), \right. \\
& \qquad \left. \frac{1}{\left(\Delta_i + \frac{\tilde{\alpha}_\epsilon}{1-\epsilon}\right)^2} \log\left(\frac{n}{\delta}\log_2\left(\frac{n}{\delta\left(\Delta_i + \frac{\tilde{\alpha}_\epsilon}{1-\epsilon}\right)^2}\right)\right)\right\}
\end{aligned}
$$

$$
\begin{aligned}
&+ c_6 \sum_{i \in M_\epsilon^c} \frac{n}{(\Delta_i - \epsilon\mu_1)^2} \\
&= c_6 \sum_{i=1}^{n} \max \left\{ \frac{1}{((1-\epsilon)\mu_1 - \mu_i)^2} \log \left(\frac{n}{\delta} \log_2 \left(\frac{n}{\delta((1-\epsilon)\mu_1 - \mu_i)^2} \right) \right), \right. \\
&\qquad \left. \frac{1}{\left(\mu_1 + \frac{\tilde{\alpha}_\epsilon}{1-\epsilon} - \mu_i\right)^2} \log \left(\frac{n}{\delta} \log_2 \left(\frac{n}{\delta \left(\mu_1 + \frac{\tilde{\alpha}_\epsilon}{1-\epsilon} - \mu_i\right)^2} \right) \right) \right\} \\
&\quad + c_6 \sum_{i \in M_\epsilon^c} \frac{n}{((1-\epsilon)\mu_1 - \mu_i)^2}
\end{aligned}
$$

for a sufficiently large constant c_6.

6.F.4.12 Step 11: High probability sample complexity bound

Finally, the Good Filter is equivalent to `EAST`, Algorithm 10, except split across rounds. `EAST` is an elimination algorithm. Note that the Good Filter is union bounded over $2n$ events whereas the bounds in `EAST` are union bounded over n events. The Good Filter and Bad Filter are given the same number of samples in each round, and the Good Filter can terminate within a round, conditioned on $\mathcal{E}_1 \cap \mathcal{E}_2$. Therefore, we can bound the complexity of `FAREAST` in terms of that of `EAST` run at failure probability $\delta/2$. If `FAREAST` terminates in the second round or later, the arguments in Steps 4 and 5 can be used to show that `FAREAST` draws no more than a factor of 18 more samples than `EAST`, though this estimate is highly pessimistic. If `FAREAST` terminates in round 1 (when gaps are large), we may still show that this is within a constant factor of the complexity of `EAST`, but the story is more complicated. In the first round, the bad filter draws at most $c'n\log(16) + 16(n+1)\log(8n/\delta)$ samples where c' is the constant from `Median Elimination`. Since we have assumed that $\max(\Delta_i, |\epsilon\mu_1 - \Delta_i|) \leqslant 6(1-\epsilon) \leqslant 6$, this sum is likewise within a constant factor of the complexity of `EAST`. Hence with probability at least $1-\delta$, by Theorem 6.28,

$$
T \leqslant c_5 \sum_{i=1}^{n} \min \left\{ \max \left\{ \frac{1}{((1-\epsilon)\mu_1 - \mu_i)^2} \log \left(\frac{n}{\delta} \log_2 \left(\frac{n}{\delta((1-\epsilon)\mu_1 - \mu_i)^2} \right) \right), \right.
$$

$$\frac{}{(\mu_1 + \frac{\tilde{\alpha}_\epsilon}{1-\epsilon} - \mu_i)^2} \log\left(\frac{n}{\delta}\log_2\left(\frac{n}{\delta(\mu_1 + \frac{\tilde{\alpha}_\epsilon}{1-\epsilon})^2}\right)\right),$$
$$\frac{1}{(\mu_1 + \frac{\tilde{\beta}_\epsilon}{1-\epsilon} - \mu_i)^2} \log\left(\frac{n}{\delta}\log_2\left(\frac{n}{\delta(\mu_1 + \frac{\tilde{\beta}_\epsilon}{1-\epsilon} - \mu_i)^2}\right)\right)\Bigg\},$$
$$\frac{(1-\epsilon+\gamma)^2}{\gamma^2\mu_1^2} \log\left(\frac{n}{\delta}\log_2\left(\frac{(1-\epsilon+\gamma)^2 n}{\delta\gamma^2\mu_1^2}\right)\right)\Bigg\}$$

samples for a sufficiently large constant c_5. □

6.F.5 An elimination algorithm for all ϵ

First, we state an elimination algorithm EAST (**E**limination **A**lgorithm for a **S**ampled **T**hreshold) and bound its sample complexity. EAST is equivalent to the good filter in FAREAST. At all times, EAST maintains an active set $\mathcal{A}$ and samples all arms $i \in \mathcal{A}$, progressively eliminating arms from $\mathcal{A}$ until termination occurs. Additionally, EAST maintains upper and lower bounds, denoted U_t and L_t, on the the threshold, $\mu_1 - \epsilon$ in the additive case and $(1-\epsilon)\mu_1$ in the multiplicative case. If $\hat{\mu}_i(t) + C_{\delta/n}(t) < L_t$, EAST may infer that $i \notin G_\epsilon$ (resp. $i \notin M_\epsilon$) and accordingly removes i from $\mathcal{A}$. If $\hat{\mu}_i(t) - C_{\delta/n}(t) > U_t$, EAST may infer that $i \in G_\epsilon$ (resp. $i \in M_\epsilon$) and adds i to a set G of good arms it has found so far. However, a good arm $i \in G$ is only removed from $\mathcal{A}$, if EAST can also certify that it is not the best arm, namely if $\hat{\mu}_i(t) + C_{\delta/n}(t) < \max_j \hat{\mu}_j(t) - C_{\delta/n}(t)$. This ensures that $\mu_1 - \epsilon \in [L_t, U_t]$ at all times in the additive case, and similarly, $(1-\epsilon)\mu_1 \in [L_t, U_t]$ in the multiplicative case. If $\mathcal{A} \subset G$, EAST may declare that $G = G_\epsilon$ (resp. $G = M_\epsilon$) and terminates. Otherwise, the algorithm terminates when $U_t - L_t < \gamma/2$ and returns $\mathcal{A} \cup G$ in the additive case or when $U_t - L_t < \frac{\gamma}{2-\epsilon} L_t$ in the multiplicative case. This limits the number of samples of any arm and ensures that no arm worse than $(\epsilon+\gamma)$-good is returned. We give pseudocode for EAST in Algorithm 10. Pieces specific to the additive case are shown in red, and pieces specific to the multiplicative case are shown in blue.

Algorithm 10 `EAST` : **E**limination **A**lgorithm for a **S**ampled **T**hreshold

Require: $\epsilon, \delta > 0$, slack $\gamma \geqslant 0$, (if multiplicative, $0 < \epsilon \leqslant 1/2$)

Let $\mathcal{A} \leftarrow [n]$ be the active set, and $G \leftarrow \emptyset$ be the set of ϵ-good arms found so far,
Let $t \leftarrow 0$
while $\mathcal{A} \not\subset G$ and $U_t - L_t \geqslant \gamma/2$ or $U_t - L_t \geqslant \frac{\gamma}{2-\epsilon} L_t$ **do**
  Pull each arm $i \in \mathcal{A}$ and update its empirical mean $\hat{\mu}_i(t)$, Update $t \leftarrow t+1$
  Update $U_t \leftarrow \max_j \hat{\mu}_j(t) + C_{\delta/n}(t) - \epsilon$ or $U_t \leftarrow (1-\epsilon)\left(\max_j \hat{\mu}_j(t) + C_{\delta/n}(t)\right)$
  Update $L_t \leftarrow \max_j \hat{\mu}_j(t) - C_{\delta/n}(t) - \epsilon$ or $L_t \leftarrow (1-\epsilon)\left(\max_j \hat{\mu}_j(t) - C_{\delta/n}(t)\right)$

  for $i \in \mathcal{A}$ **do**
    if $\hat{\mu}_i(t) - C_{\delta/n}(t) > U_t$ **then**
      add i to G ▷ Arm i is good
    end if
    if $\hat{\mu}_i(t) + C_{\delta/n}(t) < L_t$ **then**
      Remove i from $\mathcal{A}$ ▷ Arms in G_ϵ^c or M_ϵ^c are removed
    end if
    if $i \in G$ and $\hat{\mu}_i(t) + C_{\delta/n}(t) < \max_j \hat{\mu}_j(t) - C_{\delta/n}(t)$ **then**
      Remove i from $\mathcal{A}$ ▷ Arms in G_ϵ or M_ϵ are removed
    end if
  end for
end while **return** $G \cup \mathcal{A}$

Theorem 6.27. *Fix $\epsilon > 0$, $0 < \delta \leqslant 1/2$, $\gamma \in [0, 8]$ and an instance ν such that $\max(\Delta_i, |\epsilon - \Delta_i|) \leqslant 8$ for all i. In the case that $G_\epsilon = [n]$, let $\alpha_\epsilon = \min(\alpha_\epsilon, \beta_\epsilon)$. With probability at least $1-\delta$, `EAST` returns a set G such that $G_\epsilon \subset G \subset G_{(\epsilon+\gamma)}$ in at most*

$$\sum_{i=1}^{n} \min\left\{ \max\left\{ \frac{64}{(\mu_1 - \epsilon - \mu_i)^2} \log\left(\frac{2n}{\delta} \log_2\left(\frac{768n}{\delta(\mu_1 - \epsilon - \mu_i)^2}\right)\right), \right.\right.$$
$$\frac{256}{(\mu_1 + \alpha_\epsilon - \mu_i)^2} \log\left(\frac{2n}{\delta} \log_2\left(\frac{768n}{\delta(\mu_1 + \alpha_\epsilon - \mu_i)^2}\right)\right),$$
$$\left.\left. \frac{256}{(\mu_1 + \beta_\epsilon - \mu_i)^2} \log\left(\frac{2n}{\delta} \log_2\left(\frac{768n}{\delta(\mu_1 + \beta_\epsilon - \mu_i)^2}\right)\right) \right\}, \right.$$
$$\left. \frac{64}{\gamma^2} \log\left(\frac{2n}{\delta} \log_2\left(\frac{192n}{\delta\gamma^2}\right)\right) \right\}$$

samples.

Next, we a theorem bounding the complexity of EAST in the multiplicative regime.

Theorem 6.28. *Fix* $\epsilon, \delta \in (0, 1/2]$, $\gamma \in [0, \min(1, 6/\mu_1))$ *and an instance* ν *such that* $\max(\Delta_i, |\epsilon\mu_1 - \Delta_i|) \leqslant 6$ *for all* i. *Assume that* $\mu_1 \geqslant 0$. *In the case that* $M_\epsilon = [n]$, *let* $\tilde{\alpha}_\epsilon = \min(\tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon)$. *With probability at least* $1 - \delta$, *EAST returns a set* G *such that* $M_\epsilon \subset G \subset M_{(\epsilon+\gamma)}$ *in at most*

$$\sum_{i=1}^{n} \min\left\{\max\left\{\frac{64}{((1-\epsilon)\mu_1 - \mu_i)^2}\log\left(\frac{2n}{\delta}\log_2\left(\frac{192n}{\delta((1-\epsilon)\mu_1 - \mu_i)^2}\right)\right),\right.\right.$$
$$\frac{576}{(\mu_1 + \frac{\tilde{\alpha}_\epsilon}{1-\epsilon} - \mu_i)^2}\log\left(\frac{2n}{\delta}\log_2\left(\frac{1728n}{\delta(\mu_1 + \frac{\tilde{\alpha}_\epsilon}{1-\epsilon})^2}\right)\right),$$
$$\left.\left.\frac{576}{(\mu_1 + \frac{\tilde{\beta}_\epsilon}{1-\epsilon} - \mu_i)^2}\log\left(\frac{2n}{\delta}\log_2\left(\frac{1728n}{\delta(\mu_1 + \frac{\tilde{\beta}_\epsilon}{1-\epsilon} - \mu_i)^2}\right)\right)\right\},\right.$$
$$\left.\frac{144(1-\epsilon+\gamma)}{\gamma^2\mu_1^2}\log\left(\frac{2n}{\delta}\log_2\left(\frac{432(1-\epsilon+\gamma)n}{\delta\gamma^2\mu_1^2}\right)\right)\right\}$$

samples.

6.F.6 Proof of Theorem 6.27 EAST in the additive regime

Proof. **Notation for the proof:** Throughout, recall $\Delta_i = \mu_1 - \mu_i$. Recall that t counts the number of times each arm in $\mathcal{A}$ has been sampled and thus the number of times that the conditionals in Lines 6.H.2 and 6.H.2 have been evaluated. Let $\mathcal{A}(t)$ denote the state $\mathcal{A}$ at this time before the arms have been eliminated from $\mathcal{A}$ in lines 6.H.2 and 6.H.2. Let $G(t)$ be defined similarly. Therefore, the total number of samples drawn by EAST up to time t is $\sum_{s=1}^{t} |\mathcal{A}(s)|$.

For $i \in G_\epsilon$, let T_i denote the random variable of the number of times arm i is sampled before it is added to G in Line 8. For $i \in G_\epsilon^c$, let T_i denote the random variable of the number of times arm i is sampled before it is removed from $\mathcal{A}$ in Line 6.H.2. For any arm i, let T_i' denote the random variable of the of the number

of times i is sampled before $\hat{\mu}_i(t) + C_{\delta/n}(t) \leqslant \max_{j \in \mathcal{A}} \hat{\mu}_j(t) - C_{\delta/n}(t)$.

Define the event

$$\mathcal{E} = \left\{ \bigcap_{i \in [n]} \bigcap_{t \in \mathbb{N}} |\hat{\mu}_i(t) - \mu_i| \leqslant C_{\delta/n}(t) \right\}.$$

Using standard anytime confidence bound results, and recalling that that $C_\delta(t) := \sqrt{\frac{4\log(\log_2(2t)/\delta)}{t}}$, we have

$$\begin{aligned}\mathbb{P}(\mathcal{E}^c) &= \mathbb{P}\left(\bigcup_{i \in [n]} \bigcup_{t \in \mathbb{N}} |\hat{\mu}_i - \mu_i| > C_{\delta/n}(t) \right) \\ &\leqslant \sum_{i=1}^{n} \mathbb{P}\left(\bigcup_{t \in \mathbb{N}} |\hat{\mu}_i - \mu_i| > C_{\delta/n}(t) \right) \leqslant \sum_{i=1}^{n} \frac{\delta}{n} = \delta\end{aligned}$$

Hence, $\mathbb{P}(\mathcal{E}) \geqslant 1 - \delta$.

6.F.6.1 Step 0: Correctness

Claim 0: On $\mathcal{E}$, first we prove that $G(t) \subset G_\epsilon$ for all $t \in \mathbb{N}$.

In particular, this shows that `EAST` never incorrectly add arms in G_ϵ^c to the set G.

Proof. We begin by showing that on $\mathcal{E}$ the best arm is never removed from $\mathcal{A}$ for all t. Note for any i

$$\hat{\mu}_1 + C_{\delta/n}(t) \geqslant \mu_1 \geqslant \mu_i \geqslant \hat{\mu}_i(t) - C_{\delta/n}(t) > \hat{\mu}_i(t) - C_{\delta/n}(t) - \epsilon.$$

In particular this shows, $\hat{\mu}_1 + C_{\delta/n}(t) > \max_{i \in \mathcal{A}} \hat{\mu}_i(t) - C_{\delta/n}(t) - \epsilon = L_t^*$ and $\hat{\mu}_1 + C_{\delta/n}(t) \geqslant \max_{i \in \mathcal{A}} \hat{\mu}_i(t) - C_{\delta/n}(t)$ showing that 1 will never exit $\mathcal{A}$ in line 6.H.2.

Secondly, we show that at all times t, $\mu_1 - \epsilon \in [L_t, U_t]$. By the above, since μ_1

never leaves $\mathcal{A}$,

$$U_t = \max_{i \in \mathcal{A}} \hat{\mu}_i(t) + C_{\delta/n}(t) - \epsilon \geqslant \hat{\mu}_1(t) + C_{\delta/n}(t) - \epsilon \geqslant \mu_1 - \epsilon$$

and for any i,

$$\mu_1 - \epsilon \geqslant \mu_i - \epsilon \geqslant \hat{\mu}_i(t) - C_{\delta/n}(t) - \epsilon$$

Hence $\mu_1 - \epsilon \geqslant \max_i \hat{\mu}_i(t) - C_{\delta/n}(t) - \epsilon = L_t$.

Next, we show that $G(t) \subset G_\epsilon$ for all $t \geqslant 1$. Suppose not. Then $\exists, t \in N$ and $\exists i \in G_\epsilon^c \cap G(t)$ such that,

$$\mu_i \geqslant \hat{\mu}_i(t) - C_{\delta/n}(t) \geqslant U_t \geqslant \mu_1 - \epsilon > \mu_i,$$

with the last inequality following from the previous assertion, giving a contradiction. □

Claim 1: Next, we show that on $\mathcal{E}$, $G_\epsilon \subset \mathcal{A}(t) \cup G(t)$ for all $t \in \mathbb{N}$.

In particular this implies that if $\mathcal{A} \subset G$, then $G_\epsilon \subset G$. Combining this with the previous claim gives $G \subset G_\epsilon \subset G$, hence $G = G_\epsilon$. On this condition, `EAST` terminates by line 2 and returns the set $\mathcal{A} \cup G = G$. Note that by definition, $G_\epsilon \subset G_{(\epsilon+\gamma)}$ for all $\gamma \geqslant 0$. Therefore `EAST` terminates correctly on this condition.

Proof. Suppose for contradiction that there exists $i \in G_\epsilon$ such that $i \notin \mathcal{A}(t) \cup G(t)$. This occurs only if i is eliminated in line 6.H.2. Hence, there exists a $t' \leqslant t$ such that $\hat{\mu_i}(t') + C_{\delta/n}(t') < L_{t'}$. Therefore, on the event $\mathcal{E}$,

$$\mu_1 - \epsilon \overset{\mathcal{E}}{\geqslant} L_{t'} = \max_{j \in \mathcal{A}} \hat{\mu}_j(t') - C_{\delta/n}(t') - \epsilon > \hat{\mu_i}(t') + C_{\delta/n}(t') \overset{\mathcal{E}}{\geqslant} \mu_i$$

which contradicts $i \in G_\epsilon$. □

Claim 2: Finally, we show that if $U_t - L_t \leqslant \gamma/2$, then $\mathcal{A} \cup G \subset G_{(\epsilon+\gamma)}$.

Combining with the previous that $G_\epsilon \subset \mathcal{A} \cup G$, if `EAST` terminates on this condition by line 2, it does so correctly.

Proof. Assume $U_t - L_t \leqslant \gamma/2$. This implies that

$$(\max_{i \in A(t)} \hat{\mu}_i(t) + C_{\delta/n}(t) - \epsilon) - (\max_{i \in A(t)} \hat{\mu}_i(t) - C_{\delta/n}(t) - \epsilon) = 2C_{\delta/n}(t) \leqslant \gamma/2.$$

Suppose for contradiction that there exists $i \in G^c_{(\epsilon+\gamma)}$ such that $i \in \mathcal{A} \cup G$. Since $G_\epsilon \cap G^c_{(\epsilon+\gamma)} = \emptyset$ and we have previously shown than $G(t) \subset G_\epsilon$ for all t, we have that $i \in A \backslash G$. Therefore, by the condition in line 6.H.2, $\hat{\mu}_i(t) + C_{\delta/n}(t) \geqslant L_t$. Hence, $\mu_i + 2C_{\delta/n}(t) \overset{\mathcal{E}}{\geqslant} \hat{\mu}_i(t) + C_{\delta/n}(t) \geqslant L_t$. By assumption, we have that $U_t - \gamma/2 \leqslant L_t$, and the event $\mathcal{E}$ implies that $U_t \geqslant \mu_1 - \epsilon$. Therefore, $\mu_i + 2C_{\delta/n}(t) \geqslant U_t - \gamma/2 \geqslant \mu_1 - \epsilon - \gamma/2$. Combining this with the inequality $2C_{\delta/n} \leqslant \gamma/2$, we have that

$$\gamma \geqslant 2C_{\delta/n}(t) + \gamma/2 \geqslant \mu_1 - \epsilon - \mu_i \overset{i \in G^c_{(\epsilon+\gamma)}}{>} \gamma$$

which is a contradiction. □

Therefore, on the event $\mathcal{E}$, if EAST terminates due to either condition in line 2, it returns $\mathcal{A} \cup G$ such that $G_\epsilon \subset \mathcal{A} \cup G \subset G_{(\epsilon+\gamma)}$. Since $\mathbb{P}(\mathcal{E}) \geqslant 1 - \delta$, EAST terminates correctly with probability at least $1 - \delta$.

6.F.6.2 Step 1: Controlling the total number of samples given by EAST to arms in G_ϵ

To keep track of the number of samples that arms are given by EAST, we introduce random variables T_i and T_i' for all $i \in [n]$. When arm i has been given $\max(T_i, T_i')$ samples it is removed from $\mathcal{A}$ in line 6.H.2.

By Step 0, only arms in G_ϵ are added to G. Therefore, T_i is defined as

$$T_i = \min \left\{ t : \begin{array}{ll} i \in G_k(t+1) & \text{if } i \in G_\epsilon \\ i \notin \mathcal{A}(t+1) & \text{if } i \in G^c_\epsilon \end{array} \right\} \overset{\mathcal{E}}{=} \min \left\{ t : \begin{array}{ll} \hat{\mu}_i - C_{\delta/n}(t) \geqslant U_t & \text{if } i \in G_\epsilon \\ \hat{\mu}_i + C_{\delta/n}(t) \leqslant L_t & \text{if } i \in G^c_\epsilon \end{array} \right\} \tag{6.24}$$

Similarly, recall T_i' denotes the random variable of the of the number of times i is

sampled before $\hat{\mu}_i(t) + C_{\delta/n}(t) \leqslant \max_{j\in\mathcal{A}} \hat{\mu}_j(t) - C_{\delta/n}(t)$. Hence,

$$T_i' = \min\left\{t : \hat{\mu}_i(t) + C_{\delta/n}(t) \leqslant \max_{j\in\mathcal{A}(t)} \hat{\mu}_j(t) - C_{\delta/n}(t)\right\} \tag{6.25}$$

Claim 0: For $i \in G_\epsilon$, we have that $T_i \leqslant h(0.25(\epsilon - \Delta_i), \delta/n)$.

Proof. Note that, $4C_{\delta/n}(t) \leqslant \mu_i - (\mu_1 - \epsilon)$, true when $t > h\left(0.25(\epsilon - \Delta_i), \frac{\delta}{n}\right)$, implies that for all j,

$$\begin{aligned}
\hat{\mu}_i(t) - C_{\delta/n}(t) &\overset{\mathcal{E}}{\geqslant} \mu_i - 2C_{\delta/n}(t) \\
&\geqslant \mu_1 + 2C_{\delta/n}(t) - \epsilon \\
&\geqslant \mu_j + 2C_{\delta/n}(t) - \epsilon \\
&\overset{\mathcal{E}}{\geqslant} \hat{\mu}_j(t) + C_{\delta/n}(t) - \epsilon
\end{aligned}$$

so in particular, $\hat{\mu}_i(t) - C_{\delta/n}(t) \geqslant \max_{j\in\mathcal{A}} \hat{\mu}_j(t) + C_{\delta/n}(t) - \epsilon = U_t$. □

Claim 1: For $i \in G_\epsilon$, we have that $T_i' \leqslant h(0.25\Delta_i, \delta/n)$.

Proof. Note that $4C_{\delta/n}(t) \leqslant \mu_1 - \mu_i$, true when $t > h\left(0.25\Delta_i, \frac{\delta}{n}\right)$, implies that

$$\begin{aligned}
\hat{\mu}_i(t) + C_{\delta/n}(t) &\overset{\mathcal{E}}{\leqslant} \mu_i + 2C_{\delta/n}(t) \\
&\leqslant \mu_1 - 2C_{\delta/n}(t) \\
&\overset{\mathcal{E}}{\leqslant} \hat{\mu}_1(t) - C_{\delta/n}(t).
\end{aligned}$$

As shown in Step 0, $1 \in \mathcal{A}(t)$ for all $t \in \mathbb{N}$, and in particular $\hat{\mu}_1(t) \leqslant \max_{i\in\mathcal{A}(t)} \hat{\mu}_i(t)$. Hence, $\hat{\mu}_i(t) + C_{\delta/n}(t) \leqslant \max_{j\in\mathcal{A}(t)} \hat{\mu}_j(t) - C_{\delta/n}(t)$. □

6.F.6.3 Step 2: Controlling the total number of samples given by EAST to arms in G_ϵ^c

Claim: Next, we show that $T_i \leqslant h\left(0.25(\epsilon - \Delta_i), \frac{\delta}{n}\right)$ for $i \in G_\epsilon^c$

Proof. Note that, $4C_{\delta/n}(t) \leqslant \mu_1 - \epsilon - \mu_i$, true when $t > h\left(0.25(\epsilon - \Delta_i), \frac{\delta}{n}\right)$, implies

that

$$\begin{aligned}\hat{\mu}_i(t) + C_{\delta/n}(t) &\overset{\mathcal{E}}{\leqslant} \mu_i + 2C_{\delta/n}(t) \\ &\leqslant \mu_1 - 2C_{\delta/n}(t) - \epsilon \\ &\overset{\mathcal{E}}{\leqslant} \hat{\mu}_1(t) - C_{\delta/n}(t) - \epsilon\end{aligned}$$

As shown in Step 0, $1 \in \mathcal{A}(t)$ for all $t \in \mathbb{N}$, and in particular $\hat{\mu}_1(t) \leqslant \max_{i \in \mathcal{A}(t)} \hat{\mu}_i(t)$. Therefore $\hat{\mu}_i(t) + C_{\delta/n}(t) \leqslant \max_{j \in \mathcal{A}} \hat{\mu}_j(t) - C_{\delta/n}(t) - \epsilon = L_t$. □

6.F.6.4 Step 3: Bounding the total number of samples drawn by EAST

With the results of Steps 1 and 2, we may bound the total sample complexity of EAST. Note that independently of the event $\mathcal{E}$, EAST terminates if $U_t - L_t \leqslant \gamma/2$. Let the random variable of the maximum number of samples given to any arm before this occurs be T_γ. Additionally, EAST may terminate if $\mathcal{A} \subset G$. Let the random variable of maximum number of samples given to any arm before this occurs be $T_{\alpha_\epsilon \beta_\epsilon}$. Note that due to the sampling procedure, the total number of samples drawn by EAST at termination may be written as $\sum_{t=1}^{\min(T_\gamma, T_{\alpha_\epsilon\beta_\epsilon})} |\mathcal{A}(t)|$.

Now we bound $\sum_{t=1}^{\min(T_\gamma, T_{\alpha_\epsilon\beta_\epsilon})} |\mathcal{A}(t)|$. Let $S_i = \min\{t : i \notin A(t+1)\}$. Hence,

$$\begin{aligned}\sum_{t=1}^{\min(T_\gamma, T_{\alpha_\epsilon\beta_\epsilon})} |\mathcal{A}(t)| = \sum_{t=1}^{\min(T_\gamma, T_{\alpha_\epsilon\beta_\epsilon})} \sum_{i=1}^{n} \mathbb{1}[i \in \mathcal{A}(t)] &= \sum_{i=1}^{n} \sum_{t=1}^{\min(T_\gamma, T_{\alpha_\epsilon\beta_\epsilon})} \mathbb{1}[i \in \mathcal{A}(t)] \\ &= \sum_{i=1}^{n} \min\{T_\gamma, T_{\alpha_\epsilon\beta_\epsilon}, S_i\}\end{aligned}$$

For arms $i \in G_\epsilon^c$, $S_i = T_i$ by definition. For $i \in G_\epsilon$, $S_i = \max(T_i, T_i')$ by line 6.H.2 of the algorithm. Then

$$\begin{aligned}\sum_{i=1}^{n} \min\{T_\gamma, T_{\alpha_\epsilon\beta_\epsilon}, S_i\} &= \sum_{i \in G_\epsilon} \min\{T_\gamma, T_{\alpha_\epsilon\beta_\epsilon}, \max(T_i, T_i')\} + \sum_{i \in G_\epsilon^c} \min\{T_\gamma, T_{\alpha_\epsilon\beta_\epsilon}, T_i\} \\ &= \sum_{i \in G_\epsilon} \min\{T_\gamma, \min\{T_{\alpha_\epsilon\beta_\epsilon}, \max(T_i, T_i')\}\} + \sum_{i \in G_\epsilon^c} \min\{T_\gamma, T_{\alpha_\epsilon\beta_\epsilon}, T_i\}\end{aligned}$$

$$= \sum_{i \in G_\epsilon} \min\{T_\gamma, \max\{T_i, \min(T_i', T_{\alpha_\epsilon \beta_\epsilon})\}\} + \sum_{i \in G_\epsilon^c} \min\{T_\gamma, T_{\alpha_\epsilon \beta_\epsilon}, T_i\}$$

We may define $T_\gamma := \min\{t : U_t - L_t \leqslant \gamma/2\}$. Note that $4C_{\delta/n}(t) \leqslant \gamma$, true when $t > h(0.25\gamma, \delta/n)$ implies that

$$U_t - L_t = (\max_{i \in A(t)} \hat{\mu}_i(t) + C_{\delta/n}(t) - \epsilon) - (\max_{i \in A(t)} \hat{\mu}_i(t) - C_{\delta/n}(t) - \epsilon) = 2C_{\delta/n}(t) \leqslant \gamma/2.$$

Therefore, we have that $T_\gamma \leqslant h(0.25\gamma, \delta/n)$.

Next, we may define $T_{\alpha_\epsilon \beta_\epsilon} = \min\{t : \mathcal{A}(t) \subset G_\epsilon\}$. By step 0, on the event $\mathcal{E}$, $\mathcal{A} \subset G$ implies that $G = G_\epsilon$. Therefore, $T_{\alpha_\epsilon \beta_\epsilon}$ may be equivalently defined as $T_{\alpha_\epsilon \beta_\epsilon} = \min\{t : G(t) = G_\epsilon \text{ and } G_\epsilon^c \cap \mathcal{A} = \emptyset\}$. Recalling the definition of T_i, we see that $T_{\alpha_\epsilon \beta_\epsilon} = \max_i(T_i)$.

Recall that by steps 1 and 2, $T_i \leqslant h\left(0.25(\epsilon - \Delta_i), \frac{\delta}{n}\right)$ and $T_i' \leqslant h\left(0.25\Delta_i, \frac{\delta}{n}\right)$. Furthermore, by monotonicity of $h(\cdot, \cdot)$, this implies that $T_{\alpha_\epsilon \beta_\epsilon} = h(0.25 \min(\alpha_\epsilon, \beta_\epsilon), \delta/n)$. Plugging this in, we see that

$$\begin{aligned}
&\sum_{i \in G_\epsilon} \min\{T_\gamma, \max\{T_i, \min(T_i', T_{\alpha_\epsilon \beta_\epsilon})\}\} + \sum_{i \in G_\epsilon^c} \min\{T_\gamma, T_{\alpha_\epsilon \beta_\epsilon}, T_i\} \\
&\quad = \sum_{i \in G_\epsilon} \min\{T_\gamma, \max\{T_i, \min(T_i', T_{\alpha_\epsilon \beta_\epsilon})\}\} + \sum_{i \in G_\epsilon^c} \min\{T_\gamma, T_i\} \\
&\quad \leqslant \sum_{i \in G_\epsilon} \min\left\{\max\left\{h\left(0.25(\epsilon - \Delta_i), \frac{\delta}{n}\right),\right.\right. \\
&\qquad\qquad \left.\left. \min\left[h\left(0.25\Delta_i, \frac{\delta}{n}\right), h\left(0.25 \min(\alpha_\epsilon, \beta_\epsilon), \frac{\delta}{n}\right)\right]\right\}, \right. \\
&\qquad\qquad \left. h\left(0.25\gamma, \frac{\delta}{n}\right)\right\} \\
&\qquad + \sum_{i \in G_\epsilon^c} \min\left\{h\left(0.25(\epsilon - \Delta_i), \frac{\delta}{n}\right), h\left(0.25 \min(\alpha_\epsilon, \beta_\epsilon), \frac{\delta}{n}\right)\right\} \\
&\quad = \sum_{i=1}^{n} \min\left\{\max\left\{h\left(0.25(\epsilon - \Delta_i), \frac{\delta}{n}\right),\right.\right.
\end{aligned}$$

$$\min\left[h\left(0.25\Delta_i, \frac{\delta}{n}\right), h\left(0.25\min(\alpha_\epsilon, \beta_\epsilon), \frac{\delta}{n}\right)\right]\Bigg\},$$
$$h\left(0.25\gamma, \frac{\delta}{n}\right)\Bigg\}$$

where the final equality holds by definition for arms in G_ϵ. Next, by Lemma 6.33, we may bound the minimum of $h(\cdot,\cdot)$ functions.

$$\begin{aligned}
&\sum_{i=1}^{n} \min\left\{\max\left\{h\left(\frac{\Delta_i-\epsilon}{4}, \frac{\delta}{n}\right), \min\left[h\left(\frac{\Delta_i}{4}, \frac{\delta}{n}\right), h\left(\frac{\min(\alpha_\epsilon,\beta_\epsilon)}{4}, \frac{\delta}{n}\right)\right]\right\},\right.\\
&\qquad\left. h\left(\frac{\gamma}{4}, \frac{\delta}{n}\right)\right\}\\
&= \sum_{i=1}^{n} \min\left\{\max\left\{h\left(\frac{\Delta_i-\epsilon}{4}, \frac{\delta}{n}\right),\right.\right.\\
&\qquad\left.\left. \min\left[h\left(\frac{\Delta_i}{4}, \frac{\delta}{n}\right), \max\left[h\left(\frac{\alpha_\epsilon}{4}, \frac{\delta}{n}\right), h\left(\frac{\beta_\epsilon}{4}, \frac{\delta}{n}\right)\right]\right]\right\},\right.\\
&\qquad\left. h\left(\frac{\gamma}{4}, \frac{\delta}{n}\right)\right\}\\
&\leqslant \sum_{i=1}^{n} \min\left\{\max\left\{h\left(\frac{\Delta_i-\epsilon}{4}, \frac{\delta}{n}\right),\right.\right.\\
&\qquad\left.\left. \max\left[h\left(\frac{\Delta_i+\alpha_\epsilon}{8}, \frac{\delta}{n}\right), h\left(\frac{\Delta_i+\beta_\epsilon}{8}, \frac{\delta}{n}\right)\right]\right\},\right.\\
&\qquad\left. h\left(\frac{\gamma}{4}, \frac{\delta}{n}\right)\right\}\\
&= \sum_{i=1}^{n} \min\left\{\max\left\{h\left(\frac{\Delta_i-\epsilon}{4}, \frac{\delta}{n}\right), h\left(\frac{\Delta_i+\alpha_\epsilon}{8}, \frac{\delta}{n}\right), h\left(\frac{\Delta_i+\beta_\epsilon}{8}, \frac{\delta}{n}\right)\right\},\right.\\
&\qquad\left. h\left(\frac{\gamma}{4}, \frac{\delta}{n}\right)\right\}
\end{aligned}$$

Finally, we use Lemma 6.32 to bound the function $h(\cdot,\cdot)$. Since $\delta \leqslant 1/2$, $\delta/n \leqslant 2e^{-e/2}$. Further, $\max(\Delta_i, |\epsilon-\Delta_i|) \leqslant 8$ for all i, we have that $0.25\Delta_i \leqslant 2$, $0.25|\epsilon-\Delta_i| \leqslant$

2, and $0.25\min(\alpha_\epsilon, \beta_\epsilon) \leqslant 2$. Therefore,

$$\begin{aligned}
&\sum_{i=1}^{n} \min\left\{\max\left\{h\left(\frac{\Delta_i-\epsilon}{4}, \frac{\delta}{n}\right), h\left(\frac{\Delta_i+\alpha_\epsilon}{8}, \frac{\delta}{n}\right), h\left(\frac{\Delta_i+\beta_\epsilon}{8}, \frac{\delta}{n}\right)\right\},\right. \\
&\qquad\qquad \left. h\left(\frac{\gamma}{4}, \frac{\delta}{n}\right)\right\} \\
&\leqslant \sum_{i=1}^{n} \min\left\{\max\left\{\frac{64}{(\epsilon-\Delta_i)^2}\log\left(\frac{2n}{\delta}\log_2\left(\frac{192n}{\delta(\epsilon-\Delta_i)^2}\right)\right),\right.\right. \\
&\qquad\qquad \frac{256}{(\Delta_i+\alpha_\epsilon)^2}\log\left(\frac{2n}{\delta}\log_2\left(\frac{768n}{\delta(\Delta_i+\alpha_\epsilon)^2}\right)\right), \\
&\qquad\qquad \left.\left.\frac{256}{(\Delta_i+\beta_\epsilon)^2}\log\left(\frac{2n}{\delta}\log_2\left(\frac{768n}{\delta(\Delta_i+\beta_\epsilon)^2}\right)\right)\right\}, \\
&\qquad \left.\frac{64}{\gamma^2}\log\left(\frac{2n}{\delta}\log_2\left(\frac{192n}{\delta\gamma^2}\right)\right)\right\} \\
&= \sum_{i=1}^{n} \min\left\{\max\left\{\frac{64}{(\mu_1-\epsilon-\mu_i)^2}\log\left(\frac{2n}{\delta}\log_2\left(\frac{768n}{\delta(\mu_1-\epsilon-\mu_i)^2}\right)\right),\right.\right. \\
&\qquad\qquad \frac{256}{(\mu_1+\alpha_\epsilon-\mu_i)^2}\log\left(\frac{2n}{\delta}\log_2\left(\frac{768n}{\delta(\mu_1+\alpha_\epsilon-\mu_i)^2}\right)\right), \\
&\qquad\qquad \left.\left.\frac{256}{(\mu_1+\beta_\epsilon-\mu_i)^2}\log\left(\frac{2n}{\delta}\log_2\left(\frac{768n}{\delta(\mu_1+\beta_\epsilon-\mu_i)^2}\right)\right)\right\}, \\
&\qquad \left.\frac{64}{\gamma^2}\log\left(\frac{2n}{\delta}\log_2\left(\frac{192n}{\delta\gamma^2}\right)\right)\right\}.
\end{aligned}$$

□

6.F.7 Proof of Theorem 6.28, EAST in the multiplicative regime

Proof. **Notation for the proof:** Throughout, recall $\Delta_i = \mu_1 - \mu_i$. Recall that t counts the number of times each arm in $\mathcal{A}$ has been sampled and thus the number of times that the conditionals in Lines 6.H.2 and 6.H.2 have been evaluated. Let $\mathcal{A}(t)$ denote the state $\mathcal{A}$ at this time before the arms have been eliminated from $\mathcal{A}$ in lines 6.H.2 and 6.H.2. Let $G(t)$ be defined similarly. Therefore, the total number of samples drawn by EAST up to time t is $\sum_{s=1}^{t} |\mathcal{A}(s)|$.

For $i \in M_\epsilon$, let T_i denote the random variable of the number of times arm i is sampled before it is added to G in Line 8. For $i \in M_\epsilon^c$, let T_i denote the random variable of the number of times arm i is sampled before it is removed from $\mathcal{A}$ in Line 6.H.2. For any arm i, let T_i' denote the random variable of the of the number of times i is sampled before $\hat{\mu}_i(t) + C_{\delta/n}(t) \leqslant \max_{j \in \mathcal{A}} \hat{\mu}_j(t) - C_{\delta/n}(t)$.

Define the event

$$\mathcal{E} = \left\{ \bigcap_{i \in [n]} \bigcap_{t \in \mathbb{N}} |\hat{\mu}_i(t) - \mu_i| \leqslant C_{\delta/n}(t) \right\}.$$

Using standard anytime confidence bound results, and recalling that that $C_\delta(t) := \sqrt{\frac{4\log(\log_2(2t)/\delta)}{t}}$, we have

$$\begin{aligned}\mathbb{P}(\mathcal{E}^c) &= \mathbb{P}\left(\bigcup_{i \in [n]} \bigcup_{t \in \mathbb{N}} |\hat{\mu}_i - \mu_i| > C_{\delta/n}(t) \right) \\ &\leqslant \sum_{i=1}^{n} \mathbb{P}\left(\bigcup_{t \in \mathbb{N}} |\hat{\mu}_i - \mu_i| > C_{\delta/n}(t) \right) \leqslant \sum_{i=1}^{n} \frac{\delta}{n} = \delta\end{aligned}$$

Hence, $\mathbb{P}(\mathcal{E}) \geqslant 1 - \delta$.

6.F.7.1 Step 0: Correctness

Claim 0: On $\mathcal{E}$, first we prove that $G(t) \subset M_\epsilon$ for all $t \in \mathbb{N}$.

In particular, this shows that `EAST` never incorrectly add arms in M_ϵ^c to the set G.

Proof. Firstly we show $1 \in \mathcal{A}$ for all $t \in \mathbb{N}$, namely the best arm is never removed from $\mathcal{A}$. Note for any i such that $\hat{\mu}_i(t) - C_{\delta/n}(t) \geqslant 0$,

$$\hat{\mu}_1 + C_{\delta/n}(t) \geqslant \mu_1 \geqslant \mu_i \geqslant \hat{\mu}_i(t) - C_{\delta/n}(t) > (1-\epsilon)(\hat{\mu}_i(t) - C_{\delta/n}(t)).$$

For i such that $\hat{\mu}_i(t) - C_{\delta/n}(t) < 0$, if $\hat{\mu}_1 + C_{\delta/n}(t) \geqslant 0$, then

$$\hat{\mu}_1 + C_{\delta/n}(t) \geqslant 0 > (1-\epsilon)(\hat{\mu}_i(t) - C_{\delta/n}(t)).$$

Note that $\hat{\mu}_1 + C_{\delta/n}(t) < 0$ implies on the event $\mathcal{E}$ that $\mu_1 < 0$, which contradicts the assumption that $\mu_1 \geqslant 0$ made in the theorem. In particular this shows, $\hat{\mu}_1 + C_{\delta/n}(t) > (1-\epsilon)(\max_{i\in\mathcal{A}} \hat{\mu}_i(t) - C_{\delta/n}(t)) = L_t$ and $\hat{\mu}_1 + C_{\delta/n}(t) \geqslant \max_{i\in\mathcal{A}} \hat{\mu}_i(t) - C_{\delta/n}(t)$ showing that 1 will never exit $\mathcal{A}$ in line 28.

Secondly, we show that at all times t, $(1-\epsilon)\mu_1 \in [L_t, U_t]$. By the above, since μ_1 never leaves $\mathcal{A}$,

$$U_t = (1-\epsilon)(\max_{i\in\mathcal{A}} \hat{\mu}_i(t) + C_{\delta/n}(t)) \geqslant (1-\epsilon)(\hat{\mu}_1(t) + C_{\delta/n}(t)) \geqslant (1-\epsilon)\mu_1$$

and for any i,

$$(1-\epsilon)\mu_1 \geqslant (1-\epsilon)\mu_i \geqslant (1-\epsilon)(\hat{\mu}_i(t) - C_{\delta/n}(t))$$

Hence $(1-\epsilon)\mu_1 \geqslant (1-\epsilon)(\max_i \hat{\mu}_i(t) - C_{\delta/n}(t)) = L_t$.

Next, we show that $G \subset M_\epsilon$ for all $k \geqslant 1, t \geqslant 1$. Suppose not. Then $\exists, k, t \in \mathbb{N}$ and $\exists i \in M_\epsilon^c \cap G(t)$ such that,

$$\mu_i \geqslant \hat{\mu}_i(t) - C_{\delta/n}(t) \geqslant U_t \geqslant (1-\epsilon)\mu_1 > \mu_i,$$

with the last inequality following from the previous assertion, giving a contradiction. □

Claim 1: Next, we show that on $\mathcal{E}$, $M_\epsilon \subset \mathcal{A}(t) \cup G(t)$ for all $t \in \mathbb{N}$.

In particular this implies that if $\mathcal{A} \subset G$, then $M_\epsilon \subset G$. Combining this with the previous claim gives $G \subset M_\epsilon \subset G$, hence $G = M_\epsilon$. On this condition, `EAST` terminates and returns the set $\mathcal{A} \cup G = G$. Note that by definition, $M_\epsilon \subset M_{(\epsilon+\gamma)}$ for all $\gamma \geqslant 0$. Therefore `EAST` terminates correctly on this condition.

Proof. Suppose for contradiction that there exists $i \in M_\epsilon$ such that $i \notin \mathcal{A}(t) \cup G(t)$. This occurs only if i is eliminated in line 6.H.2. Hence, there exists a

$t' \leqslant t$ such that $\hat{\mu}_i(t') + C_{\delta/n}(t') < L_{t'}$. Therefore, on the event $\mathcal{E}$,

$$(1-\epsilon)\mu_1 \overset{\mathcal{E}}{\geqslant} L_{t'} = (1-\epsilon)\left(\max_{j\in\mathcal{A}} \hat{\mu}_j(t') - C_{\delta/n}(t')\right) > \hat{\mu}_i(t') + C_{\delta/n}(t') \overset{\mathcal{E}}{\geqslant} \mu_i$$

which contradicts $i \in M_\epsilon$. □

Claim 2: Finally, we show that on $\mathcal{E}$, if $U_t - L_t \leqslant \frac{\gamma}{2-\epsilon} L_t$, then $\mathcal{A} \cup G \subset M_{(\epsilon+\gamma)}$.

Combining with Claim 1 that $M_\epsilon \subset \mathcal{A} \cup G$, if `EAST` terminates on this condition, it does so correctly and returns all arms in M_ϵ and none in $M^c_{(\epsilon+\gamma)}$.

Proof. By Claim 0, $G \subset M_\epsilon \subset M_{\epsilon+\gamma}$. Hence, $G \cap M^c_{(\epsilon+\gamma)} = \emptyset$. Therefore, we wish to show that $\mathcal{A} \cap M^c_{(\epsilon+\gamma)} = \emptyset$ which implies that $G \cap \mathcal{A} \subset M_{\epsilon+\gamma}$. Assume $U_t - L_t < \frac{\gamma}{2-\epsilon} L_t$. Recall that

$$U_t = (1-\epsilon)\left(\max_{i\in\mathcal{A}} \hat{\mu}_i(t) + C_{\delta/n}(t)\right)$$

and

$$L_t = (1-\epsilon)\left(\max_{i\in\mathcal{A}} \hat{\mu}_i(t) - C_{\delta/n}(t)\right)$$

All arms in $\mathcal{A}(t)$ have received exactly t samples. Hence, $U_t - L_t = 2(1-\epsilon)C_{\delta/n}(t)$. On $\mathcal{E}$, $L_t \leqslant (1-\epsilon)\mu_1$ This implies that

$$2(1-\epsilon)C_{\delta/n}(t) < \frac{\gamma}{2-\epsilon} L_t \leqslant \frac{1-\epsilon}{2-\epsilon}\gamma\mu_1,$$

and in particular,

$$2C_{\delta/n}(t) < \frac{\gamma\mu_1}{2-\epsilon}.$$

Therefore, we wish to show that when the above is true, then for any $i \in M^c_{\epsilon+\gamma}$, $L_t - (\hat{\mu}_i(t) + C_{\delta/n}(t)) > 0$, implying that $i \notin \mathcal{A}$.

$$\begin{aligned} L_t - (\hat{\mu}_i(t) + C_{\delta/n}(t)) &= (1-\epsilon)\left(\max_{j\in\mathcal{A}} \hat{\mu}_j - C_{\delta/n}(t)\right) - (\hat{\mu}_i(t) + C_{\delta/n}(t)) \\ &\geqslant (1-\epsilon)\left(\max_{j\in\mathcal{A}} \mu_j - 2C_{\delta/n}(t)\right) - (\mu_i + 2C_{\delta/n}(t)) \end{aligned}$$

$$\begin{aligned}
&\overset{(a)}{\geqslant} (1-\epsilon)\left(\mu_1 - 2C_{\delta/n}(t)\right) - ((1-\epsilon-\gamma)\mu_1 + 2C_{\delta/n}(t)) \\
&= \gamma\mu_1 - 2(2-\epsilon)C_{\delta/n}(t) \\
&> \gamma\mu_1 - (2-\epsilon)\frac{\gamma\mu_1}{2-\epsilon} \\
&= 0
\end{aligned}$$

which implies that $i \notin \mathcal{A}$. Inequality (a) follows jointly from the fact that $1 \in \mathcal{A}$ and the fact that all arms in $\mathcal{A}$ have received t samples implies $\max_{j\in\mathcal{A}} \mu_j - 2C_{\delta/n}(t) = \mu_1 - 2C_{\delta/n}(t)$. Additionally, inequality (a) follows from $\mu_i \leqslant (1-\epsilon-\gamma)\mu_1$ since $i \in M^c_{\epsilon+\gamma}$. □

Therefore, on the event $\mathcal{E}$, if `EAST` terminates due to either condition in line 2, it returns $\mathcal{A} \cup G$ such that $M_\epsilon \subset \mathcal{A} \cup G \subset M_{(\epsilon+\gamma)}$. Since $\mathbb{P}(\mathcal{E}) \geqslant 1-\delta$, `EAST` terminates correctly with probability at least $1-\delta$.

6.F.7.2 Step 1: Controlling the total number of samples given by `EAST` to arms in M_ϵ

To keep track of the number of samples that arms are given by `EAST`, we introduce random variables T_i and T_i' for all $i \in [n]$. When arm i has been given $\max(T_i, T_i')$ samples it is removed from $\mathcal{A}$ in line 6.H.2.

By Step 0, only arms in M_ϵ are added to G. Therefore, T_i is defined as

$$T_i = \min\left\{ t : \begin{array}{ll} i \in G(t+1) & \text{if } i \in M_\epsilon \\ i \notin \mathcal{A}(t+1) & \text{if } i \in M_\epsilon^c \end{array} \right\} \overset{\mathcal{E}}{=} \min\left\{ t : \begin{array}{ll} \hat{\mu}_i - C_{\delta/n}(t) \geqslant U_t & \text{if } i \in M_\epsilon \\ \hat{\mu}_i + C_{\delta/n}(t) \leqslant L_t & \text{if } i \in M_\epsilon^c \end{array} \right\} \tag{6.26}$$

Similarly, recall T_i' denotes the random variable of the of the number of times i is sampled before $\hat{\mu}_i(t) + C_{\delta/n}(t) \leqslant \max_{j\in\mathcal{A}} \hat{\mu}_j(t) - C_{\delta/n}(t)$. Hence,

$$T_i' = \min\left\{ t : \hat{\mu}_i(t) + C_{\delta/n}(t) \leqslant \max_{j\in\mathcal{A}(t)} \hat{\mu}_j(t) - C_{\delta/n}(t) \right\} \tag{6.27}$$

Claim 0: For $i \in M_\epsilon$, we have that $T_i \leqslant h\left(\frac{\epsilon\mu_1 - \Delta_i}{4-2\epsilon}, \frac{\delta}{n}\right)$.

Proof. Note that $\mu_i - 2C_{\delta/n}(t) \geqslant (1-\epsilon)(\mu_1 + 2C_{\delta/n}(t))$ may be rearranged as $(4-2\epsilon)C_{\delta/n}(t) \leqslant \epsilon\mu_1 - \Delta_i$, and this is true when $t > h\left(\frac{\epsilon\mu_1-\Delta_i}{4-2\epsilon}, \frac{\delta}{n}\right)$. This condition implies that for all j,

$$\begin{aligned}\hat{\mu}_i(t) - C_{\delta/n}(t) &\overset{\mathcal{E}}{\geqslant} \mu_i - 2C_{\delta/n}(t) \\ &\geqslant (1-\epsilon)(\mu_1 + 2C_{\delta/n}(t)) \\ &\geqslant (1-\epsilon)(\mu_j + 2C_{\delta/n}(t)) \\ &\overset{\mathcal{E}}{\geqslant} (1-\epsilon)(\hat{\mu}_j(t) + C_{\delta/n}(t))\end{aligned}$$

so in particular, $\hat{\mu}_i(t) - C_{\delta/n}(t) \geqslant (1-\epsilon)(\max_{j\in\mathcal{A}} \hat{\mu}_j(t) + C_{\delta/n}(t)) = U_t$. □

Claim 1: For $i \in M_\epsilon$, we have that $T_i' \leqslant h(0.25\Delta_i, \delta/n)$.

Proof. Note that $4C_{\delta/n}(t) \leqslant \mu_1 - \mu_i$, true when $t > h\left(0.25\Delta_i, \frac{\delta}{n}\right)$, implies that

$$\begin{aligned}\hat{\mu}_i(t) + C_{\delta/n}(t) &\overset{\mathcal{E}}{\leqslant} \mu_i + 2C_{\delta/n}(t) \\ &\leqslant \mu_1 - 2C_{\delta/n}(t) \\ &\overset{\mathcal{E}}{\leqslant} \hat{\mu}_1(t) - C_{\delta/n}(t).\end{aligned}$$

As shown in Step 0, $1 \in \mathcal{A}(t)$ for all $t \in \mathbb{N}$, and in particular $\hat{\mu}_1(t) \leqslant \max_{i\in\mathcal{A}(t)} \hat{\mu}_i(t)$. Hence, $\hat{\mu}_i(t) + C_{\delta/n}(t) \leqslant \max_{j\in\mathcal{A}(t)} \hat{\mu}_j(t) - C_{\delta/n}(t)$. □

6.F.7.3 Step 2: Controlling the total number of samples given by EAST to arms in M_ϵ^c

Next, we bound T_i for $i \in M_\epsilon^c$. $i \in M_\epsilon^c$ is eliminated from $\mathcal{A}$ if it has received at least T_i samples.

Claim: $T_i \leqslant h\left(\frac{\Delta_i-\epsilon\mu_1}{4-2\epsilon}, \frac{\delta}{n}\right)$ for $i \in M_\epsilon^c$

Proof. Note that $\mu_i + 2C_{\delta/n}(t) \leqslant (1-\epsilon)(\mu_1 - 2C_{\delta/n}(t))$ may be rearranged as $(4-2\epsilon)C_{\delta/n}(t) \leqslant \Delta_i - \epsilon\mu_1$, and this is true when $t > h\left(\frac{\Delta_i-\epsilon\mu_1}{4-2\epsilon}, \frac{\delta}{n}\right)$. This condition

implies that

$$\begin{aligned}\hat{\mu}_i(t) + C_{\delta/n}(t) &\overset{\mathcal{E}}{\leqslant} \mu_i + 2C_{\delta/n}(t) \\ &\leqslant (1-\epsilon)(\mu_1 - 2C_{\delta/n}(t)) \\ &\overset{\mathcal{E}}{\leqslant} (1-\epsilon)(\hat{\mu}_1(t) - C_{\delta/n}(t))\end{aligned}$$

As shown in Step 0, $1 \in \mathcal{A}(t)$ for all $t \in \mathbb{N}$, and in particular $\hat{\mu}_1(t) \leqslant \max_{i\in\mathcal{A}(t)} \hat{\mu}_i(t)$. Therefore $\hat{\mu}_i(t) + C_{\delta/n}(t) \leqslant (1-\epsilon)(\max_{j\in\mathcal{A}} \hat{\mu}_j(t) - C_{\delta/n}(t)) = L_t$. □

6.F.7.4 Step 3: Bounding the total number of samples drawn by EAST

With the results of Steps 1 and 2, we may bound the total sample complexity of EAST. Note that independently of the event $\mathcal{E}$, EAST terminates if $U_t - L_t \leqslant \frac{\gamma}{2-\epsilon} L_t$. Let the random variable of the maximum number of samples given to any arm before this occurs be $T_\gamma := \min\{t : U_t - L_t \leqslant \frac{\gamma}{2-\epsilon} L_t\}$. Additionally, EAST may terminate if $\mathcal{A} \subset G$. Let the random variable of maximum number of samples given to any arm before this occurs be $T_{\tilde{\alpha}_\epsilon \tilde{\beta}_\epsilon}$. Note that due to the sampling procedure, the total number of samples drawn by EAST at termination may be written as $\sum_{t=1}^{\min(T_\gamma, T_{\tilde{\alpha}_\epsilon\tilde{\beta}_\epsilon})} |\mathcal{A}(t)|$.

Now we bound $\sum_{t=1}^{\min(T_\gamma, T_{\tilde{\alpha}_\epsilon\tilde{\beta}_\epsilon})} |\mathcal{A}(t)|$. Let $S_i = \min\{t : i \notin A(t+1)\}$. Hence,

$$\begin{aligned}\sum_{t=1}^{\min(T_\gamma, T_{\tilde{\alpha}_\epsilon\tilde{\beta}_\epsilon})} |\mathcal{A}(t)| = \sum_{t=1}^{\min(T_\gamma, T_{\tilde{\alpha}_\epsilon\tilde{\beta}_\epsilon})} \sum_{i=1}^{n} \mathbb{1}[i \in \mathcal{A}(t)] &= \sum_{i=1}^{n} \sum_{t=1}^{\min(T_\gamma, T_{\tilde{\alpha}_\epsilon\tilde{\beta}_\epsilon})} \mathbb{1}[i \in \mathcal{A}(t)] \\ &= \sum_{i=1}^{n} \min\left\{T_\gamma, T_{\tilde{\alpha}_\epsilon\tilde{\beta}_\epsilon}, S_i\right\}\end{aligned}$$

For arms $i \in M_\epsilon^c$, $S_i = T_i$ by definition. For $i \in M_\epsilon$, $S_i = \max(T_i, T_i')$ by line 6.H.2 of the algorithm. Then

$$\sum_{i=1}^{n} \min\left\{T_\gamma, T_{\tilde{\alpha}_\epsilon\tilde{\beta}_\epsilon}, S_i\right\} = \sum_{i\in M_\epsilon} \min\left\{T_\gamma, T_{\tilde{\alpha}_\epsilon\tilde{\beta}_\epsilon}, \max(T_i, T_i')\right\} + \sum_{i\in M_\epsilon^c} \min\left\{T_\gamma, T_{\tilde{\alpha}_\epsilon\tilde{\beta}_\epsilon}, T_i\right\}$$

$$= \sum_{i \in M_\epsilon} \min\left\{T_\gamma, \min\left\{T_{\tilde{\alpha}_\epsilon \tilde{\beta}_\epsilon}, \max(T_i, T_i')\right\}\right\} + \sum_{i \in M_\epsilon^c} \min\left\{T_\gamma, T_{\tilde{\alpha}_\epsilon \tilde{\beta}_\epsilon}, T_i\right\}$$
$$= \sum_{i \in M_\epsilon} \min\left\{T_\gamma, \max\left\{T_i, \min(T_i', T_{\tilde{\alpha}_\epsilon \tilde{\beta}_\epsilon})\right\}\right\} + \sum_{i \in M_\epsilon^c} \min\left\{T_\gamma, T_{\tilde{\alpha}_\epsilon \tilde{\beta}_\epsilon}, T_i\right\}$$

Next we bound T_γ.

Claim: On $\mathcal{E}$, $T_\gamma \leqslant h\left(\frac{\gamma\mu_1}{2(2-\epsilon+\gamma)}, \frac{\delta}{n}\right)$.

Proof: $C_{\delta/n}(t) < \frac{\gamma\mu_1}{2(2-\epsilon+\gamma)}$ is true when $t \geqslant h\left(\frac{\gamma\mu_1}{2(2-\epsilon+\gamma)}, \frac{\delta}{n}\right)$. Note that

$$C_{\delta/n}(t) < \frac{\gamma\mu_1}{2(2-\epsilon+\gamma)} \iff 2C_{\delta/n}(t) < \frac{\gamma}{2-\epsilon}\left(\mu_1 - 2C_{\delta/n}(t)\right).$$

This implies that

$$\begin{aligned} U_t - L_t &= 2(1-\epsilon)C_{\delta/n}(t) \\ &< 2\frac{1-\epsilon}{2-\epsilon}\gamma\left(\mu_1 - 2C_{\delta/n}(t)\right) \\ &\leqslant \frac{1-\epsilon}{2-\epsilon}\gamma\left(\hat{\mu_1}(t) - C_{\delta/n}(t)\right) \\ &\leqslant \frac{1-\epsilon}{2-\epsilon}\gamma\left(\max_{i \in \mathcal{A}} \hat{\mu}_i - C_{\delta/n}(t)\right) \\ &= \frac{\gamma}{2-\epsilon} L_t \end{aligned}$$

□

Next, we may define $T_{\tilde{\alpha}_\epsilon \tilde{\beta}_\epsilon} = \min\{t : \mathcal{A}(t) \subset M_\epsilon\}$. By step 0, on the event $\mathcal{E}$, $\mathcal{A} \subset G$ implies that $G = M_\epsilon$. Therefore, $T_{\tilde{\alpha}_\epsilon \tilde{\beta}_\epsilon}$ may be equivalently defined as $T_{\tilde{\alpha}_\epsilon \tilde{\beta}_\epsilon} = \min\{t : G(t) = M_\epsilon \text{ and } M_\epsilon^c \cap \mathcal{A} = \emptyset\}$. Recalling the definition of T_i, we see that $T_{\tilde{\alpha}_\epsilon \tilde{\beta}_\epsilon} = \max_i(T_i)$.

Recall that by steps 1 and 2, $T_i \leqslant h\left(\frac{\epsilon\mu_1 - \Delta_i}{4-2\epsilon}, \frac{\delta}{n}\right)$ and $T_i' \leqslant h\left(0.25\Delta_i, \frac{\delta}{n}\right)$. Furthermore, by monotonicity of $h(\cdot,\cdot)$, this implies that $T_{\tilde{\alpha}_\epsilon \tilde{\beta}_\epsilon} = h\left(\frac{\min(\tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon)}{4-2\epsilon}, \frac{\delta}{n}\right)$. Plugging this in, we see that

$$\sum_{i \in M_\epsilon} \min\left\{T_\gamma, \max\left\{T_i, \min(T_i', T_{\tilde{\alpha}_\epsilon \tilde{\beta}_\epsilon})\right\}\right\} + \sum_{i \in M_\epsilon^c} \min\left\{T_\gamma, T_{\tilde{\alpha}_\epsilon \tilde{\beta}_\epsilon}, T_i\right\}$$

$$
\begin{aligned}
&= \sum_{i \in M_\epsilon} \min\left\{T_\gamma, \max\left\{T_i, \min(T_i', T_{\tilde{\alpha}_\epsilon \tilde{\beta}_\epsilon})\right\}\right\} + \sum_{i \in M_\epsilon^c} \min\{T_\gamma, T_i\} \\
&\leqslant \sum_{i \in M_\epsilon} \min\left\{\max\left\{h\left(\frac{\epsilon\mu_1 - \Delta_i}{4-2\epsilon}, \frac{\delta}{n}\right), \min\left[h\left(0.25\Delta_i, \frac{\delta}{n}\right), h\left(\frac{\min(\tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon)}{4-2\epsilon}, \frac{\delta}{n}\right)\right]\right\},\right. \\
&\qquad \left. h\left(\frac{\gamma\mu_1}{2(2-\epsilon+\gamma)}, \frac{\delta}{n}\right)\right\} \\
&\quad + \sum_{i \in M_\epsilon^c} \min\left\{h\left(\frac{\epsilon\mu_1 - \Delta_i}{4-2\epsilon}, \frac{\delta}{n}\right), h\left(\frac{\gamma\mu_1}{2(2-\epsilon+\gamma)}, \frac{\delta}{n}\right)\right\} \\
&= \sum_{i=1}^{n} \min\left\{\max\left\{h\left(\frac{\epsilon\mu_1 - \Delta_i}{4-2\epsilon}, \frac{\delta}{n}\right), \min\left[h\left(0.25\Delta_i, \frac{\delta}{n}\right), h\left(\frac{\min(\tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon)}{4-2\epsilon}, \frac{\delta}{n}\right)\right]\right\},\right. \\
&\qquad \left. h\left(\frac{\gamma\mu_1}{2(2-\epsilon+\gamma)}, \frac{\delta}{n}\right)\right\}
\end{aligned}
$$

where the final equality holds by definition for arms in M_ϵ. Lastly, note that $\frac{1}{3(1-x)} \leqslant \frac{1}{2-x}$ for $x \leqslant 1/2$. By monotonicity of h, we may lower bound the denominators $\frac{1}{4-2\epsilon}$ and $\frac{1}{2(2-\epsilon+\gamma)}$ as $\frac{1}{6(1-\epsilon)}$ and $\frac{1}{6(1-\epsilon+\gamma)}$ respectively. Since $\epsilon \in (0, 1/2]$, we may likewise lower bound $\frac{1}{4-2\epsilon}$ as $1/4$. Plugging this in, we see that

$$
\begin{aligned}
&\sum_{i=1}^{n} \min\left\{\max\left\{h\left(\frac{\epsilon\mu_1 - \Delta_i}{4-2\epsilon}, \frac{\delta}{n}\right), \min\left[h\left(0.25\Delta_i, \frac{\delta}{n}\right), h\left(\frac{\min(\tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon)}{4-2\epsilon}, \frac{\delta}{n}\right)\right]\right\},\right. \\
&\qquad \left. h\left(\frac{\gamma\mu_1}{2(2-\epsilon+\gamma)}, \frac{\delta}{n}\right)\right\} \\
&\leqslant \sum_{i=1}^{n} \min\left\{\max\left\{h\left(\frac{\epsilon\mu_1 - \Delta_i}{4}, \frac{\delta}{n}\right), \min\left[h\left(0.25\Delta_i, \frac{\delta}{n}\right), h\left(\frac{\min(\tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon)}{6(1-\epsilon)}, \frac{\delta}{n}\right)\right]\right\},\right. \\
&\qquad \left. h\left(\frac{\gamma\mu_1}{6(1-\epsilon+\gamma)}, \frac{\delta}{n}\right)\right\}
\end{aligned}
$$

Next, by Lemma 6.33, we may bound the minimum of $h(\cdot, \cdot)$ functions.

$$
\sum_{i=1}^{n} \min\left\{\max\left\{h\left(\frac{\Delta_i - \epsilon\mu_1}{4}, \frac{\delta}{n}\right), \min\left[h\left(\frac{\Delta_i}{4}, \frac{\delta}{n}\right), h\left(\frac{\min(\tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon)}{6(1-\epsilon)}, \frac{\delta}{n}\right)\right]\right\},
$$

$$
\begin{aligned}
&\qquad\qquad h\left(\frac{\gamma\mu_1}{6(1-\epsilon+\gamma)}, \frac{\delta}{n}\right)\Bigg\} \\
&= \sum_{i=1}^{n} \min\Bigg\{\max\Bigg\{h\left(\frac{\Delta_i - \epsilon\mu_i}{4}, \frac{\delta}{n}\right), \\
&\qquad\qquad \min\left[h\left(\frac{\Delta_i}{4}, \frac{\delta}{n}\right), \max\left[h\left(\frac{\tilde{\alpha}_\epsilon}{6(1-\epsilon)}, \frac{\delta}{n}\right), h\left(\frac{\tilde{\beta}_\epsilon}{6(1-\epsilon)}, \frac{\delta}{n}\right)\right]\right]\Bigg\}, \\
&\qquad\qquad h\left(\frac{\gamma\mu_i}{6(1-\epsilon+\gamma)}, \frac{\delta}{n}\right)\Bigg\} \\
&\leqslant \sum_{i=1}^{n} \min\Bigg\{\max\Bigg\{h\left(\frac{\Delta_i - \epsilon\mu_i}{4}, \frac{\delta}{n}\right), \\
&\qquad\qquad \max\left[h\left(\frac{\Delta_i + \frac{\tilde{\alpha}_\epsilon}{1-\epsilon}}{12}, \frac{\delta}{n}\right), h\left(\frac{\Delta_i + \frac{\tilde{\beta}_\epsilon}{1-\epsilon}}{12}, \frac{\delta}{n}\right)\right]\Bigg\}, \\
&\qquad\qquad h\left(\frac{\gamma\mu_i}{6(1-\epsilon+\gamma)}, \frac{\delta}{n}\right)\Bigg\} \\
&= \sum_{i=1}^{n} \min\Bigg\{\max\Bigg\{h\left(\frac{\Delta_i - \epsilon\mu_i}{4}, \frac{\delta}{n}\right), h\left(\frac{\Delta_i + \frac{\tilde{\alpha}_\epsilon}{1-\epsilon}}{12}, \frac{\delta}{n}\right), h\left(\frac{\Delta_i + \frac{\tilde{\beta}_\epsilon}{1-\epsilon}}{12}, \frac{\delta}{n}\right)\Bigg\}, \\
&\qquad\qquad h\left(\frac{\gamma\mu_i}{6(1-\epsilon+\gamma)}, \frac{\delta}{n}\right)\Bigg\}
\end{aligned}
$$

Finally, we use Lemma 6.32 to bound the function $h(\cdot,\cdot)$. Since $\delta \leqslant 1/2$, $\delta/n \leqslant 2e^{-e/2}$. Further, $|\epsilon\mu_1 - \Delta_i| \leqslant 6$ for all i and $\epsilon \leqslant 1/2$ implies that $\frac{1}{6(1-\epsilon)}|\epsilon\mu_1 - \Delta_i| \leqslant 2$ and $\frac{1}{6(1-\epsilon)}\min(\tilde{\alpha}_\epsilon, \tilde{\beta}_\epsilon) \leqslant 2$. $\Delta_i \leqslant 8$ for all i, gives $0.25\Delta_i \leqslant 2$. Lastly, $\gamma \leqslant 6/\mu_1$ implies that $\frac{\gamma\mu_1}{6(1-\epsilon+\gamma)} \leqslant 2$. Therefore,

$$
\begin{aligned}
&\sum_{i=1}^{n} \min\Bigg\{\max\Bigg\{h\left(\frac{\Delta_i - \epsilon\mu_i}{4}, \frac{\delta}{n}\right), h\left(\frac{\Delta_i + \frac{\tilde{\alpha}_\epsilon}{1-\epsilon}}{12}, \frac{\delta}{n}\right), h\left(\frac{\Delta_i + \frac{\tilde{\beta}_\epsilon}{1-\epsilon}}{12}, \frac{\delta}{n}\right)\Bigg\}, \\
&\qquad\qquad h\left(\frac{\gamma\mu_i}{6(1-\epsilon+\gamma)}, \frac{\delta}{n}\right)\Bigg\} \\
&\leqslant \sum_{i=1}^{n} \min\Bigg\{\max\Bigg\{\frac{64}{(\epsilon\mu_1 - \Delta_i)^2}\log\left(\frac{2n}{\delta}\log_2\left(\frac{192n}{\delta(\epsilon\mu_1 - \Delta_i)^2}\right)\right),
\end{aligned}
$$

$$
\begin{aligned}
&\frac{576}{(\Delta_i + \frac{\tilde{\alpha}_\epsilon}{1-\epsilon})^2} \log\left(\frac{2n}{\delta}\log_2\left(\frac{1728n}{\delta(\Delta_i + \frac{\tilde{\alpha}_\epsilon}{1-\epsilon})^2}\right)\right), \\
&\frac{576}{(\Delta_i + \frac{\tilde{\beta}_\epsilon}{1-\epsilon})^2} \log\left(\frac{2n}{\delta}\log_2\left(\frac{1728n}{\delta(\Delta_i + \frac{\tilde{\beta}_\epsilon}{1-\epsilon})^2}\right)\right)\Bigg\}, \\
&\frac{144(1-\epsilon+\gamma)^2}{\gamma^2\mu_1^2} \log\left(\frac{2n}{\delta}\log_2\left(\frac{432(1-\epsilon+\gamma)^2 n}{\delta\gamma^2\mu_1^2}\right)\right)\Bigg\} \\
= \sum_{i=1}^{n} \min\Bigg\{ &\max\Bigg\{ \frac{64}{((1-\epsilon)\mu_1 - \mu_i)^2} \log\left(\frac{2n}{\delta}\log_2\left(\frac{192n}{\delta((1-\epsilon)\mu_1 - \mu_i)^2}\right)\right), \\
&\frac{576}{(\mu_1 + \frac{\tilde{\alpha}_\epsilon}{1-\epsilon} - \mu_i)^2} \log\left(\frac{2n}{\delta}\log_2\left(\frac{1728n}{\delta(\mu_1 + \frac{\tilde{\alpha}_\epsilon}{1-\epsilon})^2}\right)\right), \\
&\frac{576}{(\mu_1 + \frac{\tilde{\beta}_\epsilon}{1-\epsilon} - \mu_i)^2} \log\left(\frac{2n}{\delta}\log_2\left(\frac{1728n}{\delta(\mu_1 + \frac{\tilde{\beta}_\epsilon}{1-\epsilon} - \mu_i)^2}\right)\right)\Bigg\}, \\
&\frac{144(1-\epsilon+\gamma)^2}{\gamma^2\mu_1^2} \log\left(\frac{2n}{\delta}\log_2\left(\frac{432(1-\epsilon+\gamma)^2 n}{\delta\gamma^2\mu_1^2}\right)\right)\Bigg\}.
\end{aligned}
$$

□

6.G Comparison to top k

In the comparison to the complexity of a thresholding algorithm given the value of $\mu_1 - \epsilon$ or $(1-\epsilon)\mu_1$ is more transparent and can be read off of the upper and lower bound for ALL-ϵ directly. As we comment throughout and test in our experiments, the equivalent TOP-k problem would be if $|G_\epsilon|$ were know to the algorithm. Furthermore, we note that $(\mathtt{ST})^2$ is similar in its sampling strategy to `LUCB1` by Kalyanakrishnan et al. (2012). In fact, we show in our experiments, that $(\mathtt{ST})^2$ achieves similar performance as `LUCB1` given the value of $|G_\epsilon|$. In this section we compare the complexity of $(\mathtt{ST})^2$ to the complexity of `LUCB1` given the value of $|G_\epsilon|$ and show that they are within a constant factor. We focus on the additive case with a particular choice of ϵ but the idea is more general. As long as ϵ is such that $\alpha_\epsilon \approx \beta_\epsilon$ which is true except in pathological cases the same intuition is true for a

different constant. This equivalence is true despite LUCB1 having more information about the instance than $(\mathtt{ST})^2$! In all cases for this section, let $\gamma = 0$ for clarity.

In the case of Top-k, gaps are defined as:

$$\Delta_i^k = \begin{cases} \mu_i - \mu_{k+1}, & \text{if } 1 \leqslant i \leqslant k \\ \mu_k - \mu_i, & \text{if } k+1 \leqslant i \leqslant n \end{cases}$$

Further, define $c = \frac{\mu_k + \mu_{k+1}}{2}$. Kalyanakrishnan et al. (2012) show that

$$\Delta_i^k/2 \leqslant |\mu_i - c| \leqslant \Delta_i^k.$$

Hence, we may compare to $\frac{\mu_k + \mu_{k+1}}{2}$ as opposed to μ_k or μ_{k+1} and pay only a constant factor. We are interested in the setting where $G_\epsilon = \mu_k$. Let

$$\epsilon = \mu_1 - \frac{\mu_k + \mu_{k+1}}{2}.$$

Then

$$\mu_1 - \epsilon = \frac{\mu_k + \mu_{k+1}}{2}.$$

Furthermore,

$$\alpha_\epsilon = \beta_\epsilon = \mu_k - \frac{\mu_k + \mu_{k+1}}{2} = \frac{\mu_k - \mu_{k+1}}{2}.$$

For arms in G_ϵ^c (arms $k+1, \cdots, n$),

$$\frac{1}{(\mu_1 - \epsilon - \mu_i)^2} = \frac{1}{\left(\frac{\mu_k + \mu_{k+1}}{2} - \mu_i\right)^2},$$

matching their contribution to the complexity of returning the top-k arms up to a constant, given in Theorem 6 of Kalyanakrishnan et al. (2012), the upper bound on the complexity of LUCB.

For arms in G_ϵ, Theorem 6.5, their contribution to the sample complexity is

$$\max\left\{ \frac{1}{\left(\frac{\mu_k + \mu_{k+1}}{2} - \mu_i\right)^2}, \frac{1}{(\mu_1 + \alpha_\epsilon - \mu_i)^2}, \frac{1}{(\mu_1 + \beta_\epsilon - \mu_i)^2} \right\}$$

$$= \max\left\{ \frac{1}{\left(\frac{\mu_k+\mu_{k+1}}{2} - \mu_i\right)^2}, \frac{1}{\left(\mu_1 + \frac{\mu_k-\mu_{k+1}}{2} - \mu_i\right)^2} \right\}$$

For i such that $\mu_i \leqslant \frac{\mu_1+\mu_k}{2}$

$$\max\left\{ \frac{1}{\left(\frac{\mu_k+\mu_{k+1}}{2} - \mu_i\right)^2}, \frac{1}{\left(\mu_1 + \frac{\mu_k-\mu_{k+1}}{2} - \mu_i\right)^2} \right\} = \frac{1}{\left(\frac{\mu_k+\mu_{k+1}}{2} - \mu_i\right)^2},$$

equal to that arm's contribution to the top-k bound. For i such that $\mu_i \geqslant \frac{\mu_1+\mu_k}{2}$,

$$\begin{aligned}
\mu_1 + \frac{\mu_k - \mu_{k+1}}{2} - \mu_i &\geqslant \mu_1 + \frac{\mu_k - \mu_{k+1}}{2} - \frac{\mu_1 + \mu_k}{2} \\
&= \frac{2\mu_1 - \mu_1 - \mu_k + \mu_k - \mu_{k+1}}{2} \\
&= \frac{\mu_1 - \mu_{k+1}}{2} \\
&\geqslant \frac{1}{2}\left(\mu_1 - \frac{\mu_k + \mu_{k+1}}{2}\right).
\end{aligned}$$

Hence, their contribution to all-ϵ is at most $\frac{4}{\left(\mu_1 - \frac{\mu_k+\mu_{k+1}}{2}\right)^2}$. Note that

$$\mu_i - \frac{\mu_k + \mu_{k+1}}{2} \leqslant \mu_1 - \frac{\mu_k + \mu_{k+1}}{2}$$

Hence, for such arms, their contribution to the bound in top-k is at least

$$\frac{1}{\left(\mu_1 - \frac{\mu_k+\mu_{k+1}}{2}\right)^2}.$$

Combining all pieces, we see that the complexity of $(\texttt{ST})^2$ when $\epsilon = \mu_1 - \frac{\mu_k+\mu_{k+1}}{2}$ is within a constant factor of that of `LUCB1`.

6.H An elimination algorithm for general Lipschitz functions of the best arm

6.H.1 More general subsets of arms

EAST can be modified to find subsets of arms that satisfy a more general threshold condition. For a given $\Gamma : \mathbb{R} \to \mathbb{R}$, say we want to find the set $G_\Gamma(\nu)$ of all arms whose means are at least $\Gamma(\mu_1)$, i.e.,

$$G_\Gamma(\nu) := \{i : \mu_i \geqslant \Gamma(\mu_1)\}. \tag{6.28}$$

An example of the above good set is finding all multiplicative ϵ-good arms, which corresponds to $\Gamma(x) = (1-\epsilon)x$. Other choices that could be of interest depending on application are polynomials, exponential etc. The original good subset of arms in (6.1) is the case when $\Gamma(x) = x - \epsilon$. To obtain $G_\Gamma(\nu)$ correctly with probability $1-\delta$, we modify line 4 in EAST to be the following.

$$\text{Update } U_t \leftarrow \max_{x\in[a,b]} \Gamma(x) \text{ and } L_t \leftarrow \min_{x\in[a,b]} \Gamma(x), \tag{6.29}$$

$$\text{where } a := \max_{j\in\mathcal{A}} \hat{\mu}_j - C_{\delta/n}(t), b := \max_{j\in\mathcal{A}} \hat{\mu}_j + C_{\delta/n}(t). \tag{6.30}$$

If parameter $\gamma > 0$ is used, line 2 of EAST is modified to be the following.

$$\textbf{while } \mathcal{A} \not\subset G \text{ and } \max\{2C_{\delta/n}(t), U_t - L_t\} > \gamma/2 \textbf{ do}$$

Then the returned $G \cup \mathcal{A}$ satisfies $G_\Gamma(\nu) \subseteq G \cup \mathcal{A} \subseteq \{i : \mu_i \geqslant \Gamma(\mu_1) - \gamma\}$.

Theorem 6.29. *For any instance ν, any choice of $\delta > 0$, and any threshold function $\Gamma(\cdot)$ the modified EAST algorithm returns a set $G \cup \mathcal{A}$ that with probability $1-\delta$, contains all arms in the good set $G_\Gamma(\nu)$ defined in (6.28), and no arms with mean values less than $\Gamma(\mu_1) - \gamma$.*

Proof Sketch. The main step is to show that in the good event $\{\mu_i \in [\hat{\mu}_i - C_{\delta/n}(t), \hat{\mu}_i + C_{\delta/n}(t)] \forall i \in \mathcal{A}_t, \forall t \in \mathbb{N}\}$ the threshold value $\Gamma(\mu_1) \in [L_t, U_t]$ at all

times t, because

$$a := \max_{j \in \mathcal{A}_t} \hat{\mu}_j - C_{\delta/n}(t) \overset{\text{(i)}}{\leqslant} \mu_{\arg\max_{j \in \mathcal{A}_t} \hat{\mu}_j} \overset{\text{(ii)}}{\leqslant} \mu_1 \overset{\text{(iii)}}{\leqslant} u_1 \overset{\text{(iv)}}{\leqslant} \max_{j \in \mathcal{A}_t} \hat{\mu}_j + C_{\delta/n}(t) =: b,$$

where (i), (iii) are valid because the bounds for all arms are correct in the good event, (ii) is because μ_1 is the largest mean, and (iv) is because in the good event arm $1 \in \mathcal{A}_t$ at all t. Thus $\mu_1 \in [a, b]$ and from (4), $\Gamma(\mu_1) \in [L_t, U_t]$ at all times. So we can correctly eliminate arms using L_t and U_t.

For Lipschitz-continuous $\Gamma(\cdot)$, modified `EAST` has a sample complexity similar in form to `EAST` itself, Theorem 6.27.

Theorem 6.30. *Fix $\delta \in (0, 1/2]$. Let $\Gamma : \mathbb{R} \to \mathbb{R}$ be Lipschitz-continuous with constant $\mathcal{L} > 0$, i.e., $|\Gamma(x + y) - \Gamma(x)| \leqslant \mathcal{L}|y|$ for all $x, y \in \mathbb{R}$. Define $\Delta'_i := \Gamma(\mu_1) - \mu_i \; \forall i \neq 1$. Fix $\gamma \in [0, 8\max(1, \mathcal{L}))$ and an instance ν, such that $\max\{\frac{\Delta_i}{4}, \frac{\Delta'_i}{2+2\mathcal{L}}\} \leqslant 2$ for all i. With probability at least $1 - \delta$, the modified `EAST` algorithm returns the set $G \cup \mathcal{A}$ which satisfies $G_\Gamma(\nu) \subset G \cup \mathcal{A} \subset \{i : \mu_i \geqslant \Gamma(\mu_1) - \gamma\}$ after the following number of samples:*

$$\sum_{i=1}^{n} \min \left\{ \max \left\{ \frac{8(1+\mathcal{L})^2}{\Delta_i'^2} \log\left(\frac{2n}{\delta} \log_2\left(\frac{24(1+\mathcal{L})n}{\delta \Delta_i'^2}\right)\right), \right.\right.$$
$$\min\left[\frac{64}{\Delta_i^2}\log\left(\frac{2n}{\delta}\log_2\left(\frac{192n}{\delta\Delta_i^2}\right)\right),\right.$$
$$\left.\left.\left.\frac{8(1+\mathcal{L})^2}{\min(\alpha'_\epsilon, \beta'_\epsilon)^2}\log\left(\frac{2n}{\delta}\log_2\left(\frac{24(1+\mathcal{L})n}{\delta\min(\alpha'_\epsilon,\beta'_\epsilon)^2}\right)\right)\right]\right\},$$
$$\left.\frac{64\max(1,\mathcal{L})}{\gamma^2}\log\left(\frac{2n}{\delta}\log_2\left(\frac{192\max(1,\mathcal{L})n}{\delta\gamma^2}\right)\right)\right\} \tag{6.31}$$

where $\alpha'_\epsilon := \mu_k - \Gamma(\mu_1)$, and $\beta'_\epsilon := \Gamma(\mu_1) - \mu_{k+1}$ if $G_\Gamma^c \neq \emptyset$ and ∞ otherwise.

As expected, plugging $\Gamma(x) = x - \epsilon, \mathcal{L} = 1$ in (6.31) gives us the same expression as in Theorem 6.27.

6.H.2 Proof of Theorem 6.29

We first show that arm 1 is always present in $\mathcal{A}$. For if not, at the earliest time t when arm 1 is not in $\mathcal{A}$, we have that either

$$\hat{\mu}_1 + C_{\delta/n}(t) < L_t, \tag{6.32}$$

$$\text{or} \quad \hat{\mu}_1 + C_{\delta/n}(t) < a. \tag{6.33}$$

Inequality (6.32), if satisfied, would remove arm 1 from $\mathcal{A}$ at step of `EAST` , while inequality (6.33) would remove it at step . We show next that neither of these can be true in the good event $\mathcal{E}$ when all the bounds are correct. If (6.32) is true, then

$$\mu_1 < L_t = \min_{x \in [a,b]} \Gamma(x) \implies \mu_1 \notin [a, b].$$

Denoting $j^* := \arg\max_{j \in \mathcal{A}(t)} \hat{\mu}_j$, since $\mu_1 < b$, the above inequality implies that $\mu_1 < \mu_{j^*}$, which is a contradiction. Inequality (6.33) implies that $\mu_1 \notin [a, b]$, which leads to a contradiction in the same manner as above.

In the good event $\mathcal{E}$, the threshold value $\Gamma(\mu_1) \in [L_t, U_t]$ at all times t, because

$$a := \max_{j \in \mathcal{A}_t} \hat{\mu}_j - C_{\delta/n}(t) \overset{\text{(i)}}{\leqslant} \mu_{\arg\max_{j \in \mathcal{A}_t} \hat{\mu}_j} \overset{\text{(ii)}}{\leqslant} \mu_1 \overset{\text{(iii)}}{\leqslant} u_1 \overset{\text{(iv)}}{\leqslant} \max_{j \in \mathcal{A}_t} \hat{\mu}_j + C_{\delta/n}(t) =: b,$$

where (i), (iii) are valid because the bounds for all arms are correct in the good event, (ii) is because μ_1 is the largest mean, and (iv) is because in the good event arm $1 \in \mathcal{A}_t$ at all t. Thus $\mu_1 \in [a, b]$ and from (4), $\Gamma(\mu_1) \in [L_t, U_t]$ at all times. This allows us to show $G(t) \subseteq G_\Gamma$ at all t, for if not, there is an arm $i \in G_\Gamma^c \cap G(t)$ such that

$$\mu_i \geqslant \hat{\mu}_i(t) - C_{\delta/n}(t) \geqslant U_t \geqslant \Gamma(\mu_1) > \mu_i,$$

where the last inequality is because $i \in G_\Gamma^c$ and yields a contradiction. Next we show $G_\Gamma \subseteq \mathcal{A}(t) \cup G(t)$. For if not, suppose arm $i \in G_\Gamma$ is eliminated at time t'. On

the event $\mathcal{E}$ that implies

$$\mu_i \leqslant \hat{\mu}_i(t') + C_{\delta/n}(t') \leqslant L_{t'} \leqslant \Gamma(\mu_1),$$

which contradicts that $i \in G_\Gamma$. Finally, we show that if $\max\{2C_{\delta/n}(t), U_t - L_t\} \leqslant \gamma/2$, then $\mathcal{A}(t) \cup G(t) \subset \{i : \mu_i \geqslant \Gamma(\mu_1) - \gamma\}$. Suppose not, then there is an arm i whose $\mu_i < \Gamma(\mu_1) - \gamma$ and it is present in $\mathcal{A}(t) \setminus G(t)$, so that from lines 8 and 11

$$U_t \geqslant \hat{\mu}_i - C_{\delta/n}(t) \quad \text{and} \quad \hat{\mu}_i + C_{\delta/n}(t) > L_t.$$

Then using $\max\{2C_{\delta/n}(t), U_t - L_t\} \leqslant \gamma/2$, we have that

$$\max(U_t, \hat{\mu}_i + C_{\delta/n}(t)) - \min(L_t, \hat{\mu}_i - C_{\delta/n}(t)) \leqslant U_t - L_t + 2C_{\delta/n}(t) \leqslant \gamma,$$

which under event $\mathcal{E}$ implies $\mu_i \geqslant \Gamma(\mu_1) - \gamma$ giving a contradiction.

6.H.3 Proof of Theorem 6.30

The proof proceeds in a manner similar to the proof of Theorem 6.27by first obtaining bounds on T_i. For arm $i \in G_\Gamma^c$, we show $T_i \leqslant h(\Delta_i'/(2+2\mathcal{L}), \delta/n)$ by arguing that in the good event $\mathcal{E}$, for all $t > h(\Delta_i'/(2+2\mathcal{L}), \delta/n)$ the arm i is not in $\mathcal{A}(t)$. Let $j^* := \arg\max_{j \in \mathcal{A}(t)} \hat{\mu}_j$. Then,

$$\begin{aligned}
\hat{\mu}_i + C_{\delta/n}(t) \quad & < L_t = \min_{x \in [\hat{\mu}_{j^*} - C_{\delta/n}(t), \hat{\mu}_{j^*} + C_{\delta/n}(t)]} \Gamma(x) \\
& < \min_{x \in [\hat{\mu}_1 - C_{\delta/n}(t), \hat{\mu}_{j^*} + C_{\delta/n}(t)]} \Gamma(x), \\
\overset{\mathcal{E}}{\Leftarrow} \mu_i + 2C_{\delta/n}(t) \quad & < \min_{x \in [\mu_1 - 2C_{\delta/n}(t), \mu_{j^*} + 2C_{\delta/n}(t)]} \Gamma(x) < \min_{x \in [\mu_1 - 2C_{\delta/n}(t), \mu_1 + 2C_{\delta/n}(t)]} \Gamma(x) \\
& < \Gamma(\mu_1) - 2\mathcal{L}C_{\delta/n}(t), \\
\Leftrightarrow (2+2\mathcal{L})C_{\delta/n}(t) \quad & < \Gamma(\mu_1) - \mu_i =: \Delta_i' \Leftarrow t \geqslant h(\Delta_i'/(2+2\mathcal{L}), \delta/n).
\end{aligned}$$

The chain of inequalities above are true as we are increasing the domain of the minimization at each step. The reverse implication is true under $\mathcal{E}$. We can argue about

T_i for $i \in G_\Gamma$ in a similar manner. Furthermore, for $i \in G_\Gamma$, $T'_i \leqslant h(0.25\Delta_i, \delta/n)$ is true in exactly the same way as before. To obtain the final sample complexity, we use T_β and T_γ. As before, $T_\beta = \max_i\{T_i\} = h(\min(\alpha'_\epsilon, \beta'_\epsilon)/(2+2\mathcal{L}), \delta/n)$. The only difference is in the expression for T_γ, which because of the stopping condition in modified `EAST`, is defined as

$$T_\gamma := \min\{t : \max(2C_{\delta/n}(t), U_t - L_t) \leqslant \gamma/2\}.$$

$2C_{\delta/n}(t) \leqslant \gamma/2$ is ensured when $t > h(0.25\gamma, \delta/n)$. For the other part, we have that

$$U_t - L_t = \max_{x\in[\hat{\mu}_{j^*}-C_{\delta/n}(t),\hat{\mu}_{j^*}+C_{\delta/n}(t)]} \Gamma(x) - \min_{x\in[\hat{\mu}_{j^*}-C_{\delta/n}(t),\hat{\mu}_{j^*}+C_{\delta/n}(t)]} \Gamma(x) \leqslant 2\mathcal{L}C_{\delta/n}(t).$$

Thus $U_t - L_t \leqslant \gamma/2$ is ensured when $2\mathcal{L}C_{\delta/n}(t) \leqslant \gamma/2$, which occurs when $t \geqslant h(0.25\gamma/\mathcal{L}, \delta/n)$. Thus $T_\gamma = h(0.25\gamma/\max(1,\mathcal{L}), \delta/n)$. Plugging in the expressions we obtain the final sample complexity as

$$\sum_{i=1}^{n} \min\left\{\max\left\{h\left(\frac{\Delta'_i}{2+2\mathcal{L}}, \frac{\delta}{n}\right), \min\left[h\left(0.25\Delta_i, \frac{\delta}{n}\right), h\left(\frac{1}{2+2\mathcal{L}}\min(\alpha'_\epsilon, \beta'_\epsilon), \frac{\delta}{n}\right)\right]\right\}, h\left(\frac{0.25\gamma}{\max(1,\mathcal{L})}, \frac{\delta}{n}\right)\right\}$$

Finally, we use Lemma 6.32 to bound the function $h(\cdot,\cdot)$. Since $\delta \leqslant 1/2$, $\delta/n \leqslant 2e^{-e/2}$. Further, recall that we have assumed $\max\{\frac{\Delta_i}{4}, \frac{\Delta'_i}{2+2\mathcal{L}}\} \leqslant 2$ for all i. Likewise, this implies that $\frac{\min \alpha'_\epsilon, \beta'_\epsilon}{2+2\mathcal{L}} \leqslant 2$. Lastly, $\gamma \leqslant 8\max(1,\mathcal{L})$ implies that $\frac{0.25}{\max(1,\mathcal{L})} \leqslant 2$. Therefore,

$$\begin{aligned}
&\sum_{i=1}^{n} \min\left\{\max\left\{h\left(\frac{\Delta'_i}{2+2\mathcal{L}}, \frac{\delta}{n}\right), \min\left[h\left(0.25\Delta_i, \frac{\delta}{n}\right), h\left(\frac{\min(\alpha'_\epsilon, \beta'_\epsilon)}{2+2\mathcal{L}}, \frac{\delta}{n}\right)\right]\right\}, h\left(\frac{0.25\gamma}{\max(1,\mathcal{L})}, \frac{\delta}{n}\right)\right\} \\
&\leqslant \sum_{i=1}^{n} \min\left\{\max\left\{\frac{8(1+\mathcal{L})^2}{\Delta_i'^2}\log\left(\frac{2n}{\delta}\log_2\left(\frac{24(1+\mathcal{L})n}{\delta\Delta_i'^2}\right)\right),\right.\right.
\end{aligned}$$

$$\min\left[\frac{64}{\Delta_i^2}\log\left(\frac{2n}{\delta}\log_2\left(\frac{192n}{\delta\Delta_i^2}\right)\right),\right.$$
$$\left.\frac{8(1+\mathcal{L})^2}{\min(\alpha'_\epsilon,\beta'_\epsilon)^2}\log\left(\frac{2n}{\delta}\log_2\left(\frac{24(1+\mathcal{L})n}{\delta\min(\alpha'_\epsilon,\beta'_\epsilon)^2}\right)\right)\right]\Bigg\},$$
$$\frac{64\max(1,\mathcal{L})}{\gamma^2}\log\left(\frac{2n}{\delta}\log_2\left(\frac{192\max(1,\mathcal{L})n}{\delta\gamma^2}\right)\right)\Bigg\}$$

□

6.I Technical Lemmas

Lemma 6.31. *If* $a > 1$, $b > e$, *and* $t > \max(a\log(2b\log(ab)), e)$, *then* $\frac{a\log(b\log(t))}{t} \leqslant 1$

Proof. **Step 1:** Plug in $t = a\log(2b\log(ab))$ to the expression $\frac{a\log(b\log(t))}{t}$.

$$\frac{a\log(b\log(a\log(2b\log(ab))))}{a\log(2b\log(ab))} = \frac{\log(b\log(a\log(2b\log(ab))))}{\log(2b\log(ab))}$$

Since $\log(\cdot)$ increases monotonically, the above is less than 1 if $b\log(a\log(2b\log(ab))) \leqslant 2b\log(ab)$.

$$\begin{aligned}
&b\log(a\log(2b\log(ab))) \leqslant 2b\log(ab)\\
&\overset{(b>0)}{\iff} \log(a\log(2b\log(ab))) \leqslant 2\log(ab)\\
&\iff a\log(2b\log(ab)) \leqslant (ab)^2\\
&\iff \log(2b\log(ab)) \leqslant ab^2\\
&\iff 2b\log(ab) \leqslant e^{ab^2}
\end{aligned}$$

which is true if $a, b > 1$.

Step 2: Next, for $t > a\log(2b\log(ab))$, we wish to show that the inequality $\frac{a\log(b\log(t))}{t} \leqslant 1$ still holds. To do so, it suffices to show that $f(t) = \frac{a\log(b\log(t))}{t}$ is

decreasing for $t > a\log(2b\log(ab))$. To see this, take the derivative.

$$f'(t) = \frac{a}{t^2\log(t)} - \frac{a\log(b\log(t))}{t^2} = \frac{a}{t^2}\left(\frac{1}{\log(t)} - \log(b(\log(t))\right)$$

This is negative when $\frac{1}{\log(t)} < \log(b(\log(t))$. Let $u = b\log(t)$. The previous is equivalent to the condition $b < u\log(u)$. For $t > e$, $u > b$ and $b > e$. Hence $b < u\log(u)$ completing the proof. □

Lemma 6.32. *For $\delta < 2e^{-e/2}$, $\Delta \leqslant 2$,*

$$t \geqslant \frac{4}{\Delta^2}\log\left(\frac{2}{\delta}\log_2\left(\frac{12}{\delta\Delta^2}\right)\right) \implies C_\delta(t) = \sqrt{\frac{4\log(\log_2(2t)/\delta)}{t}} \leqslant \Delta.$$

Proof.

$$\sqrt{\frac{4\log(\log_2(2t)/\delta)}{t}} \leqslant \Delta \iff \frac{4\frac{8}{\Delta^2}\log\left(\frac{1}{\delta\log(2)}\log(2t)\right)}{t} \leqslant 1.$$

If $\Delta \leqslant 2$, then $8/\Delta^2 \geqslant 2 > 1$. Similarly, if $\delta < 2e^{-e/2} < \frac{1}{e\log(2)}$, then $\frac{1}{\delta\log(2)} > e$. Hence, by Lemma 6.31, setting $a = \frac{8}{\Delta^2}$ and $b = \frac{1}{\delta\log(2)}$, the above is true if

$$2t \geqslant \max\left(\frac{8}{\Delta^2}\log\left(\frac{2}{\delta\log(2)}\log\left(\frac{8}{\delta\Delta^2\log(2)}\right)\right), e\right).$$

Trivially, $\delta\log(2) < 2$. Hence, $\delta < 2e^{-e/2}$ and $\Delta \leqslant 2$ implies

$$\frac{8}{\Delta^2}\log\left(\frac{2}{\delta\log(2)}\log\left(\frac{8}{\delta\Delta^2\log(2)}\right)\right) \geqslant 2\log\left(\frac{2}{\delta}\log_2\left(\frac{2}{\delta\log(2)}\right)\right) \geqslant 2\log(2/\delta) > e.$$

Therefore, we may simplify the maximum as

$$t \geqslant \frac{4}{\Delta^2}\log\left(\frac{2}{\delta}\log_2\left(\frac{12}{\delta\Delta^2}\right)\right) \geqslant \frac{4}{\Delta^2}\log\left(\frac{2}{\delta}\log_2\left(\frac{8}{\delta\Delta^2\log(2)}\right)\right)$$

which implies the desired result. □

Lemma 6.33. *For any function* $h(\cdot,\cdot) : \mathbb{R}^+ \times \mathbb{R}^+ \to \mathbb{R}^+$ *that decreases monotonically in its first argument, we have that for any* $a, b, c, \delta \in \mathbb{R}^+$

$$\min\left(h(a,\delta), h(b,\delta)\right) \leqslant h\left(\frac{a+b}{2}, \delta\right)$$

and

$$\min\{h(a,\delta), \max[h(b,\delta), h(c,\delta)]\} \leqslant \max\left\{h\left(\frac{a+b}{2},\delta\right), h\left(\frac{a+c}{2},\delta\right)\right\}.$$

Proof. First, we bound the expression $\min\left(h(a,c), h(b,c)\right)$.

$$\min\left(h(a,\delta), h(b,\delta)\right) = h\left(\max(a,d),\delta\right) \leqslant h\left((a+b)/2,\delta\right)$$

Next, we bound, expressions of the form $\min\{h(a,\delta), \max[h(b,\delta), h(c,\delta)]\}$ using the above inequality.

$$\begin{aligned}\min\{h(a,\delta), \max[h(b,\delta), h(c,\delta)]\} &= \max\left\{\min\left[h(a,\delta), h(b,\delta)\right], \min\left[h(a,\delta), h(c,\delta)\right]\right\} \\ &\leqslant \max\left\{h\left((a+b)/2,\delta\right), h\left((a+c)/2,\delta\right)\right\}.\end{aligned}$$

□

7 FINDING NEAREST NEIGHBORS FROM A NOISY DISTANCE ORACLE

7.1 Introduction

In Chapter 5, we presented a method to efficiently learn the nearest neighbor graph of a set of points from noisy distance measurements. It employed triangle inequality bounds to more efficiently find nearest neighbors and built this into an active sampling algorithm. An advantage to only employing the triangle inequality is that the method automatically can be used in *any* metric space, regardless of the form of the metric. Furthermore, it used these bounds to achieve the optimal $O(n \log(n))$ rate for finding a nearest neighbor graph of n points while only assuming noisy distance estimates– the only algorithm to do so. Nearest neighbor graphs give $O(1)$ access to a query point's nearest neighbor among a set of $n-1$ other points. A downside, however is that if one wishes to identify the nearest neighbor of a query point that is not one of the n nodes of the graph, this requires $O(n)$ samples. Furthermore, in the worst case, adding or removing a node in the graph requires recomputing the entire nearest neighbor graph. Hence, though nearest neighbor graph methods can be powerful, the can also be brittle.

In this chapter, we take an alternate approach. Instead, our goal is to find a data structure that can efficiently answer nearest neighbor queries for *any* query point. In adding flexibility to which points may be queried, we will slightly sacrifice on the complexity of answering nearest neighbor queries. While Chapter 5 presents a method that answered queries for a restricted set in $O(1)$ complexity, we will instead present a method than can answer queries in $O(\log(n))$ time for any query point. In particular, this is still an exponential improvement over naive search which would require $O(n)$ complexity. Precisely, consider the following problem:

Problem Statement: Consider a set of n points $\mathcal{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$ in a metric space $(\mathcal{M}, d)$. The metric is unknown, but we can query a stochastic oracle for an estimate of any pairwise distance. Build a data structure such that for any query point $\mathbf{q}$ unknown a priori to the algorithm, it returns its nearest neighbor $\mathbf{x}_q := \min_{\mathbf{x}_i \in \mathcal{X}} d(\mathbf{q}, \mathbf{x}_i)$ with probability at least $1 - \delta$ in as few oracle queries as possible.

The performance of algorithms for this problem is measured along several axes.

1. **Query time:** Given $\mathcal{X}$ and a new query point $\mathbf{q}$, how many samples does an algorithm require to return the nearest neighbor in $\mathcal{X}$, $\mathbf{x}_q$?

2. **Build time:** All non-trivial algorithms for this problem build a data structure to answer the queries. How many initial, burn-in samples are needed to compute this data structure?

3. **Sub-linear insertion time:** If a data structure to answer nearest neighbor queries on $\mathcal{X}$ has already been computed, how many samples are needed to insert $\mathbf{x}'$ such that the data structure can now answer nearest neighbor queries to the set $\mathcal{X} \cup \{\mathbf{x}'\}$?

4. **Sub-linear removal time:** If a data structure to answer nearest neighbor queries on $\mathcal{X}$ has already been computed, how many samples are needed to remove $\mathbf{x}' \in \mathcal{X}$ such that the data structure can now answer nearest neighbor queries to the set $\mathcal{X} \backslash \{\mathbf{x}'\}$?

5. **Memory footprint:** How much memory is needed to store the data structure?

6. **Accuracy:** What is the probability that a method correctly returns $\mathbf{x}_q$?

Before proceeding with our discussion for answering nearest neighbor queries with noisy data, it is helpful to set expectations for what is optimal performance in the noiseless regime. Establishing general lower bounds for this problem is challenging as different algorithms assume different query models and different metric spaces. In general, jointly requiring only $O(\log(n))$ samples to answer queries and

$O(n \log(n))$ build time is considered optimal. $O(n)$ memory complexity is also state of the art for non-trivial algorithms that build a data structure and do not perform linear search over $\mathcal{X}$ to answer new queries. Optimal accuracy is 100%, though some algorithms trade accuracy for speed. Insertion and removal time varies for different methods, but at minimum, well performing algorithms should be able to insert of remove points without fully recomputing the data structure. $O(\log(n))$ insertion and removal time is state of the art.

7.1.1 Related work

Classical methods to answer nearest neighbor queries include kd trees (Bentley, 1975) which achieve $O(n \log(n))$ build time, $O(\log(n))$ query time, and are known to be computationally efficient. The core drawback to kd trees is that they lack a uniform accuracy guarantee. While kd trees perform well for some query points, they do not for others, and it is not computationally feasible to known which points are likely to succeed and which are not (Dasgupta and Sinha, 2013). Many methods have tried to achieve similar build and query time performance while achieving higher accuracy. Two major approaches to achieve this are to either add randomness in the build phase to achieve a uniform accuracy guarantee or to exploit measures of dimensionality and structure of metric spaces to give accuracy guarantees. Dasgupta and Sinha (2013) provide several algorithms in the first category that are similar in spirit to kd trees and achieve high accuracy. Krauthgamer and Lee (2004) and Beygelzimer et al. (2006) provide algorithms in the second category which achieve perfect accuracy for any query point. Haghiri et al. (2017) provide a method that answers nearest neighbor queries from triplet comparisons directly and can be seen as a combination of both approaches to achieve higher accuracy. In general, nearest neighbor problems (from noiseless measurements) are well studied and we direct the reader to Bhatia et al. (2010) for a general survey. Additionally, a parallel line of work has considered the approximate nearest neighbor problem where for a query point $\mathbf{q}$ and $\epsilon > 0$ one wishes to find any point in the set $\{\mathbf{x} \in \mathcal{X} : d(\mathbf{q}, \mathbf{x}) \leqslant (1+\epsilon) \min_{z \in \mathcal{X}} d(\mathbf{q}, \mathbf{z})\}$. We refer the reader to Wang and

Banerjee (2014); Andoni et al. (2018) for a survey of methods. Of particular interest to this chapter will be the `Cover Tree` algorithm presented in Beygelzimer et al. (2006). In this chapter, we extend this method to deal with noisy queries.

7.1.2 The Curse of Dimensionality for Nearest Neighbor Search

The "curse of dimensionality" greatly impacts the performance of nearest neighbor algorithms. In particular, there exist worst case arrangements of data where any algorithm either needs $\Omega(n)$ samples to find a query point $\mathbf{q}$'s nearest neighbor or suffers low accuracy. These examples occur for high dimensional data in particular. As an example, consider the following construction. Let $\mathcal{X}'$ be the set of the n vertices of the simplex in $\mathbb{R}^{n-1}$. Form $\mathcal{X}$ by applying a random ϵ-perturbation to each vertex independently. Let $\mathbf{q} = [1/n, \cdots, 1/n]^T \in \mathbb{R}^n$ be the geometric center of the simplex. By design, its distance to every point in $\mathcal{X}$ is nearly equal. One can show that in this setting, any algorithm requires $\Omega(n)$ samples to return $\mathbf{q}$'s nearest neighbor. Intuitively, this means that any algorithm must check the distance from $\mathbf{q}$ to each vertex. To see this, consider $\mathbf{x}_i \in \mathcal{X}$. Suppose one wishes to bound $d(\mathbf{q}, \mathbf{x}_i)$ without explicitly measuring this distanced. This is important, because if sufficiently tight bounds exist, then an algorithm may be able to declare that $\mathbf{x}_i$ is not $\mathbf{q}$'s nearest neighbor without querying the distance. Indeed, this was the core idea behind the `ANNTri` algorithm proposed in Chapter 5 to efficiently estimate the nearest neighbor graph of $\mathcal{X}$ from noisy distance measurements. Without additional assumptions, the only method to do this is the triangle inequality which for another $\mathbf{x}_j \in \mathcal{X}$ guarantees

$$|d(\mathbf{q}, \mathbf{x}_j) - d(\mathbf{x}_i, \mathbf{x}_j)| \leqslant d(\mathbf{q}, \mathbf{x}_i) \leqslant d(\mathbf{q}, \mathbf{x}_j) + d(\mathbf{x}_i, \mathbf{x}_j).$$

Because all distances are nearly 1 by construction, the lower bound is near 0 and the upper bound is near 2. Hence, all bounds are vacuous. Intuitively then, algorithms cannot cleverly use the triangle inequality to reduce the set of possible nearest neighbors and must query the distance to each point in $\mathcal{X}$.

In general, if the data is not in Euclidean space, a notion of dimension may

be poorly defined or ill-suited to quantify the performance of nearest neighbor algorithms. Instead, algorithms consider several measures of dimension that help quantify their performance. As a general rule, performance improves with low dimensional data. In this chapter, we will make use of the *expansion constant* as in (Haghiri et al., 2017; Krauthgamer and Lee, 2004; Beygelzimer et al., 2006).

Definition 7.1. *The expansion constant of a set of points $\mathcal{S}$ is the smallest $c \geqslant 2$ such that $|B(\mathbf{x}, 2r)| \leqslant c|b(\mathbf{x}, r)|$ for any $\mathbf{x} \in \mathcal{S}$ and any $r > 0$ where $B(\mathbf{x}, r)$ is the ball of radius $r > 0$ centered at $\mathbf{x}$ according to the distance measure associated with the ambient metric space.*

As an example, for a set of points in arranged uniformly on a surface in $\mathbb{R}^d$ have an expansion constant of at most 2^d Beygelzimer et al. (2006).

7.2 Problem setup and summary of our approach

We denote distances as $d_{i,j}$ where $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_{\geqslant 0}$ is a distance function satisfying the standard axioms and for a query point $\mathbf{q}$ define $\mathbf{x}_q := \arg\min_{\mathbf{x} \in \mathcal{X}} d(x_i, \mathbf{q})$. For a query point $\mathbf{q}$ and $\mathbf{x}_i \in \mathcal{X}$, let $d_{q,i}$ denote $d(\mathbf{q}, \mathbf{x}_i)$. For a set Q, let $d(\mathbf{q}, Q) := \min_{i \in Q} d_{p,i}$. Though the distances are unknown, we are able to draw independent samples of its true value according to a stochastic distance oracle, i.e. querying

$$Q(i, j) \quad \text{yields a realization of} \quad d_{i,j} + \eta, \tag{7.1}$$

where η is a zero-mean subGaussian random variable assumed to have scale parameter $\sigma = 1$. We let $\hat{d}_{i,j}(s)$ denote the empirical mean of the s queries of the distance oracle, $Q(i, j)$. The number of $Q(i, j)$ queries made until time t is denoted as $T_{i,j}(t)$. A possible approach to obtain the $\mathbf{q}$'s nearest neighbor is to repeatedly query $Q(q, i)$ for each i and report $\arg\min_{\mathbf{x} \in \mathcal{X}} \hat{d}_{q,i}(t)$. To improve our query efficiency, we could instead adaptively sample to focus queries on distances that we estimate are smaller. However, this would still require $O(n)$ samples to learn $\mathbf{x}_q$ as we would query the distance of each of n points to $\mathbf{q}$. Instead, we modify the Cover Tree algorithm of

Beygelzimer et al. (2006) to handle noisy inputs instead. Two core innovations that make this possible are

- Casting the problem of learning a cover as finding all ϵ-good arms in multi-armed bandits and using $(\texttt{ST})^2$ presented in Chapter 6 to develop a method to learn covers.
- Using thresholding bandits, such as Jamieson and Jain (2018), to compute the approximate distance from a query point to a set.

Our proposed algorithm `Bandit Cover Tree` (`BCT`) uses the above ideas to build a data structure to efficiently answer nearest neighbor queries from noisy distance measurements with high probability.

7.3 Cover Trees

Before presenting our method and associated results, we review core intuition behind the Cover Tree algorithm from Beygelzimer et al. (2006) in the noiseless setting ($\eta = 0$ with probability 1). As the name suggests, a cover tree is a tree-based data structure where each level of the tree forms a *cover* of $\mathcal{X}$. Each level is indexed by an integer i which decreases as one descends the tree. To avoid additional notation, we will represent the top of the tree as level ∞ and the bottom as $-\infty$ though in practice one would record integers i_{top} and $i_{\texttt{bottom}}$ denoting the top and bottom level of the tree and need only explicitly store the tree at levels between i_{top} and $i_{\texttt{bottom}}$. Each node in the tree corresponds to a point in $\mathcal{X}$, but points in $\mathcal{X}$ may correspond to multiple nodes in the tree. Reviewing Beygelzimer et al. (2006), let C_i denote the set of nodes at level i, referred to as the i^{th} cover. The cover tree algorithm is designed so that each level of the tree i obeys three invariants:

1. (nesting) $C_i \subset C_{i-1}$. Hence, the points corresponding to nodes at level i are also correspond to nodes in all lower levels.
2. (covering tree) For ever $p \in C_{i-1}$, there exists a $q \in C_i$ such that $d(p, q) \leqslant 2^i$, and the child node in level $i - 1$ is connected to its parent in level i.

3. (Separation) For any $p, q \in C_i$ with $p \neq q$, $d(p, q) > 2^i$.

These invariances are originally derived from Krauthgamer and Lee (2004). Beygelzimer et al. (2006) show that there routines obey these invariants, and we will make use of them in our proofs and to build intuition. The core idea of this method is that the i^{th} level of the tree (with i decreasing as one descends the tree) is a cover of resolution 2^i. When navigating the tree for a query point $\mathbf{q}$ at level i, one identifies all possible ancestor nodes of $\mathbf{x}_q$ at that level. When descending the tree, this set is refined until only a single parent is possible, $\mathbf{x}_q$ itself. The *nesting* invariance allows one to easily traverse the tree from parent to child. The *covering tree* invariance connects $\mathbf{x}_q$ to its parents and ancestors so that one may traverse the tree with query point $\mathbf{q}$ and end at $\mathbf{x}_q$. Lastly, the *separation* invariance ensures that there is a single parent to each node, avoiding redundant paths traversing the tree. This aids in reducing the memory necessary to store cover trees.

7.4 The Bandit Cover Tree Algorithm

The `Bandit Cover Tree` algorithm is comprised of three methods. `Noisy-Find-Nearest` finds nearest neighbors given a computed cover tree. `Noisy-Insert` allows one to insert points into a tree and can be used to construct a new tree. `Noisy-Remove` can be used to remove points from the tree.

7.4.1 Finding Nearest Neighbors with a Cover Tree

We begin by discussing how to identify the nearest neighbor of a query point $\mathbf{q}$ if the tree has already been constructed. This is to provide intuition for how cover trees work and the impact that noise has on tree traversal. After, we will show how to build a cover tree from a set of n points, $\mathcal{X}$ and how to insert and remove points to the tree when one only has access to a noisy oracle. Throughout, we take $\mathcal{T}$ to denote the cover tree. The nodes of each level i forms a cover C_i, and we may identify $\mathcal{T}$ by the covers at each level: $\mathcal{T} := \{C_\infty, \cdots, C_i, C_{i-1}, \cdots, C_{-\infty}\}$. Assume that we are given a fixed query point $\mathbf{q}$, and the expansion constant of

the set $\mathcal{X} \cup \{\mathbf{q}\}$ is bounded by c. The algorithm proceeds by keeping track of a set Q_i at each level i of all possible ancestors of $\mathbf{x}_q$, $\mathbf{q}$'s nearest neighbor. This is summarized in Algorithm 11 and adapted from Beygelzimer et al. (2006). The algorithm proceeds by exploring descending from level i to level $i-1$ and exploring all children of the nodes in Q_i. From this, the algorithm recomputes possible parents of $\mathbf{x}_q$ to form Q_{i-1}. The algorithm terminates when it reaches a level i_{bottom}. Throughout, we will represent the children of any node $p \in \mathcal{T}$ as $\text{children}(p)$. For simplicity, we assume that the nearest neighbor of $\mathbf{q}$ is unique. If this does not hold, one may instead modify what the algorithm returns to return any element of $Q_{-\infty}$. We make use of a novel subroutine to build cover sets from noisy distance measurements, based on finding all ϵ-good arms in stochastic bandits. This is given in Algorithm 12. Similar to $(\texttt{ST})^2$, the routine maintains anytime confidence widths, $C_\delta(t)$ such that for an empirical mean of the distance $d_{q,i}$: $\hat{d}_{q,i}(t)$ of t samples, we have $\mathbb{P}(\bigcup_{t=1}^{\infty} |\hat{d}_{q,i}(t) - d_{q,i}| > C_\delta(t)) \leqslant \delta$. For this work, we take $C_\delta(t) = \sqrt{\frac{c_\phi \log(\log_2(2t)/\delta)}{t}}$ for a constant c_ϕ. It suffices to take $c_\phi = 4$, though tighter bounds are known and should be used in practice, e.g. Jamieson et al. (2014); Kaufmann et al. (2016); Howard et al. (2018). Where clear, we will drop the dependence on t and $T_i(t)$.

Given the set $Q_i \subset C_i$ of $\mathbf{x}_q$'s ancestors in level i, to proceed to level $i-1$, the algorithm first computes all children of nodes in Q_i given by the set $Q \subset C_{i-1}$. It then uses Algorithm 12 to identify the subset of Q denoted Q_{i-1} that contains all ancestors of $\mathbf{x}_q$ in level $i-1$. Precisely, this is the set $\{i \in Q : d(q,i) \leqslant \min_{j \in Q} d(q,j) + 2^{i-1}\}$. In particular, this can be represented as the set of all 2^i-good arms (in the additive sense) in the set Q with the key distinction that we want the *smallest* distances not the largest. This can be achieved by multiplying each distance estimate by -1 and finding the 2^i-good arms.

Remark 7.2. *This method can be altered so that a point $p \in \mathcal{X}$ that corresponds to a node in C_i at level i of the cover tree $\mathcal{T}$ is defined to always be contained in children(p), even if a node corresponding to point p is not present in C_{i-1} (the next level down) explicitly. Traversal is identical since this is baked into the definition of the children$(\cdot)$ function, but*

Algorithm 11 `Noisy-Find-Nearest`

Require: Cover tree $\mathcal{T}$, failure probability δ, expansion constant c if known, query point $\mathbf{q}$, callable distance oracle $Q(\cdot,\cdot)$, subroutine `Identify-Cover`.

1: Let $Q_\infty = C_\infty$
2: **for** $i = \infty$ down to $i = -\infty$ **do**
3: Let $Q = \bigcup_{p \in Q_i} \text{children}(p)$
4: **if** c is known: **then**
5: Let $\alpha = \min\left\{ \left\lceil \frac{\log(n)}{\log(1+1/c^2)} + 1 \right\rceil, n \right\}$
6: **else**
7: let $\alpha = n$
8: **end if**
9: \\ Identify the set: $\{i \in Q : d(q,i) \leqslant \min_{j \in Q} d(q,j) + 2^{i-1}\}$
10: Compute $Q_{i-1} =$ `Identify-Cover` $(Q, \delta/\alpha, Q(\cdot,\cdot), \mathbf{q}, i)$
11: **end for**
12: **return** $Q_{-\infty}$, a singleton set containing $\mathbf{x}_q$.

this can save on the memory needed to store $\mathcal{T}$.

7.4.1.1 Approximate Nearest Neighbors

These methods can also easily be adapted to return an ϵ-approximate nearest neighbor though we will not analyze this setting theoretically. An ϵ-approximate nearest neighbor is defined as any point in the set $\{i : d(q,i) \leqslant (1+\epsilon)\min_j d(q,j)\}$. To find approximate nearest neighbors, add an additional line that exits the for loop if $d(q,Q_i) \geqslant 2^{i+1}(1+1/\epsilon)$ where the distance from $\mathbf{x}_q$ to the set Q_i is defined as $\min_{j \in Q_i} d(q,j)$. This requires an additional bandit routine. One could modify `FindPos` which finds positive means from Chapter 6 to perform this computation. In particular, we seek any point $j \in Q_i$, such that $2^{i+1}(1+1/\epsilon) - d(q,j) > 0$. Importantly however, `FindPos` would need to be modified so that if no such point exists, the algorithm does not run forever. This can trivially be achieved by adding a line after line 6 in `FindPos` such that if $\hat{\mu}_{i_k} < -\beta_k$, arm i_k is removed from all future rounds and adding an additional termination condition that the algorithm terminates if no points remain. If this occurs, the algorithm may declare that $d(q,Q_i) < 2^{i+1}(1+1/\epsilon)$. Finally, after exiting the 'For' loop, since Q_i may not be

Algorithm 12 `Identify-Cover`

Require: Failure probability δ, query point $\mathbf{q}$, Oracle $Q(\cdot,\cdot)$, and set $\mathbf{Q}$, Cover resolution i

Query oracle once for each point in Q
Initialize $T_i \leftarrow 1$, update $\hat{d}_{q,i}$ for each $i \in \{1, 2, \ldots, |Q|\}$
Empirical cover set: $\widehat{G} = \{i : \hat{d}_{q,i} \leqslant \max_j \hat{d}_{q,j} + 2^i\}$
$U_t = \min_j \hat{d}_{q,j}(T_j) - C_{\delta/|Q|}(T_j) + 2^i$ and $L_t = \max_j \hat{d}_{q,j}(T_j) + C_{\delta/|Q|}(T_j) + 2^i$
Known points: $K = \{i : \hat{d}_{q,i}(T_i) - C_{\delta/|Q|}(T_i) > L_t \text{ or } \hat{d}_{q,i}(T_i) + C_{\delta/|Q|}(T_i) < U_t\}$
while $K \neq Q$ **do**
    Call oracle $Q(q, i_1)$ for $i_1(t) = \arg\min_{i \in \widehat{G} \setminus K} \hat{d}_{q,i}(T_i) + C_{\delta/|Q|}(T_i)$, update $T_{i_1}, \hat{d}_{q,i_1}$
    Call oracle $Q(q, i_2)$ for $i_2(t) = \arg\max_{i \in \widehat{G}^c \setminus K} \hat{d}_{q,i}(T_i) - C_{\delta/|Q|}(T_i)$, update $T_{i_2}, \hat{d}_{q,i_2}$
    Call oracle $Q(q, i^*)$ for $i^*(t) = \arg\min_i \hat{d}_{q,i}(T_i) - C_{\delta/|Q|}(T_i)$, update $T_{i^*}, \hat{d}_{q,i^*}$
    Update bounds L_t, U_t, sets $\widehat{G}$, K
end while
return The set cover set with resolution 2^i: $\{i : \hat{d}_{q,i}(T_i) - C_{\delta/|Q|}(T_i) < U_t\}$

singleton, the algorithm should be modified to return the closest point in Q_i to q. This can simply be achieved via any best-arm identification algorithm such as lil'UCB Jamieson et al. (2014) applied to the negatives of the distance estimates (since we want the smallest distance possible, not the largest).

7.4.2 Building and Altering a Cover Tree

In this section we demonstrate how to construct a cover tree, insert new points, and remove points. We begin by discussing insertion since constructing the full tree from n points can trivially be achieved by inserting each point one at a time to an empty tree.

7.4.2.1 Insertion

Suppose we have access to a cover tree $\mathcal{T}$ on a set $\mathcal{S}$. If $\mathcal{S} = \emptyset$, $\mathcal{T}$ is empty at all levels. We wish to insert a point p into $\mathcal{T}$ such that we now have a cover tree on the set

$\mathcal{S} \cup \{p\}$. Intuitively, the insertion algorithm can be thought of as beginning at the highest resolution cover, at level $-\infty$ and climbing back up the tree, inserting p in each cover set C_i for all i until it reaches a level i_p such that $\min_{j \in C_{i_p}} d(p, j) \leqslant 2^{i_p}$ where a suitable parent node exists. The algorithm then assigns any $p' \in C_{i_p}$ such that $d(p, p') \leqslant 2^{i_p}$ and terminates. As trees are traditionally traversed via their roots not their leaves, we state this algorithm recursively beginning at the $i = \infty$ level at the top.

We provide pseudocode in Algorithm 13. The algorithm is similar to `Insert` from Beygelzimer et al. (2006) but includes additional logic to handle a noisy oracle. In particular, lines 2-8 implement a simple thresholding bandit similar to Algorithm 1 of Jamieson and Jain (2018) in the familiy-wise probability of detection – family-wise probability of error setting. In particular, it identifies all points within 2^i of the nearest in the set Q. This must be done for every candidate level i that p might be inserted into until a suitable parent node is found at level i_p. Each time we perform this comparison, there is a probability of error. This presents a challenge for the algorithm. `Noisy-Insert` is recursive and performs a thresholding bandit at each level. Hence, if it makes an error on any recursive call, the algorithm may fail. Thus, we must ensure that the probability of `Noisy-Insert` making an error in any recursive call is at most δ. Since the level i_p where p is added is unknown and depends on the query point p, the number of recursive calls before success is unknown to the algorithm. Therefore, it is not a priori obvious how to perform the appropriate Bonferroni correction to ensure the probability of an error in any recursive call is bounded by δ. A seemingly attractive approach is to use a summable sequence of δ_i depending on the level i such that $\sum_{\text{levels} i} \delta_i = \delta$. For instance, $\delta_i = 2^{-i}$ would be suitable if the root of $\mathcal{T}$ is at level 1. However, this would lead to higher sample complexity when summing the complexity of many individual bandit problems with decreasing error probabilities.

Instead, `Noisy-Insert` shares samples between rounds, similar to the technique used by `ANNTri` to find nearest neighbor graphs. By the nesting invariance of cover trees, we have that $C_i \subset C_{i-1}$. Therefore, when we descend the tree from level i to $i - 1$, we already have samples of the distance of some points in C_{i-1} to p. We

Algorithm 13 `Noisy-Insert`

Require: Cover tree $\mathcal{T}$ on n points, cover set Q_i, failure probability δ, point p to be inserted, callable distance oracle $Q(\cdot,\cdot)$, level i
Require: Empirical estimates of $\hat{d}_{p,i}$ and T_i for all $i \in C_i$ \\ let both be 0 if no samples have been colllected
Let $Q = \bigcup_{j \in Q_i} \text{children}(j)$
Query oracle once for each point in $Q \cap \{j : T_j = 0\}$
Initialize $T_i \leftarrow 1$, update $\hat{d}_{p,i}$ for each $j \in Q \cap \{j : T_j = 0\}$
\\ compute the set $\{j \in Q : d(p,j) \leqslant 2^i\}$
Known points: $K = \{j : \hat{d}_{p,j}(T_j) + C_{\delta/n}(T_j) \leqslant 2^i \text{ or } \hat{d}_{p,j}(T_i) - C_{\delta/n}(T_j) > 2^i\}$
while $|K| \neq |Q|$ **do**
    Call oracle $Q(p, i^*)$ for $i^*(t) = \arg\min_{j \notin K} \hat{d}_{p,j}(T_j) - C_{\delta/n}(T_j)$
    Update $T_{i^*}, \hat{d}_{p,i^*}$
    Update set K
end while
\\ If $d(p,Q) > 2^i$
if $\{j \in Q : \hat{d}_{p,j}(T_j) + C_{\delta/n}(T_j) \leqslant 2^i\} = \emptyset$ **then**
    Return:"no parent found"
else
    Define $Q_{i-1} = \{j \in Q : \hat{d}_{p,j}(T_j) + C_{\delta/n}(T_j) \leqslant 2^i\}$
    if $Q_i \cap Q_{i-1} \neq \emptyset$ and `Noisy-Insert`$(p, \mathcal{T}, Q_{i-1}, i-1, \delta, Q(\cdot,\cdot), \{j \in Q : \hat{d}_{p,j}(T_j)\}, \{j \in Q : T_j\})$ = "no parent found" **then**
        Choose any $p' \in Q_i \cap Q_{i-1}$
        Insert p in children(p') \\ modify the tree and insert p bc parent found
        Return:"parent found"
    else
        Return:"no parent found"
    end if
end if

simply reuse these samples and share them from round to round. Furthermore, since $\mathcal{T}$ is assumed to be a cover tree on n points, we trivially union bound each confidence width to hold with probability $1 - \delta/n$ such that all bounds for all recursive calls holds with probability at least $1 - \delta$. Note that it is possible to use the depth bound from Lemma 4.3 of Beygelzimer et al. (2006) to instead union bound by $1 - O(\delta/c^2 \log(n))$ if c were known. This trick was used for `Noisy-Find-Nearest`.

However, as `Noisy-Insert` is used for construction of the tree and adding new points to the dataset, it is unlikely that c is known to the experimenter before any data has been collected. Hence, we use the simple union bound of $1 - \delta/n$ which is always true independent of c.

7.4.2.2 Removal

Next we show how to remove a point p from level i of a cover tree $\mathcal{T}$. The process of removal is similar to insertion but slightly more complicated as we must find new parents for all of p's children in $\mathcal{T}$. We provide pseudocode in Algorithm 14 which is adapted from Beygelzimer et al. (2006) to handle noisy estimates.

Note that when we compute the distance to the set $Q_{i'}$ in lines 13 and 20, due to the nesting invariance of cover trees and the fact that samples are reused between rounds, $T_i > 0$ for all $i \in Q_{i'}$. Hence, it is unnecessary to collect any initial samples. Note that `Noisy-Remove` not only queries distances to the point p to be removed but also to p's children in order to find them new parents. Because of this, we index samples as $T_{i,j}$ denoting the number of calls to the distance oracle $Q(i, j)$. Furthermore, we union bound with δ/n^2 instead of δ/n where the additional factor of n derives from a trivial bound that p has at most $n - 1$ children nodes. If the expansion rate is known, it is possible to union bound as $\delta/c^4 n$ since Lemma 4.1 of Beygelzimer et al. (2006) bounds the number of children as c^4 for any node p. However, as the other factor of n remains unchanged by this, the improvement from $\log(n^2/\delta)$ to $\log(c^4 n/\delta)$ only impacts constant factors in the sample complexity. Hence, we ignore this. It is technically possible to use a depth bound on $\mathcal{T}$ from Lemma 4.3 of Beygelzimer et al. (2006) to achieve a dependence of $\log(c^6 \log(n)/\delta)$. In particular, this reduces the n dependence due to the union bound to be only doubly logarithmic. As the expansion constant is traditionally not known, we avoid the added complexity.

Algorithm 14 `Noisy-Remove`

Require: Cover tree $\mathcal{T}$ on n points, past cover sets $\{Q_i, \cdots, Q_\infty\}$, failure probability δ, point p to be removed, callable distance oracle $Q(\cdot,\cdot)$, level i

Require: Empirical estimates of $\hat{d}_{p,i}$ and T_i for all $i \in C_i$ \\ let both be 0 if no samples have been collected

Let $Q = \bigcup_{p \in Q_i} \text{children}(p)$

\\ Remove from lower levels first

$\mathcal{T}, \{i, j : \hat{d}_{i,j}(T_j)\}, \{i, j : T_{i,j}\} \leftarrow$ `Noisy-Remove`$(\mathcal{T}, \{Q_{i-1}, \cdots, Q_\infty\}, \delta, p, Q(\cdot,\cdot), i-1, \{i, j : \hat{d}_{i,j}(T_{i,j})\}, \{i, j : T_{i,j}\})$

if $p \in Q$ **then**

  Remove p from C_{i-1} and children(parent(p))

  for $q \in$ children(p) **do**

    $i' \leftarrow i - 1$

    \\ compute the set $\{j \in Q_{i'} : d(q, j) \leqslant 2^{i'}\}$

    Query oracle once for each point in $Q \cap \{j : T_{q,j} = 0\}$

    Initialize $T_{q,j} \leftarrow 1$, update $\hat{d}_{q,j}$ for each $j \in Q \cap \{j : T_{q,j} = 0\}$

    Known points: $K = \{j : \hat{d}_{q,j}(T_{q,j}) + C_{\delta/n^2}(T_{q,j}) \leqslant 2^{i'}$ or $\hat{d}_{q,j}(T_{q,j}) - C_{\delta/n^2}(T_{q,j}) > 2^{i'}\}$

    while $|K| \neq |Q_{i'}|$ **do**

      Call oracle $Q(q, i^*)$ for $i^*(t) = \arg\min_{j \notin K} \hat{d}_{q,j}(T_{q,j}) - C_{\delta/n^2}(T_{q,j})$

      Update T_{q,i^*}, $\hat{d}_{q,i^*}$

      Update set K

    end while

    while $\{j \in Q_{i'} : \hat{d}_{q,j}(T_{q,j}) + C_{\delta/n^2}(T_{q,j}) \leqslant 2^{i'}\} = \emptyset$ **do**

      Add q to the sets $C_{i'}$ and $Q_{i'}$. Increment i'.

      \\ for the incremented i', recompute the set $\{j \in Q_{i'} : d(q, j) \leqslant 2^{i'}\}$

      Known points: $K = \{j : \hat{d}_{q,j}(T_{q,j}) + C_{\delta/n^2}(T_{q,j}) \leqslant 2^{i'}$ or $\hat{d}_{q,j}(T_{q,j}) - C_{\delta/n^2}(T_{q,j}) > 2^{i'}\}$

      while $|K| \neq |Q_{i'}|$ **do**

        Call oracle $Q(q, i^*)$ for $i^*(t) = \arg\min_{j \notin K} \hat{d}_{q,j}(T_j) - C_{\delta/n^2}(T_j)$

        Update T_{q,i^*}, $\hat{d}_{q,i^*}$

        Update set K

      end while

    end while

    Choose any $q' \in \{j \in Q_{i'} : \hat{d}_{q,j}(T_j) + C_{\delta/n}(T_j) \leqslant 2^{i'}\}$

    make $q' = \text{parent}(q)$

  end for

end if

7.5 Theoretical Guarantees of `Bandit Cover Tree`

We wish to provide guarantees for the performance measures described in the introduction: query time, build time, insertion time, removal time, memory footprint, and accuracy. In this section, we show that `BCT`'s achieve state of the art performance on these metrics despite only having access to noisy data. All proofs are deferred to Section 7.7. As `BCT` is adapted from cover trees directly, it inherits many properties and theoretical guarantees proven in Beygelzimer et al. (2006). The core challenge becomes proving correctness for the additions we have made to make the algorithm robust to noise and accounting for the number of extra calls to the distance oracle by the algorithm. The algorithms were stated for clarity as having a root at level ∞ and descending to level ∞. All analysis will be conducted with respect to a tree with root at i_{top} and i_{bottom} however, as a real tree cannot be infinitely tall.

7.5.1 Memory

We begin by showing that a cover tree can efficiently be stored. Naively, a cover tree $\mathcal{T}$ on $\mathcal{X}$ can be stored using $O(n(i_{top} - i_{bottom}))$ memory where $n = |\mathcal{X}|$ and $i_{top} - i_{bottom}$ is the height of the tree. This follows from each level having at most n nodes trivially and there being $i_{top} - i_{bottom}$ levels. For a well balanced tree, we expect that $i_{top} - i_{bottom} = O(\log(n))$ leading to an overall memory complexity of $O(n \log(n))$. In fact, it is possible to do better.

Lemma 7.3. *A bandit cover tree requires* $O(n)$ *space to be stored.*

$O(n)$ memory is possible due to the nesting and covering tree invariants. By the nesting invariant, if a point p is present in the i^{th} cover C_i, then it is present in the cover sets of all lower levels. By the covering tree invariant, each point has a unique parent in the tree. Therefore, to store a cover tree, one need only store 1) which level of the tree each point first appears in 2) each point's parent node in the level above where it appears. After a node first appears in the tree, it may

be represented implicitly in all lower levels by defining $p \in \text{children}(p)$ for the children$(\cdot)$ function.

7.5.2 Accuracy

Next, we show that `BCT` is accurate with high probability. This requires three guarantees:

1. Search accuracy: Given a correctly constructed cover tree $\mathcal{T}$ on set $\mathcal{X}$ and query point $\mathbf{q}$, we must show that `Noisy-Find-Nearest` correctly identifies $\mathbf{q}$'s nearest neighbor with high probability.

2. Insertion accuracy: Given a correctly constructed cover tree $\mathcal{T}$ and a point p to be inserted, we must show that `Noisy-Insert` returns a valid cover tree that includes p neighbor with high probability. This will ensure accuracy in constructing a cover tree from scratch since `Noisy-Insert` can be called repeatedly to build a cover tree on a set $\mathcal{X}$.

3. Removal accuracy: Given a correctly constructed cover tree $\mathcal{T}$ and a point p to be removed, we must show that `Noisy-Remove` returns a valid cover tree that without p neighbor with high probability.

7.5.2.1 Search Accuracy

We begin by showing search accuracy. `Noisy-Find-Nearest` is an extension of the `Find-Nearest` algorithm from Beygelzimer et al. (2006). It follows the same logic as `Find-Nearest` but with an additional routine– `Identify-Cover`– to compute cover sets with from noisy data. By Theorem 2 of Beygelzimer et al. (2006), `Find-Nearest` succeeds with probability 1. If `Identify-Cover` correctly returns the cover set each time that it is called in `Noisy-Find-Nearest`, then `Noisy-Find-Nearest` will correctly return $\mathbf{x}_q$. Hence, the following guarantee is derived by showing that the probability of that `Identify-Cover` makes an error is small.

Lemma 7.4. *Fix any $\delta \leqslant 1/2$ and a query point $\mathbf{q}$. Let $\mathcal{T}$ be a cover tree on a set $\mathcal{X}$. `Noisy-Find-Nearest` returns $\mathbf{x}_q \in \mathcal{X}$ with probability at least $1 - \delta$.*

7.5.2.2 Insertion Accuracy

Next, we verify insertion accuracy with high probability. Similarly, `Noisy-Insert` is based on the `Insert` method of Beygelzimer et al. (2006) and inherits theoretical some of its properties. To handle noisy data, it makes use of a simple thresholding bandit to compute the set $\{i \in Q : d(p, i) \leqslant 2^i\}$, the subset of the set of children Q that are within 2^i of the point p to be inserted. From this set, it can check if $d(p, Q) > 2^i$ and can compute a new cover set Q_{i-1}. The following lemma ensures that `Noisy-Insert` succeeds with high probability.

Lemma 7.5. *Fix any $\delta > 0$. Let $\mathcal{T}$ be a cover tree on a set $\mathcal{X}$ and p be a point to insert. `Noisy-Insert` correctly returns a cover tree on $\mathcal{X} \cup \{p\}$ with probability at least $1 - \delta$.*

7.5.2.3 Removal Accuracy

Finally, we verify removal accuracy with high probability. `Noisy-Remove` is based on the `Remove` method of Beygelzimer et al. (2006) and inherits theoretical some of its properties as well. Throughout, assume that we have access to a completed tree $\mathcal{T}$ on the set $\mathcal{X}$ and we wish to remove a point $p \in \mathcal{X}$ from $\mathcal{T}$ to produce a new $\mathcal{T}'$ on $\mathcal{X} \backslash \{p\}$. To handle noisy data, it makes use of a simple thresholding bandit to compute the set $\{j \in Q : d(p, j) \leqslant 2^i\}$ similar to `Noisy-Insert` except for multiple values of i and for children of p as well. From this set, it can check if $d(p, Q) > 2^i$ and can compute and reassign parents to p's children nodes. The following lemma ensures that `Noisy-Remove` succeeds with high probability.

Lemma 7.6. *Fix any $\delta > 0$. Let $\mathcal{T}$ be a cover tree on a set $\mathcal{X}$ and $p \in \mathcal{X}$ be a point to remove. `Noisy-Remove` correctly returns a cover tree on $\mathcal{X} \backslash \{p\}$ with probability at least $1 - \delta$.*

7.5.3 Query Time Complexity

In the previous section, we proved that the algorithm succeeds with probability $1 - \delta$ for all routines: search, insertion, and removal. In this section we begin to answer the question of how many calls to the distance oracle it requires to perform these operations. We begin by analyzing the query time complexity of `Bandit Cover Tree`: the number of calls to the distance oracle made when answering a nearest neighbor query. To do so, we will make use of the *expansion constant*, a data-dependent measure of dimensionality. In particular, for a query point $\mathbf{q}$. We assume that the set $\mathcal{X} \cup \{\mathbf{q}\}$ has an expansion constant of c as defined in Definition 7.1. Note that this quantity is for analysis purposes only and is not required by the algorithm. `Noisy-Find-Nearest` can take in the expansion constant or a bound on it if it is known, but this is not required by the algorithm, and the algorithm can run without it.

To bound query time, we appeal to the concept of explicit and implicit nodes as discussed in 7.5.1. Each point in $\mathcal{X}$ may correspond to multiple nodes in the tree. The first time a point appears as a node at the highest level where it is present, we say that it is *explicitly represented*. Therefore, the algorithm saves in memory that the new point has entered the tree. Using the nesting invariant of cover trees, if a node's direct parent is corresponds to the same point in $\mathcal{X}$ as does the node itself, we say that the node is *implicitly represented*. In particular, the cover tree need not store this node in subsequent levels and merely records its presence when traversing the tree. Recall that the set of nodes at each level i of the cover tree is denoted C_i. `Noisy-Find-Nearest` proceeds by computing a cover set $Q_i \subset C_i$ at each level of the tree. Extending the concept of explicit and implicit representations of nodes, we say that a cover set Q_i is implicitly represented if it only contains nodes that are implicitly represented. This plays an important role in our computation of query time. We may use the size of the last explicitly represented cover in combination with a bound on the height of the tree to control the overall complexity.

Theorem 7.7. *Fix $\delta < 1/2$, a cover tree $\mathcal{T}$ on set $\mathcal{X}$, and a query point $\mathbf{q}$. Let $|\mathcal{X}| = n$ and assume that the expansion rate of $\mathcal{X} \cup \{\mathbf{q}\}$ is c (unknown to the algorithm). If*

`Noisy-Find-Nearest` succeeds, which occurs with probability $1-\delta$*, then* `Noisy-Find-Nearest` *returns* $\mathbf{q}$*'s nearest neighbor in at most*

$$O\left(c^7 \log(n) \log\left(\frac{n}{\delta}\right) \overline{\kappa}\right)$$

the total number of calls to the noisy distance oracle where parameter $\overline{\kappa}$ *is defined in the proof and depends on* X *and* $\mathbf{q}$*.*

Remark 7.8. *The term* $\overline{\kappa}$ *captures the average effect of noise on this problem. It is similar to the term* $\overline{\Delta^{-2}}$ *in Theorems 5.6 and 5.7. As the noise variance goes to* 0*, this term becomes* 1*, and* `Noisy-Find-Nearest` *converges to the behavior of* `Find-Nearest` *in Beygelzimer et al. (2006).*

Corollary 7.9. *Under the same conditions as Theorem 7.7, if* c *is known to the algorithm, the number of queries to the distance oracle is at most*

$$O\left(c^7 \log(n) \log\left(\frac{c^2 \log(n)}{\delta}\right) \overline{\kappa}\right)$$

7.5.4 Insertion Time and Build Time Complexity

Next we bound the number of calls to the distance oracle necessary to insert a new point p into a cover tree T. We analyze the case that p must be inserted within the tree. If p instead is a new root of the tree, the same bound applies, though it is possible to prove a tighter bound for this special case.

Theorem 7.10. *Fix* $\delta > 0$*, a cover tree* T *on set* X*, and a point to insert* p*. Let* $|X| = n$ *and assume that the expansion rate of* $X \cup \{p\}$ *is* c*. Run* `Noisy-Insert` *with failure probability* $1 - \delta$ *and pass it the root level cover set:* $C_{i_{top}}$ *and level* $i = i_{top}$*. If* `Noisy-Insert` *succeeds, which occurs with probability* $1 - \delta$*, then it returns a cover tree on* $X \cup \{p\}$ *in at most*

$$O\left(c^7 \log(n) \log\left(\frac{n}{\delta}\right) \overline{\kappa_p}\right)$$

calls to the noisy distance oracle where parameter $\overline{\kappa_p}$ *is defined in the proof and depends on* X *and* p*.*

Remark 7.11. *As in the statement of Theorem 7.7, the term $\overline{\kappa_p}$ captures the average effect of noise on this problem, and as the noise variance goes to 0, this term becomes 1.* `Noisy-Insert` *converges to the behavior of* `Insert` *in Beygelzimer et al. (2006).*

As discussed in Section 7.4.2.1, to construct a cover tree from scratch, one need only call `Noisy-Insert` on each point in $\mathcal{X}$ and add them the tree one at a time. The following theorem bounds the complexity of this process.

Theorem 7.12. *Fix $\delta > 0$ and set n points $\mathcal{X}$. Assume that the expansion rate of $\mathcal{X}$ is c. Calling* `Noisy-Insert` *with failure probability δ/n on each point in $\mathcal{X}$ one at a time, returns a cover tree $\mathcal{T}$ on $\mathcal{X}$ correctly with probability at least $1 - \delta$ in at most*

$$O\left(c^7 n \log(n) \log\left(\frac{n^2}{\delta}\right) \widetilde{\kappa}\right)$$

calls to the noisy distance oracle where $\widetilde{\kappa} := \frac{1}{n}\sum_{i \in \mathcal{X}} \overline{\kappa}_i$ for $\overline{\kappa}_i$ defined in the proof of Theorem 7.10.

Remark 7.13. *Note that the value of $\widetilde{\kappa}$ depends on the order in which points are inserted into the tree $\mathcal{T}$. It is possible that some insertion orders are more efficient than others. However, knowing the optimal order to insert points would require knowledge of the exact distances themselves, which is not available in this problem. Absent this knowledge, we assume that points are inserted in lexicographic order, with $\mathbf{x}_1$ first and $\mathbf{x}_n$ last.*

7.5.5 Removal Time Complexity

Next we bound the number of calls to the distance oracle necessary to insert a new point p into a cover tree $\mathcal{T}$. We analyze the case that p must be inserted within the tree. If p instead is a new root of the tree, the same bound applies, though it is possible to prove a tighter bound for this special case.

Theorem 7.14. *Fix $\delta > 0$, a cover tree $\mathcal{T}$ on set $\mathcal{X}$, and a point $p \in \mathcal{X}$ to remove. Let $|\mathcal{X}| = n$ and assume that the expansion rate of $\mathcal{X}$ is c. Run* `Noisy-Remove` *with failure*

probability $1-\delta$ *and pass it the root level cover set:* $C_{i_{top}}$. *If* `Noisy-Remove` *succeeds, which occurs with probability* $1-\delta$, *then it returns a cover tree on* $\mathcal{X}\backslash\{p\}$ *in at most*

$$O\left(c^{11}\log^2(n)\log\left(\frac{n^2}{\delta}\right)\widehat{\kappa}_p\right)$$

calls to the noisy distance oracle where parameter $\widehat{\kappa}_p$ *is defined in the proof and depends on* $\mathcal{X}$ *and* p.

Remark 7.15. *As in the statement of Theorem 7.7, the term* $\widehat{\kappa}_p$ *captures the average effect of noise on this problem, and as the noise variance goes to* 0, *this term becomes* 1. `Noisy-Remove` *converges to the behavior of* `Remove` *in Beygelzimer et al. (2006).*

Remark 7.16. *The proof of the* `Remove` *algorithm in Beygelzimer et al. (2006) follows a somewhat different logic owing to the separation invariance combined with exact knowledge of distance. It is not immediately clear if the same trick which leads to a tighter result can work in this case because distances cannot be computed exactly. We employ a somewhat more approach result to control the complexity for this bound.*

7.6 Conclusion

In this chapter, we introduced the `Bandit Cover Tree` framework. `BCT` builds on top of the `Cover Tree` algorithm by Beygelzimer et al. (2006). In particular, we extend three methods in that work, `Find-Nearest`, `Insert`, and `Remove` to handle a noisy oracle instead of the noiseless oracle assumed by the authors. Furthermore, as the noise level decreases, the performance of `Noisy-Find-Nearest`, `Noisy-Insert`, and `Noisy-Remove` converges to their noiseless counterparts. We bound the accuracy, memory footprint, build complexity, insertion and removal complexities, and query complexities for `BCT`. In particular, we show a query complexity that is $O(\log(n))$, insertion complexity is $O(\log^2(n))$, removal complexities is $O(\log^3(n))$ and a construction complexity of $O(n\log^2(n))$. The query complexity matches the state of the art n dependence for the (noiseless) nearest neighbor search problem

with additional terms accounting for the noise in this problem. The insertion, construction, and removal complexities are also near state of the art. Lastly a memory footprint of $O(n)$ and accuracy of $1 - \delta$ are both optimal in this problem. In particular, a tree with n leaves requires $\Omega(n)$ memory to store and taking $\delta \to 0$, we match the optimal accuracy of 100%.

Note that the theoretical results in this work are subtly different than those of Beygelzimer et al. (2006) beyond the obvious difference between the noisy and noiseless settings. Beygelzimer et al. (2006) bound computational complexity and assume that calls to the distance oracle require $O(1)$ computation. In our case, we directly bound the number of calls to the distance oracle with less regard for the computational overhead incurred. Some operations such as set intersections or passing vectors of empirical estimates of distances to recursive calls could lead to additional computational overhead than is present in the noiseless setting. It is an open question for future work how to bound the computational complexity in the noisy regime and if better methods exist.

Furthermore, the results depend heavily on the expansion rate, c. For Euclidean data, c may be as large as 2^d for uniformly spread out points in $\mathbb{R}^d$. Hence, for moderate values of n and larger values of d, dependences such as c^{11} in Theorem 7.14 may dominate the complexity instead of the n dependence as is traditionally assumed. Some works such as Haghiri et al. (2017); Dasgupta and Sinha (2013) trade accuracy for improved dependence on c or other measures of dimension for metric spaces. Instead, these algorithms guarantee that the correct nearest neighbor for any query point is return with probability at least $1-\delta_{c,n}$ independent of the chosen query point. δ is tunable but depends on c, n, and often other parameters. Though it can be tuned up or down, $\delta_{c,n}$ usually depends on parameters unknown to the practitioner and cannot be known precisely. These methods often achieve good empirical performance, however. In may be interesting to modify an algorithm in this regime to the noisy data setting. One especially promising approach in this vein is to modify the `Spill Tree` algorithm from Dasgupta and Sinha (2013). A generic method to do Top-k identification in multi-armed bandits such as `LUCB` from Kalyanakrishnan et al. (2012) could be used as a subroutine to handle all calls

to the distance oracle.

Lastly, we note that `Bandit Cover Tree` can be used to solve the nearest neighbor graph problem presented in Chapter 5. In particular, given a set of points $\mathcal{X}$, first build a cover tree $\mathcal{T}$ on $\mathcal{X}$. This can be done in $O(n \log(n))$ calls to the distance oracle. Next, modify `Noisy-Find-Nearest` to find 2-nearest neighbors as discussed in Beygelzimer et al. (2006). For each $p \in \mathcal{X}$, p will be its own nearest neighbor in $\mathcal{T}$ since p is contained in the graph, but $p^* = \arg\min_{j \in \mathcal{X} \setminus \{p\}} d_{p,j}$ will be its second nearest neighbor. From this, we may connect p to p^* in the nearest neighbor graph. This process can be repeated for each $p \in \mathcal{X}$. Each 2-nearest neighbor query will require $O(\log(n))$ calls to the distance oracle. Summing this over the n points in $\mathcal{X}$ with the $O(n \log(n))$ build complexity gives an overall complexity of $O(n \log(n))$ to learn a nearest neighbor graph. This matches the optimal rate given in Theorem 5.7 bounding the complexity of `ANNTri` but under *far more general* conditions on the set $\mathcal{X}$.

7.7 Proofs

7.7.1 Memory and accuracy proofs

First, we prove Lemma 7.3 which bounds the memory necessary to build, store, or search a `Bandit Cover Tree`

Proof of Lemma 7.3. By Theorem 1 of Beygelzimer et al. (2006), ordinary cover trees can be stored in $O(n)$ space. Hence, we must only show that any addition to any algorithm has not added more than $O(n)$ additional memory needed. In `Identify-Cover`, we trivially have that $|Q| \leqslant n$. In `Noisy-Insert` and `Noisy-Remove`, empirical estimates of distances and associated confidence widths are shared by all recursive calls to the algorithm. There are at most n distance estimates and n confidence widths, contributing $2n$ to the space requirement. By passing these values by reference rather than value for different recursive calls of each algorithm, the same $2n$ numbers may be shared for all recursive calls. The only other stored values, the indices for i_{top} and i_{bottom} contribute $O(1)$ space. Thus the total memory

overhead for handling noisy estimates is $O(n)$ and the total memory requirement is $O(n)$ as well. □

Next we prove several Lemmas that guarantee that the `Noisy-Find-Nearest`, `Noisy-Insert`, and `Noisy-Remove` methods all succeed with high probability. We begin with search accuracy.

Proof of Lemma 7.4. By Theorem 2 of Beygelzimer et al. (2006), `Find-Nearest` succeeds with probability 1. `Noisy-Find-Nearest` is adapted directly from `Find-Nearest` except that it uses `Identify-Cover` to identify individual cover sets at it descends the tree. Hence, we wish to show that the probability of `Identify-Cover` making an error in any call is at most δ. `Identify-Cover` is equivalent to $(\texttt{ST})^2$ from Chapter 6 except that it finds the *smallest* distances not the largest arms. It is equal to $(\texttt{ST})^2$ given the negative of all rewards. Thus, by Theorem 6.5, `Identify-Cover` given a failure probability of δ/α succeeds with probability $1 - \delta/\alpha$.

By definition, $\alpha = \min\left\{\left\lceil \frac{\log(n)}{\log(1+1/c^2)} + 1 \right\rceil, n\right\}$. By Lemma 4.3 of Beygelzimer et al. (2006), $i_{\text{top}} - i_{\text{bottom}} \leqslant \alpha$. Hence, `Identify-Cover` is called at most α times (once per level during traversal). A union bound implies that the probability of an error in any call is at most δ, completing the proof. □

Next, we show that `Noisy-Insert` succeeds with high probability.

Proof of Lemma 7.5. To show correctness, we begin by verifying that the thresholding bandit routine in lines $2-9$ does not fail in any recursive call of `Noisy-Insert`. Define the event

$$\mathcal{E} := \bigcap_{k \in [n]} \bigcap_{t=1}^{\infty} \left\{ |\hat{d}_{p,k}(t) - d_{p,k}| \leqslant C_{\delta/n}(t) \right\}$$

By definition of $C_\delta(t)$, we have that $P(\mathcal{E}^c) \leqslant \delta$ where we have used the assumption that $|\mathcal{X}| = n$ so there are only n different points explicitly represented in $\mathcal{T}$. For the remainder of the proof, we assume that $\mathcal{E}$ occurs and show that it leads to correctness in the thresholding bandit routine.

Let $\mu > 0$ denote any threshold. On $\mathcal{E}$, we have that

$$\{k : d_{p,k} < \mu\} \supset \left\{k : \hat{d}_{p,k} + C_{\delta/n}(T_k(t)) < \mu\right\}$$

for any set of values $\{k : T_k(t)\}$ at any time t. Similarly, we have that

$$\{k : d_{p,k} > \mu\} \supset \left\{k : \hat{d}_{p,k} - C_{\delta/n}(T_k(t)) > \mu\right\}.$$

Therefore, if the algorithm stops sampling when either

$$\hat{d}_{p,k} + C_{\delta/n}(T_k(t)) < \mu$$

or

$$\hat{d}_{p,k} - C_{\delta/n}(T_k(t)) > \mu$$

for every point k and any value of μ, then on event $\mathcal{E}$, we have identified the sets $\{k : d_{p,k} < \mu\}$ and $\{k : d_{p,k} > \mu\}$ correctly. In particular, the above superset relation holds with equality.

Applying this to $\mu = 2^i$ for different values of i in the algorithm we see that at all times the thresholding bandit implemented in lines $2 - 9$ succeeds. Since these bounds are shared between recursive calls of the algorithm, the routine succeeds in every recursive call.

We conclude by showing that if these sets have been computed correctly, which occurs when $\mathcal{E}$ occurs, then the algorithm correctly computes all quantities needed for the `Insert` algorithm in Beygelzimer et al. (2006).

First, note that after the thresholding bandit terminates

$$\{i \in Q : \hat{d}_{p,i}(T_i) + C_{\delta/n}(T_i) \leqslant 2^i\} = \emptyset \implies d(p, Q) > 2^i.$$

Next, note that when the thresholding bandit routine terminates at line 8, we have that

$$\left\{k \in Q : \hat{d}_{p,k} + C_{\delta/n}(T_k(t)) \leqslant 2^i\right\} = \left\{k \in Q : d_{p,k} \leqslant 2^i\right\} = Q_{i-1}$$

where the last equality holds by definition in the `Insert` algorithm.

Finally, by the nesting invariance of cover trees and the definition of children in a tree, we have that $Q_i \subset Q$. Hence $Q_i \cap Q_{i-1} \neq \emptyset$ is equivalent to the condition that $d(p, Q_i) \leqslant 2^i$. Therefore, applying Theorem 3 of Beygelzimer et al. (2006) completes the proof. □

Finally, we show that `Noisy-Remove` succeeds with high probability.

Proof of Lemma 7.6. Similar to the proof for `Noisy-Insert`, we begin by showing correctness of the thresholding bandit in lines $9 - 16$. Again, we must verify that it does not fail in any recursive call of `Noisy-Remove`. Define the event

$$\mathcal{E} := \bigcap_{j,k \in [n]} \bigcap_{t=1}^{\infty} \left\{ |\hat{d}_{j,k}(t) - d_{j,k}| \leqslant C_{\delta/n^2}(t) \right\}$$

By definition of $C_\delta(t)$, we have that $P(\mathcal{E}^c) \leqslant \delta$ where we have used the assumption that $|\mathcal{X}| = n$ so there are only n different points explicitly represented in $\mathcal{T}$ and there are at most $\binom{n}{2}$ pairs of distances. Note that especially for higher levels i of the tree it is possible that $|Q_i| < n$. Hence a weaker union bound is possible. As $|Q_i|$ is unknown a priori, we take the naive bound that $|Q_i| \leqslant n$ for any round i, though it is possible to instead alter the union bound in different recursive calls to `Noisy-Remove`.

For the remainder of the proof, we assume that $\mathcal{E}$ occurs and show that it leads to correctness in the thresholding bandit routine. Let $\mu > 0$ denote any threshold. On $\mathcal{E}$, we have that for any point q (in particular any child of node p)

$$\{k : d_{q,k} < \mu\} \supset \left\{ k : \hat{d}_{q,k}(T_{q,k}(t)) + C_{\delta/n^2}(T_{q,k}(t)) < \mu \right\}$$

for any set of values $\{k : T_{q,k}(t)\}$ at any time t. Similarly, we have that

$$\{k : d_{p,k} > \mu\} \supset \left\{ k : \hat{d}_{q,k}(T_{q,k}(t)) - C_{\delta/n^2}(T_{q,k}(t)) > \mu \right\}.$$

Therefore, if the algorithm stops sampling when either

$$\hat{d}_{q,k}(T_{q,k}(t)) + C_{\delta/n^2}(T_{q,k}(t)) < \mu$$

or

$$\hat{d}_{q,k}(T_{q,k}(t)) - C_{\delta/n^2}(T_{q,k}(t)) > \mu$$

for every point k and any value of μ, then on event $\mathcal{E}$, we have identified the sets $\{k : d_{p,k} < \mu\}$ and $\{k : d_{p,k} > \mu\}$ correctly. In particular, the above superset relation holds with equality.

Applying this to $\mu = 2^{i'}$ for different values of i' in the algorithm we see that at all times, the thresholding bandit implemented in lines $9-16$ succeeds. Since these bounds are shared between recursive calls of the algorithm, the routine succeeds in every recursive call.

We conclude by showing that if these sets have been computed correctly, which occurs when $\mathcal{E}$ occurs, then the algorithm correctly computes all quantities needed for the `Insert` algorithm in Beygelzimer et al. (2006).

First, note that when the thresholding bandit routine terminates

$$\{i \in Q : \hat{d}_{q,i}(T_i) + C_{\delta/n^2}(T_{q,i}) \leqslant 2^{i'}\} = \emptyset \implies d(q, Q) > 2^{i'}.$$

Second, note that if the set

$$\{i \in Q_{i'} : \hat{d}_{q,i}(T_{q,i}) + C_{\delta/n^2}(T_i) \leqslant 2^{i'}\}$$

is nonempty, then for any q' contained in it, we have that $d_{q,q'} \leqslant 2^{i'}$. Therefore, on $\mathcal{E}$, `Noisy-Remove` computes the same quantities as `Remove`. Applying Theorem 4 of Beygelzimer et al. (2006) completes the proof. □

7.7.2 Query Time Complexity

Now we turn our attention to the number of calls to the distance oracle needed by `Bandit Cover Tree`. We begin by proving a bound on the number of oracle calls

made by `Noisy-Find-Nearest`.

Proof of Theorem 7.7. Assume that `Noisy-Find-Nearest` succeeds, which occurs with probability $1-\delta$. Let i_{top} and i_{bottom} represent the top and bottom of $\mathcal{T}$. We begin by bounding the number of oracle calls drawn in an arbitrary round. Calls to the oracle only occur within the `Identify-Cover` routine. Suppose `Identify-Cover` is called on set Q and run with failure probability δ/n since $\delta/\alpha \geqslant \delta/n$ deterministically. Since `Identify-Cover` is equivalent to $(\mathtt{ST})^2$, Algorithm 5, with parameters $\gamma = 0$ and $\epsilon = 2^{i-1}$, Theorem 6.5 implies that the number of calls to the distance oracle made by `Identify-Cover` is bounded by

$$c_1 \log\left(\frac{n}{\delta}\right) \sum_{j \in Q} \max\left\{ \frac{1}{(d_{\min} + 2^{i-1} - d_{q,j})^2}, \frac{1}{(d_{q,j} + \kappa_i - d_{\min})^2} \right\} \tag{7.2}$$

where c_1 includes constants and doubly logarithmic terms, $d_{\min} = \min_{j \in Q} d_{q,j}$, and $\kappa_i = \min |d_{\min} + 2^{i-1} - d_{q,j}|$. κ_i combines the α_ϵ and β_ϵ terms in Theorem 6.5.

Define κ_i^{avg} to be the arithmetic means of the summands in Equation 7.2. Recall that for the cover set Q_i at level i, Q is defined as $Q = \bigcup_{p \in Q_i} \text{children}(p)$ in the `Noisy-Find-Nearest` algorithm. Hence, we may compactly write Equation 7.2 as

$$c_1 \log\left(\frac{n}{\delta}\right) \left| \bigcup_{p \in Q_i} \text{children}(p) \right| \kappa_i^{\text{avg}}.$$

Applying this, we may sum over all levels of the tree $\mathcal{T}$ and bound the total number of oracle calls as

$$c_1 \log\left(\frac{n}{\delta}\right) \sum_{i=i_{\text{top}}}^{i_{\text{bottom}}} \left| \bigcup_{p \in Q_i} \text{children}(p) \right| \kappa_i^{\text{avg}}$$

where we have written the outer sum index in order of *descending* i since $i_{\text{top}} > i_{\text{bottom}}$. This is done to reflect to the process of descending tree $\mathcal{T}$ and counting the number of oracle calls taken at each level. We proceed by bounding $\left|\bigcup_{p \in Q_i} \text{children}(p)\right|$.

By the nesting invariance, $p \in \text{children}(p)$ for any node p. Let Q^* be the final

explicit Q. That is $Q^* = Q_{i^*}$ where i^* is the lowest level such that an explicit node exists in the cover set Q_{i^*}. For all levels i such that $i > i^*$ (levels above i^*), explicit nodes have yet to be added. Hence

$$\left| \bigcup_{p \in Q_i} \text{children}(p) \right| \leqslant \left| \bigcup_{p \in Q_{i^*}} \text{children}(p) \right|.$$

For all levels below i^*, by definition, no new nodes are added as all remaining nodes are implicit. Therefore, for $i < i^*$ (lower levels of the tree),

$$\left| \bigcup_{p \in Q_i} \text{children}(p) \right| \leqslant \left| \bigcup_{p \in Q_{i^*}} \text{children}(p) \right|.$$

In particular, Q_{i^*} maximizes $\left|\bigcup_{p \in Q_i} \text{children}(p)\right|$. Following the proof of Theorem 5 in Beygelzimer et al. (2006), we have that $\left|\bigcup_{p \in Q_{i^*}} \text{children}(p)\right| \leqslant c^5$.

Next, for clarity, define

$$\overline{\kappa} := \frac{1}{i_{\text{top}} - i_{\text{bottom}}} \sum_{i=i_{\text{top}}}^{i_{\text{bottom}}} \kappa_i^{\text{avg}},$$

the average of the κ_i^{avg} terms that appears in each level. Plugging in both above pieces, we have that

$$c_1 \log\left(\frac{n}{\delta}\right) \sum_{i=i_{\text{top}}}^{i_{\text{bottom}}} \left| \bigcup_{p \in Q_i} \text{children}(p) \right| \kappa_i^{\text{avg}} \leqslant c_1 c^5 \log\left(\frac{n}{\delta}\right) (i_{\text{top}} - i_{\text{bottom}}) \overline{\kappa}$$

By Lemma 4.3 of Beygelzimer et al. (2006), we have that $i_{\text{top}} - i_{\text{bottom}} = O(c^2 \log(n))$. Plugging this in we have that the total number of oracle calls is bounded by

$$O\left(c^7 \log(n) \log\left(\frac{n}{\delta}\right) \overline{\kappa}\right)$$

completing the proof. □

Proof of Corollary 7.9. The proof follows identically, except that we may plug in

$$\alpha = \left\lceil \frac{\log(n)}{\log(1+1/c^2)} + 1 \right\rceil = O(c^2 \log(n)).$$

Hence the term from the union bound becomes $O\left(\log\left(\frac{c^2\log(n)}{\delta}\right)\right)$ instead of $\log\left(\frac{n}{\delta}\right)$. □

7.7.3 Insertion Time Complexity

Next, we bound the number of oracle calls made by `Noisy-Insert`.

Proof of Theorem 7.10. We begin by analyzing the complexity of the thresholding bandit subroutine in lines $2-9$ of `Noisy-Insert`. Assume the same event $\mathcal{E}$ from the proof of Lemma 7.5 that holds with probability $1-\delta$. We proceed by bounding the number of rounds any point j in the set Q may be $i^*(t)$ before it must enter the set K. Summing up the complexity for all points in Q bounds the complexity of this routine.

Suppose we wish to insert p into level i. Assume for point $j \in C_i$ that $d_{p,j} \leqslant 2^i$. Assume that

$$T_j \geqslant \frac{c'}{(d_{p,j}-2^i)^2} \log\left(\frac{n}{\delta}\log\left(\frac{n}{\delta(d_{p,j}-2^i)^2}\right)\right)$$

for a sufficiently large constant c'. Then

$$\hat{d}_{p,j}(T_j) + C_{\delta/n}(T_j) \overset{\mathcal{E}}{\leqslant} d_{p,j} + 2C_{\delta/n}(T_j) \overset{(a)}{<} d_{p,j} + 2^i - d_{p,j} = 2^i,$$

implying that j must be in the set K. Inequality (a) follows from Lemma 6.32 and bounds on c' are given by the same Lemma. This argument may be repeated for $j \in C_i$ such that $d_{p,j} > 2^i$ by instead considering the lower confidence bound on $\hat{d}_{p,j}(T_j)$.

Summing over all j in the set Q, no more than

$$\sum_{j \in Q} \frac{c'}{(d_{p,j} - 2^i)^2} \log\left(\frac{n}{\delta} \log\left(\frac{n}{\delta (d_{p,j} - 2^i)^2}\right)\right) \leqslant c_1 |Q| \log\left(\frac{n}{\delta}\right) \kappa_i^{\text{avg}}$$

calls to the oracle are made between lines 2 and 8 for a problem independent constant c_1. Similar to the proof of Theorem 7.7, we define κ_i^{avg} to be average of the summands including doubly logarithmic terms for brevity and clarity.

By definition, in level i $Q = \bigcup_{j \in Q_i} \text{children}(j)$. In the worst case, p is added at the leaf level, i_{bottom}. `Noisy-Insert` descends down the tree via recursive calls. This happens at most $i_{\text{top}} - t_{\text{bottom}}$ times. Summing over all levels, the total number of calls to the oracle is bounded by

$$c_1 \log\left(\frac{n}{\delta}\right) \sum_{i=i_{\text{top}}}^{i_{\text{bottom}}} \left| \bigcup_{j \in Q_i} \text{children}(j) \right| \kappa_i^{\text{avg}}$$

where the sum is indexed from the largest i to the smallest to reflect descending the tree. As in the proof of Theorem 7.7, there exists a level i^* which maximizes $\left|\bigcup_{j \in Q_i} \text{children}(j)\right|$ and we have that $\left|\bigcup_{j \in Q_{i^*}} \text{children}(j)\right| \leqslant c^5$. Define

$$\overline{\kappa}_p = \frac{1}{i_{\text{top}} - i_{\text{bottom}}} \sum_{i=i_{\text{top}}}^{i_{\text{bottom}}} \kappa_i^{\text{avg}}$$

Plugging this in,

$$\begin{aligned} c_1 \log\left(\frac{n}{\delta}\right) \sum_{i=i_{\text{top}}}^{i_{\text{bottom}}} \left| \bigcup_{j \in Q_i} \text{children}(j) \right| \kappa_i^{\text{avg}} &\leqslant c_1 c^5 \log\left(\frac{n}{\delta}\right) (i_{\text{top}} - t_{\text{bottom}}) \overline{\kappa_p} \\ &\leqslant O\left(c^7 \log(n) \log\left(\frac{n}{\delta}\right) \overline{\kappa_p}\right). \end{aligned}$$

where the inequality follows from Lemma 4.3 in Beygelzimer et al. (2006).

□

Proof of Theorem 7.12. Begin with an empty tree. `Noisy-Insert` can trivially place x_1 as a root and replace the root as necessary. We run `Noisy-Insert` with failure probability δ/n. Since placing the first node at the root is trivial, make the inductive hypothesis that we have a correct tree $\mathcal{T}$ built on a strict subset $S \subsetneq \mathcal{X}$ and wish to insert a point $p \in \mathcal{X} \backslash S$ to $\mathcal{T}$ using `Noisy-Insert`. By Lemma 7.5, this process succeeds with probability $1 - \delta/n$. By Theorem 7.10, this requires no more than

$$O\left(c^7 \log(n) \log\left(\frac{n}{\delta}\right) \overline{\kappa_p}\right)$$

calls to the noisy distance oracle.

A union bound over inserting the n points in $\mathcal{X}$ implies correctness. Summing the above expression for every point $p \in \mathcal{X}$ bounds the total sample complexity necessary for construction, stated in the Theorem. In particular, $\widetilde{\kappa}$ is the arithmetic mean of the individual $\overline{\kappa_p}$s. □

7.7.4 Removal Time Complexity

Finally, we bound the number of oracle calls needed by `Noisy-Remove`

Proof of Theorem 7.14. We begin by analyzing the complexity of the Thresholding bandit subroutine in lines $9 - 16$ of `Noisy-Remove`. Assume the same event $\mathcal{E}$ from the proof of Lemma 7.6 that holds with probability $1 - \delta$. Fix an arbitrary $q \in \text{children}(p)$. We proceed by bounding the number of rounds any point j in the set $Q_{i'}$ may be $i^*(t)$ before it must enter the set K. Summing up the complexity for all points in Q bounds the complexity of this routine.

Assume for point $j \in C_{i'}$ that $d_{q,j} \leqslant 2^{i'}$. Assume that

$$T_{q,j} \geqslant \frac{c'}{(d_{q,j} - 2^{i'})^2} \log\left(\frac{n^2}{\delta} \log\left(\frac{n^2}{\delta (d_{q,j} - 2^{i'})^2}\right)\right)$$

for a sufficiently large constant c'. Then

$$\hat{d}_{q,j}(T_{q,j}) + C_{\delta/n^2}(T_{q,j}) \overset{\mathcal{E}}{\leqslant} d_{q,j} + 2C_{\delta/n^2}(T_{q,j}) \overset{(a)}{<} d_{q,j} + 2^{i'} - d_{p,j} = 2^{i'},$$

implying that j must be in the set K. Inequality (a) follows from Lemma 6.32 and bounds on c' are given by the same Lemma. This argument may be repeated for $j \in C_i$ such that $d_{q,j} > 2^{i'}$ by instead considering the lower confidence bound on $\hat{d}_{q,j}(T_{q,j})$.

Summing over all j in the set $Q_{i'}$, no more than

$$\sum_{j \in Q_{i'}} \frac{c'}{(d_{q,j} - 2^{i'})^2} \log\left(\frac{n^2}{\delta} \log\left(\frac{n^2}{\delta(d_{q,j} - 2^{i'})^2}\right)\right) \leqslant c_1 |Q_{i'}| \log\left(\frac{n^2}{\delta}\right) \kappa^{\text{avg}}_{q,i'}$$

calls to the oracle are made between lines 9 and 16 for a problem independent constant c_1. Similar to the proof of Theorem 7.7, we define $\kappa^{\text{avg}}_{q,i'}$ to be average of the summands including doubly logarithmic terms for brevity and clarity.

If

$$\{j \in Q_{i'} : \hat{d}_{q,j}(T_{q,j}) + C_{\delta/n^2}(T_{q,j}) \leqslant 2^{i'}\} = \emptyset,$$

i' is reset to $i' + 1$ and the algorithm proceeds to the next level up the tree. An identical computation is performed as in lines $9 - 16$ and a similar bound applies as the above. The only difference is that since the Thresholding bandit is now comparing to a threshold of $2^{i'+1}$ (or alternatively $2^{i'}$ for the incremented value of i') we incur a dependence on $\kappa^{\text{avg}}_{q,i'+1}$ instead. In particular, the number of oracle queries drawn between lines 20 and 25 is at most

$$\sum_{j \in Q_{i'+1}} \frac{c'}{(d_{q,j} - 2^{i'+1})^2} \log\left(\frac{n^2}{\delta} \log\left(\frac{n^2}{\delta(d_{q,j} - 2^{i'+1})^2}\right)\right) \leqslant c_1 |Q_{i'+1}| \log\left(\frac{n^2}{\delta}\right) \kappa^{\text{avg}}_{q,i'+1}.$$

This process repeats until the conditional in the while loop is no longer satisfied. Naively, this happens at most $i_{\text{top}} - i + 1$ times as i' is initialized as $i - 1$ and is incremented until potentially it reaches the top level of $\mathcal{T}$, i_{top}. Summing this quantity over all levels of the tree, we may bound the total number of oracle calls as

$$\sum_{i'=i_{\text{top}}}^{i-1} c_1 |Q_{i'}| \log\left(\frac{n^2}{\delta}\right) \kappa^{\text{avg}}_{q,i'}.$$

As in the proof of Theorem 7.7, there exists a level i^* which maximizes $\left|\bigcup_{j\in Q_i} \text{children}(j)\right|$ and we have that $\left|\bigcup_{j\in Q_{i^*}} \text{children}(j)\right| \leqslant c^5$. As $Q_i \subset \bigcup_{j\in Q_i} \text{children}(j)$ by the nesting invariance, c^5 likewise bounds Q_i for all i. Define

$$\kappa_q^{\text{avg}}(i) = \frac{1}{i_{\text{top}} - i + 1} \sum_{i=i_{\text{top}}}^{i-1} \kappa_{q,i}^{\text{avg}}$$

Plugging this in, we may bound the total number of oracle calls (for the distance of any point to q) by

$$\begin{aligned}\sum_{i'=i_{\text{top}}}^{i-1} c_1 |Q_{i'}| \log\left(\frac{n^2}{\delta}\right) \kappa_{q,i'}^{\text{avg}} &\leqslant c_1 c^5 (i_{\text{top}} - i + 1) \log\left(\frac{n^2}{\delta}\right) \kappa_q^{\text{avg}}(i) \\ &\leqslant c_1 c^5 (i_{\text{top}} - i_{\text{bottom}}) \log\left(\frac{n^2}{\delta}\right) \kappa_q^{\text{avg}}(i).\end{aligned}$$

Next, we use Lemma 4.3 of Beygelzimer et al. (2006) to bound $i_{\text{top}} - i_{\text{bottom}} = O(c^2 \log(n))$. Therefore,

$$c_1 c^5 (i_{\text{top}} - i_{\text{bottom}}) \log\left(\frac{n^2}{\delta}\right) \kappa_q^{\text{avg}}(i) \leqslant O\left(c^7 \log(n) \log\left(\frac{n^2}{\delta}\right) \kappa_q^{\text{avg}}(i)\right)$$

The above bounds the number of calls to the distance oracle needed for any child q of p, the point to be removed. Due to the 'For' loop in line 6, this process is repeated for all $q \in \text{children}(p)$. By Lemma 4.1 of Beygelzimer et al. (2006), the number of children of any node $p \in \mathcal{T}$ is at most c^4. Define

$$\kappa^p(i) := \frac{1}{|\text{children}(p)|} \sum_{q\in\text{children}(p)} \kappa_q^{\text{avg}}(i)$$

where the superscript p and the parenthetical i denote that this quantity depends

on all children of p and level i of the tree. Then

$$\begin{aligned}\sum_{q\in\text{children}(p)} \kappa_q^{\text{avg}}(i) &= |\text{children}(p)|\frac{1}{|\text{children}(p)|}\sum_{q\in\text{children}(p)} \kappa_q^{\text{avg}}(i)\\ &\leqslant c^4\frac{1}{|\text{children}(p)|}\sum_{q\in\text{children}(p)} \kappa_q^{\text{avg}}(i)\\ &= c^4\kappa^p(i)\end{aligned}$$

Therefore, summing over all $q \in \text{children}(p)$, we can bound the total number of calls to the distance oracle drawn in lines 6 to 29 of `Noisy-Remove` by

$$O\left(c^9\log(n)\log\left(\frac{n^2}{\delta}\right)\kappa^p(i)\right).$$

As `Noisy-Remove` is recursive, it remains to sum the complexity of all recursive calls. The above bound depends on the level i on which `Noisy-Remove` is called only through the term $\kappa^p(i)$. In the worst case, p is present in every level and the 'If' condition in line 4 is true for every recursive call. Hence, we sum the above expression over every level i. Define

$$\widehat{\kappa}_p := \frac{1}{i_{\text{top}} - i_{\text{bottom}}}\sum_{i=i_{\text{top}}}^{i_{\text{bottom}}} \kappa^p(i).$$

The total number of oracle calls is bounded as

$$\begin{aligned}\sum_{i=i_{\text{top}}}^{\text{bottom}} O\left(c^9\log(n)\log\left(\frac{n^2}{\delta}\right)\kappa^p(i)\right) &= O\left(c^9(i_{\text{top}} - i_{\text{bottom}})\log(n)\log\left(\frac{n^2}{\delta}\right)\widehat{\kappa}_p\right)\\ &= O\left(c^{11}\log^2(n)\log\left(\frac{n^2}{\delta}\right)\widehat{\kappa}_p\right)\end{aligned}$$

completing the proof. □

REFERENCES

Abramovich, Felix, and Vadim Grinshtein. 2016. Model selection and minimax estimation in generalized linear models. *IEEE Transactions on Information Theory* 62(6):3721–3730.

Acevedo Nistal, Ana, Wim Van Dooren, and Lieven Verschaffel. 2013. Students' reported justifications for their representational choices in linear function problems: An interview study. *Educational Studies* 39(1):104–117.

———. 2014. Improving students' representational flexibility in linear-function problems: An intervention. *Educational Psychology* 34(6):763–786.

Agarwal, Sameer, Josh Wills, Lawrence Cayton, Gert Lanckriet, David J Kriegman, and Serge Belongie. 2007. Generalized non-metric multidimensional scaling. In *International conference on artificial intelligence and statistics*, 11–18.

Ainsworth, Shaaron. 2006. Deft: A conceptual framework for considering learning with multiple representations. *Learning and instruction* 16(3):183–198.

———. 2008. The educational value of multiple-representations when learning complex scientific concepts. In *Visualization: Theory and practice in science education*, 191–208. Springer.

———. 2014. 20–the multiple representation principle in multimedia learning. *The Cambridge handbook of multimedia learning* 464.

Ainsworth, Shaaron, Peter Bibby, and David Wood. 2002. Examining the effects of different multiple representational systems in learning primary mathematics. *The Journal of the Learning Sciences* 11(1):25–61.

Airey, John, and Cedric Linder. 2009. A disciplinary discourse perspective on university science learning: Achieving fluency in a critical constellation of modes. *Journal of Research in Science Teaching: The Official Journal of the National Association for Research in Science Teaching* 46(1):27–49.

Alemdag, Ecenaz, and Kursat Cagiltay. 2018. A systematic review of eye tracking research on multimedia learning. *Computers & Education* 125:413–428.

Alibali, Martha W, Mitchell J Nathan, Matthew S Wolfgram, R Breckinridge Church, Steven A Jacobs, Chelsea Johnson Martinez, and Eric J Knuth. 2014. How teachers link ideas in mathematics instruction using speech and gesture: A corpus analysis. *Cognition and instruction* 32(1):65–100.

Anderson, John R, C Franklin Boyle, Albert T Corbett, and Matthew W Lewis. 1990. Cognitive modeling and intelligent tutoring. *Artificial intelligence* 42(1):7–49.

Andoni, Alexandr, Piotr Indyk, and Ilya Razenshteyn. 2018. Approximate nearest neighbor search in high dimensions. *arXiv preprint arXiv:1806.09823* 7.

Antos, András, Varun Grover, and Csaba Szepesvári. 2010. Active learning in heteroscedastic noise. *Theoretical Computer Science* 411(29-30):2712–2728.

Arias-Castro, Ery, et al. 2017. Some theory for ordinal embedding. *Bernoulli* 23(3): 1663–1693.

Atzmon, Yuval, Uri Shalit, and Gal Chechik. 2015. Learning sparse metrics, one feature at a time. In *Feature extraction: Modern questions and challenges*, 30–48.

Auer, Peter, Nicolo Cesa-Bianchi, and Paul Fischer. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine learning* 47(2-3):235–256.

Baddeley, Alan. 1992. Working memory. *Science* 255(5044):556–559.

———. 2012. Working memory: Theories, models, and controversies. *Annual review of psychology* 63:1–29.

Bagaria, Vivek, Govinda M Kamath, Vasilis Ntranos, Martin J Zhang, and David Tse. 2017. Medoids in almost linear time via multi-armed bandits. *arXiv preprint arXiv:1711.00817*.

Bagaria, Vivek, Govinda M Kamath, and David N Tse. 2018. Adaptive monte-carlo optimization. *arXiv preprint arXiv:1805.08321*.

Bair, Eric, Trevor Hastie, Debashis Paul, and Robert Tibshirani. 2006. Prediction by supervised principal components. *Journal of the American Statistical Association* 101(473):119–137.

Balakrishnama, Suresh, and Aravind Ganapathiraju. 1998. Linear discriminant analysis-a brief tutorial. In *Institute for signal and information processing*, vol. 18, 1–8.

Bartlett, Peter. 2013. Lecture notes in theoretical statistics.

Bellet, Aurélien, and Amaury Habrard. 2015. Robustness and generalization for metric learning. *Neurocomputing* 151:259–267.

Bellet, Aurélien, Amaury Habrard, and Marc Sebban. 2015. Metric learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning* 9(1):1–151.

Bentley, Jon Louis. 1975. Multidimensional binary search trees used for associative searching. *Communications of the ACM* 18(9):509–517.

Beygelzimer, Alina, Sham Kakade, and John Langford. 2006. Cover trees for nearest neighbor. In *Proceedings of the 23rd international conference on machine learning*, 97–104. ACM.

Bhatia, Nitin, et al. 2010. Survey of nearest neighbor techniques. *arXiv preprint arXiv:1007.0085*.

Bian, Wei, and Dacheng Tao. 2012. Constrained empirical risk minimization framework for distance metric learning. *IEEE transactions on neural networks and learning systems* 23(8):1194–1205.

Bieda, Kristen N, and Mitchell J Nathan. 2009. Representational disfluency in algebra: Evidence from student gestures and speech. *ZDM* 41(5):637–650.

Bijmolt, Tammo HA, and Michel Wedel. 1995. The effects of alternative methods of collecting similarity data for multidimensional scaling. *International Journal of Research in Marketing* 12(4):363–371.

Bocci, Matteo, Jonas Sjölund, Ewa Kurzejamska, David Lindgren, Michael Bartoschek, Mattias Höglund, Kristian Pietras, et al. 2019. Activin receptor-like kinase 1 is associated with immune cell infiltration and regulates clec14a transcription in cancer. *Angiogenesis* 22(1):117–131.

Bodemer, Daniel, Rolf Ploetzner, Inge Feuerlein, and Hans Spada. 2004. The active integration of information during learning with dynamic and interactive visualisations. *Learning and instruction* 14(3):325–341.

Booth, Brandon M, Tiantian Feng, Abhishek Jangalwa, and Shrikanth S Narayanan. 2019. Toward robust interpretable human movement pattern analysis in a workplace setting. In *Icassp 2019-2019 ieee international conference on acoustics, speech and signal processing (icassp)*, 7630–7634. IEEE.

Brickell, Justin, Inderjit S Dhillon, Suvrit Sra, and Joel A Tropp. 2008. The metric nearness problem. *SIAM Journal on Matrix Analysis and Applications* 30(1):375–396.

Bubeck, Sébastian, Tengyao Wang, and Nitin Viswanathan. 2013. Multiple identifications in multi-armed bandits. In *International conference on machine learning*, 258–265.

Bunea, Florentina, Alexandre B Tsybakov, Marten H Wegkamp, et al. 2007. Aggregation for gaussian regression. *The Annals of Statistics* 35(4):1674–1697.

Candes, Emmanuel J, and Yaniv Plan. 2011. Tight oracle inequalities for low-rank matrix recovery from a minimal number of noisy random measurements. *IEEE Transactions on Information Theory* 57(4):2342–2359.

Chandler, Paul, and John Sweller. 1991. Cognitive load theory and the format of instruction. *Cognition and instruction* 8(4):293–332.

Chase, William G, and Herbert A Simon. 1973. Perception in chess. *Cognitive psychology* 4(1):55–81.

Chatpatanasiri, Ratthachat, Teesid Korsrilabutr, Pasakorn Tangchanachaianan, and Boonserm Kijsirikul. 2010. A new kernelization framework for mahalanobis distance learning algorithms. *Neurocomputing* 73(10-12):1570–1579.

Chaudhuri, Arghya Roy, and Shivaram Kalyanakrishnan. 2017. Pac identification of a bandit arm relative to a reward quantile. In *Thirty-first aaai conference on artificial intelligence*.

———. 2019. Pac identification of many good arms in stochastic multi-armed bandits. In *International conference on machine learning*, 991–1000.

Chechik, Gal, Varun Sharma, Uri Shalit, and Samy Bengio. 2010. Large scale online learning of image similarity through ranking. *Journal of Machine Learning Research* 11(Mar):1109–1135.

Chen, Lijie, Jian Li, and Mingda Qiao. 2017. Nearly instance optimal sample complexity bounds for top-k arm selection. In *Artificial intelligence and statistics*, 101–110.

Chi, Michelene TH, Miriam Bassok, Matthew W Lewis, Peter Reimann, and Robert Glaser. 1989. Self-explanations: How students study and use examples in learning to solve problems. *Cognitive science* 13(2):145–182.

Chi, Michelene TH, Nicholas De Leeuw, Mei-Hung Chiu, and Christian LaVancher. 1994. Eliciting self-explanations improves understanding. *Cognitive science* 18(3): 439–477.

Chiu, Jennifer L, and Marcia C Linn. 2012. The role of self-monitoring in learning chemistry with dynamic visualizations. In *Metacognition in science education*, 133–163. Springer.

Christmann-Franck, Serge, Gerard JP van Westen, George Papadatos, Fanny Beltran Escudie, Alexander Roberts, John P Overington, and Daniel Domine. 2016. Unprecedently large-scale kinase inhibitor set enabling the accurate prediction

of compound–kinase activities: A way toward selective promiscuity by design? *Journal of chemical information and modeling* 56(9):1654–1675.

Clark, Richard. 2014. Cognitive task analysis for expert-based instruction in healthcare. In *Handbook of research on educational communications and technology*, 541–551. Springer.

Clark, Richard E and Feldon, David E, and Van Merrienboer, Jeroen JG and Yates, Kenneth and Early, Sean. 2007. Cognitive task analysis. *International Journal of Educational Research* 25(5):403–417.

Clarkson, Kenneth L. 1983. Fast algorithms for the all nearest neighbors problem. In *24th annual symposium on foundations of computer science (sfcs 1983)*, 226–232. IEEE.

Conati, Cristina, Christina Merten, Kasia Muldner, and David Ternes. 2005. Exploring eye tracking to increase bandwidth in user modeling. In *International conference on user modeling*, 357–366. Springer.

Coombs, Clyde H. 1964. A theory of data.

Cope, Alexandra C, Jeff Bezemer, Roger Kneebone, and Lorelei Lingard. 2015. 'you see?' teaching and learning how to interpret visual cues during surgery. *Medical education* 49(11):1103–1116.

Cormen, Thomas H, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. 2009. *Introduction to algorithms*. MIT press.

Cox, Michael AA, and Trevor F Cox. 2008. Multidimensional scaling. In *Handbook of data visualization*, 315–347. Springer.

Cox, Richard, and Paul Brna. 2016. Twenty years on: Reflections on "supporting the use of external representations in problem solving". *International Journal of Artificial Intelligence in Education* 26(1):193–204.

Dasgupta, Sanjoy, and Kaushik Sinha. 2013. Randomized partition trees for exact nearest neighbor search. In *Conference on learning theory*, 317–337.

Dattorro, Jon. 2011. *Convex optimization & euclidean distance geometry*. Meboo Publishing USA.

Davenport, Mark A, Yaniv Plan, Ewout Van Den Berg, and Mary Wootters. 2014. 1-bit matrix completion. *Information and Inference: A Journal of the IMA* 3(3):189–223.

Davidowitz, Bette, and Gail Chittleborough. 2009. Linking the macroscopic and sub-microscopic levels: Diagrams. In *Multiple representations in chemical education*, 169–191. Springer.

Davidson, Kenneth R, and Stanislaw J Szarek. 2001. Local operator theory, random matrices and banach spaces. *Handbook of the geometry of Banach spaces* 1(317-366): 131.

De Koning, Björn B, Huib K Tabbers, Remy MJP Rikers, and Fred Paas. 2010. Attention guidance in learning from a complex animation: Seeing is understanding? *Learning and instruction* 20(2):111–122.

Degenne, Rémy, and Wouter M Koolen. 2019. Pure exploration with multiple correct answers. In *Advances in neural information processing systems*, 14564–14573.

DeLoache, Judy S. 2000. Dual representation and young children's use of scale models. *Child development* 71(2):329–338.

Dhingra, Bhuwan, Christopher J Shallue, Mohammad Norouzi, Andrew M Dai, and George E Dahl. 2018. Embedding text in hyperbolic spaces. *arXiv preprint arXiv:1806.04313*.

Disessa, Andrea A. 2004. Metarepresentation: Native competence and targets for instruction. *Cognition and instruction* 22(3):293–331.

Doroudi, Shayan, Ece Kamar, Emma Brunskill, and Eric Horvitz. 2016. Toward a learning science for complex crowdsourcing tasks. In *Proceedings of the 2016 chi conference on human factors in computing systems*, 2623–2634.

Dranchak, Patricia, Ryan MacArthur, Rajarshi Guha, William J Zuercher, David H Drewry, Douglas S Auld, and James Inglese. 2013. Profile of the gsk published protein kinase inhibitor set across atp-dependent and-independent luciferases: implications for reporter-gene assays. *PloS one* 8(3).

Dreher, Anika, and Sebastian Kuntze. 2015. Teachers facing the dilemma of multiple representations being aid and obstacle for learning: Evaluations of tasks and theme-specific noticing. *Journal fur Mathematik-Didaktik* 36(1):23–44.

Drewry, David H, Carrow I Wells, David M Andrews, Richard Angell, Hassan Al-Ali, Alison D Axtman, Stephen J Capuzzi, Jonathan M Elkins, Peter Ettmayer, Mathias Frederiksen, et al. 2017. Progress towards a public chemogenomic set for protein kinases and a call for contributions. *PloS one* 12(8).

Dreyfus, Stuart E. 2004. The five-stage model of adult skill acquisition. *Bulletin of science, technology & society* 24(3):177–181.

Eaton, Morris L, et al. 1981. On the projections of isotropic distributions. *The Annals of Statistics* 9(2):391–400.

Eilam, Billie. 2012. *Teaching, learning, and visual literacy: The dual role of visual representation*. Cambridge University Press.

Ericsson, K Anders, and Herbert A Simon. 1984. *Protocol analysis: Verbal reports as data.* the MIT Press.

Eriksson, Brian, Paul Barford, Joel Sommers, and Robert Nowak. 2010. A learning-based approach for ip geolocation. In *International conference on passive and active network measurement*, 171–180. Springer.

Even-Dar, Eyal, Shie Mannor, and Yishay Mansour. 2002. Pac bounds for multi-armed bandit and markov decision processes. In *International conference on computational learning theory*, 255–270. Springer.

———. 2006. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *Journal of machine learning research* 7(Jun):1079–1105.

Fahle, Manfred, Tomaso Poggio, et al. 2002. *Perceptual learning*. MIT Press.

Fiore, Stephen M. 1997. Verbal overshadowing of perceptual memories. *Psychology of Learning and Motivation: Advances in Research and Theory* 37:291.

Floyd, Robert W. 1962. Algorithm 97: shortest path. *Communications of the ACM* 5(6):345.

Frensch, Peter A, and Dennis Rünger. 2003. Implicit learning. *Current directions in psychological science* 12(1):13–18.

Gabillon, Victor, Mohammad Ghavamzadeh, and Alessandro Lazaric. 2012. Best arm identification: A unified approach to fixed budget and fixed confidence. In *Advances in neural information processing systems*, 3212–3220.

Garivier, Aurélien. 2013. Informational confidence bounds for self-normalized averages and applications. In *2013 ieee information theory workshop (itw)*, 1–5. IEEE.

Gegenfurtner, Andreas, Erno Lehtinen, and Roger Säljö. 2011. Expertise differences in the comprehension of visualizations: A meta-analysis of eye-tracking research in professional domains. *Educational psychology review* 23(4):523–552.

Gentner, Dedre, Jeffrey Loewenstein, and Leigh Thompson. 2003. Learning and transfer: A general role for analogical encoding. *Journal of Educational Psychology* 95(2):393.

Gentner, Dedre, and Arthur B Markman. 1997. Structure mapping in analogy and similarity. *American psychologist* 52(1):45.

Gibson, Eleanor J. 2000. Perceptual learning in development: Some basic concepts. *Ecological Psychology* 12(4):295–302.

Gibson, Eleanor Jack. 1969. Principles of perceptual learning and development.

Gilbert, Anna C, and Lalit Jain. 2017. If it ain't broke, don't fix it: Sparse metric repair. In *2017 55th annual allerton conference on communication, control, and computing (allerton)*, 612–619. IEEE.

Gilbert, John K. 2005. Visualization: A metacognitive skill in science and science education. In *Visualization in science education*, 9–27. Springer.

Globerson, Amir, and Naftali Tishby. 2003. Sufficient dimensionality reduction. *Journal of Machine Learning Research* 3(Mar):1307–1331.

Goldstone, Robert L, and Lawrence W Barsalou. 1998. Reuniting perception and conception. *Cognition* 65(2-3):231–262.

Goldstone, Robert L, David H Landy, and Ji Y Son. 2010. The education of perception. *Topics in Cognitive Science* 2(2):265–284.

Goldstone, Robert L, Douglas L Medin, and Philippe G Schyns. 1997a. *Perceptual learning.* Academic Press.

Goldstone, Robert L, Philippe G Schyns, and Douglas L Medin. 1997b. Learning to bridge between perception and cognition. *The psychology of learning and motivation* 36:1–14.

Goyal, Navin, Yury Lifshits, and Hinrich Schütze. 2008. Disorder inequality: a combinatorial approach to nearest neighbor search. In *Proceedings of the 2008 international conference on web search and data mining*, 25–32. ACM.

Grawemeyer, Beate. 2006. Evaluation of erst–an external representation selection tutor. In *International conference on theory and application of diagrams*, 154–167. Springer.

Guo, Zheng-Chu, and Yiming Ying. 2014. Guaranteed classification via regularized similarity learning. *Neural Computation* 26(3):497–522.

Gutwill, Joshua P, John R Frederiksen, and Barbara Y White. 1999. Making their own connections: Students' understanding of multiple models in basic electricity. *Cognition and Instruction* 17(3):249–282.

Haghiri, Siavash, Debarghya Ghoshdastidar, and Ulrike von Luxburg. 2017. Comparison based nearest neighbor search. *arXiv preprint arXiv:1704.01460*.

Harel, Assaf. 2016. What is special about expertise? visual expertise reveals the interactive nature of real-world object recognition. *Neuropsychologia* 83:88–99.

Hegarty, Mary, and Marcel-Adam Just. 1993. Constructing mental models of machines from text and diagrams. *Journal of memory and language* 32(6):717–742.

Heim, Eric, Matthew Berger, Lee Seversky, and Milos Hauskrecht. 2015. Active perceptual similarity modeling with auxiliary information. *arXiv preprint arXiv:1511.02254*.

Hill, Matthew, and Manjula Devi Sharma. 2015. Students' representational fluency at university: A cross-sectional measure of how multiple representations are used by physics students using the representational fluency survey. *Eurasia Journal of Mathematics, Science and Technology Education* 11(6):1633–1655.

Houle, Michael E, and Michael Nett. 2015. Rank-based similarity search: Reducing the dimensional dependence. *IEEE transactions on pattern analysis and machine intelligence* 37(1):136–150.

Howard, Steven R, Aaditya Ramdas, Jon McAuliffe, and Jasjeet Sekhon. 2018. Uniform, nonparametric, non-asymptotic confidence sequences. *arXiv preprint arXiv:1810.08240*.

Irwin, David E. 2004. Fixation location and fixation duration as indices of cognitive processing. *The interface of language, vision, and action: Eye movements and the visual world* 217:105–133.

Jain, Lalit, Kevin G Jamieson, and Rob Nowak. 2016a. Finite sample prediction and recovery bounds for ordinal embedding. In *Advances in neural information processing systems*, 2703–2711.

———. 2016b. Finite sample prediction and recovery bounds for ordinal embedding. In *Advances in neural information processing systems*, 2711–2719.

Jamieson, K., and R. Nowak. 2014. Best-arm identification algorithms for multi-armed bandits in the fixed confidence setting. In *2014 48th annual conference on information sciences and systems (ciss)*, 1–6.

Jamieson, Kevin, Daniel Haas, and Ben Recht. 2016. On the detection of mixture distributions with applications to the most biased coin problem. *arXiv preprint arXiv:1603.08037*.

Jamieson, Kevin, and Lalit Jain. 2018. A bandit approach to multiple testing with false discovery control. In *Proceedings of the 32nd international conference on neural information processing systems*, 3664–3674. NIPS'18, Red Hook, NY, USA: Curran Associates Inc.

Jamieson, Kevin, Matthew Malloy, Robert Nowak, and Sebastien Bubeck. 2013. On finding the largest mean among many. *arXiv preprint arXiv:1306.3917*.

Jamieson, Kevin, Matthew Malloy, Robert Nowak, and Sébastien Bubeck. 2014. lil'ucb: An optimal exploration algorithm for multi-armed bandits. In *Conference on learning theory*, 423–439.

Jamieson, Kevin G, Lalit Jain, Chris Fernandez, Nicholas J Glattard, and Rob Nowak. 2015. Next: A system for real-world development, evaluation, and application of active learning. In *Advances in neural information processing systems*, 2656–2664.

Jamieson, Kevin G, and Robert D Nowak. 2011. Low-dimensional embedding using adaptively selected ordinal data. In *Communication, control, and computing (allerton), 2011 49th annual allerton conference on*, 1077–1084. IEEE.

Jarodzka, Halszka, Thomas Balslev, Kenneth Holmqvist, Marcus Nyström, Katharina Scheiter, Peter Gerjets, and Berit Eika. 2012. Conveying clinical reasoning based on visual observation via eye-movement modelling examples. *Instructional Science* 40(5):813–827.

Johnson, Cheryl I, and Richard E Mayer. 2012. An eye movement analysis of the spatial contiguity effect in multimedia learning. *Journal of Experimental Psychology: Applied* 18(2):178.

Johnson, Richard M. 1973. Pairwise nonmetric multidimensional scaling. *Psychometrika* 38(1):11–18.

Kalyanakrishnan, Shivaram, and Peter Stone. 2010. Efficient selection of multiple bandit arms: Theory and practice. In *Icml*, vol. 10, 511–518.

Kalyanakrishnan, Shivaram, Ambuj Tewari, Peter Auer, and Peter Stone. 2012. Pac subset selection in stochastic multi-armed bandits. In *Icml*, vol. 12, 655–662.

Kano, Hideaki, Junya Honda, Kentaro Sakamaki, Kentaro Matsuura, Atsuyoshi Nakamura, and Masashi Sugiyama. 2019. Good arm identification via bandit feedback. *Machine Learning* 108(5):721–745.

Karnin, Zohar, Tomer Koren, and Oren Somekh. 2013. Almost optimal exploration in multi-armed bandits. In *International conference on machine learning*, 1238–1246.

Karnin, Zohar S. 2016. Verification based solution for structured mab problems. In *Advances in neural information processing systems*, 145–153.

Katz-Samuels, Julian, and Kevin Jamieson. 2020. The true sample complexity of identifying good arms. In *International conference on artificial intelligence and statistics*, 1781–1791. PMLR.

Kaufmann, Emilie, Olivier Cappé, and Aurélien Garivier. 2016. On the complexity of best-arm identification in multi-armed bandit models. *The Journal of Machine Learning Research* 17(1):1–42.

Kaufmann, Emilie, and Shivaram Kalyanakrishnan. 2013. Information complexity in bandit subset selection. In *Conference on learning theory*, 228–251.

Kazai, Gabriella, Jaap Kamps, and Natasa Milic-Frayling. 2013. An analysis of human factors and label accuracy in crowdsourcing relevance judgments. *Information retrieval* 16(2):138–178.

Kellman, Philip J, and Patrick Garrigan. 2009. Perceptual learning and human expertise. *Physics of life reviews* 6(2):53–84.

Kellman, Philip J, Christine Massey, Zipora Roth, Timothy Burke, Joel Zucker, Amanda Saw, Katherine E Aguero, and Joseph A Wise. 2008. Perceptual learning and the technology of expertise: Studies in fraction learning and algebra. *Pragmatics & Cognition* 16(2):356–405.

Kellman, Philip J, and Christine M Massey. 2013. Perceptual learning, cognition, and expertise. In *Psychology of learning and motivation*, vol. 58, 117–165. Elsevier.

Kellman, Philip J, Christine M Massey, and Ji Y Son. 2010. Perceptual learning modules in mathematics: Enhancing students' pattern recognition, structure extraction, and fluency. *Topics in Cognitive Science* 2(2):285–305.

Koedinger, Kenneth R, Albert Corbett, et al. 2006. *Cognitive tutors: Technology bringing learning sciences to the classroom*. na.

Koedinger, Kenneth R, Albert T Corbett, and Charles Perfetti. 2012. The knowledge-learning-instruction framework: Bridging the science-practice chasm to enhance robust student learning. *Cognitive science* 36(5):757–798.

Kozma, Robert, and Joel Russell. 2005. Students becoming chemists: Developing representationl competence. In *Visualization in science education*, 121–145. Springer.

Krauthgamer, Robert, and James R Lee. 2004. Navigating nets: simple algorithms for proximity search. In *Proceedings of the fifteenth annual acm-siam symposium on discrete algorithms*, 798–807. Society for Industrial and Applied Mathematics.

Kruskal, Joseph B. 1964a. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika* 29(1):1–27.

———. 1964b. Nonmetric multidimensional scaling: a numerical method. *Psychometrika* 29(2):115–129.

Lai, Meng-Lung, Meng-Jung Tsai, Fang-Ying Yang, Chung-Yuan Hsu, Tzu-Chien Liu, Silvia Wen-Yu Lee, Min-Hsien Lee, Guo-Li Chiou, Jyh-Chong Liang, and Chin-Chung Tsai. 2013. A review of using eye-tracking technology in exploring learning from 2000 to 2012. *Educational research review* 10:90–115.

LeJeune, Daniel, Richard G Baraniuk, and Reinhard Heckel. 2019. Adaptive estimation for approximate k-nearest-neighbor computations. *arXiv preprint arXiv:1902.09465*.

Li, Ker-Chau. 1991. Sliced inverse regression for dimension reduction. *Journal of the American Statistical Association* 86(414):316–327.

Li, Lisha, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. 2017. Hyperband: A novel bandit-based approach to hyperparameter optimization. *The Journal of Machine Learning Research* 18(1):6765–6816.

Linn, Marcia C, Bat-Sheva Eylon, Anna Rafferty, and Jonathan M Vitale. 2015. Designing instruction to improve lifelong inquiry learning. *Eurasia Journal of Mathematics, Science & Technology Education* 11(2).

Linn, Marcia C, and James D Slotta. 2000. Wise science. *Educational Leadership* 58(2):29–32.

Locatelli, Andrea, Maurilio Gutzeit, and Alexandra Carpentier. 2016. An optimal algorithm for the thresholding bandit problem. In *Proceedings of the 33rd international conference on international conference on machine learning-volume 48*, 1690–1698. JMLR. org.

Luxford, Cynthia Joan. 2013. Use of multiple representations to explore students' understandings of covalent and ionic bonding as measured by the bonding representations inventory. Ph.D. thesis, Miami University.

Malloy, Matthew L, and Robert D Nowak. 2014. Sequential testing for sparse recovery. *IEEE Transactions on Information Theory* 60(12):7862–7873.

Mannor, Shie, and John N Tsitsiklis. 2004. The sample complexity of exploration in the multi-armed bandit problem. *Journal of Machine Learning Research* 5(Jun): 623–648.

Mason, Blake, Lalit Jain, and Robert Nowak. 2017. Learning low-dimensional metrics. In *Advances in neural information processing systems*, 4139–4147.

Mason, Blake, Lalit Jain, Ardhendu Tripathy, and Robert Nowak. 2020. Finding all $\{\epsilon\}$-good arms in stochastic bandits. *Advances in Neural Information Processing Systems*.

Mason, Blake, Martina A. Rau, and Robert Nowak. 2019a. Cognitive task analysis for implicit knowledge about visual representations with similarity learning methods. *Cognitive Science* 43(9):e12744. https://onlinelibrary.wiley.com/doi/pdf/10.1111/cogs.12744.

Mason, Blake, Ardhendu Tripathy, and Robert Nowak. 2019b. Learning nearest neighbor graphs from noisy distance samples. In *Advances in neural information processing systems*, 9586–9596.

Mason, Lucia, Patrik Pluchino, and Maria Caterina Tornatora. 2013. Effects of picture labeling on science text processing and learning: Evidence from eye movements. *Reading Research Quarterly* 48(2):199–214.

Massey, Christin M, Philip J Kellman, Zipora Rother, and Timothy Burke. 2013. Perceptual learning and adaptive learning technology: Developing new approaches to mathematics learning in the classroom. In *Developmental cognitive science goes to school*, 249–263. Routledge.

Mayer, Richard, and Richard E Mayer. 2005. *The cambridge handbook of multimedia learning*. Cambridge university press.

McWhirter, Culver, Dustin G Mixon, and Soledad Villar. 2018. Squeezefit: Label-aware dimensionality reduction by semidefinite programming. *arXiv preprint arXiv:1812.02768*.

van der Meij, Jan. 2007. Support for learning with multiple representations designing simulation-based learning environments.

van der Meij, Jan, and Ton de Jong. 2011. The effects of directive self-explanation prompts to support active processing of multiple representations in a simulation-based learning environment. *Journal of Computer Assisted Learning* 27(5):411–423.

Mossel, Elchanan, Ryan O'Donnell, and Rocco A. Servedio. 2004. Learning functions of k relevant variables. *Journal of Computer and System Sciences* 69(3):421 – 434. Special Issue on STOC 2003.

Negahban, Sahand, Sewoong Oh, and Devavrat Shah. 2012. Iterative ranking from pair-wise comparisons. In *Advances in neural information processing systems*, 2474–2482.

Oymak, Samet, Amin Jalali, Maryam Fazel, Yonina C Eldar, and Babak Hassibi. 2015. Simultaneously structured models with application to sparse and low-rank matrices. *IEEE Transactions on Information Theory* 61(5):2886–2908.

Paivio, Allan. 1990. *Mental representations: A dual coding approach*, vol. 9. Oxford University Press.

Pape, Stephen J, and Mourat A Tchoshanov. 2001. The role of representation(s) in developing mathematical understanding. *Theory into practice* 40(2):118–127.

Peirce, Charles Sanders. 1931. *Collected papers of charles sanders peirce*. Harvard University Press.

Rappoport, Lana T, and Guy Ashkenazi. 2008. Connecting levels of representation: Emergent versus submergent perspective. *International Journal of Science Education* 30(12):1585–1603.

Rau, MA. 2016. A framework for discipline-specific grounding of educational technologies with multiple visual representations. *IEEE Transactions on Learning Technologies*.

Rau, Martina A. 2017. Conditions for the effectiveness of multiple visual representations in enhancing stem learning. *Educational Psychology Review* 29(4):717–761.

———. 2018. Making connections among multiple visual representations: how do sense-making skills and perceptual fluency relate to learning of chemistry knowledge? *Instructional Science* 46(2):209–243.

Rau, Martina A, Vincent Aleven, and Nikol Rummel. 2015a. Successful learning with multiple graphical representations and self-explanation prompts. *Journal of Educational Psychology* 107(1):30.

Rau, Martina A, Vincent Aleven, Nikol Rummel, and Stacie Rohrbach. 2013. Why interactive learning environments can have it all: resolving design conflicts between competing goals. In *Proceedings of the sigchi conference on human factors in computing systems*, 109–118. ACM.

Rau, Martina A, and Amanda L Evenstone. 2014. Multi-methods approach for domain-specific grounding: An its for connection making in chemistry. In *International conference on intelligent tutoring systems*, 426–435. Springer.

Rau, Martina A, Blake Mason, and Robert D Nowak. 2016. How to model implicit knowledge? similarity learning methods to assess perceptions of visual representations. In *Proceedings of the 9th international conference on educational data mining*, 199–206.

Rau, Martina A, Joseph E Michaelis, and Natalie Fay. 2015b. Connection making between multiple graphical representations: A multi-methods approach for

domain-specific grounding of an intelligent tutoring system for chemistry. *Computers & Education* 82:460–485.

Ren, Wenbo, Jia Liu, and Ness B Shroff. 2019. Exploring k out of top ρ fraction of arms in stochastic bandits. In *The 22nd international conference on artificial intelligence and statistics*, 2820–2828.

Richman, Howard B, Fernand Gobet, James J Staszewski, and Herbert A Simon. 1996. Perceptual and memory processes in the acquisition of expert performance: The epam model. *The road to excellence: The acquisition of expert performance in the arts and sciences, sports, and games* 167–187.

Rigollet, Philippe, and Alexandre Tsybakov. 2011. Exponential screening and optimal rates of sparse estimation. *The Annals of Statistics* 731–771.

Sankaranarayanan, Jagan, Hanan Samet, and Amitabh Varshney. 2007. A fast all nearest neighbor algorithm for applications involving large point-clouds. *Computers & Graphics* 31(2):157–174.

Schmidt-Weigand, Florian, Alfred Kohnert, and Ulrich Glowalla. 2010. Explaining the modality and contiguity effects: New insights from investigating students' viewing behaviour. *Applied Cognitive Psychology: The Official Journal of the Society for Applied Research in Memory and Cognition* 24(2):226–237.

Schnotz, Wolfgang. 2005. An integrated model of text and picture comprehension. *The Cambridge handbook of multimedia learning* 49:69.

Schnotz, Wolfgang, and Maria Bannert. 2003. Construction and interference in learning from multiple representation. *Learning and instruction* 13(2):141–156.

Schraagen, Jan Maarten, Susan F Chipman, and Valerie J Shute. 2000. State-of-the-art review of cognitive task analysis techniques. *Cognitive task analysis* 467–487.

Schroff, Florian, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the ieee conference on computer vision and pattern recognition (cvpr)*.

Settles, Burr. 2009. Active learning literature survey. Tech. Rep., University of Wisconsin-Madison Department of Computer Sciences.

Seufert, Tina. 2003. Supporting coherence formation in learning from multiple representations. *Learning and instruction* 13(2):227–237.

Shanks, DR, K Lamberts, and R Goldstone. 2005. Handbook of cognition.

Shepard, Roger N. 1962. The analysis of proximities: Multidimensional scaling with an unknown distance function. i. *Psychometrika* 27(2):125–140.

———. 1980. Multidimensional scaling, tree-fitting, and clustering. *Science* 210(4468):390–398.

Sherin, Bruce L, et al. 2000. Meta-representation: An introduction. *The Journal of Mathematical Behavior* 19(4):385–398.

Shi, Yuan, Aurélien Bellet, and Fei Sha. 2014. Sparse compositional metric learning. *arXiv preprint arXiv:1404.4105*.

Simchowitz, Max, Kevin Jamieson, and Benjamin Recht. 2017. The simulator: Understanding adaptive sampling in the moderate-confidence regime. In *Conference on learning theory*, 1794–1834.

Singla, Adish, Sebastian Tschiatschek, and Andreas Krause. 2016. Actively learning hemimetrics with applications to eliciting user preferences. In *International conference on machine learning*, 412–420.

Stalbovs, Kim, Katharina Scheiter, and Peter Gerjets. 2015. Implementation intentions during multimedia learning: Using if-then plans to facilitate cognitive processing. *Learning and Instruction* 35:1–15.

Stewart, Neil, Gordon DA Brown, and Nick Chater. 2005. Absolute identification by relative judgment. *Psychological review* 112(4):881.

Stieff, Mike. 2005. Connected chemistry-a novel modeling environment for the chemistry classroom. *Journal of Chemical Education* 82(3):489.

Su, Xiaoyuan, and Taghi M Khoshgoftaar. 2009. A survey of collaborative filtering techniques. *Advances in artificial intelligence* 2009.

Talanquer, Vicente. 2009. On cognitive constraints and learning progressions: The case of "structure of matter". *International Journal of Science Education* 31(15): 2123–2136.

Tamuz, Omer, Ce Liu, Serge Belongie, Ohad Shamir, and Adam Tauman Kalai. 2011. Adaptively learning the crowd kernel. *arXiv preprint arXiv:1105.1033*.

Tanczos, Ervin, Robert Nowak, and Bob Mankoff. 2017. A kl-lucb algorithm for large-scale crowdsourcing. In *Advances in neural information processing systems*, 5894–5903.

Tropp, Joel A. 2015. An introduction to matrix concentration inequalities. arXiv:1501.01571.

Tschopp, Dominique, Suhas Diggavi, Payam Delgosha, and Soheil Mohajer. 2011. Randomized algorithms for comparison-based search. In *Advances in neural information processing systems*, 2231–2239.

Tuckey, Helen, Mailoo Selvaratnam, and John Bradley. 1991. Identification and rectification of student difficulties concerning three-dimensional structures, rotation, and reflection. *Journal of Chemical Education* 68(6):460.

Underwood, Geoffrey, and John Everatt. 1992. The role of eye movements in reading: some limitations of the eye-mind assumption. In *Advances in psychology*, vol. 88, 111–169. Elsevier.

(US), National Academies Press. 2006. *Learning to think spatially: Gis as a support system in the k-12 curriculum*. National Academy Press.

Uttal, David H, and Katherine O'Doherty. 2008. Comprehending and learning from 'visualizations': A developmental perspective. In *Visualization: Theory and practice in science education*, 53–72. Springer.

Vaidya, Pravin M. 1989. Ano (n logn) algorithm for the all-nearest-neighbors problem. *Discrete & Computational Geometry* 4(2):101–115.

Van Der Maaten, Laurens, and Kilian Weinberger. 2012. Stochastic triplet embedding. In *Machine learning for signal processing (mlsp), 2012 ieee international workshop on*, 1–6. IEEE.

Van Gog, Tamara, Fred Paas, Jeroen JG Van Merriënboer, and Puk Witte. 2005. Uncovering the problem-solving process: Cued retrospective reporting versus concurrent and retrospective reporting. *Journal of Experimental Psychology: Applied* 11(4):237.

Van Gog, Tamara, and Katharina Scheiter. 2010. Eye tracking as a tool to study and enhance multimedia learning.

VanLehn, Kurt. 2011. The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist* 46(4):197–221.

Wang, Huahua, and Arindam Banerjee. 2014. Randomized block coordinate descent for online and stochastic optimization. *arXiv preprint arXiv:1407.0107*.

Weinberger, Kilian Q, and Lawrence K Saul. 2009. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research* 10(Feb):207–244.

Wertsch, James V, and Sibel Kazak. 2011. Saying more than you know in instructional settings. In *Theories of learning and studies of instructional practice*, 153–166. Springer.

Wise, Joseph A, Tate Kubose, Norma Chang, Arlene Russell, and Philip J Kellman. 2000. Perceptual learning modules in mathematics and science instruction. *Teaching and learning in a network world* 169–176.

Wu, Hsin-Kai, Joseph S Krajcik, and Elliot Soloway. 2001. Promoting understanding of chemical representations: Students' use of a visualization tool in the classroom.

Journal of Research in Science Teaching: The Official Journal of the National Association for Research in Science Teaching 38(7):821–842.

Wylie, Ruth, and Michelene TH Chi. 2014. 17 the self-explanation principle in multimedia learning. *The Cambridge handbook of multimedia learning* 413.

Ying, Yiming, Kaizhu Huang, and Colin Campbell. 2009. Sparse metric learning via smooth optimization. In *Advances in neural information processing systems*, 2214–2222.

Yu, A., and K. Grauman. 2014. Fine-grained visual comparisons with local learning. In *Computer vision and pattern recognition (cvpr)*.

———. 2017. Semantic jitter: Dense supervision for visual comparisons via synthetic images. In *International conference on computer vision (iccv)*.

Yuan, Ming, and Yi Lin. 2006. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 68(1):49–67.

Especially in my early years in Wisconsin, I felt somewhat out of place in a Ph.D. program. I will be eternally grateful for those lab mates who welcomed me and accepted me as one of their own. They were my northern star and my horizon line. I wrote this small poem in my third year after most of them had graduated. I record it here to express how much I appreciate them and the time we shared.

If I could have them all back

just for an evening together:
Drink gin gimlets at Lalit's
or a sloppy martini
with way too much vermouth.
Daniel playing Beethoven on Rudy's upright
and football muted on the big TV.
Coffee on the stove and bread in the oven.
Rising from the couch to fill a glass
and fill it again. Saag paneer
in the saucepan and basmati
rice on the counter by the beer.
Those were the days of excess.
Those were the days
when family was tangible.
When it wafted through the windows,
you could smell it on the exhales
mixing with the cold, winter air.
The happy drunk, the lips red
with chili oil, still buzzing
and cracked into wide smiles.

For my lab mates
Blake Mason, April 24, 2018