Leveraging Structured Sparsity for Data-Efficient and Interpretable Machine Learning

by

Urvashi K. Oswal

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

2019

Date of final oral examination: 9/13/2019

The dissertation is approved by the following members of the Final Oral Committee:

Robert D. Nowak, Professor, Electrical Engineering
Timothy Rogers, Professor, Psychology
Barry Van Veen, Professor, Electrical Engineering
Stephen Wright, Professor, Computer Science

To my parents and Vishal.

ABSTRACT

The availability of data has soared exponentially in recent years. However, human expertise has remained an expensive and time-limited resource. This thesis focuses on the development of efficient machine learning algorithms and theory that leverage redundancies and structure in the data to optimize the available human and computational resources. These efforts are motivated by applications of machine learning to human-generated data such as brain imaging, biometric analysis and recommendation systems. We exploit various notions of structure including new approaches to traditional sparsity, low-rank matrix approximations using pre-defined groups of column subsets, and an adaptive notion of sparsity based on correlated groups of variables.

First, we consider a linear bandits framework motivated by recommendation systems. This involves adaptively collecting data from users in the form of rewards and/or explanations with the aim of retrieving the most relevant items from a collection. These items can be documents (such as research papers or insurance claims) or images (such as retail products from a catalog). Traditional results on sparsity from compressed sensing break down in this framework since the actions taken are not independent. Hence, we explore a new form of the linear bandit problem in which the algorithm receives the usual stochastic rewards as well as stochastic feedback about which features are relevant to the rewards, the latter feedback

being the novel aspect.

Another notion of simplicity considered is the low-rank approximation of a matrix using a subset of its columns (and rows). Motivated by biometric applications, we generalize this approximation to incorporate known group structure in the column (and row) subsets.

Finally, we develop tools for learning and inference in the presence of correlated variables by introducing adaptive notions of sparsity, and apply them to problems in cognitive neuroscience and subspace clustering. The new regularization methods generalize the sparsity inducing regularizer, Lasso, to automatically cluster and average regression coefficients associated with strongly correlated variables. In brain imaging, the cost of acquiring data samples is high. Often the number of data samples is much fewer than the number of variables. To deal with this challenge, we propose methods to reduce complexity of solutions, as well as from a neuroscience point of view, to get a more interpretable model by including correlated variables. In subspace clustering, we build on tools developed for handling correlations to develop a new approach that is significantly more computationally efficient and scalable than existing methods using the key observation that points in the same subspace tend to be more correlated than points in different subspaces.

ACKNOWLEDGMENTS

First of all, I would like to express my sincere gratitude to my advisor, Robert Nowak, for continuous support and guidance. Thank you for introducing me to the world of research and teaching me to always trust my intuition and to step out of my comfort zone. I could not have imagined having a better advisor and mentor.

Besides my advisor, I would like to thank the rest of my committee members: Timothy Rogers, Barry Vanveen, and Stephen Wright for their insightful comments and encouragement. I would also like to thank other professors who have guided me along the way including Rebecca Willett and Po-Ling Loh.

I am grateful to my brilliant collaborators, from University of Wisconsin, Technicolor, Lands' End, and American Family Insurance, for their insightful suggestions and help. In particular, I have learnt a lot from Brian Eriksson, Kevin Xu, Swayambhoo Jain, and Christopher Cox.

Thanks to my fellow labmates, Ting-Ting Nan, Nikhil Rao, Aniruddha Bhargava, Blake Mason, Scott Sievert, Kwang-Sung Jun, Sumeet Katariya, Daniel Pimentel, Lalit Jain, Cong Han Lim, Xin Hunt, Eric Hall, and many others for stimulating discussions and fun times. I am also grateful to all my friends outside the lab who helped make my time at Madison enjoyable.

The completion of my dissertation would not have been possible with-

out the support and nurturing of my family. I am extremely grateful to my parents for their endless love, encouragement and support. I am deeply indebted to them for providing me with a positive and strong foundation. Thank you to my sister, Veenal, for being beside me through thick and thin. I am also grateful to my grandparents for their sacrifices and wisdom.

Finally, to my husband, Vishal, thank you for being a constant source of motivation and kindness. Your support and unwavering faith have made it possible for me to stay focused and grounded through grad school.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

The applications of machine learning and signal processing are diverse, from cognitive neuroscience to recommendation systems. Most applications involve dealing with huge amounts of data. However, the bottleneck in these methods is often the human response required. Human expertise is often an expensive and time-limited resource. To deal with this challenge, we resort to leveraging redundancies and structure in the data to reduce the human effort.

In the first part, we consider the setting of multi-armed bandits, in particular, linearly parameterized stochastic bandits. Linear stochastic bandit algorithms are used to sequentially select actions to maximize rewards. For instance, to model recommendation systems that help users navigate through a large collection of items (products, videos, documents). Consider the application of recommending news articles. At every time instant, the algorithm recommends an article to the user from a large database containing articles about topics like “politics", “technology", “sports". The user provides a numerical reward corresponding to her assessment of the document’s value. The goal of the algorithm is to maximize the cumulative reward over time. This can be challenging if the majority of the documents in the database are not of interest to the user. Linear bandit algorithms strike a balance between *exploration* of the database to ascertain the user’s interests and *exploitation* by retrieving documents similar to those that

have received the highest rewards. Unfortunately, standard linear bandit algorithms suffer from the curse of dimensionality. The regret (cumulative difference between optimal rewards and rewards obtained by bandit algorithms) grows linearly with the feature dimension d. The dimension d may be quite large in modern applications (*e.g.*, 1000s of features in NLP or image/vision applications). In the recommendations setting, the existing bounds easily require $T > 100$s of ratings to be meaningful, which is not realistic. The high-dimensionality also makes it challenging to employ state-of-the-art algorithms since it involves maintaining and updating a $d \times d$ matrix at every stage. We tackle the problem of linear bandits from a new perspective that incorporates feature feedback in addition to reward feedback, mitigating the curse of dimensionality. Specifically, we consider situations in which the algorithm receives a stochastic reward *and* stochastic feedback indicating which, if any, feature-dimensions were relevant to the reward value.

Another form of structured sparsity is explored in Chapter 3 where the matrices storing biometric data are often low-rank. Preserving the original structure in the data may be desirable due to many reasons including interpret-ability in case of biometric data or for storage efficiency in case of sparse matrices. This has led to the introduction of the CUR decomposition, where the factorization is performed with respect to a subset of rows and columns of the matrix itself. In many real-world environments, the ability to sample specific individual rows or columns of the matrix is limited by

either system constraints or cost. Thus, we consider matrix approximation by sampling predefined *blocks* of columns (or rows) from the matrix.

The third form of sparsity is introduced in Chapter 4. It focuses on learning and inference from data that has been collected from human subjects such as the activations in different regions of the brain in response to certain stimulus. Such applications involve several variables, often tens of thousands to millions. These variables correspond to voxels or brain regions in the case of fMRI studies. Further, the number of samples available are small due to acquisition costs. To deal with this challenge, sparsity regularizers such as Lasso and Group Lasso are used to reduce complexity of solutions, to make a solution tractable, as well as from a neuroscience point of view, to get a more interpretable model. A severe limitation of Lasso arises when some of the variables are strongly correlated. In such cases, Lasso selects an arbitrary subset of the correlated variables to preserve sparsity of the solution. This is a concern in fMRI, since certain voxels may have very correlated activation patterns and the goal is to identify *all* the voxels that are relevant to the task. Another limitation of the Lasso and Group lasso methods is that they can select at most n features (n being the number of samples), since the number of nonzero coefficients in the solution cannot exceed the number of measurements. This can be severe limitation in applications where the number of features far exceeds the number of items.We develop methods to deal with these challenges and apply them to problems in cognitive neuroscience and subspace cluster-

ing, and show that indeed the methods that take these correlations into consideration lead to more interpretable and efficient solutions.

1.1 Summary of Contributions

Chapter 2

This chapter explores a new form of the linear bandit problem in which the algorithm receives the usual stochastic rewards as well as stochastic feedback about which features are relevant to the rewards, the latter feedback being the novel aspect. The focus of this chapter is the development of new theory and algorithms for linear bandits with feature feedback which can achieve regret over time horizon T that scales like $k\sqrt{T}$, without prior knowledge of which features are relevant nor the number k of relevant features. In comparison, the regret of traditional linear bandits is $d\sqrt{T}$, where d is the total number of (relevant and irrelevant) features, so the improvement can be dramatic if $k \ll d$. The computational complexity of the new algorithm is proportional to k rather than d, making it much more suitable for real-world applications compared to traditional linear bandits. We demonstrate the performance of the new algorithm with synthetic and real human-labeled data.

Chapter 3

A common problem in large-scale data analysis is to approximate a matrix using a combination of specifically sampled rows and columns, known as CUR decomposition. In many real-world environments, the ability to sample specific individual rows or columns of the matrix is limited by either system constraints or cost. In this chapter, we consider matrix approximation by sampling predefined *blocks* of columns (or rows) from the matrix. We present an algorithm for sampling useful column blocks and provide novel guarantees for the quality of the approximation. We demonstrate the effectiveness of the proposed algorithms for computing the Block CUR decomposition of large matrices in a distributed setting with multiple nodes in a compute cluster and in a biometric data analysis setting using real-world user data from content testing.

Chapter 4

We introduce and analyze the clustering properties of the regularization methods called Ordered Weighted ℓ_1 (OWL) under less restrictive conditions. The Lasso and OSCAR regularizers are specific instances of OWL. We propose a generalization of the OWL approach to the multi-task setting, and thus call our new approach Group OWL (GrOWL). We show that GrOWL shares many of the desirable features of the OWL method, namely it automatically clusters and averages regression coefficients associated

with strongly correlated columns of the design matrix. This has two desirable effects, in terms of both model selection and prediction. First, GrOWL can select all of the relevant features, unlike standard group lasso which may not select relevant features if they happen to be strongly correlated with others. Second, GrOWL encourages the coefficients associated with strongly correlated features to be near or exactly equal. In effect, this averages strongly correlated columns which can help to denoise features and improve predictions.

Chapter 5

Representational Similarity Learning (RSL) aims to discover features that are important in representing (human-judged) similarities among objects. We formulate RSL as a sparsity-regularized multi-task regression problem. Standard methods, like group lasso, may not select important features if they are strongly correlated with others. To address this shortcoming we present a new regularizer for multitask regression called *Group Ordered Weighted* ℓ_1 (GrOWL). Another key contribution is a novel application to fMRI brain imaging. *Representational Similarity Analysis* (RSA) is a tool for testing whether localized brain regions encode perceptual similarities. Using GrOWL, we propose a new approach called *Network RSA* that can discover arbitrarily structured brain networks (possibly widely distributed and non-local) that encode similarity information. We show, in theory and fMRI experiments, how GrOWL deals with strongly correlated covariates.

Chapter 6

The main contribution in this chapter is a new approach to subspace clustering that is significantly more computationally efficient and scalable than existing state-of-the-art methods. The central idea is to modify the regression technique in sparse subspace clustering (SSC) by replacing the ℓ_1 minimization with a generalization called Ordered Weighted ℓ_1 (OWL) minimization which performs simultaneous regression and clustering of correlated variables. Using random geometric graph theory, we prove that OWL regression selects more points within each subspace than ℓ_1, resulting in better clustering results. This allows for accurate subspace clustering based on regression solutions for only a small subset of the total dataset, significantly reducing the computational complexity compared to SSC. In experiments, we find that our OWL approach can achieve a speedup of $20\times$ to $30\times$ for synthetic problems and $4\times$ to $8\times$ on real data problems.

Full List of Publications

1. Representational Similarity Learning with Application to Brain Networks. U. Oswal, C. Cox, T. Rogers, and R. Nowak. *International Conference on Machine Learning*, ICML 2016.

2. A Compressed Sensing Decomposition of Electrodermal Activity Signals. S. Jain, U. Oswal, K. S. Xu, B. Eriksson, and J. Haupt. *IEEE*

Transactions on Biomedical Engineering, 2017.

3. Scalable Sparse Subspace Clustering via Ordered Weighted ℓ_1. U. Oswal, and R. Nowak. *Allerton Conference on Communication, Control, and Computing, Allerton* 2018.

4. Block CUR: Decomposing Matrices using Groups of Columns. U. Oswal, S. Jain, K. S. Xu, and B. Eriksson. *European Conference on Machine Learning*, ECML-PKDD 2018.

5. Active Sparse Feature Selection Using Deep Convolutional Features for Image Retrieval. D. Conathan, U. Oswal, and R. Nowak. *Artificial Intelligence in Insurance, SIAM International Conference on Data Mining*, 2018.

6. Linear Bandits with Feature Feedback. U. Oswal, A. Bhargava, and R. Nowak. arXiv preprint arXiv:1903.03705, 2019.

Organization

Each chapter draws ideas from previous chapters but is mostly self-contained. The remainder of the dissertation is organized as follows. Chapter 2 provides a framework for information retrieval using linear bandits simultaneously allowing users to provide feature feedback to improve the search. In Chapter 3, we consider matrix approximation by sampling predefined *blocks* of columns (or rows) from the matrix. In Chapter 4, we motivate and

analyze the automatic clustering properties of Group Ordered Weighted ℓ_1 (GrOWL) regularization for sparse multi-task linear regression in the presence of strong correlations, and a proximal gradient descent algorithm for its computation. Chapter 5 covers the application of these penalties to discover features that are important in representing (human-judged) similarities among objects. The tools developed for handling correlations are used to develop a new approach to subspace clustering that is significantly more computationally efficient and scalable than existing state-of-the-art methods in Chapter 6. Chapter 7 presents applications of the linear bandits framework to real-world information retrieval problems, and concludes the thesis with future directions of this work. Proofs of the technical results are contained in the Appendices.

2.1 Introduction

Linear stochastic bandit algorithms are used to sequentially select actions to maximize rewards. For instance, Deshpande and Montanari (2012) propose to model recommendation systems that help users navigate through a large collection of items (products, videos, documents) using linearly parameterized multi-armed bandits. This model strikes a balance by allowing the user to explore the space of available items and probing the user's preferences. The linear bandit model assumes that the expected reward of each action is an (unknown) linear function of a (known) finite-dimensional feature associated with the action. Mathematically, if $\boldsymbol{x}_t \in \mathbb{R}^d$ is the feature associated with the action chosen at time t, then the stochastic reward is

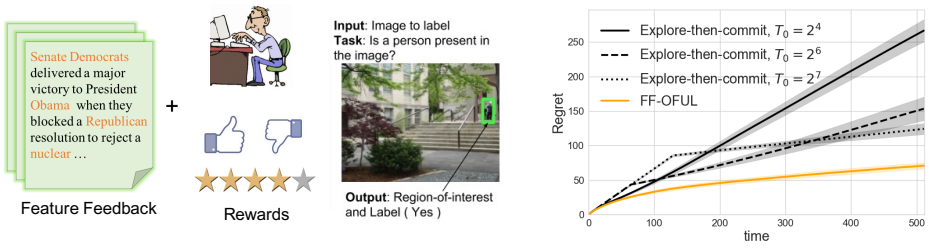
$$y_t = \boldsymbol{x}_t^\top \boldsymbol{\theta}_* + \eta_t, \tag{2.1}$$

where $\boldsymbol{\theta}_*$ is the unknown linear functional (representing the user's preferences) and η_t is a zero mean random variable. The goal is to adaptively select actions to maximize the rewards (corresponding to user's assessment of the chosen item's value). This involves (approximately) learning $\boldsymbol{\theta}_*$ and exploiting this knowledge. Linear bandit algorithms that exploit this special structure have been extensively studied and applied Abbasi-Yadkori

et al. (2011); Rusmevichientong and Tsitsiklis (2010).

Unfortunately, standard linear bandit algorithms suffer from the curse of dimensionality. The regret grows linearly with the feature dimension d. The dimension d may be quite large in modern applications (*e.g.*, 1000s of features in NLP or image/vision applications). In the recommendations setting, the existing bounds easily require $T > 100$s of ratings to be meaningful, which is not realistic. The high-dimensionality also makes it challenging to employ state-of-the-art algorithms since it involves maintaining and updating a $d \times d$ matrix at every stage. However, in many cases the linear function may only involve a sparse subset of the $k < d$ features, and this can be exploited to partially reduce dependence on d. In such cases, the regret of sparse linear bandit algorithms scales like $\sqrt{dk}$ Abbasi-Yadkori et al. (2012); Lattimore and Szepesvári (2018).

We tackle the problem of linear bandits from a new perspective that incorporates feature feedback in addition to reward feedback, mitigating the curse of dimensionality. Specifically, we consider situations in which the algorithm receives a stochastic reward *and* stochastic feedback indicating which, if any, feature-dimensions were relevant to the reward value Oswal et al. (2019). For example, consider a situation in which users rate recommended text documents and additionally highlight keywords or phrases that influenced their ratings. Figure 2.1(a) illustrates the idea. Obviously, the additional "feature feedback" may significantly improve an algorithm's ability to home-in on the relevant features. The focus of



(a) (b)

Figure 2.1: (a) (Left) Highlighted words for text-based applications and (Right) Region-of-interest feature feedback for image-based applications. (b) Comparison of the explore-then-commit strategy for different values of T_0 and our new FF-OFUL algorithm which combines exploration and exploitation steps (details of data generation in Section 2.5).

this chapter is the development of new theory and algorithms for linear bandits with feature feedback. We show that the regret of linear bandits with feature feedback scales linearly in k, the number of relevant features, without prior knowledge of which features are relevant nor the value of k. This leads to large improvements in theory and practice.

The simple feedback model, where the user directly selects a subset of the relevant features, can be generalized by allowing for an indirect form of feedback. For example, the user can select a region of an image instead of a subset of the standard deep neural network features. This form of feedback could be used in different ways. For instance, we could use methods to map deep image features to image regions. However, in this chapter we focus on the processes and benefits of incorporating direct feature feedback, and defer the development of indirect feedback models to future work.

Perhaps the most natural and simple way to leverage the feature feedback is an explore-then-commit strategy. In the first T_0 steps the algorithm selects actions at random and receives rewards and feature feedback. If T_0 is sufficiently large, then the algorithm will have learned all or most of the relevant features and it can then switch to a standard linear bandit algorithm operating in the lower-dimensional subspace defined by those features. There are two major problems with such an approach:

1. The correct choice of T_0 depends on the prevalence of relevant features in randomly selected actions, which generally is unknown. If T_0 is too small, then many relevant features will be missed and the long-run regret will scale linearly with the time horizon. If T_0 is too large, then the initial exploration period will suffer excess regret. This is depicted in Figure 2.1(b).

2. Regardless of the choice of T_0, the regret will grow linearly for $t < T_0$. The new FF-OFUL algorithm that we propose combines exploration and exploitation from the start and can lead to smaller regret initially and asymptotically as shown in Figure 2.1(b).

These observations motivate our proposed approach that dynamically adjusts the trade-off between exploration and exploitation. A key aspect of the approach is that it is automatically adaptive to the unknown number of relevant features k. Our theoretical analysis shows that its regret scales like $k\sqrt{T}$. Experimentally, we show the algorithm generally outperforms

traditional linear bandits and the explore-then-commit strategy. This is due to the fact that the dynamic algorithm exploits knowledge of relevant features as soon as they are identified, rather than waiting until all or most are found. A key consequence is that our proposed algorithm yields significantly better rewards at early stages of the process, as shown in Figure 2.1(b) and in more comprehensive experiments later in the chapter. The intuition for this is that estimating $\boldsymbol{\theta}_*$ on a fraction of the relevant coordinates can be exploited to recover a fraction of the optimal reward. Similar ideas are explored in linear bandits (without feature feedback) in Deshpande and Montanari (2012).

Definitions

For round, t, let $\mathcal{X}_t \subseteq \mathbb{R}^d$ be the set of actions/items provided to the learner. We assume the standard linear model for rewards with a hidden weight vector $\boldsymbol{\theta}_* \in \mathbb{R}^d$. If the learner selects an action, $\mathbf{x}_t \in \mathcal{X}_t$, it receives reward, y_t, defined in (2.1) where η_t is noise with a sub-Gaussian random distribution with parameter R. For the set of actions $\mathcal{X}_t$, the optimal action is given by, $\mathbf{x}_t^* := \text{argmax}_{\mathbf{x} \in \mathcal{X}_t} \mathbf{x}^\top \boldsymbol{\theta}_*$, which is unknown. We define regret as,

$$R_T = \sum_{t=1}^{T} \left(\mathbf{x}_t^{*\top} \boldsymbol{\theta}_* - \mathbf{x}_t^\top \boldsymbol{\theta}_* \right). \tag{2.2}$$

This is also called cumulative regret but, unless stated otherwise, we will refer to it as regret. We refer to the quantity $\mathbf{x}_t^{*\top}\boldsymbol{\theta}_* - \mathbf{x}_t^{\top}\boldsymbol{\theta}_*$ as the instantaneous regret which is the difference between the optimal reward and the reward received at that instant. We make the standard assumption that the algorithm is provided with an enormous action set which is only changing slowly over time, for instance, from sampling the actions without replacement ($\mathcal{X}_{t+1} = \mathcal{X}_t \backslash \mathbf{x}_t$).

Related Work

Algorithm 1 OFUL from Abbasi-Yadkori et al. (2011)

1: **for** $t = 1, 2, \ldots, T-1$ **do**
2: $(\mathbf{x}_t, \tilde{\boldsymbol{\theta}}_t) = \text{argmax}_{(\mathbf{x},\boldsymbol{\theta}) \in \mathcal{X}_t \times \mathcal{C}_{t-1}} \langle \mathbf{x}, \boldsymbol{\theta} \rangle$
3: Select action $\mathbf{x}_t$ and receive reward y_t.
4: Update $\overline{\mathbf{V}}_t = (\boldsymbol{X}_t^T \boldsymbol{X}_t + \lambda \boldsymbol{I})$ and $\widehat{\boldsymbol{\theta}}_t = \overline{\mathbf{V}}_t^{-1} \boldsymbol{X}_t^T \boldsymbol{y}_t$
5: Update ellipsoidal confidence set $\mathcal{C}_t$ as $\mathcal{C}_t = \left\{ \boldsymbol{\theta} : \|\widehat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}\|_{\overline{\mathbf{V}}_t} \leq f(R, \boldsymbol{X}_t, \boldsymbol{y}_t, \lambda) \right\}$
(for details on $f(\cdot)$ see Abbasi-Yadkori et al. (2011))
6: **end for**

The area of contextual bandits was introduced by Ginebra and Clayton (1995). The first algorithms for linear bandits appeared in Abe and Long (1999) followed by those using the optimism in the face of uncertainty principle, Auer and Long (2002); Dani et al. (2008). Rusmevichientong and Tsitsiklis (2010) showed matching upper and lower bounds when the action (feature) set is a unit hypersphere. Finally, Abbasi-Yadkori et al. (2011) gave a tight regret bound using new martingale techniques. We

use their algorithm, OFUL, as a subroutine in our work. In the area of sparse linear bandits, regret bounds are known to scale like $\sqrt{kdT}$, Abbasi-Yadkori et al. (2012); Lattimore and Szepesvári (2018), when operating in a d dimensional feature space with k relevant features. The strong dependence on the ambient dimension d is unavoidable without further (often strong and unrealistic) assumptions. These results also assume knowledge of sparsity parameter k and without it no algorithm can satisfy these regret bounds for all k simultaneously. In contrast, we propose a new algorithm that automatically adapts to the unknown sparsity level k and removes the dependence of regret on d by exploiting additional feature feedback. In terms of feature feedback in text-based applications, Croft and Das (1989) have proposed a method to reorder documents based on the relative importance of words using feedback from users. Poulis and Dasgupta (2017) consider a similar problem but for learning a linear classifier. We use a similar feedback model but focus on the bandit setting where such feedback can be naturally collected along with rewards. The idea of allowing user's to provide richer forms of feedback has been studied in the active learning literature Druck et al. (2009); Raghavan et al. (2006) and also been considered in other (interactive) learning tasks, such as cognitive science Roads et al. (2016), machine teaching Chen et al. (2018), and NLP tasks Yessenalina et al. (2010).

2.2 Problem Setting

The algorithm presents the user with an item (*e.g.*, document) and the user provides feedback in terms of whether they like the item or not (logistic model) or how much they like it (inner product model). The user also selects a few features (*e.g.*, words), if they can find them, to help orient the search. We make the following assumptions.

Assumption 2.1 (Sparsity)**.** *The hidden weight vector $\boldsymbol{\theta}_* \in \mathbb{R}^d$ is k-sparse and k is unknown. In other words, $\boldsymbol{\theta}_*$ has at most k non-zero entries or if $supp(\boldsymbol{\theta}_*) = \{i|\boldsymbol{\theta}_{*i} \neq 0\}$ then $|supp(\boldsymbol{\theta}_*)| = k \leq d$.*

Assumption 2.2 (Discoverability)**.** *For an action $\mathbf{x} \in \mathcal{X}$ selected uniformly at random, the probability that a relevant feature is present and is selected is at least $p > 0$ (unknown).*

Assumption 2.3 (Noise)**.** *Users may report irrelevant features. The number of reported irrelevant features (denoted by $0 \leq k' \leq d - k$) is unknown in advance.*

Assumption 2.1 ensures that there are at most k relevant features, however we stress that the value of k is unknown (it is possible that all d features are relevant). Assumption 2.2 ensures that while every item may not have relevant features, we are able to find them with a non-zero probability when searching through items at random. This assumption can be viewed as a (possibly pessimistic) lower bound on the rate at which relevant features are discovered. For example, it is possible that exploitative

actions may yield relevant features at a higher rate (e.g., relevant features may be correlated with higher rewards). We do not attempt to model such possibilities since this would involve making additional assumptions that may not hold in practice. Assumption 2.3 accounts for ambiguous features that are irrelevant but users erring on the side of marking as relevant.

The set up is as follows: we have a set of items or actions, $\mathcal{X} \subseteq \mathbb{R}^d$ that we can propose to the users. There is a hidden weight vector $\boldsymbol{\theta}_* \in \mathbb{R}^d$ that is k-sparse. We will further assume that $\|\boldsymbol{\theta}_*\| \leq S$ and the action vectors are bounded in norm: $\forall \mathbf{x} \in \mathcal{X}, \|\mathbf{x}\| \leq L$. Besides the reward y_t, defined in (2.1), at each time-step the learner gets $\mathcal{I}_t \subseteq \text{supp}(\boldsymbol{\theta}_*)$ which is the relevance feedback information. The model further specifies that $\forall j \in \text{supp}(\boldsymbol{\theta}_*), \Pr(j \in \mathcal{I}_t) \geq p$. That is, the probability a relevant feature is selected at random is at least p. We need this assumption to make sure that we can find all the relevant features.

2.3 Algorithm

In this section, we introduce an algorithm that makes use of feature relevance feedback by starting with a small feature space and gradually increasing the space over time without knowledge of k. We use the OFUL algorithm (stated as Algo. 1) based on the principle of optimism in face of uncertainty as a subroutine. The algorithm constructs ellipsoidal confidence sets centered around the ridge regression estimate, using observed

data such that the sets contain the unknown θ_* with high probability, and selects the action/item that maximizes the inner product with any θ from the confidence set.

Algorithm 2 Feature Feedback OFUL (FF-OFUL)

Let the set of relevant indices, $\mathcal{R}_0, \mathcal{I}_0 = \{\}$.
while $\mathcal{I}_0$ is empty **do**
  Select action at random, $\mathcal{I}_0 = \{$ indices revealed $\}$
end while
$\mathcal{R}_1 = \mathcal{R}_0 \bigcup \mathcal{I}_0$
Initialize $\mathcal{C}_0$ using actions sampled.
for $t = 1, 2, \ldots, T$ **do**
  Let $\boldsymbol{X}_t$ be the feature matrix restricted to $\mathcal{R}_t$.
  Set $\epsilon_t = 1/\sqrt{t}$. Draw b_t from bernoulli(ϵ_t)
  if $b_t = 1$ **then**
    Pick an action $\mathbf{x}_t$ uniformly at random from $\mathcal{X}_t$,
  else
    Pick $(\mathbf{x}_t, \tilde{\boldsymbol{\theta}}_t) = \text{argmax}_{(\mathbf{x},\boldsymbol{\theta}) \in \mathcal{X}_t \times \mathcal{C}_{t-1}} \langle \mathbf{x}, \boldsymbol{\theta} \rangle$
  end if
  With action $\mathbf{x}_t$ observe reward y_t and indices, $\mathcal{I}_t$.
  Update $\mathcal{R}_t = \mathcal{R}_t \bigcup \mathcal{I}_t$
  if $\mathcal{I}_t$ is empty **then**
    Rank one update to $\overline{\mathbf{V}}_t, \widehat{\boldsymbol{\theta}}_t, \mathcal{C}_t$ (see Algo. 1) using $(y_t, \mathbf{x}_t)$
  else
    Update $\boldsymbol{X}_t$ with features in $\mathcal{R}_t$.
    Recompute $\overline{\mathbf{V}}_t, \widehat{\boldsymbol{\theta}}_t, \mathcal{C}_t$ with new feature set $\boldsymbol{X}_t$.
  end if
end for

All updates are made only in the dimensions that have been marked as relevant and the space is dynamically increased as new relevant features are revealed. If nothing is marked as relevant, then by default the actions are selected at random, potentially suffering the worst possible reward

but, at the same time, increasing our chances of getting relevance feedback leading to a trade-off. Note that the algorithm is adaptive to the unknown number of relevant features k. If k were known, we could stop looking for features when all relevant ones have been selected. We find that in practice, this algorithm has an additional benefit of being more robust to changes in the ridge parameter (λ) due to its intrinsic regularization of restricting the parameter space.

Abbasi-Yadkori et al. (2011) provide a $\tilde{O}(d\sqrt{t})$ bound (restated here as Theorem 2.1) on the regret of OFUL stated as Algorithm 1 by ignoring constants and logarithmic terms.

Theorem 2.1 (Abbasi-Yadkori et al. (2011))**.** *Assume that* $\forall t > 0$ *and* $\mathbf{x} \in \mathcal{X}_t \subset \mathbb{R}^d$, $\langle \mathbf{x}, \boldsymbol{\theta}_* \rangle \in [-1, 1]$. *Then with probability at least* $1 - \delta$*, the regret of OFUL satisfies:*

$$\forall t, R_t \leq 4\sqrt{td\log(\lambda + tL/d)}(\lambda^{1/2}S + R\sqrt{2\log(1/\delta) + d\log(1 + tL/(\lambda d))})$$

where $\lambda > 0$ *is the ridge regression parameter of OFUL.*

We prove a result similar to Theorem 2.1 but reduce the dependence on the dimension from d to k. In order to do so, we must discover the support of $\boldsymbol{\theta}_*$. The idea being that we apportion a set of actions, with a form of ϵ-greedy algorithm due to Sutton and Barto (1998), to random plays in order to guarantee that we find all the relevant features, otherwise we run OFUL on the identified relevant dimensions. Reducing the proportion of random

actions over time guarantees that the regret remains sub-linear in time. We propose Algorithm 2 to exploit feature feedback. Here, at each time t, with probability proportional to $1/\sqrt{t}$, the algorithm selects an action/item to present at random, otherwise it selects the item recommended by feature-restricted-OFUL.

2.4 Main Results

In this section, we state regret bounds for the FF-OFUL algorithm along with a sketch of the proof and discuss approaches to improve the bounds while deferring proof details to supplementary material.

Regret Bound for Algorithm 2 (FF-OFUL)

Recall that $\forall \mathbf{x} \in \mathcal{X}, \|\mathbf{x}\| \leq L$ and $\|\boldsymbol{\theta}_*\| \leq S$. Therefore, for any action, the worst-case instantaneous regret can be derived using Cauchy-Schwarz as follows:

$$|\langle \mathbf{x}^*, \boldsymbol{\theta}_* \rangle - \langle \mathbf{x}, \boldsymbol{\theta}_* \rangle| \quad \leq |\langle \mathbf{x}^*, \boldsymbol{\theta}_* \rangle| + |\langle \mathbf{x}, \boldsymbol{\theta}_* \rangle| \quad \leq \|\mathbf{x}^*\| \, \|\boldsymbol{\theta}_*\| + \|\mathbf{x}\| \, \|\boldsymbol{\theta}_*\| \quad \leq 2SL$$

We provide the main result that bounds the regret (2.2) of Algorithm 2 in the following theorem.

Theorem 2.2. *Assume that* $\forall t > 0$ *and* $\mathbf{x} \in \mathcal{X}_t$, $\langle \mathbf{x}, \boldsymbol{\theta}_* \rangle \in [-1, 1]$, *with the additional assumptions 2.1, 2.2 and 2.3 (k' = 0). Then with probability* $\geq 1 - \delta$,

the cumulative regret after T steps for Algorithm 2,

$$R_T \leq \frac{8SL}{\log 6M/\delta}\left(\frac{\log 3k/\delta}{\log 1/(1-p)}\right)^2 + \log_2\frac{T}{2}\left(3SL\sqrt{T\log\frac{6M}{\delta}}\right) + 4\log_2\frac{T}{2}\sqrt{\frac{T}{2}k\log(\lambda+nL/k)}\left(\lambda^{1/2}S + R\sqrt{2\log(3M/\delta)+k\log(1+TL/(2\lambda k))}\right).$$

where $M = \log_2\frac{T}{2}$, $\lambda > 0$ is the ridge regression parameter and k is the (unknown) number of relevant features.

In other words, with high probability, the regret of Algorithm 2 (FF-OFUL) scales like $\tilde{O}(k\sqrt{T} + \frac{1}{p^2})$, by ignoring constants and logarithmic terms and using the taylor series expansion of $-\log(1-p)$, over time horizon T where k is the number of relevant features and p is the probability with which a relevant feature is marked in an action selected uniformly at random.

Remarks. The values of k and p are unknown to the algorithm and it implicitly adapts to these problem-dependent parameters. Since the regret of any algorithm is trivially bounded by O(T) (assuming bounded rewards), our new regret bound is non-trivial for $T > \max(k^2, p^{-2})$. In comparison, linear bandits without feature feedback have an O($d\sqrt{T}$) regret, which is non-trivial only when $T > d^2$. So, our new algorithm enjoys a better regret bound if $p > d^{-1}$, which is a reasonable condition in high-dimensional settings (*e.g.*, $d = 10^4$).

The three terms in the total regret come from the following events.

Regret due to: (1) exploration to guarantee observing all the relevant features (with high probability), (2) exploration after observing all relevant features (due to lack of knowledge of p or k), and (3) exploitation and exploration running OFUL (after having observed all the relevant features).

In practice, feature feedback may be noisy. Sometimes, features that are irrelevant may be marked as relevant. To account for this, we can relax our assumption to allow for subset of k' irrelevant features that are mistakenly marked as relevant. Including these features will increase the regret but the theory goes through without much difficulty as stated in the following corollary.

Corollary 2.3. *With the same assumptions as Theorem 2.2, if a fixed set of k' irrelevant features were indicated by the user (Assumption 2.3), then the regret of Algorithm 2 (FF-OFUL) scales like $\tilde{O}((k+k')\sqrt{T}+\frac{1}{p^2})$.*

The corollary follows since exploration is not affected by this noise and the regret of exploitation on the vector restricted to $k+k'$ dimensions scales like $(k+k')\sqrt{T}$. This accounts for having some features being ambiguous and users erring on the side of marking them as relevant. This only results in slightly higher regret so long as $k+k'$ is still smaller than d. One could improve this regret by making assumptions on the probabilities of feature selection to weed out the irrelevant features.

Proof Sketch of Main Result

We provide a sketch of the proof here and defer details to supplementary material. Recall, the cumulative regret is summed over the instantaneous regrets for $t = 1, \ldots, T$. We divide the cumulative regret across epochs $s = 0, \ldots, M$ of doubling size $T_s = 2^s$ for $M = \log_2 \frac{T}{2}$. This ensures that the last epoch dominates the regret and it allows for the evolving feature space. For each epoch, we bound the regret under two events, all relevant features have been identified (via user feedback) up to that epoch or not. First, we bound the regret conditioned on the event that all the relevant features have been identified in Lemma A.4. This is further, in expectation, broken down into the ϵ_s portion of random actions for pure exploration (Lemma A.1) and $1 - \epsilon_s$ modified OFUL actions on the k-dimensional feature space for exploitation-exploration (Lemma A.3). For pure exploration, we use the worst case regret bound but since ϵ_s is decreasing this does not dominate the OFUL term. Second, we bound the probability that some of the relevant features are not identified so far (Proposition A.4), which is a constant depending on k and p since it becomes zero after enough epochs have passed. Pure exploration ensures the probability that some features are not identified decreases with each passing epoch. An issue of bounding regret of the actions selected by OFUL subroutine in each epoch is that, unlike OFUL, the confidence sets in our algorithm are constructed using additional actions from exploration rounds and past epochs. To accommodate this we prove a regret bound

for this variation in Lemma A.3. Putting all this together gives us the final result.

Lower bound. The arguments from Dani et al. (2008); Rusmevichientong and Tsitsiklis (2010) can be used get a lower bound of $O(k\sqrt{T})$. To see this, assume that we know the support. Then any linear bandit algorithm that is run on that support must incur an order $k\sqrt{T}$ regret. We don't know the support but we estimate it with high probability and therefore the lower bound also applies here. Our algorithm is optimal up to log factors in terms of the dimension.

Better Early-Regret Bounds

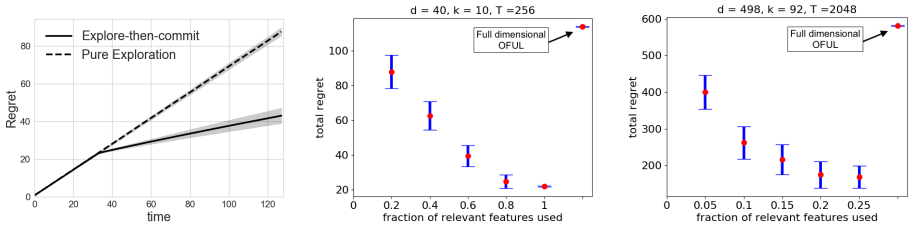
Our analysis bounds the regret of early rounds, before observing all relevant features, with the worst case regret which may be too pessimistic in practice. We present **results to support the idea of restricting the feature space in the short-term horizon and growing the feature space over time**. The results also suggest that an additional assumption on the behavior of early-regret could lead to better constants in our bounds. Any linear bandit algorithm restricted to the support of $\boldsymbol{\theta}_*$ must incur an order $k\sqrt{T}$ regret so one can only hope to improve the constants of the bound.

In Figure 2.2(a) it can be seen that the average linear regret of pure exploration has a slope that is worse than OFUL restricted to a subset of the relevant features. The $N = 1000$ actions were randomly sampled from the unit sphere in $d = 40$ dimensions and θ_* was generated with $k = 5$ sparsity.

For a pure exploration algorithm that picks actions uniformly at random, independent of the problem instance, the regret can only be bound by $2SLT$ or O(T). Let $R_{alg}^{\mathcal{K}}$ be the expected regret of algorithm alg run on a subset of relevant features $\mathcal{K} \subseteq \{1, \ldots, k\}, |\mathcal{K}| = j \leq k$. For example, alg could be the OFUL algorithm. Then $R_{alg}^{\mathcal{K}}$ represents the expected regret of OFUL restricted to features in $\mathcal{K}$. To discover relevant features ($\mathcal{K}$) we can employ an explore-then-commit strategy which first explores for $\sim \sqrt{T}$ time followed by an exploitation stage such as OFUL restricted to features in $\mathcal{K}$. The rewards in the latter exploitation stage can be divided in two parts,

$$\langle \mathbf{x}, \boldsymbol{\theta}_* \rangle = \left\langle \mathbf{x}^{\mathcal{K}}, \boldsymbol{\theta}_*^{\mathcal{K}} \right\rangle + \left\langle \mathbf{x}^{\mathcal{K}^c}, \boldsymbol{\theta}_*^{\mathcal{K}^c} \right\rangle,$$

where $\mathbf{x}^{\mathcal{K}}$ is the portion of $\mathbf{x}$ restricted to $\mathcal{K}$ and $\mathcal{K} \cup \mathcal{K}^c = [p]$. Similarly, the regret $R_{alg}^{\mathcal{K}}$ can be divided in two parts. Roughly the regret on $\mathcal{K}$ can be bounded by $j\sqrt{T}$ under certain conditions using the OFUL regret bound. For the regret on $\mathcal{K}^c$, suppose each relevant component of $\boldsymbol{\theta}_*$ has a mean square value of S^2/k (this can be achieved with a sparse gaussian model such as those described in Deshpande and Montanari (2012)). This yields $\mathbb{E}\|\boldsymbol{\theta}_*^{\mathcal{K}^c}\|^2 \approx \frac{k-j}{k} S^2$ where $j = |\mathcal{K}|$. The worst-case instantaneous regret bound on $\mathcal{K}^c$ becomes $2\sqrt{(k-j)/k}SL$ leading to an improvement in the linear regret slope by a factor of $\sqrt{(k-j)/k}$ over pure exploration (see Figure 2.2).



(a) Synthetic data (b) Synthetic data (c) Data from Poulis and Dasgupta (2017)

Figure 2.2: (a) Regret of pure exploration versus explore-then-commit strategy (b,c) Average regret of OFUL restricted to feature subsets (red dots) with 95% confidence regions (blue). The short time horizon was chosen to make the case for restricting the feature space in early rounds. In the long horizon, with more information, the relative performance of OFUL improves, but would ultimately be a factor of d/k worse than that of the low-dimensional model that includes all k relevant features.

Figure 2.2(b) shows average regret of OFUL restricted to feature subsets of different sizes with synthetic data ($N = 1000$ actions, $d = 40$ and $k = 10$). For $j \in \{2, 4, \ldots, 10\}$, we randomly picked 100 subsets of size j from the support of $\boldsymbol{\theta}_*$. We report the average regret of OFUL for a short horizon, $T = 2^8$, restricted to the 100 random subsets. We also plot average regret of OFUL on the full $d = 40$ dimensional data. Figure 2.2(c) depicts the same with real data from Poulis and Dasgupta (2017) with $d = 498$ and sparsity, $k = 92$, we choose 100 random subsets of size $j \in \{5, 10, \ldots, 25\}$ from the set of relevant features marked by users (see Section 3.5 for details) and report the average regret of OFUL restricted to the feature subsets for a relatively short time horizon, $T = 2^{11}$. The plots show that, in the short horizon, it may be more beneficial to use a subset of the relevant features than using the total feature set which may include many irrelevant

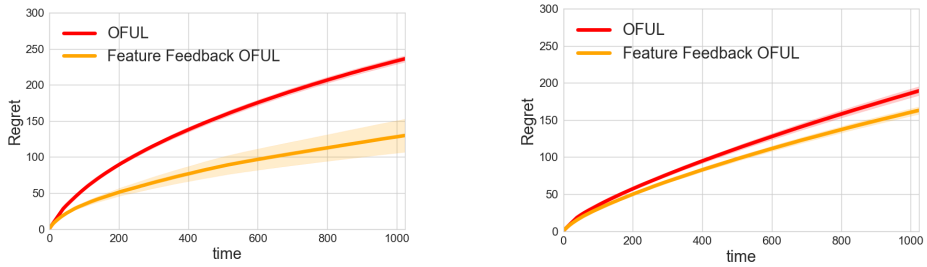
features. The intuition is that when OFUL has not seen many samples, it does not have enough information to separate the irrelevant dimensions from relevant ones. As time goes on (i.e., for longer horizons) OFUL's relative performance improves since it enjoys sublinear regret but would ultimately be a factor of d/k worse than that of the low-dimensional model that includes all k relevant features.

2.5 Experiments

Results with Synthetic Data

For synthetic data, we simulate a text categorization dataset as follows. Each action corresponds to an article. Generally an article contains only a small subset of the words from the dictionary. Therefore, to simulate documents we generate 1000 sparse actions in 40 dimensions. A 5-sparse reward generating vector, $\boldsymbol{\theta}_*$, is chosen at random. This represents the fact that in reality a document category probably contains only a few relevant words. The features represent word counts and hence are always positive. Here we have access to $\boldsymbol{\theta}_*$ therefore for any action $\mathbf{x}$, we use the standard linear model (2.1) for the reward y_t with $\eta_t \sim \mathcal{N}(0, R^2)$. The support of $\boldsymbol{\theta}_*$ is taken as the set of oracle relevant words. For every round, each word from the intersection of the support of the action and oracle relevant words is marked as relevant with probability $(p' = 0.1)$. Figure 2.4(a) shows the results averaged over 100 random trials for sparse $\boldsymbol{\theta}_*$ with

$k = 5, d = 40$, and 1000 actions. As expected, the FF-OFUL algorithm outperforms standard OFUL significantly. Figure 2.3(b) also shows that the feedback does not hurt the performance much for non-sparse $\boldsymbol{\theta}_*$ with $k = d = 40$. Figure 2.1(b) compares the performance of FF-OFUL with an explore-then-commit strategy.



(a) Synthetic data with sparse $\boldsymbol{\theta}_*$. (b) Simulated data with dense $\boldsymbol{\theta}_*$.

Figure 2.3: On simulated data ($d = 40$) **(a)** sparse $\boldsymbol{\theta}_*$ ($k = 5$), FF-OFUL outperforms OFUL significantly and **(b)** dense $\boldsymbol{\theta}_*$ ($k = d = 40$), Feature Feedback does not hurt the performance and it is close to standard OFUL. Refer introduction for explore-then-commit comparison on synthetic data.

Results with 20Newsgroup Dataset

We use the 20Newsgroup (20NG) dataset from Lang (1995). It has 2×10^5 documents covering 20 topics such as politics, sports. We choose a subset of 5 topics (misc.forsale, rec.autos, sci.med, comp.graphics, talk.politics.mideast) with approximately 4800 documents posted under these topics. For the word counts, we use the TF-IDF features for the documents which give us approximately $d = 47781$ features. For the sake of comparing our method with OFUL, we first report 500 and 1000 dimensional experiments and

then on the full $47,781$ dimensional data. To do this, we use logistic regression to train a high accuracy sparse classifier to select 153 features. Then select an additional 847 features at random in order to simulate high dimensional features. We compared OFUL and FF-OFUL algorithms on this data. This is similar to the way Poulis and Dasgupta (2017) ran experiments in the classification setting. We ran only our algorithm on the full 47781 dimension data since it was infeasible to run OFUL. For the reward model, we pick one of the articles from the database at random as $\boldsymbol{\theta}_*$ and the linear reward model in (2.1) or use the labels to generate binary, one vs many rewards to simulate search for articles from a certain category. In order to come close to simulating a noisy setting, we used the logistic model, with $q_t = 1/(1 - \exp(-\langle \mathbf{x}_t, \boldsymbol{\theta}_* \rangle), P(y_t = +1) = q_t$.

Oracle Feedback

The support of one-vs-many sparse logistic regression is used to get an "oracle set of relevant features" for each class. Each word from the intersection of the support of an action and oracle relevant words was marked as relevant with probability $p'(= 0.1)$. In our theorem statements, p is the probability that the feature is present in a random action *and* it is marked relevant. This depends on the distribution of the words, but typically $\boldsymbol{p \in (0.001, 0.01)}$ and $k \in (30, 100)$ relevant features for each category. Figure 2.4, compares OFUL, Explore-then-commit and FF-OFUL on the 20NG dataset with oracle feedback. In these simulations averaged over

100 random $\boldsymbol{\theta}_*$, FF-OFUL outperforms OFUL and Explore-then-commit significantly. OFUL parameter was tuned to $\lambda = 2^8$.

Human Feedback

Poulis and Dasgupta (2017) took 50 20Newsgroup articles from 5 categories and had users annotate relevant words. These are the same categories that we used above. This is closer to simulating human feedback since we are not using sparse logistic regression to estimate the sparse vectors. We take the user indicated relevant words instead as the relevance dimensions. There were $k \in (30, 100)$ relevant features for each category. In Figure 2.5(a), we can see that FF-OFUL is already outperforming OFUL and Explore-then-commit. This is despite the fact that it is not a very sparse regime. Surprisingly, we found that tuning had little effect on the performance of FF-OFUL whereas it had a significant effect on OFUL (see Figure 2.5). This is possibly due to the implicit regularization provided by gradually growing the number of dimensions as we receive new feedback. FF-OFUL also yields significantly better rewards at early stages by exploiting knowledge of relevant features as soon as they are identified, rather than waiting until all or most are found.

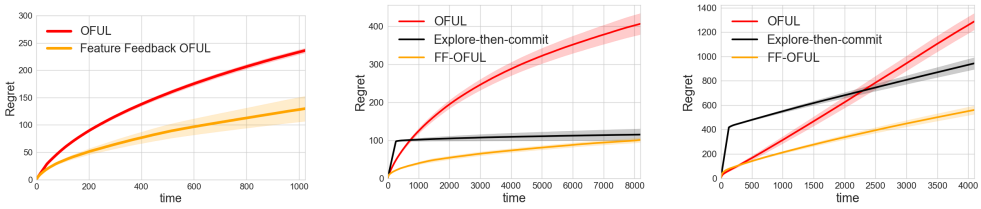
Parameter Tuning

For OFUL the ridge parameter (λ) is tuned from $\{2^i\}_{i=-7}^{10}$ to pick the one with best performance. All the tuned parameters selected for OFUL were

strictly inside this range (for $d = 40, k = 5$, $\lambda = 2^{-5}$ and for $d = 10^3$ (Newsgroup), $\lambda = 2^8$). Figure 2.5(b) demonstrates the sensitivity of OFUL to change in tuning parameter. For FF-OFUL, the remarkable feature is that it does not require parameter tuning so $\lambda = 1$ for all experiments.

Full dimension experiments

Remarkably the performance of FF-OFUL barely drops in full ($d = 47781$) feature dimensions, see Figure 2.5(c). Even though the ridge regression parameter (λ) for all the experiments was not tuned and set to $\lambda = 1$. FF-OFUL is robust to changes in the ambient dimensions and the parameter λ. Recall that we do not compare the results with OFUL on 47781 dimensional data since it would require storing and updating a $d \times d$ matrix at each stage.



(a) Synthetic data (b) Real data with replacement (c) Real data without replacement

Figure 2.4: (a) Synthetic data with sparse $\boldsymbol{\theta}_*$ ($d = 40$, $k = 5$), FF-OFUL outperforms OFUL significantly. See Figure 2.1(b) for comparison with explore-then-commit strategy. Newsgroup dataset with oracle feedback: **(b)** This plot shows that FF-OFUL outperforms OFUL and Explore-then-commit when running in $d = 1000$ dimensions, sampling actions with replacement using binary rewards model. **(c)** sampling actions without replacement and using the numerical reward model. Smallest T_0 selected such that all relevant features are marked with high probability. Note shorter time horizon for without replacement sampling since T must be less than the number of actions.

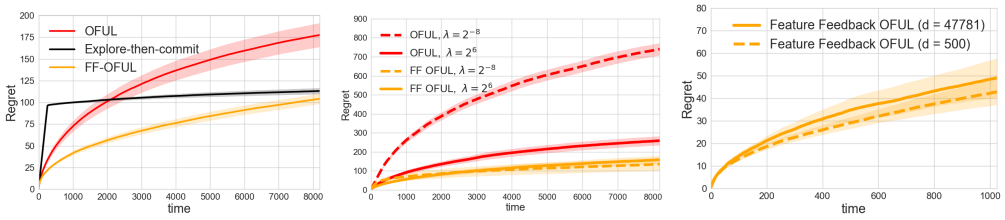


Figure 2.5: Newsgroup Dataset with Human Feedback: **(Left)** FF-OFUL outperforms OFUL and Explore-then-commit strategy in $d = 500$ dimensions. Both plots generated by tuning the parameter for OFUL. **(Center)** Sensitivity to tuning parameter λ seen by the drastic difference in performance of OFUL. In contrast, our FF-OFUL has a relatively modest difference in performance showing its robustness to the ridge regression parameter λ. **(Right)** Our algorithm for $d = 47781$ and $d = 500$ with ridge parameter $\lambda = 1$, showing its robustness to changes in dimensions and tuning.

3 MATRIX APPROXIMATION USING GROUPS OF COLUMNS

3.1 Introduction

The ability to perform large-scale data analysis is often limited by two opposing forces. The first force is the need to store data in a matrix format for the purpose of analysis techniques such as regression or classification. The second force is the inability to store the data matrix completely in memory due to the size of the matrix in many application settings. This conflict gives rise to storing factorized matrix forms, such as SVD or CUR decompositions Drineas et al. (2008).

We consider a matrix $\boldsymbol{A}$ with m rows and n columns, *i.e.*, $\boldsymbol{A} \in \mathbb{R}^{m \times n}$. Using a truncated k number of singular vectors (*e.g.*, where $k < \min\{m, n\}$), the singular value decomposition (SVD) provides the best rank-k approximation to the original matrix. The singular vectors often do not preserve the structure in original data. Preserving the original structure in the data may be desirable due to many reasons including interpret-ability in case of biometric data or for storage efficiency in case of sparse matrices. This has led to the introduction of the CUR decomposition, where the factorization is performed with respect to a subset of rows and columns of the matrix itself. This specific decomposition describes the matrix $\boldsymbol{A}$ as the product of a subset of matrix rows $\boldsymbol{R}$ and a subset of matrix columns $\boldsymbol{C}$ (along

with a matrix $\boldsymbol{U}$ that fits $\boldsymbol{A} \approx \boldsymbol{CUR}$).

Significant prior work has examined how to efficiently choose the rows and columns in the CUR decomposition and has derived worst-case error bounds (*e.g.,* Mahoney and Drineas (2009)). These methods have been applied successfully to many real-world problems including genetics Paschou et al. (2007), astronomy Yip et al. (2014), and mass spectrometry imaging Yang et al. (2015). Unfortunately, a primary assumption of current CUR techniques, that individual rows and columns of the matrix can be queried, is either impossible or quite costly in many real world problems and instead require a block approach.

In this chapter, we consider the following two applications which represent the two main motivating factors for considering block decompositions.

Biometric data analysis. In applications where the ordering of rows or columns is meaningful, such as images, video, or speech data matrices, sampling contiguous blocks of columns adds contextual information that is necessary for interpretability of the factorized representation. One emerging application is audience reaction analysis of video content using biometrics. We focus on the scenario where users watch video content while wearing sensors, and changes in biometric sensors indicate changes in reaction to the content. For example, increases in heart rate or a spike in electrodermal activity indicate an increase in content engagement. In this chapter, a matrix of biometric data such as *Electrodermal Activity* (EDA) is collected from users reacting to external stimuli, e.g., watching video

content. In prior work, EDA has shown to be useful for a variety of user analytics tasks to assess the reaction of viewersJain et al. (2017); Silveira et al. (2013). In this setting, m is the number of users and n corresponds to the number of time samples for which biometric reaction is collected. Unfortunately, there is significant cost in acquiring each user's reaction to lengthy content so instead we collect full responses (corresponding to some rows of the matrix) from only a limited number of users. For remaining users, we propose to collect responses for only a few important scenes of the video (corresponding to column blocks of the matrix) as shown in Figure 3.1 and then *approximate* their full response. An individual time sample in this use case cannot be queried in isolation due to the lack of context that caused that biometric reaction. Instead, collections of time segments (i.e., blocks) must be presented to the user. In this setting block sampling can be viewed as a **restriction** which leads to more interpretable solutions.

Distributed storage systems. Large-scale datasets often require distributed storage, a regime where there can be substantial overhead involved in querying individual rows or columns of a matrix. In these regimes, it is more efficient to retrieve predefined *blocks* of rows or columns at one time corresponding to the rows or columns stored on the same node, as shown in Figure 3.1, in order to minimize the overhead in terms of latency while keeping the throughput constant. In doing so, one forms a Block CUR decomposition, with more details provided in Section 3.5.

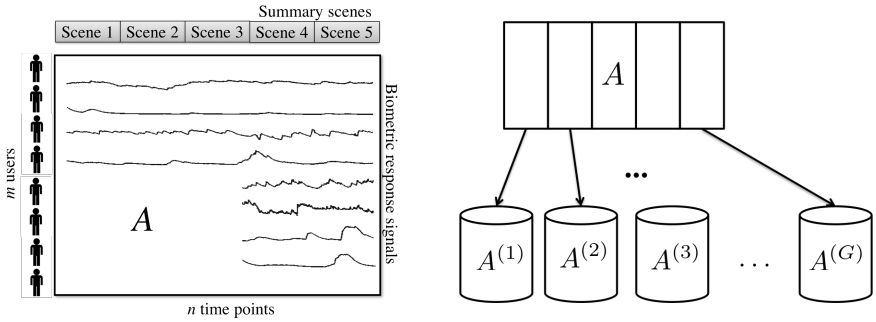


Figure 3.1: Applications: (a) Biometric data analysis. Blocks of columns or time instances correspond to scenes in a video and provide context for biometric reaction. (b) Distributed storage of a large matrix across multiple nodes in a cluster. Blocks are allocated to each of the G nodes.

Current CUR decomposition techniques do not take advantage of this predefined block structure.

Main contributions. Using these insights into real-world applications of CUR decomposition, we make a series of contributions. We propose a simple randomized Block CUR algorithm for subset selection of rows and blocks of columns and derive novel worst-case error bounds for this randomized algorithm Oswal et al. (2018). On the theory side, we present new theoretical results related to approximating matrix multiplication and generalized ℓ_2 regression in the block setting. These results are the fundamental building blocks used to derive the error bounds for the presented randomized algorithms. The sample complexity bounds feature a non-trivial dependence on the matrix partition, $i.e.$, the distribution of information in the blocks of the matrix. This dependence is non-trivial in that it cannot be obtained by simply extending the analysis of the original

individual column CUR setting to the Block CUR setting. As a result, our analysis finds a sample complexity improvement on the order of the *block stable rank* of a matrix (See Table 3.2 in Section 3.3).

On the practical side, this algorithm performs fast block sampling taking advantage of the natural storage of matrices in distributed environments (See Table 3.1 in Section 3.5). We demonstrate empirically that the proposed Block CUR algorithms can achieve a significant speed-up when used to decompose large matrices in a distributed data setting. We conduct a series of CUR decomposition experiments using Apache Spark on Amazon Elastic Map-Reduce (Amazon EMR) using both synthetic and real-world data. In this distributed environment, we find that our Block CUR approach achieves a speed-up of 2x to 6x for matrices larger than 12000×12000. This is compared with previous CUR approaches that sample individual rows and columns and while achieving the same matrix approximation error rate. We also perform experiments with real-world user biometric data from a content testing environment and present interesting use cases where our algorithms can be applied to user analytics tasks.

3.2 Problem Setting

Notation

Let $\boldsymbol{I}_k$ denote the $k \times k$ identity matrix and $\mathbf{0}$ denote a zero matrix of appropriate size. We denote vectors (matrices) with lowercase (uppercase) bold symbols like $\boldsymbol{a}$ ($\boldsymbol{A}$). The i-th row (column) of a matrix is denoted by $\boldsymbol{A}_i$ ($\boldsymbol{A}^i$). We represent the i-th block of rows of a matrix by $\boldsymbol{A}_{(i)}$ and the i-th block of columns of a matrix by $\boldsymbol{A}^{(i)}$.

Let $[n]$ denote the set $\{1, 2, \ldots, n\}$. Let $\rho = \text{rank}(\boldsymbol{A}) \leq \min\{m, n\}$ and $k \leq \rho$. The singular value decomposition (SVD) of $\boldsymbol{A}$ can be written as $\boldsymbol{A} = \boldsymbol{U}_{A,\rho}\boldsymbol{\Sigma}_{A,\rho}\boldsymbol{V}_{A,\rho}^T$ where $\boldsymbol{U}_{A,\rho} \in \mathbb{R}^{m\times\rho}$ contains the ρ left singular vectors; $\boldsymbol{\Sigma}_{A,\rho} \in \mathbb{R}^{\rho\times\rho}$ is the diagonal matrix of singular values, $\sigma_i(\boldsymbol{A})$ for $i = 1, \ldots, \rho$; and $\boldsymbol{V}_{A,\rho}^T \in \mathbb{R}^{\rho\times n}$ is an orthonormal matrix containing the ρ right singular vectors of $\boldsymbol{A}$. We denote $\boldsymbol{A}_k = \boldsymbol{U}_{A,k}\boldsymbol{\Sigma}_{A,k}\boldsymbol{V}_{A,k}^T$ as the best rank-k approximation to $\boldsymbol{A}$ in terms of Frobenius norm. The pseudoinverse of $\boldsymbol{A}$ is defined as $\boldsymbol{A}^\dagger = \boldsymbol{V}_{A,\rho}\boldsymbol{\Sigma}_{A,\rho}^{-1}\boldsymbol{U}_{A,\rho}^T$. Also, note that $\boldsymbol{C}\boldsymbol{C}^\dagger\boldsymbol{A} = \boldsymbol{U}_C\boldsymbol{U}_C^T\boldsymbol{A}$ is the projection of $\boldsymbol{A}$ onto the column space of $\boldsymbol{C}$, and $\boldsymbol{A}\boldsymbol{R}^\dagger\boldsymbol{R} = \boldsymbol{A}\boldsymbol{V}_{R,k}\boldsymbol{V}_{R,k}^T$ is the projection of $\boldsymbol{A}$ onto the row space of $\boldsymbol{R}$.

The Frobenius norm and spectral norm of a matrix are denoted by $\|\boldsymbol{A}\|_F$ and $\|\boldsymbol{A}\|_2$ respectively. The square of the Frobenius norm is given by $\|\boldsymbol{A}\|_F^2 = \sum_{i=1}^m \sum_{j=1}^n A_{i,j}^2 = \sum_{i=1}^k \sigma_i^2(\boldsymbol{A})$. The spectral norm is given by $\|\boldsymbol{A}\|_2 = \max_i \sigma_i(\boldsymbol{A})$.

The CUR problem and other related work

The need to factorize a matrix using a collection of rows and columns of that matrix has motivated the CUR decomposition literature. CUR decomposition is focused on sampling rows and columns of the matrix to provide a factorization that is close to the best rank-k approximation of the matrix. One of the most fundamental results for a CUR decomposition of a given matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ was obtained in Drineas et al. (2008). We re-state it here for the sake of completion and setting the appropriate context for our results to be stated in the next section. This relative error bound result is summarized in the following theorem.

Theorem 3.1. *(Theorem 2 from Drineas et al. (2008) applied to $\boldsymbol{A}^T$) Given $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ and an integer $k \leq \min\{m, n\}$, let $r = O(\frac{k^2}{\varepsilon^2}\ln(\frac{1}{\delta}))$ and $c = O(\frac{r^2}{\varepsilon^2}\ln(\frac{1}{\delta}))$. There exist randomized algorithms such that, if c columns are chosen to construct $\boldsymbol{C}$ and r rows are chosen to construct $\boldsymbol{R}$, then with probability $\geq 1 - \delta$, the following holds:*

$$\|\boldsymbol{A} - \boldsymbol{CUR}\|_F \leq (1 + \varepsilon)\|\boldsymbol{A} - \boldsymbol{A}_k\|_F$$

where $\varepsilon, \delta \in (0, 1)$, $\boldsymbol{U} = \boldsymbol{W}^\dagger$ and $\boldsymbol{W}$ is the scaled intersection of $\boldsymbol{C}$ and $\boldsymbol{R}$.

This theorem states that as long as enough rows and columns of the matrix are acquired (r and c, respectively), then the CUR decomposition will be within a constant factor of the error associated with the best rank-

k approximation of that matrix. Central to the proposed randomized algorithm was the concept of sampling columns of the matrix based on a *leverage score*. The leverage score measures the contribution of each column to the approximation of $\boldsymbol{A}$.

Definition 3.2. *The **leverage score** of a column is defined as the squared row norm of the top-k right singular vectors of $\boldsymbol{A}$ corresponding to the column:*

$$\ell_j = \|\boldsymbol{V}_{A,k}^T \boldsymbol{e}_j\|_2^2,\ j \in [n],$$

where $\boldsymbol{V}_{A,k}$ consists of the top-k right singular vectors of $\boldsymbol{A}$ as its rows, and $\boldsymbol{e}_j$ is the j-th column of identity matrix which picks the j-th column of $\boldsymbol{V}_{A,k}^T$.

The CUR algorithm involves randomly sampling r rows using probabilities generated by the calculated leverage scores to obtain the matrix $\boldsymbol{R}$, and thereafter sampling c columns of $\boldsymbol{A}$ based on leverage scores of the $\boldsymbol{R}$ matrix to obtain $\boldsymbol{C}$. The key technical insight in Drineas et al. (2008) is that the leverage score of a column measures "how much" of the column lies in the subspace spanned by the top-k left singular vectors of $\boldsymbol{A}$; therefore, this method of samping is also known as *subspace* sampling. By sampling columns that lie in this subspace more often, we get a relative-error low rank approximation of the matrix. The concept of sampling the important columns of a matrix based on the notion of subspace sampling first appeared in context of fast ℓ_2 regression in Drineas et al. (2006) and was

refined in Drineas et al. (2008) to obtain performance error guarantees for CUR matrix decomposition.

These guarantees were subsequently improved in follow-up work Mahoney and Drineas (2009). Modified versions of this problem have been studied extensively for adaptive sampling Wang and Zhang (2012), divide-and-conquer algorithms for parallel computations Mackey et al. (2011), and input-sparsity algorithms Boutsidis and Woodruff (2014). The authors of Wang and Zhang (2012) propose an adaptive sampling-based algorithm which requires only $c = O(k/\varepsilon)$ columns to be sampled when the entire matrix is known and its SVD can be computed. The authors of Boutsidis and Woodruff (2014) also proposed an optimal, deterministic CUR algorithm. In Boutsidis et al. (2014), the authors prove the lower bound of the column selection problem; at least $c = k/\varepsilon$ columns are selected to achieve the $(1 + \varepsilon)$ ratio.

These prior results require sampling of arbitrary rows and columns of the matrix $\boldsymbol{A}$ which may be either unrealistic or inefficient in many practical applications. In this chapter, we focus on the problem of efficiently sampling pre-defined blocks of columns (or rows) of the matrix to provide a factorization that is close to the best rank-k approximation of the matrix in the more natural environment of block sampling for biometric and distributed computation, explore the performance advantages of block sampling over individual column sampling, and provide the first non-trivial theoretical error guarantees for Block CUR decomposition. In the

following section, we propose and analyze a randomized algorithm for sampling blocks of the matrix based on *block leverage scores*.

3.3 Algorithm

A block may be defined as a collection of s columns or rows. For clarity of exposition, without loss of generality, we consider column blocks but the techniques and derivations also hold for row blocks by applying them to the transpose of the matrix. For ease of exposition, we also assume equal-sized blocks but one could easily extend the methods to blocks of varying sizes. Let $G = \lceil n/s \rceil$ be the number of possible blocks in $\boldsymbol{A}$. We consider the blocks to be predefined due to natural constraints or cost, such as data partitioning in a distributed compute cluster.

The goal of the Block CUR algorithm is to approximate the underlying matrix $\boldsymbol{A}$ using g blocks of columns and r rows, as represented in Figure 3.2. For example, in the biometric analysis setting each block could correspond to user reactions at a collection of time points corresponding to a scene in a movie. The goal is to approximate the users' reactions to the full movie using only their response to a summary of the movie (containing a subset of the scenes).

Given the new regime of submatrix blocks, we begin by defining a *block leverage score* for each block of columns.

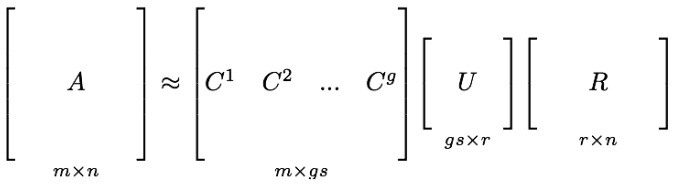


Figure 3.2: Example Block CUR decomposition, where $\boldsymbol{C}^t \in \mathbb{R}^{m \times s}$ for $t \in [g]$ is sampled from $\{\boldsymbol{A}^{(j_t)} : j_t \in [G]\}$.

Definition 3.3. *The **block leverage score** of a group of columns is defined as the sum of the squared row norms of the top-k right singular vectors of $\boldsymbol{A}$ corresponding to the columns in the block:*

$$\ell_g(\boldsymbol{A}, k) = \|\boldsymbol{V}_{A,k}^T \boldsymbol{E}_g\|_F^2, \; g \in [G],$$

where $\boldsymbol{V}_{A,k}$ consists of the top-k right singular vectors of $\boldsymbol{A}$, and $\boldsymbol{E}_g$ consists of the corresponding block of columns in the identity matrix which picks the columns of $\boldsymbol{V}_{A,k}^T$ corresponding to the elements in block g.

Much like the individual column leverage scores defined in Drineas et al. (2008), the block leverage scores measure how much a particular column block contributes to the approximation of the matrix $\boldsymbol{A}$.

Algorithm details

The Block CUR Algorithm, detailed in Algorithm 1, takes as input the matrix $\boldsymbol{A}$ and returns as output an $r \times n$ matrix $\boldsymbol{R}$ consisting of a small number of rows of $\boldsymbol{A}$ and an $m \times c$ matrix $\boldsymbol{C}$ consisting of a small number of column blocks from $\boldsymbol{A}$.

Algorithm 1: Block CUR

Input : $\boldsymbol{A}$, target rank k, size of each block s, error parameter ε, positive integers r, g

Output: $\boldsymbol{C}, \boldsymbol{R}, \widehat{\boldsymbol{A}} = \boldsymbol{CUR}$

1. *Row subset selection:* Sample r rows uniformly from $\boldsymbol{A}$ according to $p_i = 1/m$ for $i \in [m]$ and compute $\boldsymbol{R} = \boldsymbol{S}_R^T \boldsymbol{A}$.

2. *Column block subset selection:* For $t \in [g]$, select a block of columns $j_t \in [G]$ independently with probability $p_{j_t} = \frac{\ell_i(\boldsymbol{R},r)}{r} = \frac{\|\boldsymbol{V}_{R,r}^T \boldsymbol{E}_i\|_F^2}{r}$ for $i \in [G]$ and update $\boldsymbol{S}$, where $\boldsymbol{V}_{R,r}$ consists of the top-r right singular
vectors of $\boldsymbol{R}$, and $\boldsymbol{E}_i$ picks the columns $\boldsymbol{V}_{R,r}^T$ corresponding to the elements in block i. Compute $\boldsymbol{C} = \boldsymbol{AS}$.

3. *CUR approximation*: $\widehat{\boldsymbol{A}} = \boldsymbol{CUR}$ where $\boldsymbol{U} = \boldsymbol{W}^\dagger$, and $\boldsymbol{W} = \boldsymbol{RS}$ is the
scaled intersection of $\boldsymbol{R}$ and $\boldsymbol{C}$.

In Algorithm 1, for $t \in [g]$, block $j_t \in [G]$ is sampled with some probability p_{j_t} and scaled using matrix $\boldsymbol{S} \in \mathbb{R}^{n \times gs}$. The (j_t, t)-th non-zero $s \times s$ block of $\boldsymbol{S}$ is defined as $\boldsymbol{S}_{j_t,t} = \boldsymbol{I}_s / \sqrt{g p_{j_t}}$ where $g = c/s$ is the number of blocks picked by the algorithm. This sampling matrix picks the blocks of columns and scales each block to compute $\boldsymbol{C} = \boldsymbol{AS}$. A similar sampling and scaling matrix $\boldsymbol{S}_R$ is defined to pick the blocks of rows and scale each block to compute $\boldsymbol{R} = \boldsymbol{S}_R^T \boldsymbol{A}$. An example of sampling matrix $\boldsymbol{S}$ with blocks chosen in order $[1, 3, 2]$ is as follows:

$$\boldsymbol{S}_{n\times gs} = \begin{bmatrix} \frac{1}{\sqrt{gp_1}}\boldsymbol{I}_s & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \frac{1}{\sqrt{gp_2}}\boldsymbol{I}_s \\ \mathbf{0} & \frac{1}{\sqrt{gp_3}}\boldsymbol{I}_s & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}.$$

In addition to considering block sampling of columns, another advantage of this algorithm is not requiring the computation of a full SVD of $\boldsymbol{A}$. In many large-scale applications, it may not feasible to compute the SVD of the entire matrix $\boldsymbol{A}$. In these cases, algorithms requiring knowledge of the leverage scores cannot be used. Instead, we use an estimate of the block leverage scores called the *approximate block leverage scores*. A subset of the rows (corresponding to users) are chosen uniformly at random, and the block scores are calculated using the top-k right singular vectors of this row matrix instead of the entire $\boldsymbol{A}$ matrix. This step is not the focus of the experiments in this chapter so it can also be replaced with other fast approximate calculations of leverage scores involving sketching or additional sampling Drineas et al. (2012); Xu et al. (2015). The advantage of using our approximate leverage scores is that the same set of rows is used to approximate the scores and also to compute the CUR approximation. Hence no additional sampling or sketching steps are required. In terms of the biometric application, each row corresponds to a user's biometric reaction to a movie. Since collecting user reactions to lengthy content can be expensive, eliminating redundant sampling leads to huge savings in

resources.

The running time of Algorithm 1 is essentially driven by the time required to compute the SVD of $\boldsymbol{R}$, *i.e.*, $\mathcal{O}(SVD(\boldsymbol{R}))$ time, and the time to construct $\boldsymbol{R}$, $\boldsymbol{C}$ and $\boldsymbol{U}$. Construction of $\boldsymbol{R}$ requires $\mathcal{O}(rn)$ time, construction of $\boldsymbol{C}$ takes $\mathcal{O}(mc)$ time, construction of $\boldsymbol{W}$ requires $\mathcal{O}(rc)$ time and construction of $\boldsymbol{U}$ takes $\mathcal{O}(r^2c)$ time.

3.4 Main Results

The main technical contribution is a novel relative-error bound on the quality of approximation using blocks of columns or rows to approximate a matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$. Before stating the main result, we define two important quantities that measure important properties of the matrix $\boldsymbol{A}$ that are fundamental to the quality of approximation. We first define a property of matrix rank relative to the collection of matrix blocks. Specifically, we focus on the concept of *matrix stable rank* from Rudelson and Vershynin (2007) and define the *block stable rank* as the minimum stable rank across all matrix blocks.

Definition 3.4. *Let $\boldsymbol{V}_{A,k}$ consist of the top-k right singular vectors of $\boldsymbol{A}$. Then the **block stable rank** is defined as*

$$\alpha_A = \min_{g \in [G]} \frac{\|\boldsymbol{V}_{A,k}^T \boldsymbol{E}_g\|_F^2}{\|\boldsymbol{V}_{A,k}^T \boldsymbol{E}_g\|_2^2},$$

where $\boldsymbol{E}_g$ consists of the corresponding block of columns in the identity matrix that picks the columns of $\boldsymbol{V}_{A,k}^T$ corresponding to the elements in block g.

Intuitively, the above definition gives a measure of how informative the worst matrix column block is. The second property is a notion of *column space incoherence*. When we sample rows uniformly at random, we can give relative error approximation guarantees when the matrix $\boldsymbol{A}$ satisfies an incoherence condition. This avoids pathological constructions of rows of $\boldsymbol{A}$ that cannot be sampled at random.

Definition 3.5. *The top-k* ***column space incoherence*** *is defined as*

$$\mu := \mu(\boldsymbol{U}_{A,k}^T) = \frac{m}{k} \max_i \|\boldsymbol{U}_{A,k}^T \boldsymbol{e}_i\|_2^2,$$

where $\boldsymbol{e}_i$ picks the i-th column of $\boldsymbol{U}_{A,k}^T$.

The column space incoherence is used to provide a guarantee for fast approximation without computing the SVD of the entire matrix $\boldsymbol{A}$. Equipped with these definitions, we state the main result that provides a relative-error guarantee for the Block CUR approximation in Theorem 3.6.

Theorem 3.6. *Given $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ with incoherent top-k column space, i.e., $\mu \leq \mu_0$, let $r = O\left(\mu_0 \frac{k^2}{\varepsilon^2} \ln(\frac{1}{\delta})\right)$ and $g = O\left(\frac{r^2}{\alpha_R \varepsilon^2} \ln(\frac{1}{\delta})\right)$. There exist randomized algorithms such that, if r rows and g column blocks are chosen to construct $\boldsymbol{R}$*

and $\boldsymbol{C}$, respectively, then with probability $\geq 1-\delta$, the following holds:

$$\|\boldsymbol{A}-\boldsymbol{CUR}\|_F \leq (1+\varepsilon)\|\boldsymbol{A}-\boldsymbol{A}_k\|_F,$$

where $\varepsilon, \delta \in (0,1)$ and $\boldsymbol{U}=\boldsymbol{W}^{\dagger}$ is the pseudoinverse of scaled intersection of $\boldsymbol{C}$ and $\boldsymbol{R}$.

We provide a sketch of the proof and highlight the main technical challenges in proving the claim in Section 3.4 and defer the proof details to the Appendix. In Section 3.4, we first provide a relative-error guarantee (Lemma 3.8) for the approximation provided by Algorithm 1. After applying standard boosting techniques (explained in Section 3.4) we get the main result stated above.

Proof sketch of main result

In this section, we provide a sketch of the proof of Theorem 3.6 and defer the details to the Appendix. The proof of the main result rests on two important lemmas. These results are important in their own right and could be useful wherever the block sampling issue arises. The first result concerns approximate block multiplication.

Block multiplication lemma. The following lemma shows that the multiplication of two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ can be approximated by the product of the smaller sampled and scaled block matrices. This is the key lemma in proving the main result.

Lemma 3.7. *Let* $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, $\boldsymbol{B} \in \mathbb{R}^{n\times p}$,$\varepsilon, \delta \in (0,1)$, *and* α_A *be defined as* $\alpha_A := \min_{i\in[G]} \frac{\|\boldsymbol{A}^{(i)}\|_F^2}{\|\boldsymbol{A}^{(i)}\|_2^2}$. *Construct* $\boldsymbol{C}_{m\times gs}$ *and* $\boldsymbol{R}_{gs\times n}$ *using sampling probabilities* p_i *that satisfy*

$$p_i \geq \beta \frac{\|\boldsymbol{A}^{(i)}\|_F^2}{\sum_{j=1}^{G} \|\boldsymbol{A}^{(j)}\|_F^2},$$

for all $i \in [G]$ *and where* $\beta \in (0,1]$. *Then, with probability at least* $1-\delta$,

$$\|\boldsymbol{AB} - \boldsymbol{CR}\|_F \leq \frac{1}{\delta\sqrt{\beta g \alpha_A}} \|\boldsymbol{A}\|_F \|\boldsymbol{B}\|_F.$$

The proof details are provided in the Appendix. The main difficulty in proving this claim is to account for the block structure. Even though one could trivially extend individual column sampling analysis to this setting by serializing the blocks, this would lead to trivial bounds as they do not leverage the block structure. Our results exploit this knowledge and hence introduce a dependence of the sample complexity on the block stable rank of the matrix.

Using the block multiplication lemma we prove Lemma 3.8, which states a non-boosting approximation error result for Algorithm 1.

Lemma 3.8. *Given* $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ *with incoherent top-*k *column space, i.e.,* $\mu \leq \mu_0$, *let* $r = O(\mu_0 \frac{k^2}{\varepsilon^2})$ *and* $g = O(\frac{r^2}{\alpha_R \varepsilon^2})$. *If rows and column blocks are chosen according to Algorithm 1, then with probability at least 0.7, the following holds:*

$$\|\boldsymbol{A} - \boldsymbol{CUR}\|_F \leq (1+\varepsilon)\|\boldsymbol{A} - \boldsymbol{A}_k\|_F,$$

where $\varepsilon \in (0,1)$, $\boldsymbol{U} = \boldsymbol{W}^{\dagger}$ *is the pseudoinverse of the scaled intersection of* $\boldsymbol{C}$ *and* $\boldsymbol{R}$*.*

The proof of Lemma 3.8 follows standard techniques in Drineas et al. (2008) with modifications necessary for block sampling (see Appendix for the proof details). Finally, the result in Theorem 3.6 follows by applying standard boosting methods to Lemma 3.8 and running Algorithm 1 $t = \ln(\frac{1}{\delta})$ times. By choosing the solution with minimum error and observing that $0.3 < 1/e$, we have that the relative error bound holds with probability greater than $1 - e^{-t} = 1 - \delta$.

Remark. As a consequence of Lemma 3.8, we show that if enough blocks are sampled with high probability, then $\|\boldsymbol{A} - \boldsymbol{AS}(\boldsymbol{RS})^{\dagger}\boldsymbol{R}\|_F \leq (1+\varepsilon)\|\boldsymbol{A} - \boldsymbol{AR}^{\dagger}\boldsymbol{R}\|_F$. This gives a guarantee on the approximate solution obtained by solving a block-sampled regression problem $\min_{\boldsymbol{X}\in\mathbb{R}^{m\times r}} \|(\boldsymbol{AS}) - \boldsymbol{X}(\boldsymbol{RS})\|_F$ instead of the entire least squares problem. As a special case of the above result, when $\boldsymbol{R} = \boldsymbol{A}$ we get a bound for the block column subset selection problem. If $g = \mathcal{O}(\frac{k^2}{\alpha_A \varepsilon^2} \log\left(\frac{1}{\delta}\right))$ blocks are chosen, then with probability at least $1-\delta$ we have $\|\boldsymbol{A} - \boldsymbol{CC}^{\dagger}\boldsymbol{A}\|_F \leq (1+\varepsilon)\|\boldsymbol{A} - \boldsymbol{A}_k\|_F$.

3.5 Experiments

Experiments with biometric data

One emerging application is audience reaction analysis of video content using biometrics. Specifically, users watch video content while wearing sensors, with changes in biometric sensors indicating changes in reaction to the content. For example, increases in heart rate or a spike in electrodermal activity indicate an increase in content engagement. In prior work, biometric signal analysis techniques have been developed to determine valence Silveira et al. (2013) (*e.g.*, positive vs. negative reactions to films) and content segmentation Lian et al. (2014). Unfortunately these experiments require a large number of users to sit through the entire video content, which can be both costly and time-consuming.

We consider the observed biometric signals as a matrix with m users (as rows) and n biometric time samples (as columns). Matrix approximation techniques, such as CUR decomposition, point to the ability to infer the complete matrix by showing the entire content to only a subset of users (*i.e.*, rows), while the remaining users see only selected scenes of the content (*i.e.*, column blocks). To replicate a user's true reaction to content, individual columns cannot be sampled (*e.g.*, showing the user 0.25 seconds of video content) given the lack of scene context. Instead, longer scenes must be shown to the user to gather a representative response. Therefore, the Block CUR decomposition proposed in this chapter

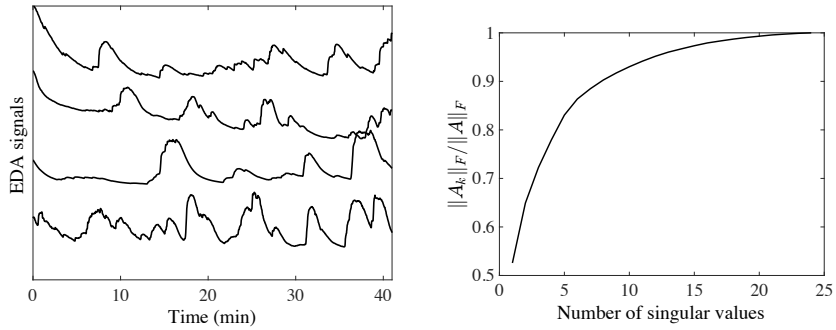


Figure 3.3: Panel (a) shows EDA data for four users watching the NCIS video and (b) demonstrates the low rank nature of $\boldsymbol{A}$

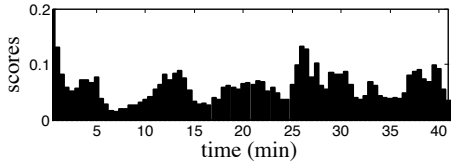
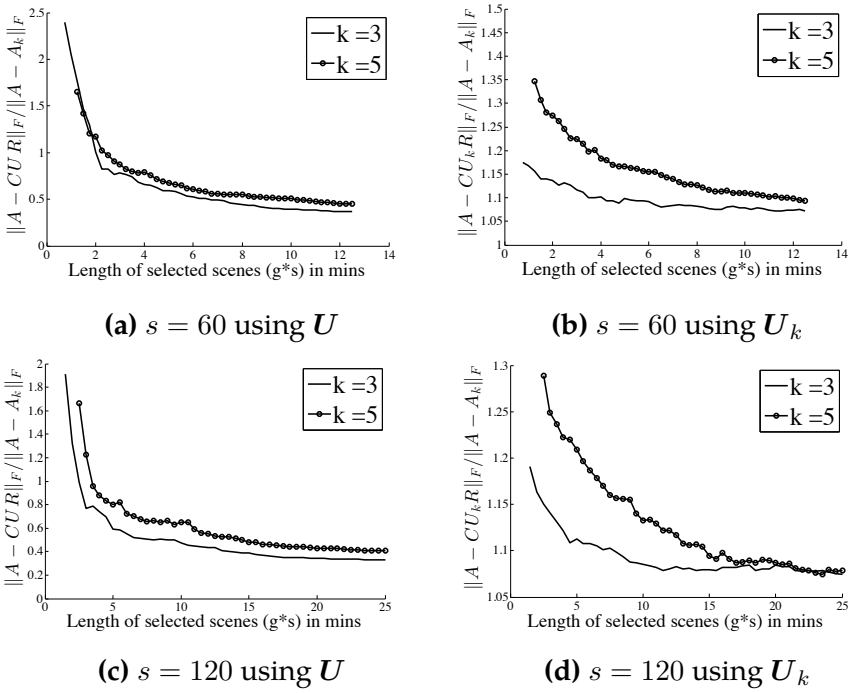


Figure 3.4: Block leverage scores for EDA data with k = 5 and s = 120 columns (30 seconds).

is directly applicable.

The biometric experiment setup is as follows. We attached 24 subjects with the Empatica E3 wearable sensor Garbarino et al. (2014) that measures electro-dermal activity (EDA) at 4 Hz. The subjects were shown a 41-minute episode of the television series "NCIS", in the genres of action and crime. The resulting biometric data matrix was 24×9929. Our goal is to use Block CUR decomposition to show only a subset of users the entire content, and to then impute the biometric data for users that have viewed only a small number of selected scenes from the content.

Results. We refer to the biometric data matrix as $\boldsymbol{A}$ and plot the EDA



(a) $s = 60$ using $\boldsymbol{U}$ **(b)** $s = 60$ using $\boldsymbol{U}_k$

(c) $s = 120$ using $\boldsymbol{U}$ **(d)** $s = 120$ using $\boldsymbol{U}_k$

Figure 3.5: Error plots for two values of target rank, $k = 3, 5$.

traces (rows) corresponding to four users in Figure 3.3a. To demonstrate the low rank nature of the data, we plot the Frobenius norm of $\boldsymbol{A}$ covered by $\boldsymbol{A}_k$ as a function of k in Figure 3.3b. We find that for this data, only 5 singular vectors are needed to capture 80% of the total Frobenius norm of the complete matrix. Next, we segment the columns of this matrix into blocks such that $s = 120$ columns (or 30 seconds). In Figure 3.4, we show the computed block leverage scores. The leverage scores seem to suggest that certain scenes are more important than others. For example, the highest leverage scores are around the 12, 26, and 38 minute marks. This corresponds to scenes of a dead body, unveiling of a clue to solving

the mystery, and the final arrest, respectively.

Using Algorithm 1, we uniformly sample EDA traces (rows) of 20 users and hold out the EDA traces of 4 users. We sample column blocks and plot the resulting error in Frobenius norm in Figure 3.5. The plots show the normalized Frobenius norm error of the CUR approximation as a function of the number of blocks, g, sampled. More precisely, the ratio $\|\boldsymbol{A} - \boldsymbol{CUR}\|_F / \|\boldsymbol{A} - \boldsymbol{A}_k\|_F$ and $\|\boldsymbol{A} - \boldsymbol{CU}_k\boldsymbol{R}\|_F / \|\boldsymbol{A} - \boldsymbol{A}_k\|_F$ are plotted for two values of the target rank, $k = 3$ and 5 and two values of block size, $s = 60$ and 120 columns per block (15 and 30 seconds), respectively. We also compare the error using $\boldsymbol{U}_k$, the rank-k approximation of $\boldsymbol{U}$, which leads to an exactly rank-k matrix approximation since this may be a restriction in some applications. We repeat Algorithm 1 ten times[1] and plot the mean normalized error over 10 trials.

The error drops sharply as we sample more blocks but quickly flattens demonstrating that a summary of the movie could suffice to approximate the full responses. The plots also show the interplay between the number of blocks sampled and the issue of context which is related to block size. To give the viewer some context we would want to make the scene as long as possible but we want to show them only a summary of the content to reduce the cost. These conflicting aims result in a trade-off of block size and the number of blocks sampled. For example, for $k = 5$, the normalized

[1]These plots were generated using *sampling without replacement* even though our theory supports *sampling with replacement* since sampling the same blocks is inefficient in practice.

Table 3.1: Table comparing the number of sampling operations needed for given ε using our Block CUR result based on block sampling and traditional CUR based on individual column sampling (note this is not the same as the vectorized block columns in Table 1). This leads to speedup since it is more efficient to retrieve predefined blocks than querying individual rows or columns in these regimes. The α_R term we introduce satisfies the bound $1 \leq \alpha_R \leq s$.

Method	No. of sampling ops.
Traditional CUR	$\mathcal{O}\left(\frac{k^2}{\varepsilon^2}\log(\frac{1}{\delta}) + \frac{k^4}{\varepsilon^6}\log^3(\frac{1}{\delta})\right)$
Block CUR	$\mathcal{O}\left(\frac{k^2}{\varepsilon^2}\log(\frac{1}{\delta}) + \frac{k^4}{\boldsymbol{\alpha_R}\varepsilon^6}\log^3(\frac{1}{\delta})\right)$

error is less than 1 when a 2.5 minute long clip is shown to the viewer, that is $g = 10$ with block size $s = 60$ columns (or 15 seconds), whereas the normalized error is less than 1 when a 3.5 minute long clip is shown to the viewer ($g = 7$) with block size $s = 120$ columns (or 30 seconds). These results demonstrate the practical use of the Block CUR algorithm.

Distributed experiments

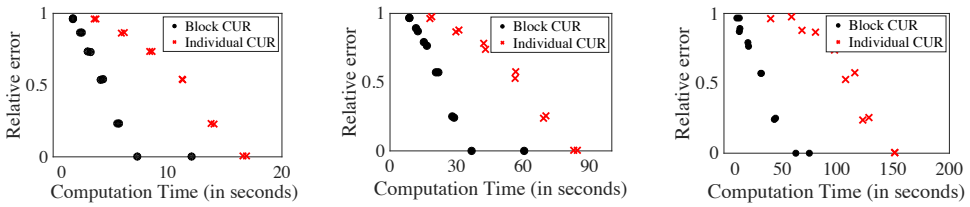
In this section we demonstrate empirically that the proposed block sampling based CUR algorithms can achieve a significant speed-up when used to decompose matrices in a distributed data setting by comparing their performance with individual column sampling based traditional CUR algorithms on both synthetic and real-world data. We report the relative-error of the decomposition (*i.e.,* $\|\boldsymbol{A} - \boldsymbol{CUR}\|_F / \|\boldsymbol{A}\|_F$) and the sampling time of each algorithm on different data-sets.

We implemented the algorithms in Scala 2.10 and Apache Spark 2.11 on Amazon Elastic Map-Reduce (Amazon EMR). The compute cluster was

constructed using four Amazon m4.4xlarge instances, with each compute node having 64 GB of RAM. Using Spark, we store the data sets as resilient distributed dataset (RDD), a collection of elements partitioned across the nodes of the cluster (see Figure 3.2). In other words, Spark partitions the data into many blocks and distributes these blocks across multiple nodes in the cluster. Using block sampling, we can approximate the matrix by sampling only a subset of the important blocks. Meanwhile, individual column sampling would require looking up all the partitions containing specific columns of interest as shown in Table 3.1. Our experiments examine the runtime speed-up from our block sampling CUR that exploits the partitioning of data.

Synthetic experiments. The synthetic data is generated by $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{V}$ where $\boldsymbol{U} \in \mathbb{R}^{m \times k}$ and $\boldsymbol{V} \in \mathbb{R}^{k \times n}$ are random matrices with i.i.d. Gaussian random entries, resulting in a low rank matrix $\boldsymbol{A}$. We perform CUR decomposition on matrices of size $m \times n$ with $m = n$, target rank k, and number of blocks G (set here across all experiments to be 100). The leverage scores are calculated by computing the SVD of the rows sampled uniformly with $\boldsymbol{R} \in \mathbb{R}^{r \times n}$. We sample one-sixth of the rows.

Figure 3.6 shows the plots for relative error achieved with respect to the runtime required to sample $\boldsymbol{C}$ and $\boldsymbol{R}$ matrices for both Block CUR and traditional CUR algorithms. To focus on the speed-up achieved by taking into account the block storage of data we compare running times of only the sampling operations of the algorithms (which excludes the time

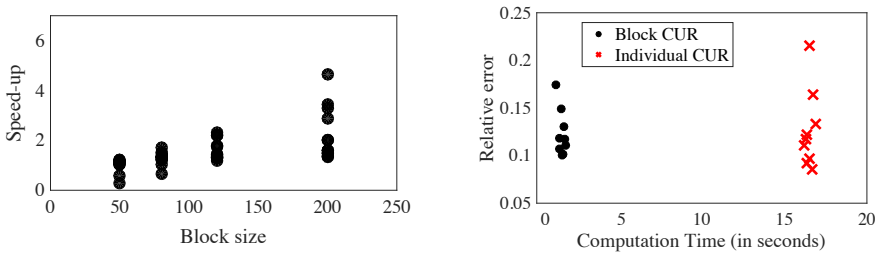


(a) $n = 8000$ **(b)** $n = 12000$ **(c)** $n = 20000$

Figure 3.6: Performance on synthetic $n \times n$ matrices with rank $n/10$.

required to compute the SVD). We note that other steps in both algorithms can be updated to include faster variants such as the approximation of leverage scores by sketching or sampling Drineas et al. (2012). We vary g, the number of blocks chosen, from 1 to 6. The number of columns chosen is thus $c = gs$, where s denotes the number of columns in a block and varies from 50 to 200. We repeat each algorithm (Block CUR and traditional CUR) twice for the specified number of columns, with each realization as a point in the plot. The proposed Block CUR algorithm samples the c columns in g blocks, while traditional CUR algorithm samples the c columns one at a time.

Consistently, these results show that block sampling achieves the relative error much faster than the individual column sampling – with performance gains increasing as the size of the matrix grows, as shown in Figure 3.6. While the same amount of data is being transmitted regardless of whether block or individual column sampling is used, block sampling is much faster because it needs to contact fewer executors to retrieve blocks of columns rather than the same number of columns individually. In the



(a) **(b)**

Figure 3.7: Performance on $900 \times 10,000$ Arcene dataset with block size 12. (a) Runtime speed-up from block sampling compared to individual column sampling for varying block sizes. (b) Block CUR achieves similar relative errors as individual CUR with much lower computation time.

worst case, sampling individual columns may need to communicate with all of the executors, while block sampling only needs to communicate with g executors. Thus, by exploiting the partitioning of the data, the Block CUR approach is able to achieve roughly the same quality of approximation as traditional column-based CUR, as measured by relative error, with significantly less computation time.

Real-world experiments. We also conduct experiments on the Arcene dataset Guyon et al. (2004) which has 900 rows and 10,000 columns. We compare the running time for both block and traditional CUR decomposition. We again find consistent improvements for the block-wise approach compared with individual column sampling. With block size $s = 12$, sampling up to 10 groups led to an average speed up of 11.2 over individual column sampling, as shown in Figure 3.7. The matrix is very low rank, and sampling a few groups gave small relative errors.

3.6 Discussion

We detail the differences between our technique and prior CUR algorithms here. This includes additional assumptions required, algorithmic trade-offs, and discussion of sampling and computational complexity.

Block stable rank

Theorem 3.6 tells us that the number of blocks required to achieve an ε relative error depends on the structure of the blocks (through α_R). Intuitively, this is saying the groups that provide more information improve the approximation faster than less informative groups. The α_R term depends on the stable or numerical rank (a stable relaxation of exact rank) of the blocks. The stable rank $\alpha = \|\boldsymbol{A}\|_F^2/\|\boldsymbol{A}\|_2^2$ is a relaxation of the rank of the matrix; in fact, it is stable under small perturbations of the matrix $\boldsymbol{A}$ Rudelson and Vershynin (2007). For instance, the stable rank of an approximately low rank matrix tends to be low. The α_R term defined in Theorem 3.6 is the minimum stable rank of the column blocks. Thus, the α_R term gives a dependence of the block sampling complexity on the stable ranks of the blocks. It is easy to check that $1 \leq \alpha_R \leq s$. In the best case, when all the groups have full stable rank with equal singular values, α_R achieves its maximum. The worst case $\alpha_R = 1$ is achieved when a group or block is rank-1. That is, sampling groups of rank s gives us a lot more information than groups of rank 1, which leads to a reduction in the total sampling

Table 3.2: Table comparing the sample complexity needed for given ε using our Block CUR result and a bound obtained by trivial extension of traditional CUR. For ease of comparison, we show the results with full SVD computation ignoring incoherence assumption stated in Corollary 1 in the Appendix. The α_R term we introduce satisfies the bound $1 \leq \alpha_R \leq s$.

Results	r	g
Traditional CUR extended to block setting	$\mathcal{O}\left(\frac{k^2}{\varepsilon^2}\log\left(\frac{1}{\delta}\right)\right)$	$\mathcal{O}\left(\frac{k^4}{\varepsilon^6}\log^3(\frac{1}{\delta})\right)$
Our Block CUR	$\mathcal{O}\left(\frac{k^2}{\varepsilon^2}\log(\frac{1}{\delta})\right)$	$\mathcal{O}\left(\frac{k^4}{\boldsymbol{\alpha_R}\varepsilon^6}\log^3(\frac{1}{\delta})\right)$

complexity.

Incoherence

The column space incoherence (Definition 3.5) is used to provide a guarantee for approximation without computing the SVD of the entire matrix $\boldsymbol{A}$. However, if it is possible to compute the SVD of the entire matrix, then the rows can be sampled using row leverage scores, and the incoherence assumption can be dropped. The relative error guarantee, independent of incoherence, for the full SVD Block CUR approximation is stated as Corollary 1 in the Appendix. The corollary follows by a similar analysis as Theorem 3.6 so we defer the proof to the Appendix. Other than block sampling, the setup of this result is equivalent to the traditional column sampling result stated in Lemma 3.8. Next, we compare the block sampling result with extensions of traditional column sampling.

Sample complexity: comparison with extensions of traditional CUR results

In order to compare the sample complexity of our block sampling results with trivial block extensions of traditional column sampling results we focus our attention on the similar leverage score based CUR result in Theorem 3.1. A simple extension to block setting could be obtained by considering a larger row space in which blocks are expanded to vectors. This would lead to a sample complexity bound obtained by Theorem 3.1. The sampling complexity of the Block CUR derived in Theorem 3.6 tells us the number of sampling operations or queries that need to be made to memory in order to construct the $\boldsymbol{R}$ and $\boldsymbol{C}$ matrices. As shown in Table 3.2 the column block sample complexity obtained by traditional CUR extensions results is always greater than or equal to those required by our Block CUR result because $1 \leq \alpha_R \leq s$. This happens since traditional CUR-based results are obtained by completely ignoring the block structure of the matrix.

4 THE GROUP ORDERED WEIGHTED ℓ_1 (GROWL)

4.1 Introduction

In the linear regression setting, we have n samples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, where $\boldsymbol{x}_i \in \mathbb{R}^p$ is a p-dimensional feature vector, and y_i is the associated response. The model approximates the response variable y_i as a linear combination of the features

$$y_i = \sum_{j=1}^{p} x_{ij}\beta_j^* + \eta_i \tag{4.1}$$

where $\boldsymbol{\beta}^*$ is the unknown linear functional and η_i is a zero mean random variable.

The usual "least-squares" estimator is based on minimizing the squared-error loss

$$\boldsymbol{\beta}_{LS} = \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^p} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2. \tag{4.2}$$

Machine learning applications such as neuroscience and computational biology involve several variables, often tens of thousands to millions. These variables correspond to voxels or brain regions in the case of fMRI studies and genes in computational biology. Further, the number of samples available are small due to acquisition costs. For example, in the case of fMRI studies, each sample might correspond to the activity in all the voxels in response to a particular stimulus. Typically, it is not feasible to obtain a

large number of data samples. Least-squares estimates in high-dimensions lead to intractable solutions by overfitting to the data. To reduce the variance, regularization penalties are often introduced to increase the bias of the estimator. The ridge estimator favors weight coefficients with small ℓ_2 norm

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^p} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_2^2, \tag{4.3}$$

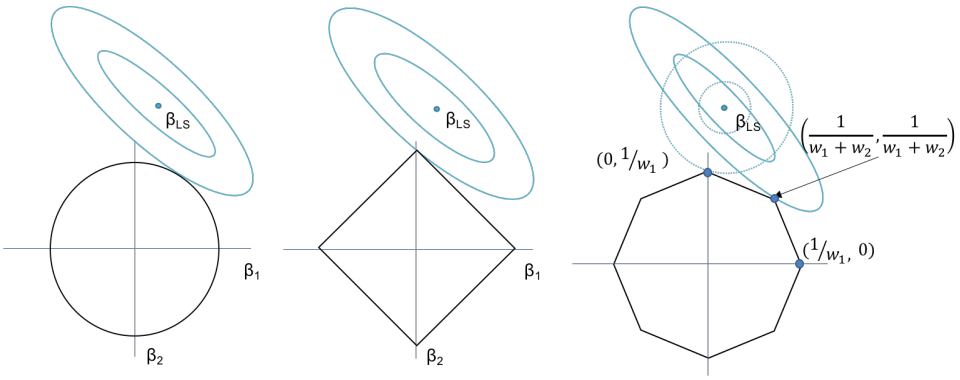
where $\lambda > 0$.

Often interpretability of the model is also desired. With a large number of variables, we often would like to identify a smaller subset that exhibit the strongest effects. To overcome this issue, typically we solve regularized problems that bias towards simpler solutions such as those with fewer number of voxels selected in the solution. Simple models make the solution more interpretable, and can be used by scientists to better understand how the different regions of the brain work. The most common model is perhaps that of Lasso Tibshirani (1996) which combines the least-squares loss with an ℓ_1-constraint, leading to sparse solutions, meaning only a few variables are assigned non-zero coefficients.

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^p} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1, \tag{4.4}$$

for $\lambda > 0$.

The constraint region for lasso is diamond-shaped, in contrast with the



(a) Ridge **(b)** Lasso **(c)** OWL

Figure 4.1: (a) Ridge (ℓ_2), (b) Lasso (ℓ_1), and (c) OWL ($w_1 > w_2 > 0$) balls in $\mathbb{R}^2$. Contours centered at Least-squares estimate, β_{LS} (4.3). Low correlation (dotted), leads to solution at $\beta_1 = 0$. High correlation (solid), leads to clustered solution at $\beta_1 = \beta_2$.

disk-shaped region of ridge regression depicted in Figure 4.1(a) and (b) respectively. The diamond has corners at sparse solutions where one of the parameters are set to zero. The key property of the ℓ_1 penalty is its ability to yield sparse solutions.

A severe limitation of Lasso arises when some of the variables are strongly correlated (*i.e.*, cases in which certain columns of X may be close to, or even exactly, collinear). In such cases, Lasso selects an arbitrary subset of the correlated variables to preserve sparsity of the solution. This is a concern in fMRI, since certain voxels may have very correlated activation patterns and the goal is to identify *all* the voxels that are relevant to the task.

In the standard (single-task) regression problem, this issue has been tackled using many techniques, including elastic net Zou and Hastie (2005),

OSCAR Bondell and Reich (2008), OWL Figueiredo and Nowak (2014), and others. The EN regularizer combines the Lasso and ridge penalties by taking the weighted sum of the ℓ_1 and ℓ_2 norms to promote grouping of correlated variables. Variants of EN (more precisely, tighter relaxations of sparsity plus ℓ_2) like the k-support norm are also introduced in Argyriou et al. (2012), Belilovsky et al. (2015). The Ordered Weighted ℓ_1 (OWL) penalty, of which the OSCAR is a special case, is given by

$$\Omega_{\boldsymbol{w}}(\boldsymbol{\beta}) \ = \sum_{i=1}^{p} w_i |\beta_{[i]}|, \tag{4.5}$$

where $\beta_{[i]}$ is the component of $\boldsymbol{\beta}$ with the i-th largest magnitude and w is a vector of non-negative and non-increasing weights such that $w_1 \geq \cdots \geq w_p \geq 0$ and $w_1 > 0$. The OSCAR was experimentally studied in Bondell and Reich (2008); Zhong and Kwok (2012); the main conclusion from their experiments is not that OSCAR clearly outperforms EN in terms of accuracy, but that while typically requiring fewer degrees of freedom due to its exact clustering behavior, it is still competitive with EN. In other words, their claim is not that OSCAR achieves higher accuracy, but that its ability to identify clusters of correlated covariates improves interpretability which is typically of interest in fMRI applications.

OWL norm and its clustering property

The Ordered weighted ℓ_1 (OWL) family of regularizers for sparse linear regression was recently introduced and studied in Bogdan et al. (2015); Bondell and Reich (2008); Figueiredo and Nowak (2016); Zeng and Figueiredo (2014). The authors show that OWL automatically clusters and averages regression coefficients associated with strongly correlated variables. This has a desirable effect of selecting more of the relevant variables than the ℓ_1 penalty. This is illustrated with the octogonal constraint region of the OWL penalty in Figure 4.1 (c).The OWL norm reduces to the ℓ_1 norm when all the weights are set to be equal and the ℓ_∞ norm when $w_2 = \cdots = w_p = 0$.

The OWL-regularized regression is stated as follows,

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^N} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \Omega_{\boldsymbol{w}}(\boldsymbol{\beta}). \tag{4.6}$$

The following result provides a sufficient condition stating that when columns of X are correlated enough, OWL automatically clusters them (i.e., correlated columns will have equal-valued coefficients in the solution). Let $\Delta_w = \min\{w_l - w_{l+1}, l = 1, ..., N-1\}$ be the minimum gap between consecutive elements of w.

Lemma 4.1 (Theorem 2.1 in Figueiredo and Nowak (2016))**.** Let $\widehat{\beta}$ be a solution of (4.6), and $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ be two columns of $\boldsymbol{X}$. If $\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2 < \Delta_w/\|\boldsymbol{y}\|_2$, then $|\widehat{\beta}_i| = |\widehat{\beta}_j|$.

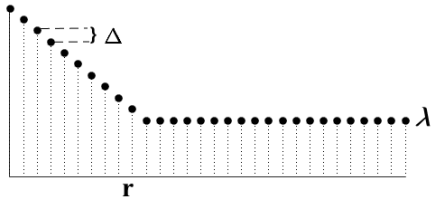


Figure 4.2: An example of OWL-Ramp weights.

New clustering properties of OWL

Recall that Δ_w is the minimum gap in the OWL weights. Usually, this gap is very small, like O($1/N$) or in some cases zero, which makes the clustering property of OWL infeasible. Hence, we extend the clustering results of OWL norm with a ramp-like sequence of weights which we call OWL-ramp.

$$\begin{aligned} w_i &= (r-i+1)\Delta + \lambda && \text{for } i \in \{1,\dots,r\}, \\ &= \lambda && \text{for } i \in \{r+1,\dots,N\}, \end{aligned}$$

where $\lambda > 0$, $\Delta \geq 0$ and $1 \leq r \leq N$. Figure 4.2 depicts an example of the OWL-Ramp weights.

We prove the following clustering result for the OWL-Ramp regularization.

Lemma 4.2. Let $\widehat{\boldsymbol{\beta}}$ be a solution of the optimization in (4.6) using OWL-Ramp weights ($\lambda, \Delta > 0$, and $1 \leq r \leq N$) and $M := \{j : |\widehat{\beta}_j| = \max_i |\widehat{\beta}_i|\}$. If elements of M belong to a Δ-connected component on the unit hypersphere in $\mathbb{R}^n$ with cardinality at least r, then $|M| \geq r$.

Proof details are provided are provided in the Appendix. This lemma provides a sufficient condition for the largest magnitude cluster in the OWL-Ramp solution to have critical mass. The Euclidean distance condition from Lemma 4.1 translates into the Δ-connected component condition in this Lemma. Note that the minimum gap Δ_w is always smaller than or equal to Δ. It is strictly smaller in most cases. This provides more room for clustering in the OWL solution.

4.2 Problem Setting

Multi-task regression is a generalization of the linear model which consists of a collection of "r-tasks" or response variables that share the same set of features or variables,

$$\boldsymbol{Y} \approx \boldsymbol{X}\boldsymbol{B}^* \tag{4.7}$$

where $\boldsymbol{Y} \in \mathbb{R}^{n\times r}$ is the collection of r response variables and $\boldsymbol{B}^* \in \mathbb{R}^{p\times r}$ corresponds to the weight coefficient matrix. In some cases, the tasks may also share the same subset of variables leading to a row-sparse weight matrix, where many rows of the coefficient matrix are zero vectors. To find row-sparse solutions to the multi-task regression problem we consider a generalization of the Lasso below.

Group Lasso

Consider the group lasso optimization

$$\min_{\boldsymbol{B}\in\mathbb{R}^{p\times r}} \|\boldsymbol{Y}-\boldsymbol{X}\boldsymbol{B}\|_F^2 \;+\; \lambda\|\boldsymbol{B}\|_{1,2}\;. \tag{4.8}$$

The parameter $\lambda > 0$ is an adjustable weight on the sparsity-promoting regularizer $\|\boldsymbol{B}\|_{1,2}$, which is defined as follows. The rows of $\boldsymbol{B}$ are denoted by $\boldsymbol{\beta}_{i*}$, $i = 1,\ldots,p$, and the norm $\|\boldsymbol{B}\|_{1,2} \;=\; \sum_{i=1}^{p} \|\boldsymbol{\beta}_{i*}\|_2$. This encourages solutions with only a few nonzero rows in $\boldsymbol{B}$ Lounici et al. (2009, 2011); Obozinski et al. (2011).

4.3 The GrOWL penalty

The main technical innovation in this chapter is a new approach to the group lasso that is designed to cope with strongly correlated covariates (i.e., cases in which certain columns of $\boldsymbol{X}$ may be close to, or even exactly, collinear). This is a concern in fMRI, since certain voxels may have very correlated activation patterns. This problem is illustrated in Figure 4.3, where we simulate a situation in which columns 5 and 7 of the data matrix $\boldsymbol{X}$ are highly correlated. Group lasso selects one of the corresponding rows in $\boldsymbol{B}$ (row 5), whereas GrOWL correctly selects both rows 5 and 7. Note that the group lasso can select at most n features (n being the number of items), since the number of nonzero rows in the solution cannot exceed

the number of measurements. This can be severe limitation of the group lasso in applications where the number of features far exceeds the number of items.

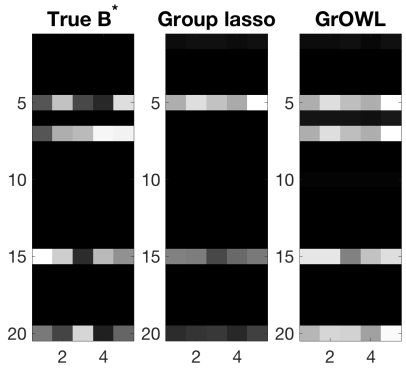


Figure 4.3: A comparison of group lasso (middle) and grOWL (right) optimization solutions with correlated columns in $\boldsymbol{X}$ showing that GrOWL selects relevant features (row 5 and 7) even if they happen to be strongly correlated and automatically cluster them by setting the corresponding coefficient rows to be equal (or nearly equal).

We propose a generalization of the OWL approach to the multi-task setting, and thus call our new approach Group OWL (GrOWL). We show that GrOWL shares many of the desirable features of the OWL method, namely it automatically clusters and averages regression coefficients associated with strongly correlated columns of $\boldsymbol{X}$. In this section, we consider the general optimization

$$\min_{\boldsymbol{B}\in\mathbb{R}^{p\times r}} \; L(\boldsymbol{B}) \; + \; G(\boldsymbol{B}) \tag{4.9}$$

where typical loss functions considered here are absolute error, $L(\boldsymbol{B}) =$

Table 4.1: Weight vectors corresponding to different instances of OWL and GrOWL

Lasso	$w_i = \lambda$ for $i = 1, \ldots, p$
Linear (OSCAR)	$w_i = \lambda + \lambda_1(p - i)/p$ for $i = 1, \ldots, p$
Spike	$w_1 = \lambda + \lambda_1, w_i = \lambda_1$ for $i = 2, \ldots, p$
Ramp	$w_i = (r - i + 1)\Delta + \lambda,$ for $i \in \{1, \ldots, r\}$, $w_i = \lambda$ for $i \in \{r + 1, \ldots, p\}$

$\|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{B}\|_1$, or squared Frobenius error, $L(\boldsymbol{B}) = \|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{B}\|_F^2$. Let $\boldsymbol{B} \in \mathbb{R}^{p \times r}$ and let $\boldsymbol{\beta}_{i*}$ and $\boldsymbol{\beta}_{*j}$ denote the ith row and jth column of $\boldsymbol{B}$. Define the GrOWL penalty

$$G(\boldsymbol{B}) = \sum_{i=1}^{p} w_i \|\boldsymbol{\beta}_{[i]*}\|_2, \tag{4.10}$$

where $\boldsymbol{\beta}_{[i]*}$ is the row of $\boldsymbol{B}$ with the i-th largest 2-norm and $\boldsymbol{w}$ is a vector of non-negative and non-increasing weights.

Proximal algorithms

We derive the proximal operator for the optimization using the GrOWL norm here. The computational algorithms to solve the GrOWL optimization based on the proximity operators can be found in Parikh and Boyd

(2013). The proximal operator of the GrOWL norm is given by

$$\text{prox}_G(\boldsymbol{V}) = \arg\min_{\boldsymbol{B}} \frac{1}{2}\|\boldsymbol{B} - \boldsymbol{V}\|_F^2 + G(\boldsymbol{B}) \tag{4.11}$$

In the following theorem, we solve for the proximity operator of GrOWL in terms of the proximity of OWL. For the exact formulation of prox_{Ω_w}, see Bogdan et al. (2013), Zeng et al. (2014).

Theorem 4.3. *Let* $\tilde{v}_i = \|\boldsymbol{v}_{i*}\|$ *for* $i = 1, \cdots, p$. *Then* $\text{prox}_G(\boldsymbol{V}) = \widehat{\boldsymbol{V}}$, *where* i*-th row of* $\widehat{\boldsymbol{V}}$ *is*

$$\widehat{\boldsymbol{v}}_{i*} = (\text{prox}_{\Omega_w}(\tilde{\boldsymbol{v}}))_i \frac{\boldsymbol{v}_{i*}}{\|\boldsymbol{v}_{i*}\|} \tag{4.12}$$

Proof Sketch: The proof proceeds by finding a lower bound for the objective function in (4.11) and then we show that the proposed solution achieves this lower bound.

Efficient O($p \log p$) algorithms to compute prox_{Ω_w} have been proposed by Bodgan *et al* Bogdan et al. (2014, 2013).

4.4 Main Results

Before we analyze the GrOWL regularization, we state a generalization of Lemma 2.1 in Figueiredo and Nowak (2014) which will be useful later in the section.

Lemma 4.4. *Consider a vector $\boldsymbol{\beta} \in \mathbb{R}^p_+$ and any two of its components β_j and β_k, such that $\beta_j > \beta_k$. Let $\boldsymbol{v} \in \mathbb{R}^p_+$ be obtained by applying a transfer of size $\varepsilon, \varepsilon'$ to $\boldsymbol{\beta}$ such that $\varepsilon \in (0, (\beta_j - \beta_k)/2]$ and $-\beta_k \leq \varepsilon' \leq \varepsilon$, that is: $v_j = \beta_j - \varepsilon, v_k = \beta_k + \varepsilon'$, and $v_i = \beta_i$, for $i \neq j, k$. Let $\boldsymbol{w}$ be a vector of non-increasing non-negative real values, $w_1 \geq w_2 \geq \cdots \geq w_p \geq 0$, and Δ be the minimum gap between two consecutive components of vector $\boldsymbol{w}$, that is, $\Delta = \min\{w_i - w_{i+1}, i = 1, \cdots, p-1\}$. $\Omega_{\boldsymbol{w}}(\cdot)$ is the OWL norm with weight vector $\boldsymbol{w}$, then*

$$\Omega_{\boldsymbol{w}}(\boldsymbol{\beta}) - \Omega_{\boldsymbol{w}}(\boldsymbol{v}) \;\geq\; \Delta\varepsilon.$$

Proof sketch. The proof is similar to that of Lemma 2.1 in Figueiredo and Nowak (2014) with different sizes $\varepsilon, \varepsilon'$. The result follows since we assume that increase in k-th component is less than decrease in j-th component *i.e.*, $\varepsilon' \leq \varepsilon$.

The following theorem states that identical variables lead to equal coefficient rows corresponding to those variables in the solution given by the optimization using GrOWL.

Theorem 4.5 (Identical columns)**.** *Let $\widehat{\boldsymbol{B}}$ denote the solution to the optimization in (4.9) with $L(\boldsymbol{B}) = \|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{B}\|_1$ or $L(\boldsymbol{B}) = \|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{B}\|_F^2$. If columns $\boldsymbol{x}_{*j}$ and $\boldsymbol{x}_{*k}$ satisfy $\boldsymbol{x}_{*j} = \boldsymbol{x}_{*k}$ and the minimum gap, $\Delta > 0$, then $\widehat{\boldsymbol{\beta}}_{j*} = \widehat{\boldsymbol{\beta}}_{k*}$.*

Proof sketch. The proof is divided into two steps. First, we show $\|\widehat{\boldsymbol{\beta}}_{j*}\| = \|\widehat{\boldsymbol{\beta}}_{k*}\|$ and then we further show that the rows are equal. We proceed by contradiction. Assume $\|\widehat{\boldsymbol{\beta}}_{j*}\| \neq \|\widehat{\boldsymbol{\beta}}_{k*}\|$ and, without loss of generality,

suppose $\|\widehat{\boldsymbol{\beta}}_{j*}\| > \|\widehat{\boldsymbol{\beta}}_{k*}\|$. We see that there exists a modification of the solution with a smaller GrOWL norm using Lemma 4.4 and same data-fitting term, and thus smaller overall objective value which contradicts our assumption that $\widehat{\boldsymbol{B}}$ is the minimizer of $L(\boldsymbol{B}) + G(\boldsymbol{B})$.

The following theorems state that nearly identical variables lead to equal norm coefficient rows, and further highly correlated coefficient rows, corresponding to those variables in the solution given by the optimization using GrOWL.

Theorem 4.6 (Correlated columns 1)**.** *Let $\widehat{\boldsymbol{B}}$ denote the solution to the optimization in (4.9) with $L(\boldsymbol{B}) = \|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{B}\|_1$ or $L(\boldsymbol{B}) = \|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{B}\|_F^2$. If $\boldsymbol{x}_{*j}$ and $\boldsymbol{x}_{*k}$ satisfy $\|\boldsymbol{x}_{*j} - \boldsymbol{x}_{*k}\|_1 \leq \frac{\Delta}{\sqrt{r}}$ or $\|\boldsymbol{x}_{*j} - \boldsymbol{x}_{*k}\|_2 \leq \frac{\Delta}{\|\boldsymbol{Y}\|_F}$ respectively, then $\|\widehat{\boldsymbol{\beta}}_{j*}\| = \|\widehat{\boldsymbol{\beta}}_{k*}\|$.*

Proof sketch. The proof is similar to the identical columns theorem. By contradiction and without loss of generality, suppose $\|\widehat{\boldsymbol{\beta}}_{j*}\| > \|\widehat{\boldsymbol{\beta}}_{k*}\|$. We show that there exists a transformation of $\widehat{\boldsymbol{B}}$ such that the increase in the data fitting term is smaller than decrease in the GrOWL norm.

Theorem 4.7 (Correlated columns 2)**.** *Let $\widehat{\boldsymbol{B}}$ denote the solution to the optimization in (4.9) with $L(\boldsymbol{B}) = \|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{B}\|_1$ or $L(\boldsymbol{B}) = \|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{B}\|_F^2$. If $\boldsymbol{x}_{*j}$ and $\boldsymbol{x}_{*k}$ satisfy $\|\boldsymbol{x}_{*j} - \boldsymbol{x}_{*k}\|_1 \leq \frac{\Delta}{\phi\sqrt{r}}$ or $\|\boldsymbol{x}_{*j} - \boldsymbol{x}_{*k}\|_2 \leq \frac{\Delta}{\phi\|\boldsymbol{Y}\|_F}$ respectively, then $\|\widehat{\boldsymbol{\beta}}_{j*} - \widehat{\boldsymbol{\beta}}_{k*}\| \leq \frac{8\phi\|\widehat{\boldsymbol{\beta}}_{k*}\|}{4\phi^2+1}$*

which further implies that

$$1 \geq \frac{\widehat{\boldsymbol{\beta}}_{j*}^T \widehat{\boldsymbol{\beta}}_{k*}}{\|\widehat{\boldsymbol{\beta}}_{j*}\| \|\widehat{\boldsymbol{\beta}}_{k*}\|} \geq 1 - \frac{1}{2}\left(\frac{8\phi}{4\phi^2+1}\right)^2 \left(\geq 1 - \frac{2}{\phi^2}\right)$$

where $\phi \geq 1$.

Proof sketch. By contradiction, suppose $\|\widehat{\boldsymbol{\beta}}_{j*} - \widehat{\boldsymbol{\beta}}_{k*}\| \geq \frac{8\phi\|\widehat{\boldsymbol{\beta}}_{k*}\|}{4\phi^2+1} \geq \frac{2\|\widehat{\boldsymbol{\beta}}_{k*}\|}{\phi}$. We show that there exists a transformation of $\widehat{\boldsymbol{B}}$ such that the increase in the data fitting term is smaller than the decrease in the GrOWL norm. This contradicts our assumption that $\widehat{\boldsymbol{B}}$ is the minimizer of $L(\boldsymbol{B}) + G(\boldsymbol{B})$ and completes the proof.

The proof details are deferred to the appendix. So far, we have seen that the GrOWL penalty has desirable clustering properties that lead to nearly identical coefficient rows. We study two variants of GrOWL with different weight sequences $\boldsymbol{w}$. We study the GrOWL-Lin weights with linear decay (equivalent to the OSCAR in single-task regression), and the GrOWL-Spike weight sequence which puts a big weight on the maximum magnitude while the rest of the coefficients are weighted equally.

5 REPRESENTATIONAL SIMILARITY LEARNING (RSL)

5.1 Introduction

This chapter considers the following learning task. Suppose we have a set of items along with human-judged pairwise similarities among them. For instance, the items could be visual stimuli such as advertisements, pictures, or diagrams. Assume that we also have a high-dimensional feature associated with each item. These could be numerical features quantifying the characteristics of each item or, in the case of fMRI, the features are voxel responses to stimuli. The learning task is to determine the subset of features that is most predictive of the human-judged similarities. This can be posed mathematically as follows. Let $\boldsymbol{S}$ be an $n \times n$ matrix of pairwise similarities between n items. Let $\boldsymbol{X}$ be an $n \times p$ matrix where the ith row is the $1 \times p$ vector of the features for item i. We then wish to find a weight matrix $\boldsymbol{W}$ such $\boldsymbol{X}\boldsymbol{W}\boldsymbol{X}^T \approx \boldsymbol{S}$. The weight matrix reveals which features are most important and how they are combined to represent the human-judged similarities. We call this *Representational Similarity Learning* (RSL).

Let us illustrate this learning problem with two applications. First, suppose the items are diagrams of chemical molecules, and that each diagram is also described by a vector comprised of many

visual features (e.g., counts of different atom types, counts of bounds,

bound angles, etc). Novice chemistry students may miss critical similarities and differences when comparing different diagrams. After gathering pairwise similarity judgments from the students, RSL can be used to identify which features they are attending to and, thus, which important features they may be overlooking. Second, consider *Representational Similarity Analysis* (RSA) in fMRI brain imaging Kriegeskorte et al. (2008). In RSA a person is scanned while viewing n different visual stimuli. Pairwise similarities are obtained through other experiments, such as asking people to look at pairs of stimuli and rate the similarity. In this case, the features are the stimuli responses of p voxels in the brain, and the goal is to determine which voxels (and hence brain regions) are encoding the similarities. RSA is depicted in Figure 5.1.

The main contributions of our work in Oswal et al. (2016) are

- We pose RSL as a sparsity-regularized multi-task regression problem. Standard methods, like group lasso, may not select important features if they are strongly correlated with others. To address this shortcoming we apply the new regularizer for multitask regression called *Group Ordered Weighted* ℓ_1 (GrOWL). We show, in theory and fMRI experiments, how GrOWL deals with strongly correlated covariates.

- Another key contribution is a novel application to fMRI brain imaging. *Representational Similarity Analysis* (RSA) is a tool for testing

whether localized brain regions encode perceptual similarities. Using GrOWL, we propose a new approach called *Network RSA* that can discover arbitrarily structured brain networks (possibly widely distributed and non-local) that encode similarity information.

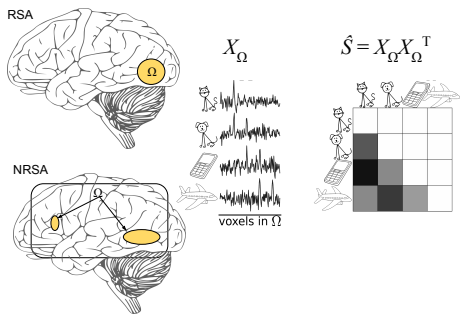


Figure 5.1: Representational Similarity Analysis. Traditional RSA methods consider only localized brain regions of interest or spherical clusters in the cortex (upper left) Kriegeskorte et al. (2006, 2008). We propose a new *Network* RSA (NRSA) method that can potentially identify non-local brain networks that encode similarity information (lower left).

5.2 Algorithm

The goal of RSL is to find a sparse and symmetric matrix $\boldsymbol{W} \in \mathbb{R}^{p \times p}$ such that $\boldsymbol{S} \approx \boldsymbol{X}\boldsymbol{W}\boldsymbol{X}^T$. By sparse we mean that at most $k < p$ rows/columns of $\boldsymbol{W}$ are nonzero. The locations of the nonzero elements indicate which features are included in the similarity representation. For instance, consider the $n \times 1$ vectors corresponding features $\boldsymbol{x}_k$ and $\boldsymbol{x}_\ell$ (i.e., the kth and ℓth columns of $\boldsymbol{X}$). It is easy to show that the contribution of these two features to the similarity representation is given by $W_{k,\ell}\, \boldsymbol{x}_k \boldsymbol{x}_\ell^T + W_{\ell,k}\, \boldsymbol{x}_\ell \boldsymbol{x}_k^T$. If

$W_{k,\ell} = W_{\ell,k} \neq 0$, then the correlations between the two features contribute to the approximation of the similarity matrix $\boldsymbol{S}$. The complete similarity representation can be expressed as

$$\boldsymbol{S} \approx \boldsymbol{X}\boldsymbol{W}\boldsymbol{X}^T = \sum_{k,\ell=1}^{p} W_{k,\ell}\, \boldsymbol{x}_k \boldsymbol{x}_\ell^T .$$

The approximation problem can be posed as the least squares optimization

$$\min_{\boldsymbol{W}} \|\boldsymbol{S} - \boldsymbol{X}\boldsymbol{W}\boldsymbol{X}^T\|_F^2 \tag{5.1}$$

where the objective is the Frobenius norm of the difference between the similarity matrix $\boldsymbol{S}$ and its approximation. Classic studies of human-produced similarity judgments in many domains of interest yield low rank matrices McRae et al. (2005); Shaver et al. (1987); Shepard (1980) due to clustering or other representational structure amongst the items under consideration. Therefore, we suppose $\boldsymbol{S}$ is a real, symmetric and approximately rank r matrix, then there exists a matrix $\boldsymbol{Y} \in \mathbb{R}^{n\times r}$ and diagonal matrix $\boldsymbol{D} \in \mathbb{R}^{r\times r}$ which satisfies $\boldsymbol{S} \approx \boldsymbol{Y}\boldsymbol{D}\boldsymbol{Y}^T$ (e.g., obtained via eigendecomposition or Cholesky decomposition) where the diagonal entries of $\boldsymbol{D}$ correspond to the sign of the r largest eigenvalues and the columns of $\boldsymbol{Y}$ are the corresponding eigenvectors of $\boldsymbol{S}$. We will assume that $\boldsymbol{S}$ is rank r in the following discussion (if not, then we will use its best rank r approximation instead). Thus, we may instead consider the

optimization

$$\min_{\boldsymbol{B}} \|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{B}\|_F^2 \tag{5.2}$$

For any coefficient matrix $\boldsymbol{B}$ the corresponding weight matrix is given by $\boldsymbol{W} = \boldsymbol{B}\boldsymbol{D}\boldsymbol{B}^T$. Both optimizations are convex, but we will work with the latter since it automatically enforces the low-rank assumption and can be easily modified to include additional constraints or regularizers.

To relate the solutions of these two optimizations, we note that any stationary point, $\widetilde{\boldsymbol{W}}$, of the optimization in (5.1) satisfies the zero-gradient condition given by

$$\nabla f(\widetilde{\boldsymbol{W}}) = \boldsymbol{X}^T(\boldsymbol{X}\widetilde{\boldsymbol{W}}\boldsymbol{X}^T - \boldsymbol{S})\boldsymbol{X} = \boldsymbol{0}.$$

Also, any stationary point, $\widehat{\boldsymbol{B}}$, of the optimization in (5.2) satisfies

$$\nabla g(\widehat{\boldsymbol{B}}) = \boldsymbol{X}^T(\boldsymbol{X}\widehat{\boldsymbol{B}} - \boldsymbol{Y}) = \boldsymbol{0}.$$

Notice that for $\widehat{\boldsymbol{W}} = \widehat{\boldsymbol{B}}\boldsymbol{D}\widehat{\boldsymbol{B}}^T$, we have

$$\begin{aligned}
\nabla f(\widehat{\boldsymbol{W}}) &= \boldsymbol{X}^T(\boldsymbol{X}\widehat{\boldsymbol{W}}\boldsymbol{X}^T - \boldsymbol{S})\boldsymbol{X} \\
&= \boldsymbol{X}^T(\boldsymbol{X}\widehat{\boldsymbol{B}}\boldsymbol{D}\widehat{\boldsymbol{B}}^T\boldsymbol{X}^T - \boldsymbol{Y}\boldsymbol{D}\boldsymbol{Y}^T)\boldsymbol{X} \\
&= \frac{1}{2}\boldsymbol{X}^T(\boldsymbol{X}\widehat{\boldsymbol{B}} - \boldsymbol{Y})\boldsymbol{D}(\boldsymbol{X}\widehat{\boldsymbol{B}} + \boldsymbol{Y})^T\boldsymbol{X} \\
&\quad + \frac{1}{2}\boldsymbol{X}^T(\boldsymbol{X}\widehat{\boldsymbol{B}} + \boldsymbol{Y})\boldsymbol{D}(\boldsymbol{X}\widehat{\boldsymbol{B}} - \boldsymbol{Y})^T\boldsymbol{X} \\
&= \boldsymbol{0}
\end{aligned}$$

Thus any stationary point of (5.2) leads us to a stationary point of (5.1). Since both optimizations are convex, we can conclude that $\widehat{\boldsymbol{B}}$ leads us to a solution of the squared optimization and under certain conditions on $\boldsymbol{X}$, say when $\boldsymbol{X}$ is full rank this solution is also unique.

In many applications the weight matrix $\boldsymbol{W}$ and the coefficient matrix $\boldsymbol{B}$ are expected to exhibit sparsity. Indeed, our hypothesis is that a small subset of the features encodes the similarity representations, hence the sparsity. Likewise classic studies of human-produced similarity judgments in many domains of interest yield low rank matrices McRae et al. (2005); Shaver et al. (1987); Shepard (1980) due to clustering or other representational structure amongst the items under consideration. Thus, we also expect $\boldsymbol{W}$ and $\boldsymbol{B}$ to be low-rank. This has an additional benefit of averaging noisy clusters of correlated features. Thus, the optimization above can be modified to obtain sparse and low-rank solutions, as described next.

RSL via Group Lasso and GrOWL

Consider the group lasso optimization in 4.8 and the GrOWL optimization in 4.9. Note that the optimization variable $\boldsymbol{B}$ is a $p \times r$ matrix, which guarantees a rank r (or less) solution, and thus similarity representation $\boldsymbol{X}\boldsymbol{B}\boldsymbol{D}\boldsymbol{B}^T\boldsymbol{X}^T$ will be rank r at most, which is a simple way to enforce the low-rank constraint. The parameter $\lambda > 0$ is an adjustable weight on the sparsity-promoting regularizer $\|\boldsymbol{B}\|_{1,2}$, which is defined as follows.

This encourages solutions with only a few nonzero rows in $\boldsymbol{B}$ Lounici et al. (2009, 2011); Obozinski et al. (2011). We note that the optimization in (5.1) can also be modified directly to obtain sparse and low-rank solutions. For instance, the nuclear norm of $\boldsymbol{W}$ could be penalized to obtain a low-rank solution. However, the nuclear norm optimization tends to be computationally expensive in practice.

5.3 Application to Brain Imaging Data

Network-based approaches to cognitive neuroscience typically assume that mental representations are encoded as distributed patterns of activation over large neural populations, with different populations encoding different kinds of representational structure and communicating this structure to other network components. Extensive research over past several years has focused on testing such hypotheses using data from functional brain imaging techniques such as fMRI. The best-known approach in this vein has been RSA Kriegeskorte et al. (2008). RSA is typically applied either to a specific brain region of interest (ROI) or to many localized regions throughout the brain in a process called *searchlight analysis* Kriegeskorte et al. (2006). For a given region, RSA computes the cosine distances between the evoked responses for all stimulus pairs. The resulting dissimilarity matrix is correlated with a target matrix of known psychophysical distances amongst stimuli. If these correlations are reliably non-zero, this

suggests the corresponding region may encode the similarity information.

A drawback of ROI and searchlight RSA is that these methods place strong assumptions on the anatomical structure of the regions thought to encode the similarities of interest (predefined ROIs or spherical clusters). We propose a new approach called *Network RSA* (NRSA) that can discover arbitrarily structured brain networks (possibly widely distributed and non-local) that encode similarity information. The key insight behind our method is that RSA can be posed as a multi-task regression problem which, in conjunction with sparsity regularization methods, can automatically detect networks of voxels that appear to jointly encode similarity information.

Network RSA is summarized as follows. Consider a set of n items and suppose we are given an $n \times n$ similarity matrix $\boldsymbol{S}$, where the ij-th element $\boldsymbol{S}_{ij}$ is the known psychophysical similarity Tversky and Gati (1982) between item i and item j. For example, these may come from human judgments of perceptual similarity between pairs of stimuli. RSA is based on the hypothesis that there exists a set of voxels whose correlations across stimuli encode the similarities in $\boldsymbol{S}$, as depicted in Figure 5.1. In RSA, the features are $\boldsymbol{X} \in \mathbb{R}^{n \times p}$, a matrix of voxel activations. Each row corresponds to activations in all p voxels in response to a stimulus, and each column corresponds to the activations in specific voxel to the n stimuli. Our generalized notion of RSA, which encompasses conventional ROI Kriegeskorte et al. (2008) and searchlight Kriegeskorte et al. (2006) approaches, involves

finding a sparse and symmetric matrix $\boldsymbol{W} \in \mathbb{R}^{p \times p}$ such that $\boldsymbol{S} \approx \boldsymbol{X}\boldsymbol{W}\boldsymbol{X}^T$. The locations of the nonzero elements indicate which voxels are included in the similarity-encoding brain network, and the weights in $\boldsymbol{W}$ indicate the strength of the edges in the network.

Network RSA application: Simulated Data

Before applying our framework to real fMRI data, we consider a simulation study that allows us to compare results against a known ground-truth. We compare group lasso and GrOWL by analyzing synthetic data generated from a deep neural network model trained to generate distributed representations of a word's sound (phonology) and meaning (semantics) from its spelling (orthography; Figure 5.2 top left). The network structure is motivated by the influential "triangle" model of the human reading system Plaut et al. (1996). Specifically, phonological outputs receive contributions from two separate pathways: a *direct* route mediated by a single hidden layer, and an "indirect" route composed of three hidden layers, which must first compute mappings from orthography to semantics, then project onward to contribute to the phonological outputs. This architecture is interesting because different kinds of similarity structure emerge through learning in different network components. The central idea is that orthographic and phonological similarities are highly systematic: items that are similar in spelling are likely (though not guaranteed) to be similar in pronunciation. In contrast, orthographic and semantic similarity struc-

tures are unsystematic: similarity of word spelling does not necessarily predict similarity of meaning and vice versa. In learning to map from orthography to semantics and on to phonology, the indirect path thus comes to encode quite different similarity relations amongst the words than does the direct path Harm and Seidenberg (2004); Plaut et al. (1996).

To capture these properties we generated model "orthographic" representations as patterns sampled from 6 overlapping clusters of binary input features, roughly corresponding to different orthographic neighborhoods. For every word a "phonological" pattern was generated by flipping each orthographic feature with probability 0.1. Thus phonological patterns were distorted variants of orthographic patterns, creating high systematicity between these. We also created a "semantic" pattern for each word from a set of binary features also organized into clusters. Across items, these vectors expressed a hierarchical similarity structure with two broad superordinate clusters each composed of three tighter clusters. Importantly, the similarity structure expressed by the semantic vectors was independent of the structure expressed in the orthographic/phonological patterns. The left bottom panel in Figure 5.2 shows the cosine distances encoded amongst the 30 "words" in each layer of one trained model. Layers in the direct path each encode roughly the same distances amongst items, while the semantic layer encodes a quite different set of distances that is weakly reflected in two of the three hidden layers in the indirect path. Thus the different components of this simple

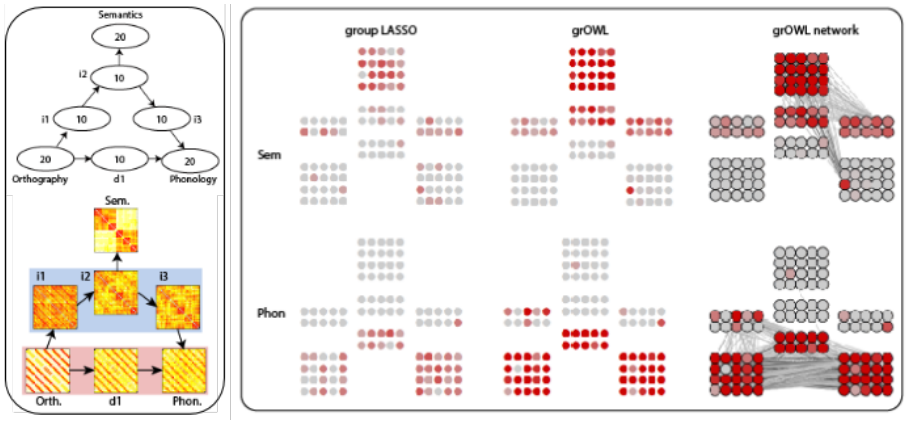


Figure 5.2: Left panel: Network architecture (top) and the similarity structure expressed in each layer (bottom). Red background shows the direct pathway and blue the indirect pathway from orthography to phonology. Layers in the two pathways encode different similarity structures. The target similarity matrices for the analysis express either the semantic structure (top layer) or the phonological structure (bottom right layer). Arrows indicate feed-forward connectivity. Right panel: Units selected by group LASSO (right) and GrOWL (middle) when decoding semantic (top) or phonological (bottom) structure. Colors show the proportion of times across subjects and unit concatenations that the unit received a non-zero weight, with red indicating 1 and gray 0. The rightmost plots show the largest weights in the associated matrix W for each GrOWL model, which pick out two subnetworks in the model.

word-reading network contribute differentially to the encoding of semantic versus ortho-phonological similarity structure.

We trained 100 models with different initial weights, corresponding to 100 model subjects, and presented each with 30 orthographic inputs. Each input generated a vector of activations over the 100 model units. To ensure high redundancy amongst units this vector was concatenated 5 times and perturbed with independent noise, yielding 500 measurements per model subjects. These were treated as analogs of the estimated BOLD response at

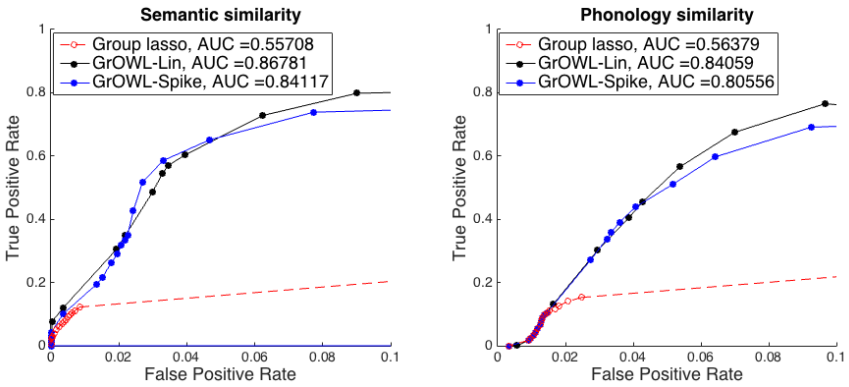


Figure 5.3: Trade-off curves for FPR ≤ 0.1 generated by sweeping through (λ, λ_1) values (for $\lambda = 0$, all units are selected and as λ is increased fewer units are given non-zero weight). Each point corresponds to a combination of λ and λ_1 that gives the best trade-off (where setting $\lambda_1 = 0$ results in the group lasso). The pareto-frontier for group lasso (red), GrOWL-Lin (black), GrOWL-Spike (blue) is averaged across 100 participants for each method, considering both similarity structures, Semantics (left panel) and Phonology (right panel). Note for any $\lambda > 0$, the group lasso solution will include *at most* $n = 30$ voxels, since the number of selected voxels will not exceed the number of measurements. If $\lambda = 0$, then the group lasso will select all voxels. Thus, group lasso curve beyond $n = 30$ selections (around 0.01 FPR) is shown as a dashed line, which extends linearly to the point $(FPR, TPR) = (1, 1)$.

each of 500 model voxels in a brain imaging study. We then applied group lasso and GrOWL to find the voxel subsets that encode either the semantic or phonological distances (derived from target values for the output layers of the network). We fit models by searching a grid of parameters (λ, λ_1), including $\lambda_1 = 0$ as the special case of GrOWL that is group lasso. For each grid point we counted a voxel as "selected" if it received a non-zero weight, and assessed how accurately the model selected the voxels encoding phonological structure (all those along the direct pathway) or semantic structure (the semantic layer hidden layers 2 and 3 in the indirect path) by

computing hit rates and false alarm rates). All three models showed low and equivalent cross-validation error; however GrOWL achieved this error rate while selecting considerably more voxels. The ROC plots in Figure 5.3 further show that GrOWL did not select additional voxels at random: it outperformed group lasso considerably in discriminating signal-carrying from non-signal carrying voxels. The right panel of Figure 5.2 shows the frequency with which each model unit is selected for the best-performing solution of each method and structure type. The strong sparsity enforced by group lasso is clearly apparent: target units are selected less consistently than with GrOWL, which consistently discovers more of the signal.

Finally, we considered the ability of GrOWL to reveal the network structure encoding each kind of similarity, treating the weights in the matrix $\boldsymbol{W}$ as direct estimates of the joint participation of pairs of units in expressing the target similarity. The rightmost plots of Figure 5.2 show the estimated connectivity, thresholded to show the 25% of the non-zero weights with the largest magnitudes. The detected edges clearly express the network representational substructure: units in the direct pathway are shown as highly interconnected with one another and weakly or disconnected from those in the indirect pathway, and vice versa. Thus the search for different kinds of similarity reveals different functional subnetworks in the model.

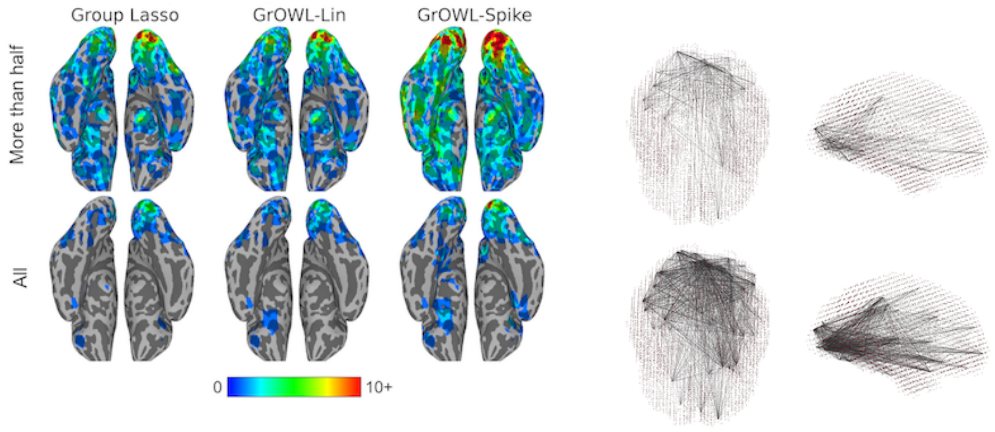
Network RSA application: Real Data

We next consider the application of group lasso and GrOWL to the discovery of similarity structure in neural responses measured by fMRI across the whole brain while participants perform a cognitive task. As with the well-known searchlight RSA Kriegeskorte et al. (2008), we begin with a measurement of the $n \times n$ similarities existing amongst a set of n items in some cognitive domain. Using fMRI, we measure the neural responses evoked by each item at the scale of single voxels (3mm cubes), and treat these p voxels as features of the n items. We then compute a rank-r approximation of the target similarity matrix $\boldsymbol{S} = \boldsymbol{Y}\boldsymbol{Y}^T$, and use this as the target $\boldsymbol{Y} \in \mathbb{R}^{n \times r}$ matrix for a sparse-regression analysis of the $n \times p$ matrix of fMRI responses, $\boldsymbol{X}$, evoked by each item across the whole cortex. The model is then fit to optimize the objective functions specified in (4.8) for group lasso and (4.9) with squared Frobenius loss for GrOWL. The best regularization parameter is selected through cross-validation, and a final model is fit with that parameter and used to predict the similarities existing amongst a set of items in an independent hold-out set. Model predictions are compared to results expected from a null hypothesis that no features encode the target similarity structure. If predictions are more accurate than expected from random data, this provides evidence that the model has discovered voxel subsets that jointly encode some of the target similarity structure. Moreover, because the model is constrained to be sparse, most voxels will receive coefficients of zero, and the presence

of non-zero coefficients can be taken as evidence that the corresponding voxel encodes information important to representing the target similarity structure.

The current experiment aims to answer three questions. (1) Does either approach learn a model from whole-brain fMRI that can accurately predict the pairwise similarities among stimuli? (2) Does group lasso or GrOWL learn a more accurate model? (3) Do the fitted models identify voxels in areas that are consistent with known neural representations? To answer these questions, we applied the approach to discover voxels that work to encode the visual similarities existing amongst a set of line drawings of common objects. We chose this task and dataset because (a) there exist well-understood methods for objectively measuring the degree of visual similarity amongst such items Antani et al. (2002) and (b) it is well known that visual similarity is encoded by neural responses in occipital and posterior temporal cortices Kriegeskorte and Kievit (2013).

fMRI dataset. The data were collected as part of a larger study from 23 participants at the University of Manchester who were compensated for their time. Each participant viewed a series of line drawings depicting common objects while their brains were scanned with fMRI. The line drawings included 37 items, each repeated 4 times for a total of 148 unique stimulus events. At each trial participants pressed a button to indicate whether the item could fit in a trash can. Scans were collected in a sparse event-related design and underwent standard pre-processing to align func-



(a) **(b)**

Figure 5.4: Panel (a) shows surface maps corresponding to group lasso (left), GrOWL-Lin (middle) and GrOWL-Spike (right) showing the voxels selected for the tuning parameters with smallest prediction error on the hold-out data for *at least five* and *all nine* cross-validations in the top and bottom rows respectively. The heat map shows the number of subjects for which those voxels were picked. Blue is the least (1 subject) and red is the most (10 or more subjects). Panel (b) is a network plot showing the top edges from the $\boldsymbol{W}$ matrix for the best-performing parameterization of group LASSO (top) and GrOWL-Spike (bottom) in one subject. The thickness of the edges is proportional to the edge weights.

tional images to the anatomy and to remove movement and scanner artifact and temporal drift. Responses to each stimulus event were estimated at each voxel using a deconvolution procedure with a standard HRF kernel. For each participant a cortical surface mask was generated based on T1-weighted anatomical images, and functional data were filtered to exclude non-cortical voxels. Voxels with estimated responses more than 5 standard deviations from the mean response across voxels were excluded from the analysis. 10k-15k voxels were selected for each participant, and neural responses across all voxels for each of 148 stimulus events were entered

into the analysis. The mean response across the 4 repeated observations of each item were taken to give 37 item responses for each participant. Each column corresponding to a voxel was normalized to be of standard deviation equal to one and a column of ones was added for bias correction.

Target similarities. Each stimulus was a bitmap of a black-and-white line drawing. We took pairwise Chamfer distance Borgefors (1988) as a proxy for inter-item visual dissimilarities. r = 3 is the smallest value to attain $\|\boldsymbol{S} - \boldsymbol{Y}\boldsymbol{Y}^T\|_F / \|\boldsymbol{S}\|_F \leq 0.15$. This 37×3 matrix $\boldsymbol{Y}$ was used as the target matrix for the analysis.

Model fitting. For each participant, training data were divided into 9 subsets containing 4-5 stimulus events each. One subset was selected as a final hold-out set. Models were then fit at each of 10 increasing values of each λ and λ_1 parameter (grid points) using 8-fold cross validation. At each fold we assessed the model using the Frobenius norm of the difference between the target $\boldsymbol{Y}$ entries and the predicted $\widehat{\boldsymbol{Y}} = \boldsymbol{X}\widehat{\boldsymbol{B}}$ entries for hold-out items (henceforth the model error). We selected the λ with the lowest mean error for each subject, then fit a full model for each subject at this value and assessed it against the final hold-out set, considering the model error on hold-out items. We repeat this with 9 different final hold-out sets.

Results. The table below shows performance on the final hold-out sets (H) for each participant and each method, considering error between predicted ($\widehat{\boldsymbol{Y}}$) and actual dissimilarities ($\boldsymbol{Y}$) where MSE = $\|\boldsymbol{Y}_H - \widehat{\boldsymbol{Y}}_H\|_F / \|\boldsymbol{Y}_H\|_F$. Both approaches show significantly non-random predic-

tion. As in our simulations, all methods show comparable prediction error on hold-out sets. We also note that, as in the simulations, GrOWL selected almost double the number of voxels in each participant. Our assertion that both approaches show significantly non-random predictions is based on a permutation-based paired t-test, where chance would have been zero difference. By this measure, for example, GrOWL-Spike's performance is significantly better than chance (t-value=8.59, p<0.0001).

Method	MSE (p-value)
Group Lasso	0.5266 (1.045e-07)
GrOWL-Lin	0.5271 (6.456e-08)
GrOWL-Spike	0.5213 (1.774e-08)

Figure 5.4(a) shows the locations of selected voxels (i.e., those with non-zero coefficients) across all 23 participants for the tuning parameters with smallest mean prediction error on hold-out data, mapped into a common anatomical space with 4mm full-width-half-max spatial smoothing and projected onto a model of the cortical surface. To look into the stability of selection across different training sets, the top row shows the voxels selected for *at least five* (out of nine) cross-validation runs while the bottom row shows the voxels selected for *all* the nine cross-validation runs. As seen in the maps, both methods pick voxels prominently in the occipital and posterior temporal cortices and GrOWL picks consistently more voxels than group lasso.

Figure 5.4(b) shows the largest magnitude edges in the $\boldsymbol{W}$ matrix for the best-performing parameterization of group LASSO (top) and GrOWL-Spike (bottom) in one subject. Two observations are of note. First, both methods uncover a similar network structure, with many interconnections in visual cortical regions and some edges connecting to anterior regions in frontal and temporal cortex. Second, as in the simulations, GrOWL reveals a much denser network. The results suggest the possibility that subregions of frontal and temporal cortex may, together with occipito-temporal cortex, participate in networks that serve to encode visual similarity structure.

5.4 Discussion

In Cox (2016), this series of whole brain analyses is extended to studying visual, audio and semantic pairwise similarities with fMRI and ECoG data to test hypotheses of a “hub-and-spoke" model which supports the role of the anterior temporal lobe (ATL) in semantic cognition, a controversial topic in cognitive neuroscience. In summary, it demonstrates that the network RSA procedure is capable of identifying both well localized and radically distributed representations in the brain. While the visual model and the semantic model of audio data resemble solutions obtained with other methods, the semantic model of visual data marks a major departure from the standard set of results.

Until recently, no method in the literature has been capable of identify-

ing distributed representations that encode semantic similarity structure that span multiple regions of the brain. It has therefore been difficult to adjudicate between these hypotheses with functional neuroimaging. Network RSA enables adjudicating the different hypotheses about how semantic similarity structure is encoded in neural activity, and the role of the ATL in that encoding.

6 SCALABLE SPARSE SUBSPACE CLUSTERING WITH OWL

6.1 Introduction

Subspace clustering refers to the task of grouping high-dimensional data points into distinct subspaces. This generalizes classical, single-subspace approaches to data modeling like principal components analysis (PCA). Effectively, subspace clustering aims to represent data in terms of a union of subspace (UoS). Many applications, ranging from computer vision (*e.g.*, image segmentation Yang et al. (2008), motion segmentation Vidal et al. (2008) and face clustering Ho et al. (2003)) to network analysis Eriksson et al. (2012), have demonstrated the advantages of this generalization. Unlike classical PCA, the problem of fitting a UoS model to data is a computationally challenging task, and numerous approaches have been proposed. For a comprehensive review of these algorithms, see Vidal (2011). The state-of-the-art is Sparse Subspace Clustering (SSC) Elhamifar and Vidal (2013) which provides both tractability and provable guarantees under mild conditions Soltanolkotabi et al. (2012).

SSC is a computationally intensive method. It requires performing a sparse regression for each of the N points in the dataset of interest. The main contribution of Oswal and Nowak (2018) is a new approach that has the potential to significantly reduce the computational complexity, making it more applicable to large-scale problems. The central idea is to modify the regression technique so that accurate clustering is possible using only

the results of a $k \ll N$ regressions, instead of all N. This reduces the complexity by a factor of N/k. The modified regression is based on the Ordered Weighted ℓ_1 (OWL) regularizer, which performs simultaneous regression and clustering of correlated variables. The clustering property of the OWL, combined with ideas from random geometric graph theory, allows us to prove that the new approach, called OWL Subspace Clustering (OSC), tends to select more points from the correct subspaces in each regression compared to SSC. In the ideal case, where L subspaces are orthogonal and the number of points per subspace is sufficiently large, then OSC can succeed with just L optimizations (gain factor of roughly N/L) as detailed later in Section 6.1. This key feature of OSC makes accurate clustering possible based on regression solutions for only a small subset of the total dataset, significantly reducing the computational complexity compared to SSC. In experiments, we find that OSC can achieve a speedup of 20$\times$ to 30$\times$ even for small scale synthetic problems.

Related work

When the data is high-dimensional or the number of data points is large, solving the N Lasso problems (each in $N-1$ variables) in SSC can be computationally challenging. Greedy algorithms for computing sparse representations of the data points (in terms of all the other data points) have therefore been popular alternatives. Broadly speaking, three kinds of such algorithms have been proposed in the literature, namely Thresh-

olded subspace clustering Heckel and Bölcskei (2015), subspace clustering using Orthogonal Matching Pursuit (OMP) Dyer et al. (2013); You et al. (2016), and Dimensionality-reduced subspace clustering Heckel et al. (2017); Wang et al. (2015). TSC relies on the nearest neighbors–in spherical distance–of each data point to construct the adjacency matrix. SSC-OMP employs OMP instead of the Lasso to compute sparse representations of the data points. Similar in spirit to SSC-OMP, Nearest subspace neighbor (NSN) Park et al. (2014) greedily assigns to each data point a subset of the other data points by iteratively selecting the data point closest (in Euclidean distance) to the subspace spanned by the previously selected data points. In this chapter, we propose OWL Subspace Clustering (OSC), which employs the Ordered Weighted ℓ_1 (OWL) regularizer instead of the Lasso to compute sparse representations of the data points. The clustering property of OWL (discussed next) leads to a hybrid approach between SSC and spherical distance based methods like TSC (illustrated in Figure 6.1). The effect is that OSC tends to assign non-zero weights to the same points as SSC and additionally selects neighbors of these points in terms of Euclidean distance. All of the above methods, including OSC, can be carried out on dimensionality reduced data points, in the spirit of Heckel et al. (2017); Wang et al. (2015). We compare the performance of these methods in Section 6.5.

Review of SSC

The key idea of SSC is representing each data point as a sparse linear combination of the remaining data points. The rationale is that points in the same subspace are likely to be selected to represent a given point, and thus the selected points provide an indication of the cluster. Sparse linear regression using ℓ_1 minimization is used to determine the representations for each point. SSC solves N sparse linear regressions over N data points to form an $N \times N$ adjacency matrix, which in turn defines a graph where the vertices are data points with edges indicated by the adjacencies. Spectral clustering is used to partition the graph, and hence the data, into clusters. Each sparse linear regression solves

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^N} \|\boldsymbol{\beta}\|_1 \text{ such that } \boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta}, \tag{6.1}$$

where the columns of $\boldsymbol{X} \in \mathbb{R}^{n \times N}$ represent the N n-dimensional data points and $\boldsymbol{y} \in \mathbb{R}^n$ is a data point that is to be represented as a linear combination of the columns using the sparse coefficient vector β. Another sparse regression commonly used in practice is the Lasso given by (4.4). It is sometimes referred to as the Lagrangian or “noisy" version since it is used for subspace clustering with noise Soltanolkotabi et al. (2014); Wang and Xu (2016).

In this chapter, we propose to replace the ℓ_1 minimization in SSC with the OWL regularizer (refer Section 4.1) in order to discover many more

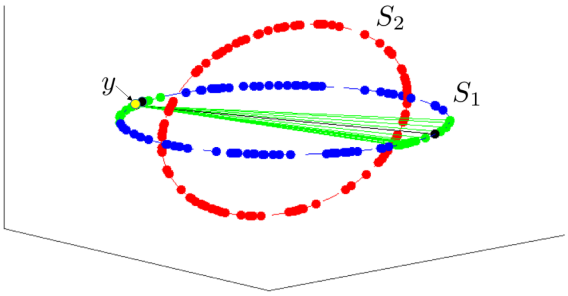


Figure 6.1: 2-dimensional subspaces in $\mathbb{R}^3$, $\mathcal{S}_1$ and $\mathcal{S}_2$. The points with non-zero coefficients in solutions of ℓ_1 and Ordered Weighted ℓ_1 sparse regressions on y (yellow point) depicted as black and green points (and edges) respectively. The automatic clustering property of OWL leads to a hybrid approach that tends to select same points as SSC and their Euclidean distance based neighbors.

points from the true subspace in each optimization. The key observation is that points in the same subspace will be more correlated (in the sense above), than points in different subspaces. OWL tends to select more points from the common subspace in each regression, compared to ℓ_1 methods (see Figure 6.2), which consequently improves the performance of spectral clustering. We formalize this idea using tools from random geometric graph theory. Next, we demonstrate the benefit of the clustering property of OWL in a simple setting where the subspaces are orthogonal to each other.

Orthogonal subspaces example

To build some intuition, let $\boldsymbol{X} \in \mathbb{R}^{n \times N}$ be a matrix whose columns are drawn from a union of L *orthogonal* linear subspaces, $\mathcal{S}_1 \cup \cdots \cup \mathcal{S}_L$. Let y be a new point from subspace $\mathcal{S}_\ell$. To keep the notation simple, let us

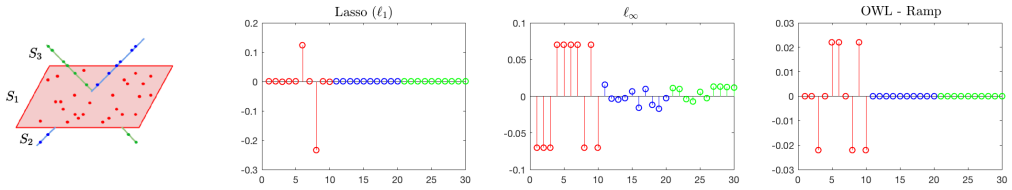


Figure 6.2: Solutions of the ℓ_1, ℓ_∞, and OWL minimizations for y_i lying in $\mathcal{S}_1$. 10 points selected from each of the three subspaces ordered such that the first and the last 10 points belong to $\mathcal{S}_1$ and $\mathcal{S}_3$ respectively. The ℓ_1 solution corresponds to choosing two other points lying in $\mathcal{S}_1$ whereas the ℓ_∞ solution selects points from all subspaces. OWL selects more points from $\mathcal{S}_1$ than ℓ_1.

assume that the dimension of each subspace is d. Suppose $\mathcal{T}$ contains the indices of columns belonging to subspace $\mathcal{S}_\ell$ and $|\mathcal{T}| = N_\ell$.

It is easy to see that any solution of (4.6), $\widehat{\beta}$, satisfies $\widehat{\beta}_j = 0$ for all $j \notin \mathcal{T}$, since columns from orthogonal subspaces cannot reduce the residual error and including them will increase the OWL penalty. Additionally, assume that points in the subspace are uniformly drawn from the unit hypersphere in that subspace. Then consider the graph constructed by placing an edge between pairs of points (vertices) that are within Δ_w Euclidean distance of each other (where Δ_w is the minimum gap in the OWL weights). A simple argument (developed in Section D.2) shows that the resulting random geometric graph is connected with probability at least $1 - \delta$, if the number of points in the subspace is large enough, specifically $N_\ell \propto \Delta_w^{-d} \log(\Delta_w^{-d}/\delta)$. It then follows from Lemma 4.1 that $|\widehat{\beta}_j| = |\widehat{\beta}_i|$ for $\forall i, j \in \mathcal{T}$. In other words, all the columns within the subspace will be selected and have equal-valued coefficients.

Figure 6.3 shows an example of the coefficients generated by perform-

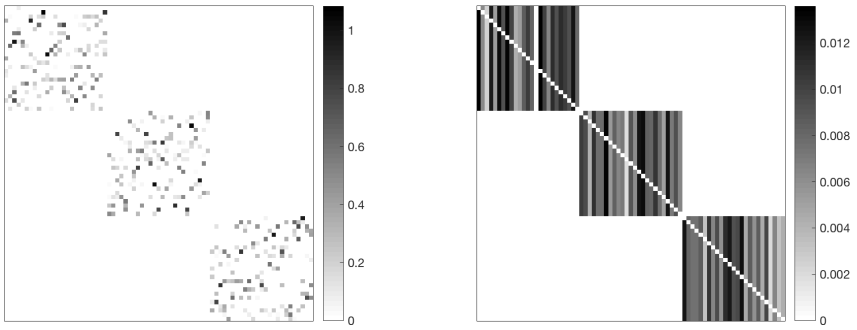


Figure 6.3: Examples of coefficient matrices $|\boldsymbol{B}| = [|\widehat{\beta}_1| \dots |\widehat{\beta}_N|]$ for exact ℓ_1 minimizations (left) and OWL optimizations (right) with the contiguous columns lying in three orthogonal subspaces each of dimension $d = 5$ in $\mathbb{R}^{15}$. The plots were generated using OWL-Ramp weights defined in Section 6.4.

ing the optimizations in (6.1) and (4.6) on a collection of points from three orthogonal subspaces. The results obtained with orthogonal subspaces suggest that, in an ideal scenario, we could perform subspace clustering by running only **one** OWL optimization per subspace without the need for spectral clustering leading to a total of only L optimizations where L is the number of subspaces. This can easily be achieved by running one OWL optimization on a randomly selected point, removing the points chosen by OWL and repeating the process till no points remain. This works since *OWL has the potential to find all the points in the same subspace* in a single run. This has the potential of significantly reducing the computational complexity of SSC. However, OWL could lead to clustering of points from other subspaces when subspaces are not orthogonal. This tradeoff is studied in more detail in the rest of the chapter by providing theoretical and empirical evidence suggesting that an OWL-based fast subspace clustering

algorithm (stated in Section 6.3) can reduce the computational complexity of SSC by a factor depending on the nature of the subspaces.

6.2 Problem Setting

Notation and model

We are given data points lying in a union of unknown linear subspaces; there are L subspaces $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_L$ of $\mathbb{R}^n$ of dimensions $d_1, d_2, \ldots, d_L$. We are given a collection of N data points as columns of $\boldsymbol{X} \subset \mathbb{R}^{n \times N}$, which may be partitioned as $\boldsymbol{X} = [\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots \boldsymbol{X}_L]$ without loss of generality; for each $\ell \in \{1, \ldots, L\}$, $\boldsymbol{X}_\ell$ is a collection of N_ℓ vectors that belong to subspace $\mathcal{S}_\ell$. The goal is to approximate the underlying subspaces using the points in X. We also assume the columns are normalized to have unit norm, $\|\boldsymbol{x}_i\|_2 = 1$. The notation used is summarized in Table 6.1.

We consider the intuitive *semi-random model* introduced in Soltanolkotabi et al. (2012) where the subspaces are fixed, and points are distributed randomly on each of the subspaces. To measure the notion of closeness or correlation between two subspaces, the affinity between subspaces is used.

Definition 6.1. The principal angles $\{\theta^{(i)}\}_{i=1}^{(d \wedge d')}$ between subspaces $\mathcal{S}$ and $\mathcal{S}'$Âœ of dimensions d and d'ÂIJ, are defined by

$$\cos(\theta^{(i)}) = \max_{\boldsymbol{u} \in \mathcal{S}} \max_{\boldsymbol{v} \in \mathcal{S}'} \frac{\boldsymbol{u}^T \boldsymbol{v}}{\|\boldsymbol{u}\| \|\boldsymbol{v}\|} := \frac{\boldsymbol{u}_i^T \boldsymbol{v}_i}{\|\boldsymbol{u}_i\| \|\boldsymbol{v}_i\|}$$

Table 6.1: Notation and parameters

L	Number of subspaces
d_ℓ	Dimension of each subspace for $\ell = 1 \dots L$
N_ℓ	Number of points sampled from each subspace
ρ_ℓ	Sampling density $\rho_\ell = N_\ell / d_\ell$
N	Total number of points, $N = \sum_\ell N_\ell$
n	Ambient dimension
λ	ℓ_1 component of OWL-Ramp, $\lambda > 0$
Δ	Slope of OWL-Ramp, $\Delta \geq 0$
r	Length of OWL-Ramp, $1 \leq r \leq N$
α	Normalized maximum affinity between subspaces, $0 \leq \alpha \leq 1$
k	Number of optimizations $1 \leq k \leq N$
σ	Noise level

with orthogonality constraints $\boldsymbol{u}^T \boldsymbol{u}_j = \boldsymbol{v}^T \boldsymbol{v}_j = 0, j = 1, \dots, i-1$.

Definition 6.2. The normalized affinity between subspaces is

$$\text{aff}(\mathcal{S}, \mathcal{S}') = \sqrt{\frac{\cos^2(\theta^{(1)}) + \cdots + \cos^2(\theta^{(d \wedge d')})}{d \wedge d'}}$$

The affinity is low when the subspaces are nearly orthogonal and high when the subspaces overlap significantly (it is equal to one when one subspace is contained in the other). Hence, when the affinity is high, clustering is hard whereas it becomes easier as the affinity decreases.

Performance metrics

To quantify performance of the algorithm, we use the following metrics from Soltanolkotabi et al. (2014)

Algorithm 2: OSC with k random seeds

Input : A set of data points $X \in \mathbb{R}^{n \times N}$, $k \in \{1, \ldots, N\}$.

Initialize $\boldsymbol{B} = \boldsymbol{0}_{N \times N}$.

For $i \in \{1, \ldots, k\}$,

  Randomly select an index j_i from $[N]$.

  Obtain $\widehat{\beta}$ by regressing $\boldsymbol{y} = \boldsymbol{X}_{\cdot j_i}$ onto the remaining columns of X using the OWL minimization (4.6).

  Store $\boldsymbol{B}_{\cdot j_i} = \widehat{\beta}$ with $B_{j_i, j_i} = 0$.

Form affinity matrix $\boldsymbol{W} = |\boldsymbol{B}| + |\boldsymbol{B}|^T$.

Apply spectral clustering to the Laplacian of W to obtain a partition.

Output: Subspaces $\{\mathcal{S}_\ell\}_1^L$, Cluster labels.

False discovery: Fix i and $j \in \{1, \ldots, N\}$ and let $\boldsymbol{B}$ be the outcome of Step1 in Algorithm 1. Then we say that (i, j) obeying $B_{ij} \neq 0$ is a false discovery if $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ do not belong to the same subspace.

True discovery: Similarly, we say that (i, j) obeying $B_{ij} \neq 0$ is a true discovery if $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ belong to same subspace.

6.3 Algorithm

The SSC procedure can easily be modified by replacing the sparse linear regression step with the OWL optimization in (4.6). As seen in the example with orthogonal subspaces, OWL solutions are denser due to its clustering property. This leads to an intuitive extension of the algorithm where we run only a subset of the total N possible optimizations. By running a random subset of k optimizations the computational complexity can be reduced. We formalize this version of OSC in Algorithm 2.

The intuition developed in the case of orthogonal subspaces suggests that taking $k = L$, where L is the number of subspaces may suffice for the OWL approach. Thus, the proposed algorithm requires solving $O(L)$ sparse regressions, whereas traditional SSC involves solving $O(N)$, and typically $L \ll N$. It is also worth noting that the computational complexity of lasso and OWL is essentially the same. Obviously, one could also consider reducing the computational complexity of traditional SSC by also using only $O(L)$ regressions instead of all N. However, as shown in our experiment section, SSC performance quickly degrades when the number of regressions is reduced, while the proposed OWL-based algorithm's performance does not, leading to a speedup of up to $30\times$ in some cases.

6.4 Main Results

In this section, we begin by stating the main results analyzing the behavior of the solution of the optimization (4.6). We also provide examples and remarks to understand the theoretical results along with proof sketches. The proof sketches also provide the intermediate results ranging from sparse regression with correlated variables to random geometric graph theory used to prove the main results.

We state the first main result showing that the solution of the OWL optimization does not include false discoveries if the the affinity between the subspaces is small enough, *i.e.,* the subspaces are not too close.

Let $\bar{w}_{N_\ell+1} := \frac{1}{N-N_\ell-1}\sum_{j=N_\ell+1}^{N} w_j$ be average of the tail of the OWL weights used.

Theorem 6.3. If $\mathcal{S}_\ell$, the subspace to which the i-th column belongs, obeys

$$\alpha_\ell := \max_{k:k\neq\ell} \text{aff}(\mathcal{S}_\ell, \mathcal{S}_k) \;\leq\; \kappa_0 \frac{\bar{w}_{N_\ell+1}}{w_1} \frac{\sqrt{\log N_\ell/d_\ell}}{\log N} \tag{6.2}$$

where κ_0 is a fixed numerical constant, then $\widehat{\beta}_j = 0$ for all $x_j \notin \mathcal{S}_\ell$, *i.e.*, there is no false discovery in the i-th column of B with probability at least $1 - L(4/N^2 + e^{-\sqrt{N_\ell d_\ell}})$.

Roughly stated the result says that with high probability the OWL solution contains no false discoveries if the ratio of OWL parameters is big enough for fixed subspaces. The affinity is higher for overlapping subspaces, then we must reduce the OWL weight gaps (Δ_w in Section 4.1) and in turn making the ratio of weights nearly 1. As seen in the introduction, the clustering property of OWL depends on gaps in consecutive weights. OWL clusters columns that are at most the gap away from each other. Hence, making the weight ratio close to 1 results in reducing the weight gaps and in turn reduces the radius of OWL clustering. For smaller values of affinity, the OWL gap can be made bigger and OWL will then group (assign equal coefficients to) more points. Thus, there is a tension in the closeness of the subspaces and the radius or gap of OWL clusters. For higher values of affinity, OWL with bigger weight gaps does not provide any benefit over Lasso hence we must set the weight ratio to one. The

weight ratio is exactly one for the ℓ_1 penalty and our bound for the special case of Lasso matches Soltanolkotabi et al. (2012) for SSC.

As foreshadowed in the introduction, we show that all the columns within the subspace of interest will have equal-valued coefficients if enough points are sampled, specifically $N_\ell \propto 1/\Delta_w^d$. Recall that Δ_w is the minimum gap in the OWL weights. Usually, this gap is very small, like O($1/N$) or in some cases zero, which makes the requirement infeasible. Hence, as part of the second main result, we will prove new clustering bounds for a specific type of the OWL norm called OWL-Ramp with weights

$$\begin{aligned} w_i &= (r-i+1)\Delta + \lambda && \text{for } i \in \{1,\dots,r\}, \\ &= \lambda && \text{for } i \in \{r+1,\dots,N\}, \end{aligned}$$

where $\lambda > 0$, $\Delta \geq 0$ and $1 \leq r \leq N$. Figure 4.2 depicts an example of the OWL-Ramp weights. The OSCAR regularizer Bondell and Reich (2008) is a special case obtained by setting $r = N$. Note that if $r \leq N_\ell$, then $\bar{w}_{N_\ell+1} = \lambda$ and the affinity condition in Theorem 6.3 becomes

$$\alpha_\ell \;\leq\; \kappa_0 \frac{\lambda}{\lambda + r\Delta} \frac{\sqrt{\log \rho_\ell}}{\log N}$$

Theorem 6.4. Let $\widehat{\boldsymbol{\beta}}$ be a solution of (4.6) with OWL-Ramp weights. If $r \leq N_\ell$, the conditions in Theorem 6.3 are satisfied, and the

$$N_\ell > \kappa_1 \Delta^{-d_\ell} \log(\Delta^{-d_\ell}/\delta) \tag{6.3}$$

points within the subspace of interest are sampled uniformly at random from the unit hypersphere, then the set $M = \{j : |\widehat{\beta}_j| = \max_i |\widehat{\beta}_i|\}$ has cardinality greater than or equal to the ramp parameter, r, with probability at least $1 - \delta - L(4/N^2 + e^{-\sqrt{N_\ell d_\ell}})$. ($\kappa_1$ is a fixed numeric constant.)

The result roughly says that if enough points are sampled from the corresponding subspace then with high probability the top r coefficients in the OWL solution have equal magnitude. Combining this with Theorem 6.3, if the solution to OWL optimization is non-trivial, this is equivalent to making r true discoveries.

It is easy to see that $\widehat{\boldsymbol{\beta}} = \underline{0}$, if $\Omega^*_{\boldsymbol{w}}(\boldsymbol{X}^T\boldsymbol{y}) < 1$ where $\Omega^*_{\boldsymbol{w}}(\boldsymbol{\beta})$ is the dual norm of OWL defined later in the section. In order to ensure that the solution is non-trivial we need at least $\bar{w} \leq \|\boldsymbol{X}^T\boldsymbol{y}\|_\infty$ for $\bar{w} = \sum_{j=1}^N w_j/N$. The $\|\boldsymbol{X}^T\boldsymbol{y}\|_\infty$ term scales at most like $\sqrt{(\log N)/d}$ and for OWL-Ramp, $\bar{w} \approx \lambda$. Intuitively, the ℓ_1 component needs to be made small enough to achieve a non-trivial solution.

6.5 Experiments

Numerical Experiments

In this section we present numerical results on synthetic data corroborating the theoretical guarantees and providing a better understanding of the behavior of OSC particularly we focus on the OWL-Ramp norm defined

in Section 6.4. The subspaces $\mathcal{S}_1, \mathcal{S}_2$ and $\mathcal{S}_3$ are generated with dimension $d = 20$ in $\mathbb{R}^n$ with ambient dimension $n = 40$.

Bases generation method, B1: The bases $\boldsymbol{U}_1$, $\boldsymbol{U}_2$ and $\boldsymbol{U}_3$ are obtained by choosing, uniformly at random, from the set of all sets of orthonormal vectors in $\mathbb{R}^n$. Since sum of the subspace dimensions exceeds ambient dimension, $n < 3d$, this ensures that the subspaces overlap and leads to $\alpha \approx 0.3$.

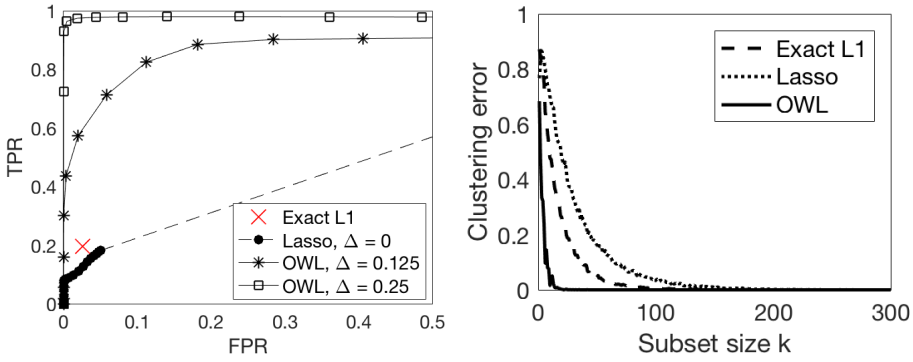
Bases generation method, B2: To generate subspaces of varying affinity, we follow the method described in Soltanolkotabi et al. (2012). The subspaces $\mathcal{S}_1$, $\mathcal{S}_2$ and $\mathcal{S}_3$ are generated using the bases

$$\boldsymbol{U}_1 = \begin{bmatrix} I_d \\ \mathbf{0}_{d\times d} \end{bmatrix}, \boldsymbol{U}_2 = \begin{bmatrix} \mathbf{0}_{d\times d} \\ I_d \end{bmatrix}, \boldsymbol{U}_3 = \begin{bmatrix} \text{diag}(\cos\boldsymbol{\theta}) \\ \text{diag}(\sin\boldsymbol{\theta}) \end{bmatrix},$$

respectively. Where the principal angles are set in such a way that the normalized affinity decreases linearly from 1 to 0.75 and diag(cos $\boldsymbol{\theta}$) corresponds to a diagonal matrix with diagonal entries equal to cos θ_i.

We select $\rho = 5$ points per subspace dimension (unless stated otherwise), *i.e*, $N_i = \rho d = 100$ points uniformly at random from each subspace using the respective bases.

Choice of hyper-parameters. See Section 6.7 for discussion on the choice of tuning parameters Δ, λ and r guided by the theory in this paper and past literature. We fix the ramp parameter, $r = N/3 = 100$, in the



(a) **(b)**

Figure 6.4: (a) Trade-off curves for FPR ≤ 0.5 generated by sweeping through (λ, Δ) values. Empirical averages of (FPR,TPR) are shown for Exact ℓ_1, Lasso, and OWL, over 100 random points. (b) Clustering error for varying k in Algorithm 2. The affinity matrix is generated for each method by running only a random subset k of the total $N = 300$ optimizations.

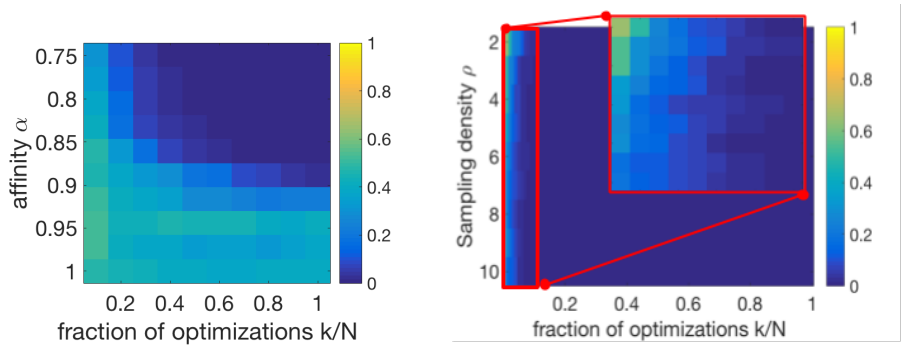


Figure 6.5: Clustering error for varying k/N in Algorithm 2 with varying affinity (left) and number of points sampled (right) for OWL regularized subspace clustering.

experiments.

The FPR-TPR trade off. In order to compare the performance of the optimizations in (6.1), (4.4) and (4.6), we generate N columns from subspaces generated using B1 to run N optimizations of each method. For

each data point, we sweep through different values of the tuning parameters (λ, Δ). For $\lambda = 0$, all points are selected and as λ is increased fewer points are given non-zero weight. Let $\widehat{\boldsymbol{\beta}}$ denote the solution to one of the optimizations. We plot the empirical averages of False Positive Rate (FPR) $= \|\widehat{\boldsymbol{\beta}}_{S^c}\|_0/|S^c|$ and True Positive Rate (TPR) $= \|\widehat{\boldsymbol{\beta}}_S\|_0/|S|$ where $\widehat{\boldsymbol{\beta}}_S$ is the part of $\widehat{\boldsymbol{\beta}}$ supported on indices of the points from the same subspace and $\widehat{\boldsymbol{\beta}}_{S^c}$ is supported on its complement. A non-zero entry in $\widehat{\boldsymbol{\beta}}_S$ is a true discovery and likewise a non-zero entry in $\widehat{\boldsymbol{\beta}}_{S^c}$ is a false discovery. By definition $|S| = N_i$ and $|S^c| = N - N_i$.

Figure 6.4 (a) shows the Receiver Operating Characteristic (ROC) curve plotting TPR versus FPR. The solution of exact ℓ_1 minimization is shown as one point on the curve since the sparsity of the solution cannot be changed by tuning parameters. Note for any $\lambda > 0$, the lasso solution will include *at most* $n = 40$ non-zero entries, since the number of selected columns will not exceed the ambient dimension. If $\lambda = 0$, then lasso selects all columns. Thus, lasso curve beyond d selections is shown as a dashed line, which extends linearly to the point (FPR,TPR) = $(1, 1)$. As suggested by theory and demonstrated by the plots, for a fixed FPR, OWL can achieve a much higher ($\mathbf{9}\times$) TPR than Lasso.

Effect of the size of the optimization subset k. To observe the effect of the size of the optimization subset k, we look at the clustering error produced by running a subset k of the total N optimizations for each method for different values of k. The subspaces are generated using B1.

We take the symmetrized affinity matrix, W, generated by each method, assuming knowledge of number of subspaces, apply a spectral clustering method to obtain the clusters. The clustering error is measured as the fraction of misclassified points from the total number of points. Figure 6.4 (b) shows how the clustering error varies with k for SSC with Exact ℓ_1, Lasso, and OWL regularized sparse linear regression steps. We report the empirical average of the clustering error over 100 random choices of subsets. By virtue of the clustering property of OWL, we see that the clustering error is low for subset sizes of the order of the number of subspaces. As demonstrated in Figure 6.4 (b), to achieve a clustering error of less than 0.01, Lasso requires **8** times more optimizations than OWL. The plots suggest that the OWL approach could potentially achieve a speedup of up to $\mathbf{20}\times$.

Effect of affinity and number of points sampled. We vary the amount of correlation between subspaces and the number of points sampled from each subspace and study its effect on the clustering error for different values of k in Figure 6.5. The subspaces with varying affinity are generated using the method described in B2 and subspaces for varying ρ are generated using B1. The affinity is varied in the range $\alpha \in [0.75, 1]$ where $\alpha = \max_\ell \alpha_\ell$ and we vary the sampling density $\rho \in [2, 10]$. Recall that the affinity is low when the subspaces are nearly orthogonal and high when the subspaces overlap significantly (it is equal to one when one subspace is contained in the other). The tuning parameters are fixed throughout the

experiment. As expected, the clustering error increases for higher values of affinity and OWL ($\Delta = 0.01$) provides no benefit over Lasso ($\Delta = 0$). On the other hand, OWL produces small values of error for most values of ρ.

Real Data Experiments

We compare our algorithm with the existing ones in the applications of motion segmentation Elhamifar and Vidal (2009); Tron and Vidal (2007) and clustering handwritten digits Hastie and Simard (1998).

For motion segmentation, we used Hopkins155 dataset Tron and Vidal (2007). It contains 155 video sequences of 2 or 3 motions. Table 6.2 summarizes clustering errors of different algorithms on the Hopkins155 dataset. For ease of comparison, the error values for TSC, SSC-OMP and NSN are populated from experiments conducted in Park et al. (2014) where the authors optimized the parameters for the existing algorithms. The parameters for SSC were set as provided in the source code. The OWL parameters are set as follows: for all real data experiments we use the same λ value as SSC, we set $r = N/4$ (rounded off to nearest integer) and set Δ such that $w_1 = \lambda + r\Delta = 2\lambda$ or 4λ, whichever, if any, leads to a non-trivial solution. Figure 6.6(a) shows the effect of running a subset of k optimizations selected at random from the set of all N optimizations for $k \in \{N, N/2, N/4, \ldots, N/32\}$. OSC outperforms or performs about the same as SSC in most cases with comparable running times. The SSC

Table 6.2: Clustering error (%) of different algorithms on the Hopkins 155 dataset.

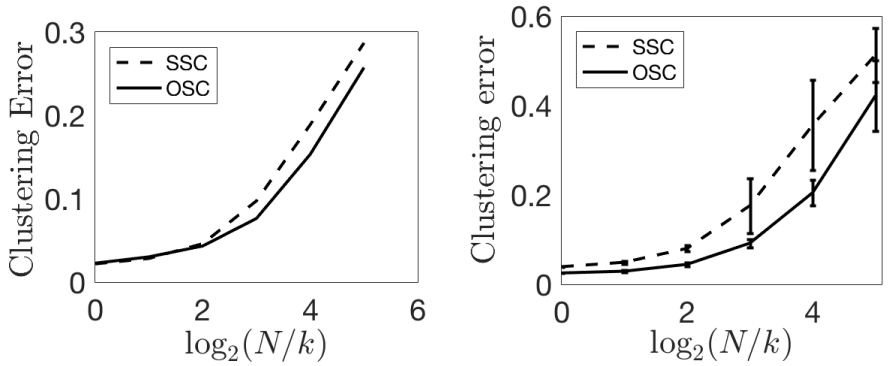
L	Algorithm	SSC	OMP	TSC	NSN	OSC
2	Mean	1.52	16.92	18.44	3.62	1.39
	Median	0.00	12.77	16.92	0.00	0.00
3	Mean	4.40	27.96	28.58	8.28	5.21
	Median	0.56	30.98	29.67	2.76	1.02

and OSC parameters in this application lead to relatively dense affinity matrices in both cases. This results in low clustering errors with fewer optimizations supporting the theory. Table 6.3 summarizes results of applying the SSC and OSC algorithms to the dataset projected into a $4L$-dimensional subspace using PCA suggesting further speedups without loss in performance.

We also use the MNIST test data set LeCun et al. that contains 10,000 centered 28 $\times$ 28 pixel images of handwritten digits, *i.e.,* it contains many more points per subspace. The empirical mean and standard deviation of the CE are computed by averaging over 100 of the following problem instances. We choose the digits {2, 4, 8} and we choose k optimizations uniformly at random from the set of all optimizations for $k \in \{N, N/2, N/4, \ldots, N/32\}$. Here we use the default parameters for SSC and set OWL parameters in similar fashion to the motion segmentation experiments. The results are summarized in Figure 6.6(b).

Table 6.3: Mean clustering error (%) on the Hopkins 155 dataset with the $4L$-dimensional data points obtained by applying PCA.

Motions (L)	SSC	OSC
2	1.83	1.49
3	4.40	5.16
All	2.41	2.32



(a) **(b)**

Figure 6.6: (a) Mean clustering error for Hopkins dataset. (b) Mean clustering error for MNIST dataset.

6.6 Proofs

In the rest of the chapter, we state the important intermediate results used to prove the main results and provide proof sketches. The intermediate results are stated in generality since they may be useful in other settings.

Proof of Theorem 6.3

We start by proving a deterministic lemma that introduces the OWL dual feasibility condition. First we define the dual norm of the OWL norm

given by Zeng and Figueiredo (2014)

$$\Omega^*_{\boldsymbol{w}}(\boldsymbol{\beta}) = \max\{\tau_i \|\beta_{(i)}\|_1, i = 1, \ldots, N\}$$

where $\beta_{(i)}$ is the sub-vector of β, consisting of the i largest magnitude elements of $\boldsymbol{\beta}$ and $\tau_i = (\sum_{j=1}^{i} w_j)^{-1}$.

Lemma 6.5. Fix $\boldsymbol{X} \in \mathbb{R}^{n \times N}$ and $\mathcal{T} \subset \{1, \ldots, N\}$. Suppose β^* is a solution to

$$\min_{\boldsymbol{\beta}} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \Omega_{\boldsymbol{w}}(\boldsymbol{\beta}) \text{ subject to } \boldsymbol{\beta}_{\mathcal{T}^c} = 0$$

obeying $\Omega^*_{\boldsymbol{w}'}(\boldsymbol{X}^T_{\mathcal{T}^c}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}^*)) < 1$ where $\boldsymbol{w}' = [w_{|\mathcal{T}|+1}, \ldots, w_N]$. Then any optimal solution $\widehat{\boldsymbol{\beta}}$ to

$$\min_{\boldsymbol{\beta}} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \Omega_{\boldsymbol{w}}(\boldsymbol{\beta})$$

must also satisfy $\widehat{\boldsymbol{\beta}}_{\mathcal{T}^c} = 0$.

Proof details are provided in the Appendix. The lemma says that if the OWL dual feasibility condition, $\Omega^*_{\boldsymbol{w}'}(\boldsymbol{X}^T_{\mathcal{T}^c}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}^*)) < 1$, is satisfied for $\mathcal{T} = \{j : \boldsymbol{X}_j \in \mathcal{S}_\ell\}$, then there are no false discoveries. To prove Theorem 6.3, it suffices to show that the dual feasibility condition is satisfied. A sufficient condition to satisfy this dual feasibility condition is $\|\boldsymbol{X}^T_{\mathcal{T}^c}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}^*)\|_\infty < \bar{w}_{|\mathcal{T}|+1}$ since it can be shown easily that for $\bar{w} = \sum_{j=1}^{N} w_j/N$, we have $\frac{1}{w_1}\|\boldsymbol{\beta}\|_\infty \leq \Omega^*_w(\boldsymbol{\beta}) \leq \frac{1}{\bar{w}}\|\boldsymbol{\beta}\|_\infty$.

We want to show the OWL dual feasibility is satisfied. Using Theorem 7.5 in Soltanolkotabi et al. (2012), we can show that,

$\|\boldsymbol{X}_j^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}^*)\|_\infty \leq \sqrt{32}\log N \frac{\text{aff}(\mathcal{S}_\ell, \mathcal{S}_j)}{\sqrt{d_\ell}}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}^*\|_2$.

To show the dual feasibility is satisfied we use Theorem 7.5 in Soltanolkotabi et al. (2012) as follows. Suppose $\boldsymbol{\Sigma} = \boldsymbol{U}^{(j)^T}\boldsymbol{U}^{(\ell)}$, where $\boldsymbol{U}^{(j)}$ is an orthogonal basis for $\mathcal{S}_j$ and $\boldsymbol{U}^{(\ell)}$ for $\mathcal{S}_\ell$ respectively. By definition, $\|\Sigma\|_F = \sqrt{d_\ell \wedge d_j}\text{aff}(\mathcal{S}_j, \mathcal{S}_\ell)$. Consider

$$\|\boldsymbol{X}_j^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}^*)\|_\infty = \|\boldsymbol{A}^T\boldsymbol{\Sigma}\boldsymbol{v}\|_\infty\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}^*\|_2.$$

Using the Lemma with $a = N^4, b = N^8$, we have the above probability at least $1 - 4/N^2$.

We next prove a bound on the size of the residual $\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}^*\|$ where $\boldsymbol{\beta}^*$ is defined in Lemma 6.5.

Lemma 6.6. If $w_1 > 0$, then with probability at least $1 - e^{-\sqrt{N_\ell d_\ell}}$, we have $\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}^*\|_2 \leq cw_1\sqrt{\frac{d_\ell}{\log N_\ell/d_\ell}}$.

Proof details are provided in the Appendix. Lemma 6.6 gives with high probability,

$$\|\boldsymbol{X}_j^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}^*)\|_\infty \leq w_1 L_N \text{aff}(\mathcal{S}_\ell, \mathcal{S}_j)$$

where $L_N = c_0\frac{\log N}{\sqrt{\log \rho_\ell}}$, for $j \neq \ell$. Finally using the assumption on the affinity of the subspaces along with union bound and Lemma 6.5 completes the proof.

Proof of Theorem 6.4

The N_ℓ points on the unit hypersphere can be viewed as forming a d_ℓ-dimensional Random Geometric Graph by placing an edge between points that are Δ-close. To formalize this notion we define a Δ-RGG on the unit hypersphere below (slightly different from the typical RGG defined in the literature Dall and Christensen (2002); Penrose (2003)).

Definition 6.7. Δ-Random Geometric Graph (RGG): Place N_i points uniformly at random on the surface of unit hypersphere $\mathbb{S}^{d_i}$. Place an edge between pairs of points (vertices) that are within Δ Euclidean distance of each other.

It can be shown that if enough points are sampled uniformly at random from the unit sphere leads to a fully connected Δ-RGG with high probability. The connectivity and percolation of a typical RGG has been studied (Penrose (2003) and references therein) in the past but the proof of the main result hinges on a fully connected graph defined on the unit hypersphere. Hence, we derive a new bound next.

Lemma 6.8. If $N_\ell > \kappa_1 \Delta^{-d_\ell} \log(\Delta^{-d_\ell}/\delta)$ points are sampled uniformly at random from the unit hypersphere in $\mathbb{R}^{d_i}$, then the Δ-RGG formed by these points is fully connected with probability at least $1 - \delta$.

κ_1 is a fixed numeric constant. To prove the result we use a covering argument to divide the surface of the unit hypersphere into m equal area

patches such that the distance between points in adjacent patches is at most Δ, followed by a multiplicative form of Chernoff's bound to show with high probability at least one point of the uniformly sampled points falls into each patch leading to a fully connected Δ-RGG. Proof details are provided in an extended version of the paper on arXiv Oswal and Nowak (2018). Using the fact that the Euclidean norm is invariant under multiplication with orthonormal matrix, the distances between points within the subspace can be translated to the ambient subspace. Similarly, we can define a Δ-RGG on the unit hypersphere in $\mathbb{R}^n$.

We prove a new clustering property for the OWL-Ramp regression in Lemma 4.2. Proof details are provided in an extended version of the paper on arXiv Oswal and Nowak (2018) and the Appendix. This lemma provides a sufficient condition for the largest magnitude cluster in the OWL-Ramp solution to have critical mass. The Euclidean distance condition from Lemma 4.1 translates into the Δ-connected component condition in this Lemma. Note that the minimum gap Δ_w is always smaller than or equal to Δ. It is strictly smaller in most cases. This provides more room for clustering in the OWL solution.

The proof of Theorem 6.4 follows by combining the results from Lemma 4.2 and Lemma 6.8 with Theorem 6.3 as follows. Let $\mathcal{T}$ be set of indices of columns belonging to the subspace, $\mathcal{S}_\ell$. From Theorem 6.3, we have with high probability the coefficients of $\widehat{\boldsymbol{\beta}}_{\mathcal{T}^c} = 0$. Lemma 6.8 with the assumption on the number of points sampled gives us that the Δ-RGG formed by

the points $\boldsymbol{X}_{\mathcal{T}}$ on the unit hypersphere is fully connected. The connected component has cardinality at least N_ℓ. If $\max_i |\widehat{\beta}_i| = 0$, then the claim is trivial so suppose $\max_i |\widehat{\beta}_i| > 0$, then $M \subseteq \mathcal{T}$ and the claim follows from $r \leq N_\ell$ and Lemma 4.2.

6.7 Discussion

To make sense of the results, the condition in (6.2) can be rewritten as an upper bound on Δ and the condition in (6.3) as a lower bound on Δ. Ignoring constants and $\log$ terms

$$\Delta \lesssim \frac{\lambda}{r}\left(\frac{1}{\alpha_\ell} - 1\right) \text{ and } \Delta \gtrsim \left(\frac{1}{N_\ell}\right)^{1/d_\ell}$$

Intuitively, we want to make Δ small enough so that nothing from outside of the true subspace is selected, but at the same time we want to make Δ as big as possible so many points are selected from the true subspace. This leads to a trade-off between the number of false discoveries and true discoveries. For orthogonal subspaces or $\alpha_\ell = 0$, the upper bound is trivially satisfied and we can make Δ big enough to group all the points in the same subspace as seen in the introduction. For $\alpha_\ell \approx 1$ or when the subspace is contained in another subspace, Δ needs to be set to zero to ensure no false discoveries. Equivalently, the best we can do in this situation is the Lasso solution. Guided by the theory we state rules of

thumb for selecting the parameters of OSC followed in the experiments.

Choice of hyper-parameters.

The λ and Δ parameters are varied in Section 6.5 to demonstrate the trade off between the number of false discoveries and true discoveries. In some applications like motion segmentation, the dimensions of the subspaces are equal and known in advance or can be roughly estimated. In such cases we recommend setting the tuning parameters as follows.

- **Ramp length**, r: In general, we recommend setting r proportional to an estimate of number points per cluster since this parameter is related to the number of points clustered by OWL.

- ℓ_1 **component**, λ: Informed by theory and experiments in this chapter and from SSC literatureSoltanolkotabi et al. (2014), we set λ proportional to $1/\sqrt{d}$.

- **Ramp slope or gap**, Δ: Since the gap in the OWL weights control the clustering behavior the theory suggests setting $\Delta \approx N_i^{-1/d}$ for many true discoveries. As the affinity of the subspaces increases Δ needs to be reduced in order to avoid false discoveries. In the experiments, we observe that even small values such as $\Delta = 0.01$ lead to enough clustering in OWL solutions given that there are enough points in each subspace.

The clustering property of the OWL, combined with ideas from random geometric graph theory, allows us to prove that OSC tends to select more points from the correct subspaces in each regression compared to SSC. In the ideal case, where L subspaces are orthogonal and the number of points per subspace is sufficiently large, then OSC can succeed with just $L \ll N$ optimizations (gain factor of roughly N/L). This key feature of OSC makes accurate clustering possible based on regression solutions for only a small subset of the total dataset, significantly reducing the computational complexity compared to SSC.

7 APPLICATIONS AND FUTURE DIRECTIONS

7.1 Applications

Content-based image retrieval (CBIR) is an area of computer vision dealing with the task of finding desired images within a large corpus. For example, someone could be searching for similar images to a certain "seed" image, or searching for images with some specific content (e.g. a product or animal). CBIR refers to algorithms that approach this task using only the data within the image itself, as opposed to a more general image retrieval approach which could include metadata or natural-language descriptions in its search.

The general industry applications of CBIR are numerous. For this chapter, we consider the following task. Often, a data scientist can easily acquire or train an image classification model for some general-purpose task, either by using a publicly available dataset or using a pretrained model. This model may not solve their exact problem, but may solve a very similar or more general version of the problem. Generally in these situations, the practitioner may employ a technique like transfer learning Pan and Yang (2010) to take this more general model and make it applicable to their domain to extract task-specific feature vectors with a small number of labeled images. These features can then be used to build our linearly parameterized bandits framework to retrieve specific types of images on demand.

For example, we apply the same technique to a large corpus of roof images obtained from inspections as part of the process of obtaining home insurance. The roof images are labeled generally as "good condition" and "bad condition", and our goal is to leverage these labels to generate a dataset with labels indicating more specific defects, such as "missing shingle(s)". The algorithm recommends images from the corpus using the linear bandits algorithms. The user provides rewards specific to the task at hand which the algorithm uses to find other similar images from the corpus.

Application to Insurance Industry

American Family Insurance has a corpus of about 400,000 images of roofs obtained from roof surveys as part of the home inspection process. As part of the survey, the roofs are graded by experts on a scale from 1 to 4, indicating condition. Using this survey, we are able to train a deep learning model (using the VGG16 architecture) to predict which roofs have "good" vs. "poor" condition.

In this application, we wish to find all the images that have the quality "missing shingle(s)". Figure 7.1 shows an example image with that quality. Since this condition is rare (in our experiments, about 2% of the images in the "poor" condition had a missing shingle), we use linear bandits to most efficiently use human labeling resources. We treat the model trained from the good/poor labels as a feature extractor, treating the last two

Figure 7.1: An example image from a roof survey that would get a positive reward for the "missing shingle" task in our experiment.

fully-connected layers of the network as an embedding for the images. We then use these features for the linear bandits process.

As mentioned in the previous chapter, linear bandits suffer from the curse of dimensionality. In this work, we explore various feature-selection routines to alleviate the high-dimensionality problem by selecting a small subset of relevant features. In our examples, we use the VGG16 architecture Liu and Deng (2015), which has two fully connected layers with 4096 neurons each. These layers probably contain redundant and unnecessary information.

To alleviate the problem posed by these high-dimensional features, we introduce a feature selection subroutine for each query. The goal is to select a small subset of features that are most relevant to our search to increase the effectiveness of our image selection algorithm. By selecting a smaller number of features, a bias is created towards simpler models. The simple models lead to more interpretable solutions. We use the following

two feature-selection methods:

1. LASSO: Tibshirani (1996), introduced in Chapter 4, constrains the solutions to be sparse, meaning only a few variables are selected. This is done by using the ℓ_1 penalty which is a convex proxy to the ℓ_0 norm representing cardinality of the coefficient vector. In practice, λ is chosen via cross-validation.

2. Marginal Regression: Genovese et al. (2009, 2012) offers a faster alternative to the Lasso regularization since it regresses the label vector separately on each feature. The marginal regression estimates for feature selection are computed using the following coefficients. Assuming the features are standardized,

$$\widehat{\alpha} := X^T y.$$

Using a tuning parameter, $\tau > 0$, we estimate the subset of relevant features by

$$\widehat{S} := \{j : |\widehat{\alpha}_j| \geq \tau\}$$

Roughly, this amounts to picking the top features that are most correlated with the reward or label vector. In practice, we choose the top $n/\log(n)$ $\hat{\alpha}$ with the largest magnitude instead of specifying a τ. We experimented with tuning τ through cross-validation but the performance difference was negligible.

Application to Fashion Retail Industry

Lands' End is a Wisconsin based clothing retail company. Their catalogs consist of thousands of fashion retail products. Similar to the insurance company, we work on content-based retrieval of products from the catalog using deep convolutional features extracted from fashion-tuned CNN using transfer learning.

7.2 Future Directions

1. The simple feedback model, where the user directly selects a subset of the relevant features, can be generalized by allowing for an indirect form of feedback. For example, the user can select a region of an image instead of a subset of the standard deep neural network features. This form of feedback could be used in different ways. For instance, we could use methods to map deep image features to image regions. This would also allow the use of neural network based representations of text used in NLP such as BERT, Devlin et al. (2018). A mapping of the highlighted words or phrases to the BERT feature dimensions could facilitate using the feature feedback model in this case.

2. In Chapter 2, we see that in the short-term horizon, when the algorithm does not have enough feedback, it starts with a small subset

of the dimensions and gradually grows the number of dimensions as it receives new feedback. This leads to better early-regret. The choice of the dimensions is based on feature feedback provided by the user. This could be generalized by combining ideas from compressed sensing and/or dimensionality reduction, alleviating the need for feature feedback from the user in the future.

In this chapter, arm and action are used interchangeably.

Feature Feedback Epoch OFUL

This second algorithm, Feature Feedback Epoch OFUL (Algorithm 3), is an epoch version of Algorithm 2 which runs in epochs of doubling length so the last epoch dominates the regret. It is essentially the same as Algorithm 2 written in a different format which facilitates proving the main result. The main difference in the algorithms is the choice of ϵ_t depicted in Figure A.1.

Algorithm 3 Feature Feedback Epoch OFUL

Let the set of relevant indices, $\mathcal{R}_0, \mathcal{I}_0 = \{\}$.
while $\mathcal{I}_0$ is empty **do**
  Pull arm at random, $\mathcal{I}_0 = \{$ indices revealed $\}$
end while
$\mathcal{R}_1 = \mathcal{R}_0 \bigcup \mathcal{I}_0$
Initialize $\mathcal{C}_0$.
for $s = 1, 2, \ldots, M-1$ **do**
  Set $\epsilon_s = c/\sqrt{2^s}$
  Let $\boldsymbol{X}_s$ be the original feature matrix with only the features in $\mathcal{R}_s$.
  for $t = 1, \ldots, 2^s$ **do**
    Draw b_t from bernoulli(ϵ_s)
    if $b_t = 1$ **then**
      Pick an arm $\mathbf{x}_t$ uniformly at random from $\mathcal{X}$,
    else
      Pick arm $\mathbf{x}_t$ such that $(\mathbf{x}_t, \tilde{\boldsymbol{\theta}}_t) = \text{argmax}_{(\mathbf{x},\boldsymbol{\theta}) \in \mathcal{X}_t \times \mathcal{C}_{t-1}} \langle \mathbf{x}, \boldsymbol{\theta} \rangle$
    end if
    Play arm $\mathbf{x}_t$ to observe reward y_t and indices revealed for this arm, $\mathcal{I}_t$.
    Update $\mathcal{R}_s = \mathcal{R}_s \bigcup \mathcal{I}_t$
    if $\mathcal{I}_t$ is empty **then**
      Rank one update to OFUL confidence set $\mathcal{C}_t$ using $(y_t, \mathbf{x}_t)$
    else
      Update $\boldsymbol{X}_s$ with features in $\mathcal{R}_s$
      Recompute the OFUL confidence set $\mathcal{C}_t$ with new feature set $\boldsymbol{X}_s$.
    end if
  end for
  $\mathcal{R}_{s+1} = \mathcal{R}_s$
  $\mathcal{C}_1 = \mathcal{C}_t$
end for

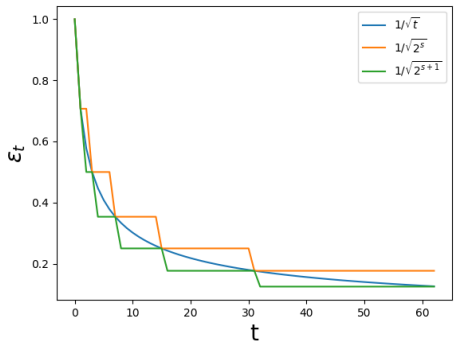


Figure A.1: Choice of ϵ_t for both the algorithms. $s = \lfloor \log_2 t \rfloor$ Recall that ϵ_t. controls the number of pure exploration steps in the algorithms.

A.1 Proof of Theorem 2.2

We begin by proving intermediate results for three different events followed by the proof details.

1. The number of times we pull a random arm during an epoch is close to its expectation.

2. We have seen all the relevant arms before the current epoch.

3. Modified OFUL regret bound using arms from both exploration and exploitation.

Bounding the number of times we pull a random arm

Lemma A.1. *During epoch s, there are $T_s = 2^s$ time steps. Let N_s be the number of random arm pulls during epoch s. Given that the probability of pulling a*

random arm during epoch s is $\epsilon_s = c/\sqrt{T_s}$, then for any $\delta_1 > 0$:

$$\Pr\left(\left|N_s - c\sqrt{T_s}\right| \geq \sqrt{\frac{T_s}{2}\log\frac{2}{\delta_1}}\right) \leq \delta_1$$

Proof. We can see N_s as the sum of T_s i.i.d. Bernoulli random variables with probability of success of ϵ_s. It is easy to see that $\mathbb{E}N_s = T_s \cdot c/\sqrt{T_s} = c\sqrt{T_s}$. Finish by applying the Hoeffding's inequality to the sum of the Bernoulli random variables. □

Corollary A.1. *With probability $\geq 1 - \delta_1$:*

$$\sqrt{\frac{T_s}{2}\log\frac{2}{\delta_1}} \leq N_s \leq 3\sqrt{\frac{T_s}{2}\log\frac{2}{\delta_1}}$$

Proof. This is a simple consequence of taking $c = \sqrt{2\log\frac{2}{\delta_1}}$ in Lemma A.1. □

Probability of having identified all the relevant arms

Proposition A.2. *Let $\alpha_0 = \sqrt{2}$ and $\alpha_i = \sqrt{\frac{T_s}{2}} = \sqrt{2^{i-1}}$ for $i > 0$. Then:*

$$\sum_{i=0}^{s-1} \alpha_i \geq \sqrt{2^s}$$

Proposition A.3. *The number of random arms pulled before an epoch s can be bounded as:*

$$\sqrt{2^s \log \frac{2}{\delta_1}} \leq \sum_{i=0}^{s-1} N_i \leq 3\sqrt{2^s \log \frac{2}{\delta_1}}$$

with probability $\geq 1 - s\delta_1$.

Proof. This is a direct result of Corollary A.1 and Proposition A.2. □

Definition A.2. *Let E_s^j be a random variable:*

$$E_s^j = \begin{cases} 1 & \text{if the } j \text{ marked as relevant up till epoch } s \\ 0 & \text{otherwise} \end{cases}$$

Let $\bigcap_{j=1}^{k} E_s^j = 1$ be the event that all the relevant features are marked.

Proposition A.4. *The probability that we have not seen all the relevant arms goes down quickly. Here we characterize how quickly. Note the assumption here that at every around, we assume that each relevant feature is revealed with some probability at least p independent of other relevant features.*

The probability that some of the k relevant features have not been marked up to epoch s, $\Pr(E_s = 0)$ is bounded as follows.

$$\Pr(E_s = 0) \leq k \exp\left(-\log \frac{1}{1-p} \sum_{i=0}^{s-1} N_i\right)$$

Proof. The proof follows by union bound.

$$\begin{aligned}
\Pr(E_s = 0) &= \Pr\left(\bigcap_{j=1}^{k} E_s^j = 1\right)^c \\
&= \Pr\left(\bigcup_{j=1}^{k} E_s^j = 0\right) \\
&\leq \sum_{j=1}^{k} \Pr(E_s^j = 0) \\
&\leq k(1-p)^{\sum_{i=0}^{s-1} N_i} \\
&= k \exp\left(-\log \frac{1}{1-p} \sum_{i=0}^{s-1} N_i\right)
\end{aligned}$$

□

Now we can find the number of epochs that need to pass after which we have observed all the features with high probability:

Proposition A.5. *After:*

$$s = \left\lceil \log_2 \left(\frac{1}{\log 2/\delta_1} \left(\frac{\log k/\delta_2}{\log 1/(1-p)} \right)^2 \right) \right\rceil := s_{observed}$$

epochs, we have observed all the relevant features with probability $\geq 1 - \delta_2$.

Proof.

$$
\begin{aligned}
\Pr(E_s = 0) &\leq k \exp\left(-\log \frac{1}{1-p} \sum_{i=0}^{s-1} N_i\right) \\
&\leq k \exp\left(-\log \frac{1}{1-p} \sqrt{2^s \log \frac{2}{\delta_1}}\right) \\
&\leq \delta_2 \text{ , we desire this} \\
&\Rightarrow \exp\left(-\log \frac{1}{1-p} \sqrt{2^s \log \frac{2}{\delta_1}}\right) \leq \frac{\delta_2}{k} \\
&\Rightarrow \log \frac{1}{1-p} \left(\sqrt{2^s \log \frac{2}{\delta_1}}\right) \geq \log \frac{k}{\delta_2} \\
&\Rightarrow \sqrt{2^s \log \frac{2}{\delta_1}} \geq \frac{\log k/\delta_2}{\log 1/(1-p)} \\
&\Rightarrow 2^s \geq \frac{1}{\log 2/\delta_1} \left(\frac{\log k/\delta_2}{\log 1/(1-p)}\right)^2 \\
&\Rightarrow s \geq \log_2 \left(\frac{1}{\log 2/\delta_1} \left(\frac{\log k/\delta_2}{\log 1/(1-p)}\right)^2\right) \\
&\Rightarrow s_{observed} = \left\lceil \log_2 \left(\frac{1}{\log 2/\delta_1} \left(\frac{\log k/\delta_2}{\log 1/(1-p)}\right)^2\right) \right\rceil \geq s
\end{aligned}
$$

□

Regret for a modification of OFUL after epoch $s_{observed}$

We cannot use the OFUL regret bound directly since our algorithm involves additional random arms sampled during the epoch along with arms sampled in previous epochs. To bound the regret of arms pulled using OFUL, we prove the following regret bound for the modified OFUL

algorithm, stated as Algorithm 4, where some additional arms are sampled in addition to the OFUL ones:

Lemma A.3. *Assume that* $\forall t > 0$ *and* $\mathbf{x} \in \mathcal{X}_t \subset \mathbb{R}^d$, $\langle \mathbf{x}, \boldsymbol{\theta}_* \rangle \in [-1, 1]$. *Then with probability at least* $1 - \delta$, *the regret of Extended OFUL (Algorithm 4) satisfies:*

$$\forall t, R_t \leq 4\sqrt{td\log(\lambda + tL/d)}(\lambda^{1/2}S + R\sqrt{2\log(1/\delta) + d\log(1 + tL/(\lambda d))t})$$

where $\lambda > 0$ *is the ridge regression parameter of OFUL.*

This lemma shows that the additional arms sampled between of OFUL turns do not harm the regret of OFUL.

Algorithm 4 Extended OFUL

1: Begin with some initial arms $\boldsymbol{X}_0$ which could be empty.
2: **for** $t = 1, 2, \ldots, T-1$ **do**
3: $(\mathbf{x}_t, \tilde{\boldsymbol{\theta}}_t) = \text{argmax}_{(\mathbf{x},\boldsymbol{\theta}) \in \mathcal{X}_t \times \mathcal{C}_{t-1}} \langle \mathbf{x}, \boldsymbol{\theta} \rangle$
4: Play $\mathbf{x}_t$ and receive reward y_t.
5: Update $\boldsymbol{X}_t, \overline{\mathbf{V}}_t = (\boldsymbol{X}_t^T \boldsymbol{X}_t + \lambda \boldsymbol{I})$ and $\widehat{\boldsymbol{\theta}}_t = \overline{\mathbf{V}}_t^{-1} \boldsymbol{X}_t^T \boldsymbol{y}_t$
6: Update ellipsoidal confidence set $\mathcal{C}_t$ as $\mathcal{C}_t = \left\{ \mathfrak{p} \in \mathbb{R}^d : \|\widehat{\boldsymbol{\theta}}_t - \mathfrak{p}\|_{\overline{\mathbf{V}}_t} \leq f(R, \boldsymbol{X}_t, \boldsymbol{y}_t, \lambda, \delta, S) \right\}$ (for details on $f(\cdot)$ see Abbasi-Yadkori et al. (2011))
7: Add some arms (randomly or otherwise) to the set $\boldsymbol{X}_t$.
8: **end for**

We will require the following result to prove the theorem.

Proposition A.6. *For symmetric positive definite matrices* $\boldsymbol{W}$, $\boldsymbol{Q}$ *and* $\mathbf{V} = \boldsymbol{W} + \boldsymbol{Q}$, *we have*

$$\|\mathbf{x}\|_{\mathbf{V}^{-1}} \leq \|\mathbf{x}\|_{\boldsymbol{W}^{-1}}$$

Proof. Let $\boldsymbol{y} = \boldsymbol{W}^{-1/2}x$. Then we have

$$\begin{aligned}
\|\mathbf{x}\|^2_{\mathbf{V}^{-1}} &= \mathbf{x}^T\mathbf{V}^{-1}\mathbf{x} \\
&= \mathbf{x}^T(\boldsymbol{W}+\boldsymbol{Q})^{-1}\mathbf{x} \\
&= \boldsymbol{y}^T\boldsymbol{W}^{1/2}(\boldsymbol{W}+\boldsymbol{Q})^{-1}\boldsymbol{W}^{1/2}y \\
&= \boldsymbol{y}^T(\boldsymbol{I}+\boldsymbol{W}^{-1/2}\boldsymbol{Q}\boldsymbol{W}^{-1/2})^{-1}y \quad \text{Let } \boldsymbol{W}^{-1/2}\boldsymbol{Q}\boldsymbol{W}^{-1/2} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{U}^T \\
&= \boldsymbol{y}^T(\boldsymbol{U}\boldsymbol{U}^T+\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{U}^T)^{-1}\boldsymbol{y} \qquad\qquad \boldsymbol{U}\boldsymbol{U}^T = \boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I} \\
&= \boldsymbol{y}^T\boldsymbol{U}^T(\boldsymbol{I}+\boldsymbol{\Sigma})^{-1}\boldsymbol{U}\boldsymbol{y} \\
&\leq \boldsymbol{y}^T\boldsymbol{U}^T\boldsymbol{I}\boldsymbol{U}\boldsymbol{y} \qquad\qquad \text{since } \forall i, \frac{1}{1+\sigma_i} \leq 1 \\
&\leq \boldsymbol{y}^T\boldsymbol{y} \\
&= \|\mathbf{x}\|^2_{\boldsymbol{W}^{-1}}
\end{aligned}$$

□

The remaining proof follows the proof of Theorem 3 and we state it here for the sake of completeness.

Proof. Let $\overline{\boldsymbol{X}}_t = [\mathbf{x}_1^T, \ldots, \mathbf{x}_t^T], \overline{\boldsymbol{W}}_t = (\overline{\boldsymbol{X}}_t^T\overline{\boldsymbol{X}}_t + \lambda\boldsymbol{I})$.

We will follow the proof of Theorem 3 in [Abbasi-Yadkori et al. (2011)] which is divided into 2 parts: first they prove that with high probabiliy $\boldsymbol{\theta}_*$ lies inside the confidence set constructed by OFUL at that time. Notice that the super martingale arguments used to prove that $\boldsymbol{\theta}_*$ is inside the confidence set with high probability do not make an assumption on how the previous arms were sampled so the argument goes through without any modification.

As in Abbasi-Yadkori et al. (2011) we can decompose the instantaneous regret as follows:

$$\begin{aligned}
r_t &= \langle \boldsymbol{\theta}_*, \mathbf{x}_* \rangle - \langle \boldsymbol{\theta}_*, \mathbf{x}_t \rangle \\
&\leq \langle \tilde{\boldsymbol{\theta}}_t, \mathbf{x}_t \rangle - \langle \boldsymbol{\theta}_*, \mathbf{x}_t \rangle \\
&= \langle \tilde{\boldsymbol{\theta}}_t - \boldsymbol{\theta}_*, \mathbf{x}_t \rangle \\
&= \langle \widehat{\boldsymbol{\theta}}_{t-1} - \boldsymbol{\theta}_*, \mathbf{x}_t \rangle + \langle \tilde{\boldsymbol{\theta}}_t - \widehat{\boldsymbol{\theta}}_{t-1}, \mathbf{x}_t \rangle \\
&= \|\widehat{\boldsymbol{\theta}}_{t-1} - \boldsymbol{\theta}_*\|_{\overline{\mathbf{V}}_{t-1}^{-1}} \|\mathbf{x}_t\|_{\overline{\mathbf{V}}_{t-1}^{-1}} + \|\tilde{\boldsymbol{\theta}}_t - \widehat{\boldsymbol{\theta}}_{t-1}\|_{\overline{\mathbf{V}}_{t-1}^{-1}} \|\mathbf{x}_t\|_{\overline{\mathbf{V}}_{t-1}^{-1}} \\
&\leq 2\sqrt{\beta_{t-1}(\delta)} \|\mathbf{x}_t\|_{\overline{\mathbf{V}}_{t-1}^{-1}}
\end{aligned}$$

where we use the fact that $(\tilde{\boldsymbol{\theta}}_t, \mathbf{x}_t)$ is optimistic and that $\widehat{\boldsymbol{\theta}}_t, \tilde{\boldsymbol{\theta}}_t, \boldsymbol{\theta}_*$ all lie in the confidence set with high probability. Thus with probability at least $1 - \delta$, for all $t \geq 0$

$$R_t \leq \sqrt{t \sum_{s=1}^{t} r_t^2} \leq \sqrt{8\beta_t(\delta) t \sum_{s=1}^{t} \|\mathbf{x}_t\|_{\overline{\mathbf{V}}_{t-1}^{-1}}} \leq \sqrt{8\beta_t \delta t \sum_{s=1}^{t} \|\mathbf{x}_t\|_{\overline{\boldsymbol{W}}_{t-1}^{-1}}}$$

where we used Proposition A.6 stated above.

By Lemma 11 in Abbasi-Yadkori et al. (2011) we have,

$$\begin{aligned} R_t &\leq \sqrt{8\beta_t(\delta) t \sum_{s=1}^{t} \|\mathbf{x}_t\|_{\overline{\boldsymbol{W}}_{t-1}^{-1}}} \\ &\leq \sqrt{8\beta_t \delta t \log(\det(\overline{\boldsymbol{W}}_t))} \\ &\leq 4\sqrt{td \log(\lambda + nL/d)} \left(\lambda^{1/2} S + R\sqrt{2\log(1/\delta) + d\log(1 + tL/(\lambda d))}\right) \end{aligned}$$

□

Regret after epoch $s_{observed}$

During each epoch after $s_{observed}$, we have at most $3\sqrt{\frac{T_s}{2} \log \frac{2}{\delta_1}}$ random arm pulls.

Lemma A.4. *For epochs $s \geq \boldsymbol{s}_{observed}$, the cumulative regret is bounded by:*

$$\begin{aligned} \sum_{s=\boldsymbol{s}_{observed}}^{M-1} R_s \leq 6SL \sum_{s=\boldsymbol{s}_{observed}}^{M-1} \sqrt{\frac{T_s}{2} \log \frac{2}{\delta_1}} + 4\sqrt{T_s k \log(\lambda + nL/k)}(\lambda^{1/2} S \\ + R\sqrt{2\log(1/\delta_3) + k\log(1 + T_s L/(\lambda k))}) \end{aligned}$$

with probability $\geq 1 - \delta_3$.

Proof. The regret during the epoch is the sum of the regret when we pull the random arms added to the regret when we pull OFUL arms.

Now, we just have to use the upper bound on the number of times we pull a random arm in Corollary A.1. During each random arm pull the worst case regret is $2SL$.

The number of times we pull an OFUL arm in epoch s, T_s^{OFUL}, is trivially upper bounded by T_s. Apply Lemma A.3 stated above with $\delta \to \delta_3$, $t \to T_s^{OFUL}$, $d \to k$ to get the result. Recall, we cannot apply the OFUL regret bound directly here since our algorithm involves additional random arms sampled during the epoch along with arms sampled in previous epochs. □

Proof of main result

We are now ready to prove the regret bound of Feature Feedback Epoch OFUL.

Proof. The regret can be summed over the epochs as:

$$\begin{aligned}
R_T &= \sum_{s=0}^{M-1} R_s \\
&= \sum_{s=0}^{s_{observed}} R_s + \sum_{s=s_{observed}+1}^{M-1} R_s \\
&\leq \sum_{s=0}^{s_{observed}} 2SLT_s + \sum_{s=s_{observed}+1}^{M-1} R_s \\
&\leq 2SL2^{s_{observed}+1} + \sum_{s=s_{observed}+1}^{M-1} R_s
\end{aligned}$$

Now, note that:

$$\begin{aligned}
2^{s_{observed}+1} &= 2 \cdot 2^{\left\lceil \log_2 \left(\frac{1}{\log 2/\delta_1} \left(\frac{\log k/\delta_2}{\log 1/(1-p)} \right)^2 \right) \right\rceil} \\
&\leq 2 \cdot 2^{\log_2 \left(\frac{1}{\log 2/\delta_1} \left(\frac{\log k/\delta_2}{\log 1/(1-p)} \right)^2 \right) + 1} \\
&= \frac{4}{\log 2/\delta_1} \left(\frac{\log k/\delta_2}{\log 1/(1-p)} \right)^2
\end{aligned}$$

Now, setting $\delta_1 = \delta_3 = \delta/3M$ and $\delta_2 = \delta/3$, we get the final regret expression using Lemma A.4. The multiplicative factor of $\log \frac{T}{2}$ comes from bounding the sum of regrets over the epochs by the max regret over all the epochs (which occurs during the last epoch) multiplied by the number of epochs, which is $\log \frac{T}{2}$.

□

The proof for Feature Feedback OFUL follows similarly by noticing that the Algorithms are essentially the same with different ϵ_t and using

the fact that $\frac{1}{\sqrt{2^{s+1}}} \leq \epsilon_t = \frac{1}{\sqrt{t}} \leq \frac{1}{\sqrt{2^s}}$ for $s = \lfloor \log_2(t) \rfloor$.

B APPROXIMATING MATRICES USING GROUPS OF COLUMNS

B.1 Proof of Theorem 3.6

As stated in Section 3.4, the result in Theorem 3.6 follows by applying standard boosting methods to Lemma 3.8 and running Algorithm 1 $t = \ln(\frac{1}{\delta})$ times. By choosing the solution with minimum error and observing that $0.3 < 1/e$, we have that the relative error bound holds with probability greater than $1 - e^{-t} = 1 - \delta$. Hence, it suffices to prove Lemma 3.8 to prove the main result.

Proof of Lemma 3.8

First, note $\boldsymbol{U} = (\boldsymbol{RS})^{\dagger}$ and $\boldsymbol{C} = \boldsymbol{AS}$.

$$\|\boldsymbol{A} - \boldsymbol{CUR}\|_F = \|\boldsymbol{A} - \boldsymbol{AS}(\boldsymbol{RS})^{\dagger}\boldsymbol{R}\|_F$$

Recall that $\boldsymbol{R} \in \mathbb{R}^{r \times n}$ has rank no greater than r; $\boldsymbol{A} \in \mathbb{R}^{m \times n}; \varepsilon \in (0, 1)$; and that the same column blocks from $\boldsymbol{R}$ and $\boldsymbol{A}$ are picked with the following probability distribution:

$$p_i = \frac{\|\boldsymbol{V}_{R,r}^T \boldsymbol{E}_i\|_F^2}{r}, \ \forall i \in [G].$$

We can use Lemma B.1 (stated and proved in next section) with proba-

bility at least 0.85 we have

$$\|\boldsymbol{A} - \boldsymbol{A}\boldsymbol{S}(\boldsymbol{R}\boldsymbol{S})^{\dagger}\boldsymbol{R}\|_F \leq (1+\varepsilon)\|\boldsymbol{A} - \boldsymbol{A}\boldsymbol{R}^{\dagger}\boldsymbol{R}\|_F.$$

Next, we bound $\|\boldsymbol{A} - \boldsymbol{A}\boldsymbol{R}^{\dagger}\boldsymbol{R}\|_F$. Since $\boldsymbol{A}$ has incoherent column space, the uniform sampling distribution $p_j = 1/m$ satisfies eqn. (13) in Drineas et al. (2008) with $\beta = 1/\mu_0$. Consequently, we can apply modified version of Theorem 1 in Drineas et al. (2008) we get with probability at least 0.85, $\|\boldsymbol{A} - \boldsymbol{A}\boldsymbol{R}^{\dagger}\boldsymbol{R}\|_F \leq (1+\varepsilon)\|\boldsymbol{A} - \boldsymbol{A}_k\|_F$. Finally, we get with probability 0.7,

$$\begin{aligned}\|\boldsymbol{A} - \boldsymbol{C}\boldsymbol{U}\boldsymbol{R}\|_F &\leq (1+\varepsilon')^2\|\boldsymbol{A} - \boldsymbol{A}_k\|_F, \\ &\leq (1+\varepsilon'')\|\boldsymbol{A} - \boldsymbol{A}_k\|_F, \qquad \text{letting } \varepsilon'' = 3\varepsilon'.\end{aligned}$$

This completes the proof of Lemma 1.

Approximating generalized ℓ_2 regression in the block setting

In this section, we give theory for generalized least squares using block subset selection that is used to prove the main results for the algorithms but applies to arbitrary matrices $\boldsymbol{A}$ and $\boldsymbol{B}$. Given matrices $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ and $\boldsymbol{B} \in \mathbb{R}^{r\times n}$, the generalized least squares problem is

$$\min_{\boldsymbol{X}\in\mathbb{R}^{m\times r}} \|\boldsymbol{A} - \boldsymbol{X}\boldsymbol{B}\|_F.$$

It is well-known that the solution to this optimization problem is given by $\widehat{\boldsymbol{X}} = \boldsymbol{A}\boldsymbol{B}^\dagger$. To approximate this problem by a subsampled problem, we sample some blocks of columns from $\boldsymbol{A}$ and $\boldsymbol{B}$ to approximate the standard ℓ_2 regression by the following optimization:

$$\min_{\boldsymbol{X}\in\mathbb{R}^{m\times r}} \|(\boldsymbol{A}\boldsymbol{S}) - \boldsymbol{X}(\boldsymbol{B}\boldsymbol{S})\|_F.$$

The solution of this problem is given by $\tilde{\boldsymbol{X}} = \boldsymbol{A}\boldsymbol{S}(\boldsymbol{B}\boldsymbol{S})^\dagger$. In the following lemma, we give a guarantee stating that, when enough blocks are sampled with the specified probability, the approximate solution is close to the actual solution to the ℓ_2 regression.

Lemma B.1. *Suppose $\boldsymbol{B} \in \mathbb{R}^{r\times n}$ has rank no greater than k; $\boldsymbol{A} \in \mathbb{R}^{m\times n}$; $\varepsilon, \delta \in (0,1)$; and let the same column blocks from $\boldsymbol{B}$ and $\boldsymbol{A}$ be picked with the following probability distribution:*

$$p_i = \frac{\|(\boldsymbol{V}_{B,k})_{(i)}\|_F^2}{k},\ \forall i \in [G].$$

If $g = \mathcal{O}(\frac{k^2}{\alpha_B \delta^4 \varepsilon^2})$ blocks are chosen, then with probability at least $1-\delta$ we have

$$\|\boldsymbol{A} - \boldsymbol{A}\boldsymbol{S}(\boldsymbol{B}\boldsymbol{S})^\dagger\boldsymbol{B}\|_F \le (1+\varepsilon)\|\boldsymbol{A} - \boldsymbol{A}\boldsymbol{B}^\dagger\boldsymbol{B}\|_F.$$

Proof. Let $\boldsymbol{B} = \boldsymbol{U}_k\boldsymbol{\Sigma}_k\boldsymbol{V}_k^T$ and $\alpha = \max_i \left(\frac{\|\boldsymbol{V}_k^T E_i\|_2}{\|\boldsymbol{V}_k^T E_i\|_F}\right)^2$

We start by showing $\boldsymbol{V}_k^T\boldsymbol{S}$ is full rank. Using Lemma 2, if $g \ge 8\alpha_R^{-1}k^2\delta^{-2}\varepsilon_1^{-2}$

and $0 < \varepsilon_1 < 1$, we get the following with probability $\geq 1 - \delta_1$,

$$\|\boldsymbol{V}_k^T \boldsymbol{V}_k - \boldsymbol{V}_k^T \boldsymbol{S}\boldsymbol{S}^T \boldsymbol{V}_k\|_2 = \|\boldsymbol{I}_k - \boldsymbol{V}_k^T \boldsymbol{S}\boldsymbol{S}^T \boldsymbol{V}_k\|_2 \leq 4\frac{k}{\delta\sqrt{\alpha_R g}} \leq \frac{\varepsilon_1}{2}.$$

This further gives us a bound on the singular values of $\boldsymbol{V}_k^T \boldsymbol{S}$, for all i,

$$|1-\sigma_i^2(\boldsymbol{V}_k^T \boldsymbol{S})| = |\sigma_i(\boldsymbol{V}_k^T \boldsymbol{V}_k) - \sigma_i(\boldsymbol{V}_k^T \boldsymbol{S}\boldsymbol{S}^T \boldsymbol{V}_k)| \leq \|\boldsymbol{I}_k - \boldsymbol{V}_k^T \boldsymbol{S}\boldsymbol{S}^T \boldsymbol{V}_k\|_2 \leq \varepsilon_1. \tag{B.1}$$

Thus, it follows for all singular values of $\boldsymbol{V}_k^T \boldsymbol{S}$,

$$\sqrt{1-\varepsilon_1} \leq \sigma_i(\boldsymbol{V}_k^T \boldsymbol{S}) \leq \sqrt{1+\varepsilon_1}. \tag{B.2}$$

Now, consider

$$\begin{aligned}
\|\Omega\|_2 &= \|(\boldsymbol{V}_k^T\boldsymbol{S})^\dagger - (\boldsymbol{V}_k^T\boldsymbol{S})^T\|_2 \\
&= \|\boldsymbol{\Sigma}^{-1}_{\boldsymbol{V}_k^T\boldsymbol{S}} - \boldsymbol{\Sigma}_{\boldsymbol{V}_k^T\boldsymbol{S}}\|_2 \\
&= \max_i \left|\sigma_i(\boldsymbol{V}_k^T\boldsymbol{S}) - \frac{1}{\sigma_i(\boldsymbol{V}_k^T\boldsymbol{S})}\right| \\
&= \max_i \frac{|\sigma_i^2(\boldsymbol{V}_k^T\boldsymbol{S}) - 1|}{|\sigma_i(\boldsymbol{V}_k^T\boldsymbol{S})|} \\
&\leq \frac{\|\boldsymbol{V}_k^T\boldsymbol{V}_k - \boldsymbol{V}_k^T\boldsymbol{S}\boldsymbol{S}^T\boldsymbol{V}_k\|_2}{\sqrt{1 - \|\boldsymbol{V}_k^T\boldsymbol{V}_k - \boldsymbol{V}_k^T\boldsymbol{S}\boldsymbol{S}^T\boldsymbol{V}_k\|_2}} \\
&\leq \frac{\varepsilon_1/2}{\sqrt{1-\varepsilon_1/2}} \\
&\leq \varepsilon_1/\sqrt{2},
\end{aligned}$$

where the first inequality follows from equation (B.1), the second inequality follows by applying Lemma 3.7 and the last inequality follows since $\varepsilon_1 < 1$ implies $\sqrt{1-\varepsilon_1/2} > 1/\sqrt{2}$

Also, for any $\boldsymbol{Q}$ we have,

$$\begin{aligned}
\mathbb{E}[\|\boldsymbol{Q}\boldsymbol{S}\|_F^2] &= \mathbb{E}\left[\sum_{t=1}^{g}\left\|\frac{1}{\sqrt{gp_{j_t}}}\boldsymbol{Q}^{(j_t)}\right\|_F^2\right] \\
&= \sum_{t=1}^{g}\mathbb{E}\left[\frac{1}{gp_{j_t}}\left\|\boldsymbol{Q}^{(j_t)}\right\|_F^2\right] \\
&= \sum_{t=1}^{g}\sum_{i=1}^{G} p_i\frac{1}{gp_i}\left\|\boldsymbol{Q}^{(i)}\right\|_F^2 \\
&= \|\boldsymbol{Q}\|_F^2.
\end{aligned}$$

By Jensen's inequality,

$$\mathbb{E}[\|\boldsymbol{QS}\|_F]^2 \leq \mathbb{E}[\|\boldsymbol{QS}\|_F^2] = \|\boldsymbol{Q}\|_F^2.$$

By applying Markov's inequality, we get with probability $\geq 1 - \delta'$,

$$\|\boldsymbol{QS}\|_F \leq \frac{1}{\delta'}\mathbb{E}[\|\boldsymbol{QS}\|_F] \leq \frac{1}{\delta'}\|\boldsymbol{Q}\|_F. \tag{B.3}$$

The following will be useful later,

$$\begin{aligned}
\boldsymbol{AS}(\boldsymbol{BS})^\dagger\boldsymbol{B} &= \boldsymbol{AS}(\boldsymbol{U}_k\boldsymbol{\Sigma}_k\boldsymbol{V}_k^T\boldsymbol{S})^\dagger\boldsymbol{U}_k\boldsymbol{\Sigma}_k\boldsymbol{V}_k^T \\
&= \boldsymbol{AS}(\boldsymbol{V}_k^T\boldsymbol{S})^\dagger\boldsymbol{\Sigma}_k^{-1}\boldsymbol{U}_k^T\boldsymbol{U}_k\boldsymbol{\Sigma}_k\boldsymbol{V}_k^T \\
&= \boldsymbol{AS}(\boldsymbol{V}_k^T\boldsymbol{S})^\dagger\boldsymbol{V}_k^T
\end{aligned}$$

Using this result and observing that $(\boldsymbol{V}_k\boldsymbol{V}_k^T + \boldsymbol{V}_k^\perp\boldsymbol{V}_k^{\perp T}) = \boldsymbol{I}$, we break down the left hand term into 3 manageable components,

$$\begin{aligned}
&\|\boldsymbol{A} - \boldsymbol{AS}(\boldsymbol{BS})^\dagger\boldsymbol{B}\|_F \\
&= \|\boldsymbol{A} - \boldsymbol{AS}(\boldsymbol{V}_k^T\boldsymbol{S})^\dagger\boldsymbol{V}_k^T\|_F \\
&= \|\boldsymbol{A} - \boldsymbol{AV}_k\boldsymbol{V}_k^T\boldsymbol{S}(\boldsymbol{V}_k^T\boldsymbol{S})^\dagger\boldsymbol{V}_k^T + \boldsymbol{AV}_k^\perp\boldsymbol{V}_k^{\perp T}\boldsymbol{S}(\boldsymbol{V}_k^T\boldsymbol{S})^\dagger\boldsymbol{V}_k^T\|_F
\end{aligned}$$

As seen before, with high probability, $\boldsymbol{V}_k^T\boldsymbol{S}$ is full rank. Using this fact

along with triangle inequality gives us

$$\begin{aligned}
&\|\boldsymbol{A} - \boldsymbol{A}\boldsymbol{S}(\boldsymbol{B}\boldsymbol{S})^{\dagger}\boldsymbol{B}\|_F \\
&= \|\boldsymbol{A} - \boldsymbol{A}\boldsymbol{V}_k\boldsymbol{V}_k^T + \boldsymbol{A}\boldsymbol{V}_k^{\perp}\boldsymbol{V}_k^{\perp T}\boldsymbol{S}(\boldsymbol{V}_k^T\boldsymbol{S})^{\dagger}\boldsymbol{V}_k^T\|_F \\
&\leq \|\boldsymbol{A} - \boldsymbol{A}\boldsymbol{V}_k\boldsymbol{V}_k^T\|_F + \|\boldsymbol{A}\boldsymbol{V}_k^{\perp}\boldsymbol{V}_k^{\perp T}\boldsymbol{S}(\boldsymbol{V}_k^T\boldsymbol{S})^{\dagger}\boldsymbol{V}_k^T\|_F
\end{aligned}$$

Define $\Omega := (\boldsymbol{V}_k^T\boldsymbol{S})^{\dagger} - (\boldsymbol{V}_k^T\boldsymbol{S})^T$,

$$\begin{aligned}
&\|\boldsymbol{A} - \boldsymbol{A}\boldsymbol{S}(\boldsymbol{B}\boldsymbol{S})^{\dagger}\boldsymbol{B}\|_F \\
&= \|\boldsymbol{A}\boldsymbol{V}_k^{\perp}\boldsymbol{V}_k^{\perp T}\|_F + \|\boldsymbol{A}\boldsymbol{V}_k^{\perp}\boldsymbol{V}_k^{\perp T}\boldsymbol{S}(\Omega + (\boldsymbol{V}_k^T\boldsymbol{S})^T)\|_F \\
&\leq \|\boldsymbol{A}\boldsymbol{V}_k^{\perp}\boldsymbol{V}_k^{\perp T}\|_F + \|\boldsymbol{A}\boldsymbol{V}_k^{\perp}\boldsymbol{V}_k^{\perp T}\boldsymbol{S}\|_F\|\Omega\|_2 + \|\boldsymbol{A}\boldsymbol{V}_k^{\perp}\boldsymbol{V}_k^{\perp T}\boldsymbol{S}\boldsymbol{S}^T\boldsymbol{V}_k\|_F
\end{aligned}$$

By (B.3) and since $\boldsymbol{V}_k^{\perp T}\boldsymbol{V}_k = 0$,

$$\begin{aligned}
&\|\boldsymbol{A} - \boldsymbol{A}\boldsymbol{S}(\boldsymbol{B}\boldsymbol{S})^{\dagger}\boldsymbol{B}\|_F \\
&\leq \left(1 + \frac{1}{\delta'}\|\Omega\|_2\right)\|\boldsymbol{A}\boldsymbol{V}_k^{\perp}\boldsymbol{V}_k^{\perp T}\|_F + \|\boldsymbol{A}\boldsymbol{V}_k^{\perp}\boldsymbol{V}_k^{\perp T}\boldsymbol{V}_k - \boldsymbol{A}\boldsymbol{V}_k^{\perp}\boldsymbol{V}_k^{\perp T}\boldsymbol{S}\boldsymbol{S}^T\boldsymbol{V}_k\|_F
\end{aligned}$$

Using Lemma 3.7 and $\|\boldsymbol{V}_k\|_F = \sqrt{k}$,

$$\begin{aligned}
&\|\boldsymbol{A} - \boldsymbol{A}\boldsymbol{S}(\boldsymbol{B}\boldsymbol{S})^\dagger \boldsymbol{B}\|_F \\
&\leq \left(1 + \frac{1}{\delta'}\|\Omega\|_2\right) \|\boldsymbol{A}\boldsymbol{V}_k^\perp \boldsymbol{V}_k^{\perp T}\|_F + \frac{1}{\delta_2\sqrt{\alpha_R g}}\|\boldsymbol{A}\boldsymbol{V}_k^\perp \boldsymbol{V}_k^{\perp T}\|_F \|\boldsymbol{V}_k\|_F \\
&\leq \left(1 + \frac{1}{\delta'}\|\Omega\|_2\right) \|\boldsymbol{A}\boldsymbol{V}_k^\perp \boldsymbol{V}_k^{\perp T}\|_F + \frac{\sqrt{k}}{\delta_2\sqrt{\alpha_R g}}\|\boldsymbol{A}\boldsymbol{V}_k^\perp \boldsymbol{V}_k^{\perp T}\|_F \\
&\leq \left(1 + \frac{1}{\delta'}\|\Omega\|_2 + \frac{\varepsilon_1}{\sqrt{8}\delta_2}\right) \|\boldsymbol{A}\boldsymbol{V}_k^\perp \boldsymbol{V}_k^{\perp T}\|_F
\end{aligned}$$

where the second inequality follows since $\|\boldsymbol{A} - \boldsymbol{A}\boldsymbol{B}^\dagger\boldsymbol{B}\|_F = \|\boldsymbol{A}\boldsymbol{V}_k^\perp \boldsymbol{V}_k^{\perp T}\|_F$ and the last inequality follows since $\frac{\sqrt{k}}{\sqrt{\alpha_R g}} \leq \frac{k}{\delta_1\sqrt{\alpha_R g}} \leq \frac{\varepsilon_1}{\sqrt{8}}$.

Finally, using $\boldsymbol{A}\boldsymbol{V}_k\boldsymbol{V}_k^T = \boldsymbol{A}\boldsymbol{B}^\dagger\boldsymbol{B}$, we have

$$\|\boldsymbol{A} - \boldsymbol{A}\boldsymbol{S}(\boldsymbol{B}\boldsymbol{S})^\dagger \boldsymbol{B}\|_F \leq \left(1 + \frac{1}{\delta'}\|\Omega\|_2 + \frac{\varepsilon_1}{\sqrt{8}\delta_2}\right) \|\boldsymbol{A} - \boldsymbol{A}\boldsymbol{B}^\dagger\boldsymbol{B}\|_F$$

Thus, we can conclude the following with probability $\geq 1 - (\delta' + \delta_1 + \delta_2) = 1 - \delta$

$$\begin{aligned}
\|\boldsymbol{A} - \boldsymbol{A}\boldsymbol{S}(\boldsymbol{B}\boldsymbol{S})^\dagger \boldsymbol{B}\|_F &\leq \left(1 + \left(\frac{1}{\sqrt{2}\delta'} + \frac{1}{2\delta_2}\right)\varepsilon_1\right) \|\boldsymbol{A} - \boldsymbol{A}\boldsymbol{B}^\dagger\boldsymbol{B}\|_F \\
&\leq (1 + \varepsilon) \|\boldsymbol{A} - \boldsymbol{A}\boldsymbol{B}^\dagger\boldsymbol{B}\|_F
\end{aligned}$$

by setting $\delta' = \delta_1 = \delta_2 = \delta/3$ and $\varepsilon = \frac{6\varepsilon_1}{\delta}$.

Lemma ?? is used, then $g \geq 36 * 8\alpha_R \frac{k^2}{\varepsilon^2\delta^4}$. Finally, note that $\varepsilon_1 \leq \varepsilon < 1$ by assumption.

□

Proof of Lemma 3.7

Proof. Note that,

$$\boldsymbol{E}\left[\left(\frac{\boldsymbol{A}^{(j_t)}\boldsymbol{B}_{(j_t)}}{gp_{j_t}}\right)_{i_1i_2}\right] = \sum_{k=1}^{G} p_k \left(\frac{\boldsymbol{A}^{(k)}\boldsymbol{B}_{(k)}}{gp_k}\right)_{i_1i_2} = \frac{1}{g}(\boldsymbol{AB})_{i_1i_2}$$

Since each block is picked independently we have,

$$\begin{aligned}
var[(\boldsymbol{CR})_{i_1i_2}] &= var\left[\sum_{t=1}^{g}\left(\frac{\boldsymbol{A}^{(j_t)}\boldsymbol{B}_{(j_t)}}{gp_{j_t}}\right)_{i_1i_2}\right] \\
&= \sum_{t=1}^{g} var\left[\left(\frac{\boldsymbol{A}^{(j_t)}\boldsymbol{B}_{(j_t)}}{gp_{j_t}}\right)_{i_1i_2}\right] \\
&= \sum_{t=1}^{g}\left(\boldsymbol{E}\left[\left(\frac{\boldsymbol{A}^{(j_t)}\boldsymbol{B}_{(j_t)}}{gp_{j_t}}\right)^2_{i_1i_2}\right] - \boldsymbol{E}\left[\left(\frac{\boldsymbol{A}^{(j_t)}\boldsymbol{B}_{(j_t)}}{gp_{j_t}}\right)_{i_1i_2}\right]^2\right) \\
&= g\left(\sum_{k=1}^{G} p_k\left(\frac{\boldsymbol{A}^{(k)}\boldsymbol{B}_{(k)}}{gp_k}\right)^2_{i_1i_2} - \frac{(\boldsymbol{AB})^2_{i_1i_2}}{g^2}\right) \\
&= \frac{1}{g}\left(\sum_{k=1}^{G}\frac{\left(\boldsymbol{A}^{(k)}\boldsymbol{B}_{(k)}\right)^2_{i_1i_2}}{p_k} - (\boldsymbol{AB})^2_{i_1i_2}\right)
\end{aligned}$$

$$
\begin{aligned}
\mathbb{E}[\|\boldsymbol{AB}-\boldsymbol{CR}\|_F^2] &= \sum_{i_1=1}^{m}\sum_{i_2=1}^{p} var[(\boldsymbol{CR})_{i_1 i_2}] \\
&= \sum_{i_1=1}^{m}\sum_{i_2=1}^{p} \frac{1}{g}\left(\sum_{k=1}^{G}\frac{\left(\boldsymbol{A}^{(k)}\boldsymbol{B}_{(k)}\right)^2_{i_1 i_2}}{p_k} - (\boldsymbol{AB})^2_{i_1 i_2}\right) \\
&= \left(\sum_{k=1}^{G}\frac{1}{gp_k}\sum_{i_1=1}^{m}\sum_{i_2=1}^{p}\left(\boldsymbol{A}^{(k)}\boldsymbol{B}_{(k)}\right)^2_{i_1 i_2}\right) - \frac{\|\boldsymbol{AB}\|_F^2}{g} \\
&= \sum_{k=1}^{G}\frac{\|\boldsymbol{A}^{(k)}\boldsymbol{B}_{(k)}\|_F^2}{gp_k} - \frac{\|\boldsymbol{AB}\|_F^2}{g} \\
&\leq \sum_{k=1}^{G}\frac{\|\boldsymbol{A}^{(k)}\|_2^2\|\boldsymbol{B}_{(k)}\|_F^2}{gp_k} \\
&\leq \sum_{k=1}^{G}\left(\frac{\|\boldsymbol{A}^{(k)}\|_2}{\|\boldsymbol{A}^{(k)}\|_F}\right)^2\frac{\|\boldsymbol{A}^{(k)}\|_F^2\|\boldsymbol{B}_{(k)}\|_F^2}{gp_k} \\
&\leq \frac{1}{\beta\alpha_A g}\|\boldsymbol{A}\|_F^2\|\boldsymbol{B}\|_F^2
\end{aligned}
$$

where $\alpha_A = \min_k\left(\frac{\|\boldsymbol{A}^{(k)}\|_F}{\|\boldsymbol{A}^{(k)}\|_2}\right)^2$. Also, note $1 \leq \alpha_A \leq s$.

By Jensen's inequality,

$$
\begin{aligned}
\mathbb{E}[\|\boldsymbol{AB}-\boldsymbol{CR}\|_F]^2 &\leq \mathbb{E}[\|\boldsymbol{AB}-\boldsymbol{CR}\|_F^2] \\
&\leq \frac{1}{\beta\alpha_A g}\|\boldsymbol{A}\|_F^2\|\boldsymbol{B}\|_F^2
\end{aligned}
$$

And by Markov's inequality, with probability $\geq 1 - \delta$, we have

$$\|\boldsymbol{AB} - \boldsymbol{CR}\|_F \leq \frac{1}{\delta}\mathbb{E}[\|\boldsymbol{AB} - \boldsymbol{CR}\|_F] \leq \frac{1}{\delta\sqrt{\beta\alpha_A g}}\|\boldsymbol{A}\|_F\|\boldsymbol{B}\|_F$$

□

Proof of Corollary B.2

Here we state and prove the corollary mentioned in the chapter. If it is possible to compute the SVD of the entire matrix, then the rows can be sampled using row leverage scores, and the incoherence assumption can be dropped. The relative error guarantee for the full SVD Block CUR approximation is stated below.

Corollary B.2. *Given $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, let $r = O(\frac{k^2}{\varepsilon^2}\ln(\frac{1}{\delta}))$ and $g = O(\frac{r^2}{\alpha_R\varepsilon^2}\ln(\frac{1}{\delta}))$. There exist randomized algorithms such that, if r rows and g column blocks are chosen to construct $\boldsymbol{R}$ and $\boldsymbol{C}$, respectively, then with probability $\geq 1 - \delta$, the following holds:*

$$\|\boldsymbol{A} - \boldsymbol{CUR}\|_F \leq (1 + \varepsilon)\|\boldsymbol{A} - \boldsymbol{A}_k\|_F,$$

where $\varepsilon, \delta \in (0, 1)$, and $\boldsymbol{U} = \boldsymbol{W}^\dagger$ is the pseudoinverse of the scaled intersection of $\boldsymbol{C}$ and $\boldsymbol{R}$.

First, note $\boldsymbol{U} = (\boldsymbol{RS})^{\dagger}$ and $\boldsymbol{C} = \boldsymbol{AS}$.

$$\|\boldsymbol{A} - \boldsymbol{CUR}\|_F = \|\boldsymbol{A} - \boldsymbol{AS}(\boldsymbol{RS})^{\dagger}\boldsymbol{R}\|_F.$$

Similar to the proof of Lemma 1, we can use Lemma 1.1 with probability at least 0.85 we have

$$\|\boldsymbol{A} - \boldsymbol{AS}(\boldsymbol{RS})^{\dagger}\boldsymbol{R}\|_F \leq (1+\varepsilon)\|\boldsymbol{A} - \boldsymbol{AR}^{\dagger}\boldsymbol{R}\|_F.$$

Recall that $\boldsymbol{A} \in \mathbb{R}^{m\times n}; \varepsilon \in (0,1)$; and that the rows $\boldsymbol{R}$ are picked from $\boldsymbol{A}$ with the following probability distribution:

$$p_i = \frac{\|e_i^T \boldsymbol{U}_{A,k}\|_2^2}{k},\ \forall i \in [m].$$

We bound $\|\boldsymbol{A} - \boldsymbol{AR}^{\dagger}\boldsymbol{R}\|_F$ using Theorem 1 in Drineas et al. (2008) we get with probability at least 0.85, $\|\boldsymbol{A} - \boldsymbol{AR}^{\dagger}\boldsymbol{R}\|_F \leq (1+\varepsilon)\|\boldsymbol{A} - \boldsymbol{A}_k\|_F$. Finally, we get with probability 0.7

$$\begin{aligned}\|\boldsymbol{A} - \boldsymbol{CUR}\|_F &\leq (1+\varepsilon')^2\|\boldsymbol{A} - \boldsymbol{A}_k\|_F \\ &\leq (1+\varepsilon'')\|\boldsymbol{A} - \boldsymbol{A}_k\|_F \qquad \text{(letting } \varepsilon'' = 3\varepsilon')\end{aligned}$$

This completes the proof of Corollary B.2.

C.1 Clustering properties with Absolute error loss function

Proof of Theorem 4.5

Proof. The proof is divided into two steps. First, we show $\|\widehat{\boldsymbol{\beta}}_{j*}\| = \|\widehat{\boldsymbol{\beta}}_{k*}\|$ and then we further show that the rows are equal. We proceed by contradiction. Assume $\|\widehat{\boldsymbol{\beta}}_{j*}\| \neq \|\widehat{\boldsymbol{\beta}}_{k*}\|$ and, without loss of generality, suppose $\|\widehat{\boldsymbol{\beta}}_{j*}\| > \|\widehat{\boldsymbol{\beta}}_{k*}\|$. We see that there exists a modification of the solution with a smaller GrOWL norm and same data-fitting term, and thus smaller overall objective value which contradicts our assumption that $\widehat{\boldsymbol{B}}$ is the minimizer of $L(\boldsymbol{B}) + G(\boldsymbol{B})$.

Consider the modification, $\boldsymbol{V} = \widehat{\boldsymbol{B}}$ except $\widehat{\boldsymbol{v}}_{j*} = \widehat{\boldsymbol{\beta}}_{j*} - \boldsymbol{\varepsilon}$ and $\widehat{\boldsymbol{v}}_{k*} = \widehat{\boldsymbol{\beta}}_{k*} + \boldsymbol{\varepsilon}$ where $\boldsymbol{\varepsilon} = \delta\widehat{\boldsymbol{\beta}}_{j*}$ and δ is chosen such that $\|\boldsymbol{\varepsilon}\| \in \left(0, \frac{\|\widehat{\boldsymbol{\beta}}_{j*}\| - \|\widehat{\boldsymbol{\beta}}_{k*}\|}{2}\right]$
Let $L(\boldsymbol{B}) = \|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{B}\|_1 = \|\boldsymbol{Y}' - \boldsymbol{x}_{*j}\widehat{\boldsymbol{\beta}}_{j*} - \boldsymbol{x}_{*k}\widehat{\boldsymbol{\beta}}_{k*}\|_1$ where $\boldsymbol{Y}'$ is the residual term given by $\boldsymbol{Y}' = \boldsymbol{Y} - \sum_{i\neq j,k} \boldsymbol{x}_{*i}\widehat{\boldsymbol{\beta}}_{i*}$. Since $\boldsymbol{x}_{*j} = \boldsymbol{x}_{*k}$, L is invariant under this transformation, *i.e.*, $L(\boldsymbol{V}) = L(\widehat{\boldsymbol{B}})$. Same is true for $L(\boldsymbol{B}) = \|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{B}\|_F^2$.

Observe that the GrOWL norm of $\boldsymbol{B}$ is equal to the OWL norm of the vector of euclidean norms of rows of $\boldsymbol{B}$. Since $\|\boldsymbol{v}_{k*}\| = \|\boldsymbol{\beta}_{k*} + \boldsymbol{\varepsilon}\| \leq \|\boldsymbol{\beta}_{k*}\| + \|\boldsymbol{\varepsilon}\|$, this transformation is equivalent to that defined in Lemma

4.4 and we have

$$G(\widehat{\boldsymbol{B}}) - G(\boldsymbol{V}) \;\; \geq \;\; \Delta\|\boldsymbol{\varepsilon}\|$$

This leads to a contradiction to our assumption that $\widehat{\boldsymbol{B}}$ is the minimizer of $L(\boldsymbol{B}) + G(\boldsymbol{B})$ and completes the proof that $\|\widehat{\boldsymbol{\beta}}_{j*}\| = \|\widehat{\boldsymbol{\beta}}_{k*}\|$. Now, let $\widehat{\boldsymbol{\beta}}_{j*} + \widehat{\boldsymbol{\beta}}_{k*} = \boldsymbol{z}$, then the minimizer satisfies

$$\min_{\widehat{\boldsymbol{\beta}}_{j*}, \widehat{\boldsymbol{\beta}}_{k*}} \; w_j\|\widehat{\boldsymbol{\beta}}_{j*}\| + w_k\|\widehat{\boldsymbol{\beta}}_{k*}\|$$

$$\text{such that } \widehat{\boldsymbol{\beta}}_{j*} + \widehat{\boldsymbol{\beta}}_{k*} = \boldsymbol{z} \text{ and } \|\widehat{\boldsymbol{\beta}}_{j*}\| = \|\widehat{\boldsymbol{\beta}}_{k*}\|$$

It is easy to see that the solution to this optimization is $\widehat{\boldsymbol{\beta}}_{j*} = \widehat{\boldsymbol{\beta}}_{k*} = \boldsymbol{z}/2 \quad \square$

Proof of Theorem 4.6

Proof. The proof is similar to the identical columns theorem. By contradiction and without loss of generality, suppose $\|\widehat{\boldsymbol{\beta}}_{j*}\| > \|\widehat{\boldsymbol{\beta}}_{k*}\|$. We show that there exists a transformation of $\widehat{\boldsymbol{B}}$ such that the increase in the data fitting term is smaller than the decrease in the GrOWL norm.

Consider the modification, $\boldsymbol{V}$, as defined in the proof of Theorem 4.5. By triangle inequality, the difference in loss function L that results from this modification satisfies

$$L(\boldsymbol{V}) - L(\widehat{\boldsymbol{B}}) \;\; \leq \;\; \left\|\boldsymbol{x}_{*j} - \boldsymbol{x}_{*k}\right\|_1 \|\boldsymbol{\varepsilon}\|_1$$

Invoking Lemma 4.4 as in the previous theorem and $\|\boldsymbol{\varepsilon}\|_1 \leq \sqrt{r}\|\boldsymbol{\varepsilon}\|$, we get

$$\begin{aligned} L(\boldsymbol{V}) + G(\boldsymbol{V}) - (L(\widehat{\boldsymbol{B}}) + G(\widehat{\boldsymbol{B}})) \\ \leq (\left\|\boldsymbol{x}_{*j} - \boldsymbol{x}_{*k}\right\|_1 - \frac{\Delta}{\sqrt{r}})\|\boldsymbol{\varepsilon}\| < 0 \end{aligned}$$

This contradicts our assumption that $\widehat{\boldsymbol{B}}$ is the minimizer of $L(\boldsymbol{B})+G(\boldsymbol{B})$ and completes the proof. □

Proof of Theorem 4.7

Proof. The proof is similar to the identical columns theorem. By contradiction, suppose $\|\widehat{\boldsymbol{\beta}}_{j*} - \widehat{\boldsymbol{\beta}}_{k*}\| \geq \frac{8\phi\|\widehat{\boldsymbol{\beta}}_{k*}\|}{4\phi^2+1} \geq \frac{2\|\widehat{\boldsymbol{\beta}}_{k*}\|}{\phi}$. We show that there exists a transformation of $\widehat{\boldsymbol{B}}$ such that the increase in the data fitting term is smaller than the decrease in the GrOWL norm.

Consider the modification, $\boldsymbol{V}$, as defined in the proof of Theorem 4.5 with $\boldsymbol{\varepsilon} = \frac{\widehat{\boldsymbol{\beta}}_{j*} - \widehat{\boldsymbol{\beta}}_{k*}}{2}$. By triangle inequality, the difference in loss function L that results from this modification satisfies

$$L(\boldsymbol{V}) - L(\widehat{\boldsymbol{B}}) \;\; \leq \;\; \left\|\boldsymbol{x}_{*j} - \boldsymbol{x}_{*k}\right\|_1 \|\boldsymbol{\varepsilon}\|_1$$

We now bound the decrease in the GrOWL norm. Note by parallelo-

gram law,

$$
\begin{aligned}
&\|\widehat{\boldsymbol{\beta}}_{j*} + \widehat{\boldsymbol{\beta}}_{k*}\|^2 \\
&= 2\|\widehat{\boldsymbol{\beta}}_{j*}\|^2 + 2\|\widehat{\boldsymbol{\beta}}_{k*}\|^2 - \|\widehat{\boldsymbol{\beta}}_{j*} - \widehat{\boldsymbol{\beta}}_{k*}\|^2 \\
&\leq 2\|\widehat{\boldsymbol{\beta}}_{j*}\|^2 + 2\|\widehat{\boldsymbol{\beta}}_{k*}\|^2 + \left(\frac{1}{4\phi^2} - \frac{1}{4\phi^2} - 1\right)\|\widehat{\boldsymbol{\beta}}_{j*} - \widehat{\boldsymbol{\beta}}_{k*}\|^2 \\
&\leq 4\|\widehat{\boldsymbol{\beta}}_{j*}\|^2 + \left(\frac{\|\widehat{\boldsymbol{\beta}}_{j*} - \widehat{\boldsymbol{\beta}}_{k*}\|}{2\phi}\right)^2 - \frac{1+4\phi^2}{4\phi^2}\|\widehat{\boldsymbol{\beta}}_{j*} - \widehat{\boldsymbol{\beta}}_{k*}\|^2 \\
&\leq 4\|\widehat{\boldsymbol{\beta}}_{j*}\|^2 + \left(\frac{\|\widehat{\boldsymbol{\beta}}_{j*} - \widehat{\boldsymbol{\beta}}_{k*}\|}{2\phi}\right)^2 - 2\frac{\|\widehat{\boldsymbol{\beta}}_{j*}\|\|\widehat{\boldsymbol{\beta}}_{j*} - \widehat{\boldsymbol{\beta}}_{k*}\|}{\phi} \\
&\leq \left(\|\widehat{\boldsymbol{\beta}}_{j*}\| + \|\widehat{\boldsymbol{\beta}}_{k*}\| - \frac{\|\widehat{\boldsymbol{\beta}}_{j*} - \widehat{\boldsymbol{\beta}}_{k*}\|}{2\phi}\right)^2
\end{aligned}
$$

Thus, we have

$$
\begin{aligned}
G(\widehat{\boldsymbol{B}}) - G(\boldsymbol{V}) &\geq \Delta\left(\|\widehat{\boldsymbol{\beta}}_{j*}\| + \|\widehat{\boldsymbol{\beta}}_{k*}\| - \|\widehat{\boldsymbol{\beta}}_{j*} + \widehat{\boldsymbol{\beta}}_{k*}\|\right) \\
&\geq \frac{\Delta\|\widehat{\boldsymbol{\beta}}_{j*} - \widehat{\boldsymbol{\beta}}_{k*}\|}{2\phi} = \frac{\Delta\|\boldsymbol{\varepsilon}\|}{\phi}
\end{aligned}
$$

Combining this with $\|\boldsymbol{\varepsilon}\|_1 \leq \sqrt{r}\|\boldsymbol{\varepsilon}\|$, we get

$$
\begin{aligned}
&L(\boldsymbol{V}) + G(\boldsymbol{V}) - (L(\widehat{\boldsymbol{B}}) + G(\widehat{\boldsymbol{B}})) \\
&\qquad \leq \left(\sqrt{r}\left\|\boldsymbol{x}_{*j} - \boldsymbol{x}_{*k}\right\|_1 - \frac{\Delta}{\phi}\right)\|\boldsymbol{\varepsilon}\| < 0
\end{aligned}
$$

This contradicts our assumption that $\widehat{\boldsymbol{B}}$ is the minimizer of $L(\boldsymbol{B})+G(\boldsymbol{B})$ and completes the proof. □

C.2 Clustering properties with squared Frobenius loss function

In this section, we consider the optimization

$$\min_{\boldsymbol{X}} \|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{B}\|_F^2 \; + \; G(\boldsymbol{B}) \tag{C.1}$$

Here we derive an upper bound on the increase in the squared loss term after applying the transformation, $\boldsymbol{V}$. We assume that the columns of the matrix, $\boldsymbol{X}$, are normalized to a common norm, *i.e.*, ($\|\boldsymbol{x}_{*i}\| = c$ for $i = 1, \cdots, p$). Define $L(\boldsymbol{X}) = \|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{B}\|_F^2 = \|\boldsymbol{Y}' - \boldsymbol{x}_{*j}\boldsymbol{\beta}_{j*} - \boldsymbol{x}_{*k}\boldsymbol{\beta}_{k*}\|_F^2$ where $\boldsymbol{Y}'$ is again the residual term.

Lemma C.1. *Let $\widehat{\boldsymbol{B}} \in \mathbb{R}^{p\times r}$ and if $\boldsymbol{V}$ is as defined in Theorem 4.5, then we have*

$$L(\boldsymbol{V}) - L(\widehat{\boldsymbol{B}}) \leq \|\boldsymbol{\varepsilon}\|\|\boldsymbol{Y}'\|_F\|\boldsymbol{x}_{*j} - \boldsymbol{x}_{*k}\|$$

Proof.

$$\begin{aligned} L(\boldsymbol{V}) - L(\widehat{\boldsymbol{B}}) = \frac{1}{2}\|\boldsymbol{Y}' - \boldsymbol{x}_{*j}(\widehat{\boldsymbol{\beta}}_{j*} - \boldsymbol{\varepsilon}) - \boldsymbol{x}_{*k}(\widehat{\boldsymbol{\beta}}_{k*} + \boldsymbol{\varepsilon})\|_F^2 \\ - \frac{1}{2}\|\boldsymbol{Y}' - \boldsymbol{x}_{*j}\widehat{\boldsymbol{\beta}}_{j*} - \boldsymbol{x}_{*k}\widehat{\boldsymbol{\beta}}_{k*}\|_F^2 \end{aligned}$$

Expanding the Frobenius norm terms, canceling the common $\frac{1}{2}\|\boldsymbol{Y}'\|_F^2$ terms and using the common norm of columns ($\|\boldsymbol{x}_{*i}\| = c$ for $i = 1, \cdots, p$)

we get

$$
\begin{aligned}
& L(\boldsymbol{V}) - L(\widehat{\boldsymbol{B}}) \\
& = \frac{c^2}{2}\text{tr}((\widehat{\boldsymbol{\beta}}_{j*} - \boldsymbol{\varepsilon})(\widehat{\boldsymbol{\beta}}_{j*} - \boldsymbol{\varepsilon})^T + (\widehat{\boldsymbol{\beta}}_{k*} + \boldsymbol{\varepsilon})(\widehat{\boldsymbol{\beta}}_{k*} + \boldsymbol{\varepsilon})^T \\
& - \widehat{\boldsymbol{\beta}}_{j*}\widehat{\boldsymbol{\beta}}_{j*}^T - \widehat{\boldsymbol{\beta}}_{k*}\widehat{\boldsymbol{\beta}}_{k*}^T) + \text{tr}(\boldsymbol{Y}'^T(\boldsymbol{x}_{*j} - \boldsymbol{x}_{*k})\boldsymbol{\varepsilon}) \\
& + \text{tr}((\widehat{\boldsymbol{\beta}}_{j*} - \boldsymbol{\varepsilon})\boldsymbol{x}_{*j}^T\boldsymbol{x}_{*k}(\widehat{\boldsymbol{\beta}}_{k*} + \boldsymbol{\varepsilon})^T - \widehat{\boldsymbol{\beta}}_{j*}\boldsymbol{x}_{*j}^T\boldsymbol{x}_{*k}\widehat{\boldsymbol{\beta}}_{k*}^T)
\end{aligned}
$$

Expanding terms and making further cancellations gives

$$
\begin{aligned}
& L(\boldsymbol{V}) - L(\widehat{\boldsymbol{B}}) \\
& = \text{tr}(\boldsymbol{Y}'^T(\boldsymbol{x}_{*j} - \boldsymbol{x}_{*k})\boldsymbol{\varepsilon}) - (c^2 - \boldsymbol{x}_{*j}^T\boldsymbol{x}_{*k})\ \text{tr}((\widehat{\boldsymbol{\beta}}_{j*} - \widehat{\boldsymbol{\beta}}_{k*} - \boldsymbol{\varepsilon})\boldsymbol{\varepsilon}^T) \\
& \leq \text{tr}(\boldsymbol{Y}'^T(\boldsymbol{x}_{*j} - \boldsymbol{x}_{*k})\boldsymbol{\varepsilon}) \\
& - (c^2 - \boldsymbol{x}_{*j}^T\boldsymbol{x}_{*k})\|\boldsymbol{\varepsilon}\|(\|\widehat{\boldsymbol{\beta}}_{j*}\| - \|\widehat{\boldsymbol{\beta}}_{k*}\| - \|\boldsymbol{\varepsilon}\|) \\
& \leq \text{tr}(\boldsymbol{Y}'^T(\boldsymbol{x}_{*j} - \boldsymbol{x}_{*k})\boldsymbol{\varepsilon}^T) \\
& \leq \|\boldsymbol{Y}'\|_F\|(\boldsymbol{x}_{*j} - \boldsymbol{x}_{*k})\boldsymbol{\varepsilon}\|_F \\
& = \|\boldsymbol{\varepsilon}\|\|\boldsymbol{Y}'\|_F\|\boldsymbol{x}_{*j} - \boldsymbol{x}_{*k}\|
\end{aligned}
$$

where the first inequality follows from simplification and Cauchy-Schwarz inequality. The second inequality follows from $c^2 > \boldsymbol{x}_{*j}^T\boldsymbol{x}_{*k}$ and $\|\widehat{\boldsymbol{\beta}}_{j*}\|_2 - \|\widehat{\boldsymbol{\beta}}_{k*}\|_2 - \|\boldsymbol{\varepsilon}\| > 0$ (by assumption). The third inequality follows, again, by Cauchy-Schwarz inequality.

□

Using this Lemma one can easily extend the clustering properties of GrOWL to the optimization in (C.1).

C.3 Proof of Theorem 4.3

Outline: the proof proceeds by finding a lower bound for the objective function in (4.11) and then we show that the proposed solution achieves this lower bound.

Proof. First, note that the following is true for any $\boldsymbol{B}$ and $\boldsymbol{V}$,

$$\begin{aligned}\|\boldsymbol{B}-\boldsymbol{V}\|_F^2 &= \sum_{i=1}^{p}\|\boldsymbol{\beta}_{i*}-\boldsymbol{v}_{i*}\|^2 \\ &\geq \sum_{i=1}^{p}(\|\boldsymbol{\beta}_{i*}\|-\|\boldsymbol{v}_{i*}\|)^2 = \|\tilde{\boldsymbol{\beta}}-\tilde{\boldsymbol{v}}\|^2\end{aligned}$$

where the inequality follows from reverse triangle inequality.

Combining this with $G(\boldsymbol{B}) = \Omega_{\boldsymbol{w}}(\tilde{\boldsymbol{\beta}})$, we have a lower bound on the objective function in (4.11). For all $\boldsymbol{B} \in \mathbb{R}^{p\times r}$

$$\frac{1}{2}\|\boldsymbol{B}-\boldsymbol{V}\|_F^2 + G(\boldsymbol{B}) \geq \frac{1}{2}\|\text{prox}_{\Omega_{\boldsymbol{w}}}(\tilde{\boldsymbol{v}})-\tilde{\boldsymbol{v}}\|^2 + \Omega_{\boldsymbol{w}}(\text{prox}_{\Omega_{\boldsymbol{w}}}(\tilde{\boldsymbol{v}}))$$

Finally, we show that $\boldsymbol{B} = \widehat{\boldsymbol{V}}$ achieves this lower bound,

$$
\begin{aligned}
&\frac{1}{2}\|\widehat{\boldsymbol{V}} - \boldsymbol{V}\|_F^2 + G(\widehat{\boldsymbol{V}}) \\
&= \frac{1}{2}\sum_{i=1}^{p} \|(\text{prox}_{\Omega_{\boldsymbol{w}}}(\tilde{\boldsymbol{v}}))_i \frac{\boldsymbol{v}_{i*}}{\|\boldsymbol{v}_{i*}\|} - \boldsymbol{v}_{i*}\|_2^2 + \Omega_{\boldsymbol{w}}(\text{prox}_{\Omega_{\boldsymbol{w}}}(\tilde{\boldsymbol{v}})) \\
&= \frac{1}{2}\|\text{prox}_{\Omega_{\boldsymbol{w}}}(\tilde{\boldsymbol{v}}) - \tilde{\boldsymbol{v}}\|_2^2 + \Omega_{\boldsymbol{w}}(\text{prox}_{\Omega_{\boldsymbol{w}}}(\tilde{\boldsymbol{v}}))
\end{aligned}
$$

□

Properties of OWL

Subgradient condition for OWL:

$$z_i = \text{sign}(\beta_i)(P_\beta w)_i, \text{ if } \beta_i \neq 0, \text{ and } z_i = 0, \text{otherwise.}$$

When defined as above, we can write the subdifferential vector as

$$z = (P_\beta w) \cdot \text{sign}(\beta)$$

Proof of Lemma 6.5

Proof. Let $\mathcal{T} = \{i | \beta_i^* \neq 0\}$. Note that $\mathcal{T} \subseteq \mathcal{T}$.

From optimality conditions, we have $X_{\mathcal{T}}^T(y - X\beta^*) = (P_{\beta^*} w)_{\mathcal{T}} \cdot \text{sign}(\beta_{\mathcal{T}}^*)$.

Consider a perturbation $\beta^* + th$.

Let P_{β^*} and P_{β^*+th} be permutation matrices such that $\Omega_w(\beta^*) = (P_{\beta^*} w)^T |\beta^*|$ and $\Omega_w(\beta^* + th) = (P_{\beta^*+th} w)^T |\beta^* + th|$. Notice that these permutation matrices may not be unique.

For $t > 0$ sufficiently small such that $\text{sign}(\beta_{\mathcal{T}}^*) = \text{sign}(\beta_{\mathcal{T}}^* + th_{\mathcal{T}})$ and the group ordering doesn't change i.e, there exist P_{β^*} and P_{β^*+th} such that

$(P_{\beta^*}w)_{\mathcal{T}} = (P_{\beta^*+th}w)_{\mathcal{T}}$,

$$
\begin{aligned}
&\Omega_w(\beta^* + th) \\
&= \langle (P_{\beta^*+th}w) \cdot \text{ sign } (\beta^* + th), \beta^* + th \rangle \\
&= \langle (P_{\beta^*+th}w)_{\mathcal{T}} \cdot \text{sign}(\beta^*_{\mathcal{T}} + th_{\mathcal{T}}), \beta^*_{\mathcal{T}} + th_{\mathcal{T}} \rangle + t(P_{\beta^*+th}w)^T_{\mathcal{T}^c}|h_{\mathcal{T}^c}| \\
&= \langle (P_{\beta^*+th}w)_{\mathcal{T}} \cdot \text{sign}(\beta^*_{\mathcal{T}}), \beta^*_{\mathcal{T}} + th_{\mathcal{T}} \rangle + t(P_{\beta^*+th}w)^T_{\mathcal{T}^c}|h_{\mathcal{T}^c}| \\
&= \Omega_w(\beta^*) + t\langle (P_{\beta^*}w)_{\mathcal{T}} \cdot \text{sign}(\beta^*_{\mathcal{T}}), h_{\mathcal{T}} \rangle + t(P_{\beta^*+th}w)^T_{\mathcal{T}^c}|h_{\mathcal{T}^c}| \\
&= \Omega_w(\beta^*) + t\langle (P_{\beta^*}w)_{\mathcal{T}} \cdot \text{sign}(\beta^*_{\mathcal{T}}), h_{\mathcal{T}} \rangle + t\Omega_{(P_{\beta^*+th}w)_{\mathcal{T}^c\downarrow}}(h_{\mathcal{T}^c}) \\
&\geq \Omega_w(\beta^*) + t\langle (P_{\beta^*}w)_{\mathcal{T}} \cdot \text{sign}(\beta^*_{\mathcal{T}}), h_{\mathcal{T}} \rangle + t\Omega_{w'}(h_{\mathcal{T}^c})
\end{aligned}
$$

where the last inequality follows since $\mathcal{T}^c \subseteq \mathcal{T}^c$ and $(P_{\beta^*+th}w)_{\mathcal{T}^c\downarrow} = [w_{|\mathcal{T}|+1}, \ldots, w_{|\mathcal{T}|}, w_{|\mathcal{T}|+1}, \ldots, w_N]$. The weights on $h_{\mathcal{T}^c}$ will only decrease. Recall from optimality conditions we have $X^T_{\mathcal{T}}(y - X\beta^*) = (P_{\beta^*}w)_{\mathcal{T}} \cdot \text{sign}(\beta^*_{\mathcal{T}})$, this gives us

$$
\begin{aligned}
&\frac{1}{2}\|y - X(\beta^* + th)\|_2^2 + \Omega_w(\beta^* + th) \\
&= \frac{1}{2}\|y - X\beta^*\|_2^2 + \frac{t^2}{2}\|h\|_2^2 - t\langle X^T(y - X\beta^*), h\rangle + \Omega_w(\beta^* + th) \\
&\geq \frac{1}{2}\|y - X\beta^*\|_2^2 + \Omega_w(\beta^*) + \frac{t^2}{2}\|h\|_2^2 + t\Omega_{w'}(h_{\mathcal{T}^c}) - \langle X^T_{\mathcal{T}^c}(y - X\beta^*), h_{\mathcal{T}^c}\rangle
\end{aligned}
$$

Let $X^T_{\mathcal{T}^c}(y - X\beta^*) =: \epsilon_{\mathcal{T}^c}$ and if $h_{\mathcal{T}^c} \neq 0$. Consider the last term,

$$\Omega_{w'}(h_{\mathcal{T}^c}) - \langle \epsilon_{\mathcal{T}^c}, h_{\mathcal{T}^c} \rangle \geq \Omega_{w'}(h_{\mathcal{T}^c}) - \Omega^*_{w'}(\epsilon_{\mathcal{T}^c})\Omega_{w'}(h_{\mathcal{T}^c})$$
$$> 0$$

from assumption $\Omega^*_{w'}(\epsilon_{\mathcal{T}^c}) < 1$.

This implies $\frac{1}{2}\|y - X(\beta^* + th)\|_2^2 + \Omega_w(\beta^* + th) > \frac{1}{2}\|y - X\beta^*\|_2^2 + \Omega_w(\beta^*)$ and the claim follows.

□

Proof of Lemma 6.6

Proof. Define:

$$\widehat{\beta}^{(1)} = \arg\min_{\beta^{(1)}} \frac{1}{2}\|y - X^{(1)}\beta^{(1)}\|_2^2 + \Omega_w(\beta^{(1)})$$

$$\bar{\beta}^{(1)} = \arg\min_{\beta^{(1)}} \Omega_w(\beta^{(1)}) \text{ s.t. } y = X^{(1)}\beta^{(1)}$$

Since $\widehat{\beta}^{(1)}$ minimizes the first optimization and $y = X^{(1)}\bar{\beta}^{(1)}$, we have the following

$$\frac{1}{2}\|y - X^{(1)}\widehat{\beta}^{(1)}\|_2^2 + \Omega_w(\widehat{\beta}^{(1)}) \;\leq\; \frac{1}{2}\|y - X^{(1)}\bar{\beta}^{(1)}\|_2^2 + \Omega_w(\bar{\beta}^{(1)}) \;=\; \Omega_w(\bar{\beta}^{(1)})$$

Let $h = \widehat{\beta}^{(1)} - \bar{\beta}^{(1)}$, then

$$\frac{1}{2}\|y - X^{(1)}\widehat{\beta}^{(1)}\|_2^2 \;=\; \frac{1}{2}\|X^{(1)}h\|_2^2 \;\leq\; \Omega_w(\bar{\beta}^{(1)}) - \Omega_w(\bar{\beta}^{(1)} + h)$$

Let P and Q be permutation matrices such that:

$$\Omega_w(\widehat{\beta}^{(1)}) = \langle P_{\widehat{\beta}^{(1)}} w, |\widehat{\beta}^{(1)}| \rangle$$

$$\Omega_w(\bar{\beta}^{(1)}) = \langle P_{\bar{\beta}^{(1)}} w, |\bar{\beta}^{(1)}| \rangle$$

Note that by definition of OWL norm $\langle P_{\bar{\beta}^{(1)}} w, |\widehat{\beta}^{(1)}| \rangle \leq \langle P_{\widehat{\beta}^{(1)}} w, |\widehat{\beta}^{(1)}| \rangle$. Let S be the support of $\bar{\beta}$, then

$$\begin{aligned}
\Omega_w(\bar{\beta}^{(1)} + h) - \Omega_w(\bar{\beta}^{(1)}) &= \langle P_{\widehat{\beta}^{(1)}} w, |\bar{\beta}^{(1)} + h| \rangle - \langle P_{\bar{\beta}^{(1)}} w, |\bar{\beta}^{(1)}| \rangle \\
&\geq \langle P_{\bar{\beta}^{(1)}} w, |\bar{\beta}^{(1)} + h| - |\bar{\beta}^{(1)}| \rangle \\
&= \langle (P_{\bar{\beta}^{(1)}} w)_S, |\bar{\beta}^{(1)}_S + h_S| - |\bar{\beta}^{(1)}_S| \rangle + \langle (P_{\bar{\beta}^{(1)}} w)_{S^c}, |h_{S^c}| \rangle \\
&\geq \langle \mathsf{sign}(\bar{\beta}^{(1)}_S) \cdot (P_{\bar{\beta}^{(1)}} w)_S, h_S \rangle + \langle (P_{\bar{\beta}^{(1)}} w)_{S^c}, |h_{S^c}| \rangle
\end{aligned}$$

Plugging into the inequality above gives

$$\frac{1}{2}\|X^{(1)}h\|_2^2 \leq -\langle \mathsf{sign}(\bar{\beta}^{(1)}_S) \cdot (P_{\bar{\beta}^{(1)}} w)_S, h_S \rangle - \langle (P_{\bar{\beta}^{(1)}} w)_{S^c}, |h_{S^c}| \rangle$$

since $\bar{\beta}^{(1)}$ is optimal there exists ν such that

$$v = X^{(1)^T}\nu, v_S = \text{sign}(\bar{\beta}_S^{(1)}) \cdot (P_{\bar{\beta}^{(1)}}w)_S \text{ and } \Omega^*_{(P_{\bar{\beta}^{(1)}}w)_{S^c}}(v_{S^c}) \leq 1$$

using this

$$\langle \text{sign}(\bar{\beta}_S^{(1)}) \cdot (P_{\bar{\beta}^{(1)}}w)_S, h_S\rangle = \langle v_S, h_S\rangle = \langle \nu, X^{(1)}h\rangle - \langle v_{S^c}, h_{S^c}\rangle$$

Thus we have

$$\begin{aligned}|\langle \text{sign}(\bar{\beta}_S^{(1)}) \cdot (P_{\bar{\beta}^{(1)}}w)_S, h_S\rangle| &\leq |\langle \nu, X^{(1)}h\rangle| + |\langle v_{S^c}, h_{S^c}\rangle| \\ &\leq \|X^{(1)}h\|_2\|\nu\|_2 + \Omega^*_{(P_{\bar{\beta}^{(1)}}w)_{S^c}}(v_{S^c})\Omega_{(P_{\bar{\beta}^{(1)}}w)_{S^c}}(h_{S^c}) \\ &\leq \|X^{(1)}h\|_2\|\nu\|_2 + \Omega_{(P_{\bar{\beta}^{(1)}}w)_{S^c}}(h_{S^c})\end{aligned}$$

Plugging back into

$$\frac{1}{2}\|X^{(1)}h\|_2^2 \leq \|X^{(1)}h\|_2\|\nu\|_2 + \Omega_{(P_{\bar{\beta}^{(1)}}w)_{S^c}}(h_{S^c}) - \langle (P_{\bar{\beta}^{(1)}}w)_{S^c}, |h_{S^c}|\rangle$$

Note that since $\bar{\beta}_{S^c} = 0$, we have $h_{S^c} = \widehat{\beta}_{S^c}$ and $\Omega_{(P_{\bar{\beta}^{(1)}}w)_{S^c}}(h_{S^c}) = \langle (P_{\bar{\beta}^{(1)}}w)_{S^c}, |h_{S^c}|\rangle$. Thus it follows that

$$\frac{1}{4}\|X^{(1)}h\|_2^2 \leq \|\nu\|_2^2 \leq cw_1^2\frac{d}{\log(\frac{N}{d})}$$

where proof Last inequality is shown next.

□

Consider the exact OWL norm minimization problem,

$$\min_{\beta} \ \Omega_w(\beta) \ \text{ s.t. } \ y = X\beta$$

and its dual

$$\max_{\nu} \langle y, \nu \rangle \text{ s.t. } \ \Omega_w^*(X^T\nu) \ \leq 1$$

Any dual feasible point ν satisfies,

$$\frac{1}{w_1}\|X^T\nu\|_\infty \leq \Omega_w^*(X^T\nu) \ \leq 1$$

Thus, $\frac{1}{w_1}\nu \in K^o$ which implies

$$\|\frac{1}{w_1}\nu\|_2 \leq R(K^o) = \frac{1}{r(K)}$$

where K =conv(x_i), $r(K)$ is its inradius, K^o is the polar set and $R(K^o)$ is its circumradius. Equality follows since $R(K^o).r(K) = 1$.

We get

$$\|\nu\|_2 \leq \frac{w_1}{r(K)}$$

finally, using standard results (like Lemma 7.4 in Soltanolkotabi et al. (2012)) we get with probability at least $1 - e^{-\sqrt{Nd}}$,

$$\|\nu\|_2^2 \leq cw_1^2 \frac{d}{\log(\frac{N}{d})}$$

where c is a constant.

Proof of Lemma 6.8

Proof. Suppose we divide the surface of the unit hypersphere into m patches of equal surface area. Then the probability that a uniformly sampled point falls into a particular patch is $p = 1/m$.

Let $C =$ the number of times a sample falls in a particular patch in N_ℓ independent trials. The multiplicative form of Chernoff's bound is

$$P(C \leq (1-b)\mu) \leq \exp\left(-\frac{b^2\mu}{2}\right), \text{ for any } 0 < b \leq 1$$

where $\mu = E[C] = N_\ell p = N_\ell/m$. Taking $b = 1$ we get

$$P(C \leq 0) \leq \exp\left(-\frac{N_\ell}{2m}\right)$$

Let $\delta' \geq \exp\left(-\frac{N_\ell}{2m}\right)$. It follows that if $N_\ell \geq 2m\log(1/\delta')$, then with probability at least $1-\delta'$ there is at least one sample falling in the patch. Now if we want this to hold for all m patches, we can union bound to get the following.

$$P(\cup_{i=1}^m C_i \leq 0) \leq \sum_{i=1}^m P(C_i \leq 0) \leq m\delta' =: \delta$$

If $N_\ell \geq 2m\log(m/\delta)$, then with probability at least $1-\delta$ there is at least one sample in each of the m patches.

If we construct the patches such that the distance between any point in a certain patch and any point in adjacent patches is always less than or equal to Δ, this gives us a completely connected Δ-RGG.

Construction of the patches: ϵ-covering number ($N_{d_\ell}^{\epsilon}(\mathbb{S}^{d_\ell})$) of unit sphere in $\mathbb{R}^{d_\ell}$ has the following property:

$$N_{d_\ell}^{\epsilon}(\mathbb{S}^{d_\ell}) \leq 3\epsilon^{-d_\ell}$$

so we need at least $m = O(\Delta^{-d_\ell})$ patches for $\epsilon < \Delta/2$ and $N_\ell > \kappa_1\Delta^{-d_\ell}\log(\Delta^{-d_\ell}/\delta)$ points in the subspace to say with probability at least $1-\delta$, there will be a point in every patch on the surface of the unit sphere leading to a connected Δ-RGG. □

Proof of Lemma 4.2

Proof. By contradiction, assume $|M| < r$.

Let $r' = r - |M|$ and let G be the Δ-connected component containing elements of M.

Define $G_0 := M$. For $m = 1, \ldots, r'$, define

$G_m = G_{m-1} \cup j$ for some $j \in G, j \notin G_{m-1}, \exists i \in G_{m-1}$ such that $\|x_i - x_j\| < \Delta$.

By definition, we have for $m = 0, \ldots, r'$ that $G_m \subseteq G$.

If for some $m \in [r']$, the set $G_{m-1} = M = \arg\max_i |\widehat{\beta}_i|$, then by definition there exists $j \in G_m$ such that $|\widehat{\beta}_j| < |\widehat{\beta}_i|$, $\forall i \in G_{m-1}$.

In the rest of the proof, we show a contradiction arises and completes the proof by induction.

Consider the following alternative solution, $\tilde{\beta} \in \mathbb{R}^p$, such that

$$\begin{aligned}
&\tilde{\beta}_j = \widehat{\beta}_j + \epsilon, \\
&\tilde{\beta}_i = \widehat{\beta}_i - \epsilon, \text{ for the } i \in G_{m-1} \text{ that is } \Delta \text{ close to } j \\
&\tilde{\beta}_k = \widehat{\beta}_k \text{ for } k \neq i, j
\end{aligned}$$

where $\epsilon \in (0, \min\{\frac{|\widehat{\beta}_i| - |\widehat{\beta}_j|}{2}, |\widehat{\beta}_i| - |\widehat{\mu}_2|\})$ where $|\widehat{\mu}_2|$ is the second largest unique magnitude of $\widehat{\beta}$, this ensures the components of M stay in the top r in the alternative solution.

From Lemma 2.3 of Figueiredo and Nowak (2016) we have

$$\begin{aligned}
L(\tilde{\beta}) - L(\widehat{\beta}) &= \|y - X\tilde{\beta}\|_2^2 - \|y - X\widehat{\beta}\|_2^2 \\
&\leq \epsilon \|y\| \|x_i - x_j\| \\
&< \epsilon \Delta
\end{aligned}$$

where the last inequality follows by assumption $\|x_i - x_j\| < \Delta$ and $\|y\| = 1$.

Also, we have

$$\begin{aligned}\Omega_w(\tilde{\beta}) - \Omega_w(\widehat{\beta}) &= \lambda\|\tilde{\beta}\|_1 + \Delta\sum_{i=1}^{d}(d-i+1)|\tilde{\beta}_{[i]}| - \lambda\|\widehat{\beta}\|_1 - \Delta\sum_{i=1}^{d}(r-i+1)|\widehat{\beta}_{[i]}| \\ &= \Delta\sum_{i=1}^{r}(r-i+1)|\tilde{\beta}_{[i]}| - \Delta\sum_{i=1}^{r}(r-i+1)|\widehat{\beta}_{[i]}| \\ &\leq -\Delta\epsilon\end{aligned}$$

where the second last equality follows since $|\widehat{\beta}_i|$ is in the top r magnitudes of $\widehat{\beta}$ and the definition of ϵ ensures $|\tilde{\beta}_i|$ is in the top r magnitudes of $\tilde{\beta}$ along with a variant of Lemma 2.1 when $\text{sign}(\widehat{\beta}_i) = \text{sign}(\widehat{\beta}_j)$ or a variant of Lemma 2.2 when $\text{sign}(\widehat{\beta}_i) \neq \text{sign}(\widehat{\beta}_j)$ from Figueiredo and Nowak (2016) (using the fact that $w_{\ell+a} - w_{m-b} \geq \Delta$ and $w_{\ell+a} - w_{m+b} \geq \Delta$ since we ensure that $\ell + a > r$ and $m - b \geq r$.)

Putting these together we have

$$L(\tilde{\beta}) - L(\widehat{\beta}) + \Omega_w(\tilde{\beta}) - \Omega_w(\widehat{\beta}) < 0$$

This contradicts our assumption that $\widehat{\beta}$ is the minimizer of $L(\beta) + \Omega_w(\beta)$. □

D.1 Proof of Theorem 6.3

We start with the deterministic lemma stated in Lemma 6.5 that introduces the OWL dual feasibility condition.

The lemma tells us that if the OWL dual feasibility condition $\Omega^*_{w'}(X^T_{\mathcal{T}^c}(y - X\beta^*)) < 1$ is satisfied for $\mathcal{T} = \{j : X_j \in \mathcal{S}_\ell\}$, then we have no false discoveries. To prove Theorem 6.3, it suffices to show that the dual feasibility condition is satisfied. One sufficient condition to satisfy the OWL dual feasibility condition is $\|X^T_{\mathcal{T}^c}(y - X\beta^*)\|_\infty < \bar{w}_{|\mathcal{T}|+1}$ since it can be shown easily that for $\bar{w} = \sum_{j=1}^N w_j/N$, we have

$$\frac{1}{w_1}\|\beta\|_\infty \leq \Omega^*_w(\beta) \leq \frac{1}{\bar{w}}\|\beta\|_\infty$$

To show the dual feasibility is satisfied we use the following result.

Lemma D.1 (Theorem 7.5 in Soltanolkotabi et al. (2012))**.** *Let $A \in \mathbb{R}^{d_\ell \times N_\ell}$ be a matrix with columns sampled uniformly at random from the unit sphere of $\mathbb{R}^{d_\ell}$, $v \in \mathbb{R}^{d_j}$ be a vector sampled uniformly at random from the unit sphere of $\mathbb{R}^{d_j}$ and independent of A and $\Sigma \in \mathbb{R}^{d_\ell \times d_j}$ be a deterministic matrix. We have*

$$\|A^T\Sigma v\|_\infty \leq \sqrt{\log a \log b}\frac{\|\Sigma\|_F}{\sqrt{d_\ell}\sqrt{d_j}},$$

with probability at least $1 - \frac{2}{\sqrt{a}} - \frac{2N_\ell}{\sqrt{b}}$.

We can use this as follows. Suppose $\Sigma = {U^{(j)}}^T U^{(\ell)}$, where $U^{(j)}$ is an orthogonal basis for $\mathcal{S}_j$ and $U^{(\ell)}$ for $\mathcal{S}_\ell$ respectively. By definition, $\|\Sigma\|_F = \sqrt{d_\ell \wedge d_j}\text{aff}(\mathcal{S}_j, \mathcal{S}_\ell)$. Consider

$$\|X_j^T(y - X\beta^*)\|_\infty = \|A^T\Sigma v\|_\infty \|y - X\beta^*\|_2$$

Using the Lemma with $a = N^4, b = N^8$, we have with probability at least $1 - 4/N^2$

$$\|X_j^T(y - X\beta^*)\|_\infty \leq \sqrt{32}\log N \frac{\text{aff}(\mathcal{S}_\ell, \mathcal{S}_j)}{\sqrt{d_\ell}}\|y - X\beta^*\|_2$$

Lemma 6.6 gives, with probability at least $1 - e^{-\sqrt{N_\ell d_\ell}} - 4/N^2$,

$$\|X_j^T(y - X\beta^*)\|_\infty \leq w_1 L_N \text{aff}(\mathcal{S}_\ell, \mathcal{S}_j)$$

where $L_N = c_0 \frac{\log N}{\sqrt{\log \rho_\ell}}$, for $j \neq \ell$.

Finally using the assumption on the affinity of the subspaces we get $\|X_j^T(y - X\beta^*)\|_\infty \leq \bar{w}_{|\mathcal{T}|+1}$ with probability at least $1 - e^{-\sqrt{N_\ell d_\ell}} - 4/N^2$ which along with union bound and Lemma 6.5 completes the proof of Theorem 6.3.

D.2 Proof of Theorem 6.4

Let $\mathcal{T}$ be set of indices of columns belonging to the subspace, $\mathcal{S}_\ell$. From Theorem 6.3, we have with high probability the coefficients of $\widehat{\beta}_{\mathcal{T}^c} = 0$. Lemma 6.8 with the assumption on the number of points sampled gives us that the Δ-RGG formed by the points $X_\mathcal{T}$ on the unit hypersphere is fully connected. The connected component has cardinality at least N_ℓ. If $\max_i |\widehat{\beta}_i| = 0$, then the claim is trivial so suppose $\max_i |\widehat{\beta}_i| > 0$, then

$M \subseteq \mathcal{T}$ and the claim follows from $r \leq N_\ell$ and Lemma 4.2.

REFERENCES

Abbasi-Yadkori, Yasin, David Pal, and Csaba Szepesvari. 2011. Improved Algorithms for Linear Stochastic Bandits. *Advances in Neural Information Processing Systems (NIPS)* 1–19.

———. 2012. Online-to-Confidence-Set Conversions and Application to Sparse Stochastic Bandits. In *Proceedings of the international conference on artificial intelligence and statistics (aistats)*.

Abe, Naoki, and Philip M. Long. 1999. Associative reinforcement learning using linear probabilistic concepts. In *Proceedings of the international conference on machine learning (icml)*, 3–11.

Antani, Sameer, Rangachar Kasturi, and Ramesh Jain. 2002. A survey on the use of pattern recognition methods for abstraction, indexing and retrieval of images and video. *Pattern recognition* 35(4):945–965.

Argyriou, Andreas, Rina Foygel, and Nathan Srebro. 2012. Sparse prediction with the k-support norm. In *Advances in neural information processing systems*, 1457–1465.

Auer, Peter, and M Long. 2002. Using Confidence Bounds for Exploitation-Exploration Trade-offs. *Journal of Machine Learning Research* 3:2002.

Belilovsky, Eugene, Andreas Argyriou, Gaël Varoquaux, and Matthew Blaschko. 2015. Convex relaxations of penalties for sparse correlated

variables with bounded total variation. *Machine Learning* 100(2-3):533–553.

Bogdan, Małgorzata, Ewout van den Berg, Chiara Sabatti, Weijie Su, and Emmanuel J Candès. 2015. Slope?adaptive variable selection via convex optimization. *The annals of applied statistics* 9(3):1103.

Bogdan, Malgorzata, Ewout van den Berg, Chiara Sabatti, Weijie Su, and Emmanuel J Candes. 2014. Slope–adaptive variable selection via convex optimization. *arXiv preprint arXiv:1407.3824*.

Bogdan, Malgorzata, Ewout van den Berg, Weijie Su, and Emmanuel Candes. 2013. Statistical estimation and testing via the sorted l1 norm. *arXiv preprint arXiv:1310.1969*.

Bondell, Howard D, and Brian J Reich. 2008. Simultaneous regression shrinkage, variable selection, and supervised clustering of predictors with oscar. *Biometrics* 64(1):115–123.

Borgefors, Gunilla. 1988. Hierarchical chamfer matching: A parametric edge matching algorithm. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 10(6):849–865.

Boutsidis, Christos, Petros Drineas, and Malik Magdon-Ismail. 2014. Near-optimal column-based matrix reconstruction. *SIAM Journal on Computing* 43(2):687–717.

Boutsidis, Christos, and David P Woodruff. 2014. Optimal cur matrix decompositions. In *Proceedings of the 46th annual acm symposium on theory of computing*, 353–362. ACM.

Chen, Yuxin, Oisin Mac Aodha, Shihan Su, Pietro Perona, and Yisong Yue. 2018. Near-optimal machine teaching via explanatory teaching sets. In *International conference on artificial intelligence and statistics*, 1970–1978.

Cox, Christopher R. 2016. Testing neurocognitive predictions of the hub-and-spoke model of semantic memory with network representational similarity analysis. Ph.D. thesis, University of Wisconsin-Madison.

Croft, W Bruce, and Raj Das. 1989. Experiments with query acquisition and use in document retrieval systems. In *Proceedings of the 13th annual international acm sigir conference on research and development in information retrieval*, 349–368. ACM.

Dall, Jesper, and Michael Christensen. 2002. Random geometric graphs. *Physical review E* 66(1):016121.

Dani, Varsha, Thomas P Hayes, and Sham M Kakade. 2008. Stochastic linear optimization under bandit feedback.

Deshpande, Yash, and Andrea Montanari. 2012. Linear bandits in high dimension and recommendation systems. In *2012 50th annual allerton conference on communication, control, and computing (allerton)*, 1750–1754. IEEE.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Drineas, Petros, Malik Magdon-Ismail, Michael W Mahoney, and David P Woodruff. 2012. Fast approximation of matrix coherence and statistical leverage. *Journal of Machine Learning Research* 13(Dec):3475–3506.

Drineas, Petros, Michael W Mahoney, and S Muthukrishnan. 2006. Sampling algorithms for l 2 regression and applications. In *Proceedings of the seventeenth annual acm-siam symposium on discrete algorithm*, 1127–1136. Society for Industrial and Applied Mathematics.

———. 2008. Relative-error cur matrix decompositions. *SIAM Journal on Matrix Analysis and Applications* 30(2):844–881.

Druck, Gregory, Burr Settles, and Andrew McCallum. 2009. Active learning by labeling features. In *Proceedings of the 2009 conference on empirical methods in natural language processing: Volume 1-volume 1*, 81–90. Association for Computational Linguistics.

Dyer, Eva L, Aswin C Sankaranarayanan, and Richard G Baraniuk. 2013. Greedy feature selection for subspace clustering. *The Journal of Machine Learning Research* 14(1):2487–2517.

Elhamifar, Ehsan, and René Vidal. 2009. Sparse subspace clustering. In *Computer vision and pattern recognition, 2009. cvpr 2009. ieee conference on*, 2790–2797. IEEE.

Elhamifar, Ehsan, and Rene Vidal. 2013. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE transactions on pattern analysis and machine intelligence* 35(11):2765–2781.

Eriksson, Brian, Laura Balzano, and Robert Nowak. 2012. High-rank matrix completion. In *Artificial intelligence and statistics*, 373–381.

Figueiredo, Mario, and Robert Nowak. 2016. Ordered weighted l1 regularized regression with strongly correlated covariates: Theoretical aspects. In *Artificial intelligence and statistics*, 930–938.

Figueiredo, Mario AT, and Robert D Nowak. 2014. Sparse estimation with strongly correlated variables using ordered weighted l1 regularization. *arXiv preprint arXiv:1409.4005*.

Garbarino, M., et al. 2014. Empatica e3 - a wearable wireless multi-sensor device for real-time computerized biofeedback and data acquisition. In *Proc. 4th int. conf. wirel. mob. commun. healthc.*, 39–42.

Genovese, Christopher, Jiashun Jin, and Larry Wasserman. 2009. Revisiting marginal regression. *arXiv preprint arXiv:0911.4080*.

Genovese, Christopher R, Jiashun Jin, Larry Wasserman, and Zhigang Yao. 2012. A comparison of the lasso and marginal regression. *Journal of Machine Learning Research* 13(Jun):2107–2143.

Ginebra, Josep, and Murray K Clayton. 1995. Response surface bandits. *Journal of the Royal Statistical Society. Series B (Methodological)* 771–784.

Guyon, Isabelle, Steve R Gunn, Asa Ben-Hur, and Gideon Dror. 2004. Result analysis of the nips 2003 feature selection challenge. In *Nips*, vol. 4, 545–552.

Harm, Michael W, and Mark S Seidenberg. 2004. Computing the meanings of words in reading: cooperative division of labor between visual and phonological processes. *Psychological review* 111(3):662.

Hastie, Trevor, and Patrice Y Simard. 1998. Metrics and models for handwritten character recognition. *Statistical Science* 54–65.

Heckel, Reinhard, and Helmut Bölcskei. 2015. Robust subspace clustering via thresholding. *IEEE Transactions on Information Theory* 61(11):6320–6342.

Heckel, Reinhard, Michael Tschannen, and Helmut Bölcskei. 2017. Dimensionality-reduced subspace clustering. *Information and Inference: A Journal of the IMA* iaw021.

Ho, Jeffrey, Ming-Husang Yang, Jongwoo Lim, Kuang-Chih Lee, and David Kriegman. 2003. Clustering appearances of objects under varying

illumination conditions. In *Computer vision and pattern recognition, 2003. proceedings. 2003 ieee computer society conference on*, vol. 1, I–I. IEEE.

Jain, Swayambhoo, Urvashi Oswal, Kevin Shuai Xu, Brian Eriksson, and Jarvis Haupt. 2017. A compressed sensing based decomposition of electrodermal activity signals. *IEEE Transactions on Biomedical Engineering* 64(9):2142–2151.

Kriegeskorte, Nikolaus, Rainer Goebel, and Peter Bandettini. 2006. Information-based functional brain mapping. *Proceedings of the National Academy of Sciences of the United States of America* 103(10):3863–3868.

Kriegeskorte, Nikolaus, and Rogier A Kievit. 2013. Representational geometry: integrating cognition, computation, and the brain. *Trends in cognitive sciences* 17(8):401–412.

Kriegeskorte, Nikolaus, Marieke Mur, and Peter Bandettini. 2008. Representational similarity analysis–connecting the branches of systems neuroscience. *Frontiers in systems neuroscience* 2.

Lang, Ken. 1995. Newsweeder: Learning to filter netnews. In *Machine learning proceedings 1995*, 331–339. Elsevier.

Lattimore, Tor, and Csaba Szepesvári. 2018. Bandit algorithms.

LeCun, Yann, Corinna Cortes, and Christopher J.C. Burges. The mnist database. `http://yann.lecun.com/exdb/mnist/`. Accessed: 2018.

Lian, Wenzhao, Vinayak Rao, Brian Eriksson, and Lawrence Carin. 2014. Modeling correlated arrival events with latent semi-markov processes. In *Proceedings of the international conference on machine learning (icml)*.

Liu, S., and W. Deng. 2015. Very deep convolutional neural network based image classification using small training sample size. In *2015 3rd iapr asian conference on pattern recognition (acpr)*, 730–734.

Lounici, Karim, Massimiliano Pontil, Alexandre B Tsybakov, and Sara Van De Geer. 2009. Taking advantage of sparsity in multi-task learning. *arXiv preprint arXiv:0903.1468*.

Lounici, Karim, Massimiliano Pontil, Sara Van De Geer, and Alexandre B Tsybakov. 2011. Oracle inequalities and optimal inference under group sparsity. *The Annals of Statistics* 2164–2204.

Mackey, Lester W, Michael I Jordan, and Ameet Talwalkar. 2011. Divide-and-conquer matrix factorization. In *Advances in neural information processing systems*, 1134–1142.

Mahoney, Michael W, and Petros Drineas. 2009. Cur matrix decompositions for improved data analysis. *Proceedings of the National Academy of Sciences* 106(3):697–702.

McRae, Ken, George S Cree, Mark S Seidenberg, and Chris McNorgan. 2005. Semantic feature production norms for a large set of living and nonliving things. *Behavior research methods* 37(4):547–559.

Obozinski, Guillaume, Martin J Wainwright, and Michael I Jordan. 2011. Support union recovery in high-dimensional multivariate regression. *The Annals of Statistics* 1–47.

Oswal, Urvashi, Aniruddha Bhargava, and Robert Nowak. 2019. Linear bandits with feature feedback. *arXiv preprint arXiv:1903.03705*.

Oswal, Urvashi, Christopher Cox, Matthew Lambon-Ralph, Timothy Rogers, and Robert Nowak. 2016. Representational similarity learning with application to brain networks. In *International conference on machine learning*, 1041–1049.

Oswal, Urvashi, Swayambhoo Jain, Kevin S Xu, and Brian Eriksson. 2018. Block cur: Decomposing matrices using groups of columns. In *Joint european conference on machine learning and knowledge discovery in databases*, 360–376. Springer.

Oswal, Urvashi, and Robert Nowak. 2018. Scalable sparse subspace clustering via ordered weighted l 1 regression. In *2018 56th annual allerton conference on communication, control, and computing (allerton)*, 305–312. IEEE.

Pan, Sinno Jialin, and Qiang Yang. 2010. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering* 22(10):1345–1359.

Parikh, Neal, and Stephen Boyd. 2013. Proximal algorithms. *Foundations and Trends in optimization* 1(3):123–231.

Park, Dohyung, Constantine Caramanis, and Sujay Sanghavi. 2014. Greedy subspace clustering. In *Advances in neural information processing systems*, 2753–2761.

Paschou, Peristera, Elad Ziv, Esteban G Burchard, Shweta Choudhry, William Rodriguez-Cintron, Michael W Mahoney, and Petros Drineas. 2007. Pca-correlated snps for structure identification in worldwide human populations. *PLoS Genet* 3(9):e160.

Penrose, Mathew. 2003. *Random geometric graphs*. 5, Oxford University Press.

Plaut, David C, James L McClelland, Mark S Seidenberg, and Karalyn Patterson. 1996. Understanding normal and impaired word reading: computational principles in quasi-regular domains. *Psychological review* 103(1):56.

Poulis, Stefanos, and Sanjoy Dasgupta. 2017. Learning with feature feedback: from theory to practice. In *Artificial intelligence and statistics*, 1104–1113.

Raghavan, Hema, Omid Madani, and Rosie Jones. 2006. Active learning with feedback on features and instances. *Journal of Machine Learning Research* 7(Aug):1655–1686.

Roads, Brett, Michael C Mozer, and Thomas A Busey. 2016. Using highlighting to train attentional expertise. *PloS one* 11(1):e0146266.

Rudelson, Mark, and Roman Vershynin. 2007. Sampling from large matrices: An approach through geometric functional analysis. *Journal of the ACM (JACM)* 54(4):21.

Rusmevichientong, Paat, and John N Tsitsiklis. 2010. Linearly Parameterized Bandits. *Math. Oper. Res.* 35(2):395–411.

Shaver, Phillip, Judith Schwartz, Donald Kirson, and Cary O'connor. 1987. Emotion knowledge: further exploration of a prototype approach. *Journal of personality and social psychology* 52(6):1061.

Shepard, Roger N. 1980. Multidimensional scaling, tree-fitting, and clustering. *Science* 210(4468):390–398.

Silveira, Fernando, Brian Eriksson, Anmol Sheth, and Adam Sheppard. 2013. Predicting audience responses to movie content from electrodermal activity signals. In *Proceedings of the 2013 acm ubicomp*, 707–716. ACM.

Soltanolkotabi, Mahdi, Emmanuel J Candes, et al. 2012. A geometric analysis of subspace clustering with outliers. *The Annals of Statistics* 40(4): 2195–2238.

Soltanolkotabi, Mahdi, Ehsan Elhamifar, Emmanuel J Candes, et al. 2014. Robust subspace clustering. *The Annals of Statistics* 42(2):669–699.

Sutton, Richard S, and Andrew G Barto. 1998. *Reinforcement learning: An introduction*, vol. 1. MIT press Cambridge.

Tibshirani, Robert. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)* 58(1): 267–288.

Tron, Roberto, and René Vidal. 2007. A benchmark for the comparison of 3-d motion segmentation algorithms. In *Computer vision and pattern recognition, 2007. cvpr'07. ieee conference on*, 1–8. IEEE.

Tversky, Amos, and Itamar Gati. 1982. Similarity, separability, and the triangle inequality. *Psychological review* 89(2):123.

Vidal, René. 2011. Subspace clustering. *IEEE Signal Processing Magazine* 28(2):52–68.

Vidal, René, Roberto Tron, and Richard Hartley. 2008. Multiframe motion segmentation with missing data using powerfactorization and gpca. *International Journal of Computer Vision* 79(1):85–105.

Wang, Shusen, and Zhihua Zhang. 2012. A scalable cur matrix decomposition algorithm: lower time complexity and tighter bound. In *Advances in neural information processing systems*, 647–655.

Wang, Yining, Yu-Xiang Wang, and Aarti Singh. 2015. A deterministic analysis of noisy sparse subspace clustering for dimensionality-reduced data. In *Proceedings of the 32nd international conference on machine learning (icml-15)*, 1422–1431.

Wang, Yu-Xiang, and Huan Xu. 2016. Noisy sparse subspace clustering. *Journal of Machine Learning Research* 17(12):1–41.

Xu, Miao, Rong Jin, and Zhi-Hua Zhou. 2015. Cur algorithm for partially observed matrices. In *Proceedings of the international conference on machine learning (icml)*, 1412–1421.

Yang, Allen Y, John Wright, Yi Ma, and S Shankar Sastry. 2008. Unsupervised segmentation of natural images via lossy data compression. *Computer Vision and Image Understanding* 110(2):212–225.

Yang, Jiyan, Oliver RÃ¼bel, Michael W Mahoney, and Benjamin P Bowen. 2015. Identifying important ions and positions in mass spectrometry imaging data using cur matrix decompositions. *Analytical chemistry* 87(9): 4658–4666.

Yessenalina, Ainur, Yisong Yue, and Claire Cardie. 2010. Multi-level structured models for document-level sentiment classification. In *Proceedings of the 2010 conference on empirical methods in natural language processing*, 1046–1056. Association for Computational Linguistics.

Yip, Ching-Wa, Michael W Mahoney, Alexander S Szalay, István Csabai, Tamás Budavári, Rosemary FG Wyse, and Laszlo Dobos. 2014. Objective identification of informative wavelength regions in galaxy spectra. *The Astronomical Journal* 147(5):110.

You, Chong, Daniel Robinson, and René Vidal. 2016. Scalable sparse subspace clustering by orthogonal matching pursuit. In *Proceedings of the ieee conference on computer vision and pattern recognition*, 3918–3927.

Zeng, Xiangrong, Mario Figueiredo, et al. 2014. The atomic norm formulation of oscar regularization with application to the frank-wolfe algorithm. In *Proceedings of the european signal processing conference, lisbon, portugal*.

Zeng, Xiangrong, and Mário AT Figueiredo. 2014. The ordered weighted ℓ_1 norm: Atomic formulation, projections, and algorithms. *arXiv preprint arXiv:1409.4271*.

Zhong, Leon Wenliang, and James T Kwok. 2012. Efficient sparse modeling with automatic feature grouping. *Neural Networks and Learning Systems, IEEE Transactions on* 23(9):1436–1447.

Zou, Hui, and Trevor Hastie. 2005. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67(2):301–320.