Critical Approaches to Inverse Optimization and Other Algorithmic Technologies for Modeling Values and Preferences

by

Ari Smith

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

Doctor of Philosophy

(Industrial and Systems Engineering)

at the

UNIVERSITY OF WISCONSIN–MADISON

2025

Date of final oral examination: 05/08/2025

The dissertation is approved by the following members of the Final Oral Committee:
Justin Boutilier, Assistant Professor, Industrial and Systems Engineering
Yonatan Mintz, Assistant Professor, Industrial and Systems Engineering
Jeffrey Linderoth, Professor, Industrial and Systems Engineering
Alan Rubel, Professor, Library and Information Sciences

Contents

Abstract

This dissertation studies and develops models and solution methods for two new applications of inverse optimization in areas concerning analyzing fairness in system design and decision making. One such case is for the analysis of congressional districtings to identify partisan gerrymandering, and the second is focused on detecting imbalances in representation in training sets of black-box machine learning predictors across social groups. In developing these applications we propose and study new methods for solving generalized inverse optimization problems with subgradient methods, as well as producing heuristic approaches for the districting application domain and methods for performing hypothesis tests on inferred degrees of imbalance for the machine learning application domain. Additionally, this thesis studies preference elicitation techniques and recommender systems, which have structural similarities to inverse optimization. First, a comparative study of algorithmic methods in preference elicitation is conducted through the theoretical frameworks of feminist science and technology studies scholars, particularly those of Susan Leigh Star and Karen Barad, to understand how different techniques infrastructure notions of agency and preference differently in users and how power dynamics may enter in the enacting of agential cuts in preference elicitation methods as apparatuses of observation. Finally, an applied study in developing a recommender system is carried out, producing a recommender system for family caregivers of people with Alzheimer's disease and related dementias, helping caregivers find support resources for care topics with which they need the most assistance.

Chapter 1

Introduction

1.1 Thematic Outline of This Thesis

This thesis is primarily concerned with algorithmic methods that concern preference elicitation, broadly construed, or that are structurally similar to such methods. As recommender systems increasingly pervade everyday life, especially in online/digital spaces, it is necessary that implementation and evaluation of preference elicitation methods, which often constitute one element of recommender systems, be done with a critical eye. Preference elicitation is often discursively deployed by its implementers as a means of producing knowledge claims about a human subject. Particularly, it is often posed as a method of understanding an internal state of said subject, such as a utility function or some personal value structure, through observation of external actions, behaviors, and choices.

Taking inspiration from scholars in science and technology studies (STS), if one adopts the stance that non-human agents can engage with value structures, whether it be holding, promoting, enacting such values in interacting with other agents, or being otherwise value-laden, then one can entertain the possibility of using algorithmic preference elicitation technologies, or structurally similar technologies, to produce data-driven insights into the values promoted by such objects. With this in mind, we investigate the application of a model fitting method known as *inverse optimization* that has structural similarities to preference elicitation tools, towards two scenarios of analyzing sociotechnical agents.

Chapter 2 of this thesis is concerned with one application of this approach. Particularly, the sociotechnical agent under scrutiny is legislative districtings in U.S. states. Legislative districts determine how a state population is partitioned into groups that each elect one representative, both in state legislatures and in the federal House of Representatives. Every ten years, following the U.S. census, these districts are redrawn such that shifting populations do not imbalance the voting power of different citizens within a state. However,

in the U.S. public political discourse, these districtings are frequently accused of being *gerrymandered*, that is, being designed with the intent of unevenly distributing voting power and the resulting legislative representation across social groups. Partisan gerrymandering is the term given to gerrymandering wherein one political party is given an advantage over another. In high-profile court cases surrounding allegations of gerrymandering, courts have indicated a desire for the production of quantitative methods for identifying partisan gerrymandering.

Predominant methods for quantitatively measuring and identifying gerrymandering have drawn on the production of metrics that are broadly designed to track democratic sociopolitical values, such as the metrics of *partisan bias* (Grofman and King 2007), *efficiency gap* (Stephanopoulos and McGhee 2015), or *election competitiveness* (Nagle 2017). There has been debate between legal scholars concerning which metrics and their underlying democratic values are the most valuable in defining gerrymandering, or which are centered around rights that could be deemed fictitious. Legal scholars that have introduced such metrics have often advocated for the use of cutoff values for delineating the legality of a districting, without nuanced consideration for the possibility that the underlying political geography of a state could inhibit such districting metrics from reaching desirable values for any possible districting. More recent quantitative methods have used statistical simulation methods to evaluate how districtings compare to a statistical distribution of average case possible districtings, which can overcome this criticism (Duchin 2018). Although the necessary supposition that districtings derived from Markov Chain Monte Carlo random walk methods represent a valuable average case of what is possible is not necessarily obviously justified. Furthermore, when there are multiple metrics in consideration, it is not necessarily possible that they can be simultaneously optimized, so analyses must also consider how a districting measures up when multiple metrics are simultaneously held to be important for democratic districtings.

Since it is critical to understand how an underlying political geography restricts what is possible to achieve in metrics, particularly when there are multiple metrics that are considered important and are simultaneously competing with each other, our method of analysis uses methods structurally similar to preference elicitation technologies to understand how a choice of legislative districtings enacts a prioritization of some such sociopolitical values *over* others, with reference to what combinations of metrics can be feasibly achieved over a given political geography. As such, even if metrics that are motivated to track some notion of partisan unfairness are deemed fictitious or not a compelling legal argument in themselves, an inverse optimization analysis may show that a districting prioritizes unfairness specifically to the detriment of other constitutionally mandated objectives, such as

creating compact districts or reducing population imbalance. We argue that this approach can produce a more nuanced data-driven argument that a proposed districting should be considered gerrymandered.

As such we utilize generalized inverse optimization to examine the design of congressional districtings, to see how they enact prioritizations of some sociopolitical values over others. (Generalized) inverse optimization is process that in which model parameters for an optimization model are inferred such that a known outcome is the optimal solution to said, model, or as close to optimal as possible. In this case, we infer parameters for how an objective for districting decisions can be constituted as a weighted combination of relevant metrics, namely population imbalance, districti compactness, and efficiency gap, to examine how these metrics and their underlying sociopolitical values are prioritized by a districting decision.

This analysis can be used to add new dimensions to discourses surrounding partisan gerrymandering in the US, and gives a tool for assessing how gerrymandering could be considered illegal; not because partisan representation is inequal to population proportion (which may not be a right that political parties as such hold), but rather because the design of districts enacts hierarchies of political values that prioritize uneven partisan representation specifically to the detriment of other rights of the people of the electorate themselves, such as the right to compact districts, which is specifically discussed in the U.S. Constitution.

Inverse optimization is computationally intensive when the space of feasible decision for the input is modeled as a mixed integer linear program (Bulut and Ralphs (2021)), which is how we quantitatively model political districting decisions and their resulting metric measurements. As such, we develop new algorithmic methods for solving generalized mixed integer inverse optimization problems, as well as further heuristic methods for achieving faster approximate solutions inspired by ensemble methods in machine learning, that are specific to the case of graph-partitioning based decisions such as political districting.

We apply these methods to the real world setting of the current congressional districts of the state of Iowa. Iowa's districts are drawn by a non-partisan independent committee, whose decisions are then approved by the state legislature. In 2021, the state legislature rejected the first proposed districts and then approved the second proposed set, citing the improved reduction in population imbalance. However, the approved districtings also yielded one more Republican seat in the ensuing election compared to the rejected plan, raising possible suspicions of an attempt to gerrymander. We apply inverse optimization analyses to both proposed districtings to evaluate how both plans prioritize population imbalance, compactness, and efficiency gap (a measure of imbalanced partisan representa-

tion). Examining the inferred multi-objective weights for both districting plans, we find that the accepted districting does in fact increase the weight on minimizing population imbalance, and does not increase weight on maximizing efficiency gap in Republicans' favor.

In Chapter 3, we investigate another application of generalized inverse optimization for evaluating fair prioritizations in decision-making. Namely, we apply inverse optimization to detect the if machine learning models are trained on datasets that under-represent a vulnerable demographic group, without being able to directly observe the dataset itself. Imbalanced datasets are a known issue that can lead to biased decision-making in machine learning contexts. Many machine learning models are trained by minimizing some measure of prediction error on their training set, and when one demographic group is underrepresented in said dataset, it is possible that the corresponding trained model may not sufficiently prioritize predicting accurately for observations in said group. As such, in many modeling scenarios it is common to adjust training weight coefficients for observations that are rare in the dataset, but important to predict accurately.

One obstacle the public faces in addressing issues of imbalanced datasets in machine learning is the privacy of widely used proprietary models and their training data. Understanding the weights of classes of training observations as an adjustable parameter in model training, for machine learning models that are trained by an optimization process minimizing prediction error on a training set, we can apply inverse optimization using a known dataset to infer what relative weightings of observations classes a non-transparent model comes closest to optimizing. This can indicate to us the relative representation of certain classes in a hidden training dataset compared to a known dataset, which could be caused by either adjusted training weightings, or, more likely, biased processes in the collection of data to under/over-represent certain groups.

In our approach, we develop subgradient-based methods for solving a generalized inverse optimization problem for three different machine learning models; ridge regression (as well as unregularized linear regression), soft margin support vector machines, and optimal decision trees. We also formulate a constrained generalized inverse optimization method for this setting wherein observation training weights are explained by a linear function of some constituent attributes of each observation (which may or may not be features of the machine learning model), allowing for an explanatory interpretation of how relative representation in a hidden training set is associated with a given attribute in the form of regression coefficients. We show how a permutation test can be used to evaluate the statistical significance in hypothesis tests of these coefficients, in order to evaluate claims of bias in training set representation to a desired significance level.

This work presents a novel approach to detecting unfairness in opaque machine learning models, seeking to detect root causes of unfairness rather than symptoms. On a set of dataset with varying amounts of simulated undersampling for a minority class, our methods out-compete baseline methods that compare prediction errors across groups, when classifying opaque models based on whether they are trained on datasets that significantly undersample the minority group. We additionally find that many models types exhibit a strong and significant correlation (up to a Pearson ρ of 0.68) between the inferred degree of imbalance and the ground truth degree of imbalance. Furthermore, pairwise comparisons show that in many cases when comparing models trained on different datasets, the model with more extreme undersampling will exhibit a more extreme inferred bias measurement, indicating that inverse optimization analysis can be used as a tool to compare and select between two models with unknown amounts of training set imbalance.

Keeping with the mindset of understanding non-human sociotehnical actors as value-laden, Chapter 4, written in the form of a paper with a target journal such as *Big Data and Society*, is motivated by the notion that this disposition towards value-analysis of technological agents can be reflexively applied to preference elicitation technologies themselves.

Chapter 4 focuses on a comparative analysis of three predominant algorithmic implementations of preference elicitation within recommender systems; *content-based recommendation*, *collaborative filtering*, and more recently developed *latent-factor methods*. Particularly, this analysis draws on the methodological frameworks of two scholars in feminist science and technology studies; Karen Barad's *agential realism* (Barad (2007)), and Susan Leigh Star's *sociotechnical infrastructuring* Star and Ruhleder (1996), to understand how the deployment of these methods across different social fields delineate variable conceptions and attributions of agency and individuality in order to produce their respective knowledge claims of preference within subjects. Under Barad's framework, technologies of observation necessarily produce an 'agential cut' between the observed and the agencies of observation, that are not always obvious choices and can have implications for what resulting concepts and narratives we are able to use to understand the world around us. Examining the multiple algorithmic methods used for preference elicitation, we see that they enact different agential cuts with respect to consumers/preference holders, preferred items, and the algorithm/platform itself. The implications within this variety of possible cuts range from how elements of taste/preference are seen as inherent attributes of people, objects, both, or neither, how/if we understand groups of people to hold coherent understandings of the same object they engage with, and if a person is seen as a coherent agent across different social contexts.

We also observe that the comparison of the first two methods is coextensive with

the progression of power differentials within the social fields examined, that are also seen in Donna Haraway's theorized shift from 'Organics of Domination' to 'Informatics of Domination' (Haraway (2016)), which was proposed around the same time as the advent of collaborative filtering methods, and thus propose a new category, 'Latencies of Domination' that corresponds to the more recent latent factor methods. As the deployment of these various approaches to preference elicitation varies across fields, we pose that understanding the specific choice of methods deployed can be an important tool for understanding the underlying power relations that prefigure subjects as either individuals or members of masses, and that observation of this phenomenon in newly developing fields can open up critical opportunities for intervention in newly forming power structures.

One such field that we might consider to be broadening its conceptions and operationalizations of preference is in healthcare. While patient-centered care paradigms have encouraged the inviting the patient into the process of defining and participating in their own vision of health, we consider the possible role of preference elicitation for the caregiver, and how it can be used to achieve more successful patient care as well as lower burden on caregivers. We specifically focus on the domain of family caregiving for patients with Alzheimer's Disease and Related Dementias (ADRD), where family caregivers are unlikely to have formal training, and may have a broad range of expertise and comfortability across different areas of care.

Chapter 5 of this thesis concerns ongoing work in partnership with CareVirtue, a healthcare startup that provides an online platform for caregivers for people with ADRD. The platform is oriented towards assisting a network of caregivers surrounding one patient with coordinating their care, as well as with assisting caregivers in learning about and managing care tasks for which they might be under-prepared, such as managing finances for care costs. One feature under development in this platform is the production of a recommender system that will help caregivers identify areas where they may need additional support, and recommend additional resources and trainings in these areas.

Keeping mindful of the concerns from Chapter 4, we develop a recommender system that does not infer any taxonomic structure of areas of care, but rather utilizes survey data from the lived experiences across a population of caregivers to infer how confidence in some areas may impact levels of confidence in other areas. Applying supervised machine learning methods, we ask every caregiver onboarding the platform their comfort level with a subset of care areas, and use this data, along with demographic features, to predict the comfort levels with all other care areas, based on relationships inferred from the survey data.

This machine learning approach avoids importing any fixed hierarchies in the organiza-

tion of caregiving work that could result from content-based recommendation methods. Additionally, the cold-start problems that individual users may experience with collaborative filtering-based recommendation methods are also avoided due to the existence of dense survey data training the preference elicitation model. This is particularly helpful in this domain setting since the moment of onboarding is when caregivers will have the least overall experience in caregiving, and can benefit the most from recommendations on trainings and resources.

We evaluate the success of our recommender methods by using test sets from the initial survey data, as well as through implementation on a trial of the platform with new caregivers. We compare test set evaluations against baseline methods of recommending the most popularly unfamiliar topics, and a content-based recommender that assumes that familiarity with a sub-topic is equal to familiarity of another subtopic in the same topic according to a naive organization of care areas. We find that our machine learning based recommender outperforms compared to the baseline methods when predicting confidence levels of care areas, testing on both out-of-sample survey data, and observations of resource ratings that correspond to the care topics in a trial implementation of the recommender system. We also find in the trial implementation that recommended resources in the top tier of recommendations have a significantly larger click-through rate than resources supplied in a general bank, indicating the efficacy of providing recommendations for caregivers.

1.2 Additional Research Not Covered in this Thesis

In addition to the prospective thesis material included in the following chapters, I have also completed research projects in non-thematically-related areas that I feel are worth mentioning here.

Through the Wisconsin Partnership Program, I have completed a project in conjunction with UW-Health and the UW-Madison SMPH Emergency Medicine Department concerning the use of machine learning methods to forecast the likelihood of surges in crowding in the Emergency Department (ED) in the near future, using data collected at patient-level, health system-level, and community-level. The project evaluated the efficacy of predicting patient volume over 12-hour periods at three different points in patient flow through the ED as indicators of crowding. Additionally, the study also investigated the impact of data drift across two changes. First, we approached spatial data drift as we investigated the efficacy of models in two Wisconsin hospitals — one urban hospital and one rural hospital — and the impact of transferring models trained at one location to be tested in another. Second, we investigate temporal data drift across the 'sudden drift event' of the onset of the COVID-19

pandemic, by comparing model performance in datasets from both before and during the COVID-19 pandemic, as well as the value of predicting COVID-era outcomes using models trained on pre-COVID observations when there are few COVID-era observations to train on. Our study concludes: that ED Boarding is a more predictable metric than ED arrivals or admissions through the ED, which indicates crowding in the ED; that spatial transfer of models should be done with extreme caution if at all; and that in the case of the sudden drift event of COVID-19, using pre-COVID models is valuable up to a point where enough COVID-era data was collected to out-perform pre-COVID models.

This research has been published in the Journal of the American Medical Informatics Association, under the title "Multisite Evaluation of Prediction Models for Emergency Department Crowding Before and During the COVID-19 Pandemic". A copy of this paper has been included in Appendix D, rather than in the main body of the thesis.

1.3 Acknowledgements

Thanks to:

1. My family and my partner Carson
2. My advisor Justin, my labmates, and my thesis committee
3. My dearest friends with whom I've bounced off and developed ideas (Luke L., Josh M., Char V.D.B., Kristina B., Seare F.)
4. The creative/experimental music and arts community of Madison, WI, that has shown me so much love and always listened to what I've needed to express (Tim R. and Liz S., Devin D., Jakob H., Matt B., Hanah Jon T., Maggie C., Sahada B., Andy J., Matty A., Frank M., Greg R., Helen F., Anthony U.)
5. The faculty and affiliates of the Holtz Center who have introduced me to an supported my interest in pivotal ideas for my work (Sai S., Keith W., Xerxes M., Bennett M., etc.)

Chapter 2

Generalized Inverse Optimization as a Technology of Observing Congressional Gerrymandering

2.1 Introduction

Inverse optimization (IO) has received significant attention in recent years; see Chan et al. (2023) for a thorough review. At a high level, inverse optimization infers unknown parameters (e.g., cost vector, constraint matrix, etc.) of a *forward optimization problem* (FOP) such that an observed solution is rendered optimal (or nearly optimal) for the FOP. Inverse optimization has been applied to a wide range of problems including radiation therapy (Chan et al. 2014), transportation (Patriksson 2015), and cellular biology (Zhao et al. 2015).

The IO literature can be partitioned into two general approaches, depending on the problem setting. The first approach ("classical IO") is to assume that the observed solution is optimal for the FOP and then leverage results from duality theory to design an appropriate IO problem (Ahuja and Orlin 2001, Schaefer 2009). The second approach ("generalized IO") is to make no assumptions about the optimality of the observed solution (i.e., it may not be optimal or even feasible) and then use a loss function to design an appropriate IO problem (Chan et al. 2014).

To date, most of the IO literature has focused on convex FOPs with relatively little research on purely integer or mixed-integer FOPs. In the context of integer IO problems, there are two streams of literature. First, Schaefer (2009) and Lamperski and Schaefer (2015) characterize the polyhedral representation of purely integer and mixed-integer IO problems, respectively. Second, Wang (2009) propose a cutting plane approach for

efficiently solving mixed-integer IO problems. Recent research has extended this cutting plane approach to be more efficient using parallel computing (Wang 2013) or trust regions (Bodur et al. 2022), and to solve generalized mixed-integer IO problems (Moghaddass and Terekhov 2020).

In this paper, we propose a new approach for solving generalized mixed-integer IO problems based on sub-gradient methods. Our method leverages the observation that the generalized IO problem is equivalent to minimizing a loss function over a bounded domain and many commonly used loss functions permit sub-gradient calculation. We present the conditions for when a generalized IO problem can be solved using sub-gradient methods and we test various implementations, including gradient descent and the Frank-Wolfe method.

2.1.1 Motivating example: political gerrymandering

In the United States of America, representation in federal and state legislatures is apportioned by democratic elections in geographic sub-regions known as legislative districts. Every 10 years, a nationwide census is conducted and the data gathered is used to redraw new legislative districts. This redrawing process, commonly referred to as *political districting*, is regulated by federal and state laws. The United States Constitution requires that political districts must be of equal population, and state constitutions enforce additional requirements upon district design. For example, the state of Iowa requires that congressional districts do not split up counties (Iowa Constitution Article III §37) and that districts must be as compact as practicable (Iowa State Code, Chapter 42.4.4).

Political districting is often politically contentious, as there are frequent allegations – in high-profile court cases and in the broader public political sphere – that districts are drawn to create imbalanced voting power across populations (Lieb 2022, Villenueve 2022, Sherman 2019). When such imbalance is used to privilege the representation of one political party over another, it is referred to as *partisan gerrymandering*. Allegations of partisan gerrymandering often result in trials, where judges are tasked with deciding if a particular political districting is illegal, and as such must be redrawn. In these cases, the judicial system has indicated a desire for quantitative methods of measuring partisan gerrymandering, allowing for judicial rulings to be made with more confidence (and based on defensible data) (Vieth v. Jubelirer 2004). However, quantitatively identifying gerrymandering is challenging because public and judicial discourse around gerrymandering typically invokes common sense notions — particularly of *fairness* and *compactness* — which are often not quantified or critically elaborated.

The foundation of quantitative methods for identifying political gerrymandering relies on the development of district-level metrics that quantify and align with sociopolitical values; either those legally required (e.g., compactness in many states) or those in the public discourse (e.g., "fair" representation). Two constitutional requirements – contiguity and equal population – are easily quantified. However, a frequent third requirement – compactness – has been subject to much debate. For example, Young (1988) demonstrated that the common sense notions of compactness should not be expected to match a single quantitative metric, because nearly (if not) all compactness metrics can be met with counterexamples that are quantitatively compact but do not feel compact to qualitative perceptions, or vice versa. Similarly, "fair" partisan representation has been quantified with a multitude of metrics, under such names as partisan bias (Grofman and King 2007), competitiveness of elections (Nagle 2017), and the efficiency gap of wasted votes (Stephanopoulos and McGhee 2015).

Initial research on identifying political gerrymandering invoked hard cutoffs for a single (chosen) metric as an indicator of gerrymandering (Stephanopoulos and McGhee 2015). However, a univariate approach fails to consider the contingency of what values are actually attainable for real political geographies of state populations, or that a multitude of competing metrics may inhibit each other from simultaneously meeting some benchmark, even if they all align with different democratic values that are important to courts. To address this shortcoming, researchers have used computational methods to develop Markov Chain Monte Carlo approaches that create a set of "reasonable districtings" and corresponding probability distributions for each districting metric. A proposed political districting is then compared against this information to determine whether or not it is a statistical outlier; if so, this may indicate that a districting prioritizes an undemocratic value and should be considered gerrymandering (Duchin et al. 2019, Daryl DeFord and Solomon 2020).

In this paper, we propose an inverse optimization approach to quantitatively identify partisan gerrymandering. By formulating the process of drawing political districtings as a multi-objective optimization problem, inverse optimization allows us to consider how a proposed districting enacts a prioritization of some values over others with respect to what is possible in a given political geography. Thus, even if metrics that are motivated to track some notion of partisan unfairness are deemed fictitious or not a compelling legal argument in themselves, an inverse optimization analysis may show that a districting prioritizes unfairness specifically to the detriment of other constitutionally mandated objectives, such as creating compact districts or reducing population imbalance. We argue that this approach can produce a more nuanced data-driven argument that a proposed districting should be considered gerrymandered.

2.1.2 Contributions

We summarize our contributions as follows:

1. We propose a new approach – *gap-gradient methods* – for generalized inverse integer optimization that leverages the problem's similarity to minimizing a bounded convex function where querying the function gradient is computationally difficult. We characterize when a generalized IO problem can be solved using sub-gradient methods and we prove that modifications to classic sub-gradient algorithms can return exact solutions in finite time. We evaluate our methods using a set of instances from the MIPLIB 2017 library and we find that our best implementation is able to improve solution time by up to 90%, compared to the best performing method from the literature.

2. We develop a custom heuristic method for graph-based inverse problems using a combination of coarsening methods from the field of graph partitioning and ensemble methods from the field of machine learning. Our methods use ensembles of smaller problem instances to produce approximate solutions that provably converge towards the optimal inverse solution as ensemble size increases, while saving computational expense and allowing for tractable run times for large real-world problems. Our heuristic is able to reduce the median solution time by 52%, while still producing near-optimal solutions.

3. We propose a new application domain – quantitatively identifying gerrymandering – for generalized inverse integer optimization. We apply our overall solution approach to analyze the congressional districts of the State of Iowa using real-world data. We show that accepted and previously rejected district plans both greatly prioritize minimizing population imbalance over district compactness or partisan efficiency gap. However, the increased priority on population imbalance in the accepted district plan results in a 16% decrease in population imbalance while creating a 372% increase in partisan efficiency gap compared to the rejected plan.

Many of the graph-based IO methods formulated and implemented in this paper (Contribution 2) can also be used in several other applications, such as flexible job shop scheduling, employee scheduling, and facility location, particularly in settings where there are competing objectives to be considered (e.g., distribution of polling locations).

2.2 Literature Review

Our work contributes to three primary streams of literature: inverse optimization (Section 2.2.1), optimization approaches for political districting (Section 2.2.2), and quantitative methods for identifying gerrymandering (Section 2.2.3).

2.2.1 Inverse Optimization

The idea of IO was proposed by Ahuja and Orlin (2001) who studied linear FOPs in a classical IO setting. Since then, there has been much research on IO, including for conic FOPs (Iyengar and Kang 2005), general convex FOPs (Zhang and Xu 2010), purely integer FOPs (Schaefer 2009), and mixed-integer FOPs (Wang 2009, Bulut and Ralphs 2021). See Chan et al. (2023) for a recent review.

In recent years, there have been several extensions to IO, often inspired by machine learning and data-driven optimization. Importantly, Chan et al. (2014) developed the generalized IO paradigm, where no assumptions are made about the optimality or feasibility of the observed solutions. Other extensions include IO with multiple observations (Keshavarz et al. 2011, Babier et al. 2021), IO with uncertain data (Ghobadi et al. 2018, Aswani et al. 2018), and goodness of fit measures for IO (Chan et al. 2019).

Our work is most closely related to the IO literature where the FOP is a mixed-integer linear program. In this vein, Wang (2009) proposed the first algorithmic approach (a cutting plane algorithm) for solving mixed integer IO problems. Since then, there have been several extensions that seek to improve tractability (Bodur et al. 2022, Moghaddass and Terekhov 2020, Wang 2013). Most similar to our paper is the work of Moghaddass and Terekhov (2020) who develop the first approach for generalized inverse mixed integer optimization by leveraging Wang (2009)'s cutting plane method, while assuming a loss function where the optimality gap is formulated as an absolute gap. Recently (and in parallel to this work), Scroccaro et al. (2023) develop descent methods for learning solutions to mixed integer IO problems. However, they use a more general augmented suboptimality loss function that does not necessarily permit provable finite time convergence to the optimal solution.

We contribute to this literature in three key ways: (i) we propose a new solution method for solving generalized inverse multi-objective mixed integer optimization in finite time that uses first order methods of minimizing a bounded convex function, (ii) we propose a set of heuristic approaches inspired by ensemble methods in machine learning that are tailored to the setting where the FOP is graph-based (i.e., amenable to graph coarsening),

and (iii) we propose a new and potentially impactful application domain for generalized inverse integer optimization.

2.2.2 Optimization approaches to political districting

Political districting has a long history as an optimization problem in the operations research community. See Ricca et al. (2013) or Swamy et al. (2022) for a recent literature review. The literature in this area typically uses mixed-integer programming to design political districtings (Hess et al. 1965). Most approaches focus on nonpolitical metrics such as population imbalance, contiguity, and compactness (Garfinkel and Nemhauser 1970). Since these problems are very computationally challenging, research has focused on developing efficient formulations and both exact and heuristic solution methods (Mehrotra et al. 1998, Validi et al. 2022).

More recently, researchers have developed districting models that include or optimize for notions of political fairness such as efficiency gap, partisan symmetry, and competitiveness (Swamy et al. 2022, King et al. 2015). For example, Swamy et al. (2022) produce a formulation that considers districting design as a multi-objective optimization problem between district compactness and variety of fairness-related metrics, and uses the formulation to explore the Pareto frontier of tradeoffs between pairs of metrics over a state's political geography. While we do not directly contribute another districting formulation, we make minor modifications to Swamy et al. (2022)'s formulation and their multi-objective approach, and use it as the FOP for our generalized inverse optimization model, demonstrating another use of their formulation.

2.2.3 Quantitative Methods for Identifying Gerrymandering

Initial research on identifying political gerrymandering invoked hard cutoffs for a single (chosen) metric as an indicator of gerrymandering; see Stephanopoulos and McGhee (2018) for a review. In recent years, research has focused on how metrics can quantitatively identify gerrymandering using approaches that are not cutoff rules. For example, Duchin (2018) propose the use of Markov Chain Monte Carlo methods to produce a set of reasonable districtings. This set of districtings is then used to create statistical distributions of particular partisan fairness (or compactness) metrics. If the districting under scrutiny presents a partisan fairness metric that is a strong statistical outlier compared to the distribution, then an argument can be made that the districting explicitly prioritizes disproportionate representation (i.e., partison "unfairness"). Note that this line of argument requires the assumption that the constructed set of districtings accurately represents average or per-

missible districtings. These methods have been applied to the study of various underlying metrics for district design, including partisan seat share (Duchin et al. 2019), partisan symmetry (Daryl DeFord and Solomon 2020), and proportion of majority-minority districts (Becker et al. 2021).

In contrast to previous approaches, we contribute a new method for quantitatively measuring how district design impacts partisan fairness. Rather than comparing a districting to a constructed distribution of permissible districtings like the Markov Chain Monte Carlo methods, our inverse optimization methods analyze how a district compares to the boundary of what is feasible, while critically recognizing that multiple metrics can be in competition with each other. For example, if a districting that meets a certain benchmark for one metric limits the best possible performance on other metrics, then producing judgements based on the evaluation of a single metric may be unhelpful when multiple metrics are valued. With inverse optimization, we can understand how a districting choice enacts a prioritization of metrics and their corresponding democratic values. For example, if one argued that high political unfairness is not sufficient to disqualify a districting, then one may show that the districting under scrutiny prioritizes political unfairness explicitly to the detriment of another legal right, such as district compactness or population balance. We beleive our approach may provide a more robust argument that a districting should be disqualified.

2.3 Models

In this section, we introduce the general structure of the forward multi-objective mixed-integer linear optimization problem (Section 2.3.1) and the corresponding inverse optimization problem (Section 2.3.2)

2.3.1 The Forward Problem

To apply inverse optimization, we must first define the structure of the forward optimization problem (FOP). Inverse optimization implicitly assumes that the observed solutions were generated by a decision making process similar to the FOP. In the context of political districting, the decision making process is typically represented by a multi-objective mixed-integer linear optimization problem (Swamy et al. 2022).

We first introduce the general problem structure and provide the specific FOP for political districting in Section 2.6. Let $\mathbf{y} \in \mathbb{R}^n \times \mathbb{Z}^{n-q}$, $q = 0, 1, 2, ..., n$ represent the decision variables for the FOP, and let $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\mathbf{b} \in \mathbb{R}^m$ represent the constraints. We let $\mathcal{K}$

denote the set of objective functions. Then, the general multi-objective mixed-integer linear optimization problem can be written as

$$\begin{aligned} \underset{\mathbf{y}}{\text{minimize}} \quad & \boldsymbol{\alpha}^T \mathbf{C}\mathbf{y} && \text{(2.1a)} \\ \text{subject to} \quad & A\mathbf{y} \leq \mathbf{b}, && \text{(2.1b)} \\ & \mathbf{y} \in \mathbb{R}^n \times \mathbb{Z}^{n-q}, && \text{(2.1c)} \end{aligned}$$

where the rows of $\mathbf{C} \in \mathbb{R}^{|\mathcal{K}| \times n}$ represent different linear objective functions and $\boldsymbol{\alpha} \in \mathbb{R}^{|\mathcal{K}|}$ denotes the cost vector. Let $\mathcal{S} = \{A\mathbf{y} \leq \mathbf{b}, \mathbf{y} \in \mathbb{R}^n \times \mathbb{Z}^{n-q}\}$ denote the feasible region of the FOP, let $\mathcal{B}(\mathcal{S}) = \{\mathbf{C}\mathbf{y} |\ \mathbf{y} \in \mathcal{S}\}$ denote the set of feasible sub-objective values (also known as the FOP objective feasible space), and let $\mathcal{F}(\boldsymbol{\alpha}, \mathcal{S})$ represent the set of optimal solutions. Without loss of generality, we assume that $\mathcal{S}$ is non-empty (Chan et al. 2014).

2.3.2 Inverse Optimization Model

In its classical form, inverse optimization seeks to, given a feasible region for a forward problem ($\mathcal{S}$) and an observed solution ($\hat{\mathbf{y}}$), return an objective function to the forward problem ($\bar{\boldsymbol{\alpha}}$) for which the given solution is optimal – a property called *inverse feasibility* (Chan et al. 2023). Given a feasible solution to the forward problem, $\hat{\mathbf{y}} \in \mathcal{S}$, we let $\mathcal{C}(\hat{\mathbf{y}}, \mathcal{S}) = \{\boldsymbol{\alpha} \in \mathbb{R}^{|\mathcal{K}|} \mid \hat{\mathbf{y}} \in \mathcal{F}(\boldsymbol{\alpha}, \mathcal{S})\}$ denote the inverse-feasible region, which includes all cost vectors $\boldsymbol{\alpha}$ that render $\hat{\mathbf{y}}$ optimal.

In data-driven contexts, it may not be reasonable to assume that $\hat{\mathbf{y}}$ is optimal for any possible objective (i.e., $\mathcal{C}(\hat{\mathbf{y}}, \mathcal{S}) = \varnothing$) or even feasible for the FOP (i.e., $\hat{\mathbf{y}} \notin \mathcal{S}$), so instead we seek to minimize some loss function of the forward objective that penalizes the extent to which inverse-feasibility is not satisfied. We denote the loss function $\ell(\hat{\mathbf{y}}, \mathcal{S}, \boldsymbol{\alpha})$, and note that if a solution $\hat{\mathbf{y}}$ does have some classically inverse feasible objective, then for any $\boldsymbol{\alpha} \in \mathcal{C}(\hat{\mathbf{y}}, \mathcal{S})$, we have $\ell(\hat{\mathbf{y}}, \mathcal{S}, \boldsymbol{\alpha}) = 0$. Chan et al. (2023) enumerate five possible loss functions that are useful in the data-driven inverse optimization context.

For any such loss function, the corresponding inverse optimization problem seeks to minimize the loss function, with constraints placed on the allowable objectives $\boldsymbol{\alpha}$, i.e., $\boldsymbol{\alpha} \in \mathcal{A}$. We can write this problem as:

$$\begin{aligned} \underset{\boldsymbol{\alpha}}{\text{minimize}} \quad & \ell(\hat{\mathbf{y}}, \mathcal{S}, \boldsymbol{\alpha}) && \text{(GIO}(\hat{\mathbf{y}}, \mathcal{S})\text{)} \\ \text{subject to} \quad & \boldsymbol{\alpha} \in \mathcal{A}. \end{aligned}$$

Many choices of loss functions have inverse optimization formulations that can be

more practically expressed. For example, for the absolute sub-optimality loss function $\ell_{\text{ABS}}(\hat{\mathbf{y}}, \mathcal{S}, \boldsymbol{\alpha}) = \boldsymbol{\alpha}^\intercal \mathbf{C}\hat{\mathbf{y}} - \min_{\mathbf{y}\in\mathcal{S}} \boldsymbol{\alpha}^\intercal \mathbf{C}\mathbf{y}$, which measures the difference in FOP objective value between the input $\hat{\mathbf{y}}$ and an optimal FOP solution for the given value of $\boldsymbol{\alpha}$, the equivalent inverse optimization formulation can be written as:

$$\begin{aligned} \underset{\boldsymbol{\alpha},\, \xi_{\text{ABS}}}{\text{minimize}} \quad & \xi_{\text{ABS}} && (\text{GIO}_{\text{ABS}}(\hat{\mathbf{y}}, \mathcal{S})) \\ \text{subject to} \quad & \boldsymbol{\alpha}^\intercal \mathbf{C}\hat{\mathbf{y}} \leq \boldsymbol{\alpha}^\intercal \mathbf{C}\mathbf{y} + \xi_{\text{ABS}}, \quad \forall\, \mathbf{y} \in \mathcal{S}, \\ & \boldsymbol{\alpha} \in \mathcal{A}. \end{aligned}$$

One potentially useful choice of $\mathcal{A}$ for many applications is the unit simplex in dimension $|\mathcal{K}|$, because then $\boldsymbol{\alpha}$ may be interpreted as a convex combination of weights for multiple objectives, allocating fractional amounts of importance to each objective in a form that adds up to 1. We also note that when C is the identity matrix, the generalized multi-objective inverse optimization problem $\text{GIO}(\hat{\mathbf{y}}, \mathcal{S})$ becomes a standard generalized inverse optimization problem, and all following solution approaches remain applicable without loss of generality.

2.4 Solution Approaches

In this section, we recap existing cutting plane methods (Section 2.4.1) and introduce a new approach for solving inverse mixed integer optimization problems (Section 2.4.2).

2.4.1 Cutting Planes

The first cutting plane algorithm for inverse mixed integer linear optimization was tailored to the classical inverse case that assumes the existence of an inverse-feasible objective (Wang 2009). The algorithm alternates between a master problem and a separation problem. The master problem produces an optimal forward problem objective ($\boldsymbol{\alpha}$) that is hypothesized to be inverse feasible (i.e., $\boldsymbol{\alpha} \in \mathcal{C}(\hat{\mathbf{y}}, \mathcal{S})$) under partial knowledge of the forward feasible region ($\mathcal{S}$). The separation problem solves the forward optimization problem with the hypothesized objective ($\boldsymbol{\alpha}$) to produce an extreme point that can be added to the set $\mathcal{S}$. With each new extreme point that is added to $\mathcal{S}$, the set of potentially inverse-feasible objectives $\mathcal{C}(\hat{\mathbf{y}}, \mathcal{S})$ decreases in size. When the forward problem returns an extreme point that is already known to be in $\mathcal{S}$, further iterations will not remove elements from $\mathcal{C}(\hat{\mathbf{y}}, \mathcal{S})$, and so the output of the master problem is known to be an inverse-feasible solution. Thus, the algorithm returns the solution to the master problem and terminates. Since the forward

problem is a mixed integer linear program, it is guaranteed to have a finite number of extreme points, and so the algorithm is guaranteed to (eventually) terminate.

More recently, Moghaddass and Terekhov (2020) proposed an extension to Wang's algorithm that is suitable for data-driven generalized inverse optimization, in the case of an absolute sub-optimality loss function. As such, they use $\text{GIO}_{\text{ABS}}(\hat{\mathbf{y}}, \mathcal{S})$ as their master problem, and alternate between the master problem and the separation problem until an objective is found that minimizes the loss function. We present their algorithm structure (Algorithm 3) using our established notation in A.7. In A.2, we contribute a minor modification to the algorithm proposed by Moghaddass and Terekhov (2020) that utilizes the relative sub-optimality loss function, rather than absolute sub-optimality.

2.4.2 Gap-Gradient Methods

In this section, we describe our proposed solution methods – *gap-gradient methods* – and highlight various extensions. Our methods are applicable to a wide range of loss functions and constraint sets for the cost vector.

We first define what we call the *gap function* for a given generalized inverse optimization problem. The gap function is simply the loss function values over the feasible region of $\boldsymbol{\alpha}$, which we formally defined as follows.

Definition 2.1 (Gap Function)**.** *For a generalized inverse optimization problem with fixed $\hat{\mathbf{y}}$ and $\mathcal{S}$, and some loss function $\ell(\hat{\mathbf{y}}, \mathcal{S}, \boldsymbol{\alpha})$, the gap function is defined as $\xi(\boldsymbol{\alpha}) = \ell(\hat{\mathbf{y}}, \mathcal{S}, \boldsymbol{\alpha}), \forall \boldsymbol{\alpha} \in \mathcal{A}$.*

Note that the gap function will be a convex function as long as the loss function is a convex function of $\boldsymbol{\alpha}$, given fixed $\hat{\mathbf{y}}$ and $\mathcal{S}$. For the example where the loss function is absolute sub-optimality and $\boldsymbol{\alpha}$ is constrained to the unit simplex, the gap function is $\xi_{ABS}(\boldsymbol{\alpha}) = \boldsymbol{\alpha}^\intercal \mathbf{C}\hat{\mathbf{y}} - \min_{\mathbf{y} \in \mathcal{S}} \boldsymbol{\alpha}^\intercal \mathbf{C}\mathbf{y}$, where $\boldsymbol{\alpha} \in \{||\boldsymbol{\alpha}||_1 = 1, \boldsymbol{\alpha} \geq 0\}$. To keep the rest of the paper succinct, all discussion henceforth concerns the case in which the loss function ℓ is absolute sub-optimality and the gap function domain $\mathcal{A}$ is the unit simplex.

Proposition 2.2. *The absolute gap function $\xi_{ABS}(\boldsymbol{\alpha})$ is a convex function.*

Proof. Proof of Proposition 2.2: First, note that the epigraph of $\xi_{ABS}(\boldsymbol{\alpha})$ is $\{\xi_{ABS} \in \mathbb{R} \mid \xi_{ABS} \geq \boldsymbol{\alpha}^\intercal \mathbf{C}\hat{\mathbf{y}} - \min_{\mathbf{y} \in \mathcal{S}} \boldsymbol{\alpha}^\intercal \mathbf{C}\mathbf{y}\}$, which is equivalent to $\{\xi_{ABS} \in \mathbb{R} \mid \xi_{ABS}(\boldsymbol{\alpha}) \geq \boldsymbol{\alpha}^\intercal \mathbf{C}\hat{\mathbf{y}} - \boldsymbol{\alpha}^\intercal \mathbf{C}\mathbf{y}, \forall \mathbf{y} \in \mathcal{S}\}$. This space is the intersection of half-planes, and as such, is convex. Therefore, the gap function is a convex function because its epigraph is convex. □

Figure 2.1 depicts an example inverse optimization problem with two sub-objectives, c_1 and c_2. Figure 2.1(a) displays the convex hull of the set of feasible subobjective values

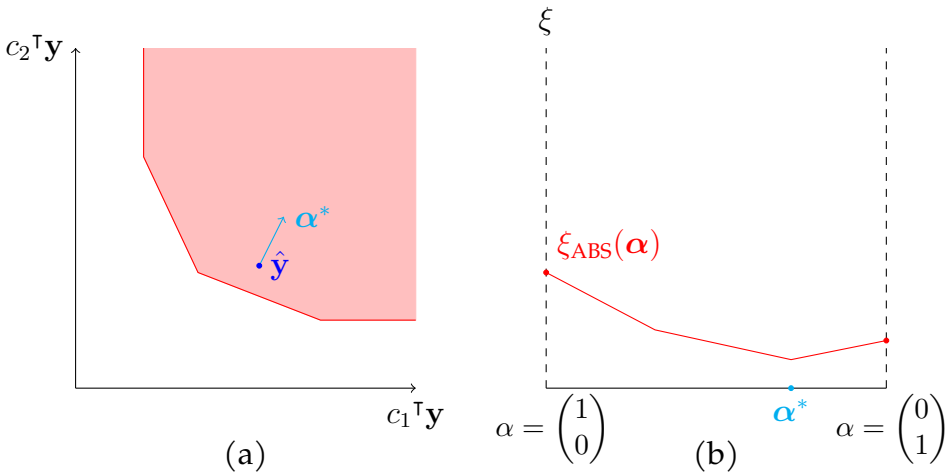


Figure 2.1: An example inverse optimization problem with two sub-objectives. (a) The FOP objective feasible space ($conv(\mathcal{B})$), inverse input ($\hat{\mathbf{y}}$), and the inverse solution ($\boldsymbol{\alpha}^*$), and (b) the corresponding absolute gap function (ξ_{ABS}) and its minimizer ($\boldsymbol{\alpha}^*$).

for the forward problem ($conv(\mathcal{B})$), the inverse input ($\hat{\mathbf{y}}$), and the FOP multi-objective weighting ($\boldsymbol{\alpha}^*$) that minimizes the absolute sub-optimality loss function ℓ_{ABS} given $\hat{\mathbf{y}}$ and $\mathcal{S}$ (i.e., the optimal inverse solution). Figure 2.1(b) displays the corresponding absolute gap function ($\xi_{ABS}(\boldsymbol{\alpha})$). The horizontal axis of Figure 2.1(b) describes the set of allowable multi-objective weightings for the two sub-objectives as a convex combination ranging from $\begin{pmatrix}1\\0\end{pmatrix}$ to $\begin{pmatrix}0\\1\end{pmatrix}$. Note that the vertices of $conv(\mathcal{B})$ correspond to the piecewise linear components of ξ, and that $\boldsymbol{\alpha}^*$ (the solution to $\text{GIO}_{\text{ABS}}(\hat{\mathbf{y}}, \mathcal{S})$) is also the minimizer of $\xi_{ABS}(\boldsymbol{\alpha})$.

Now, consider the cutting plane algorithm used to solve GIO_{abs} (e.g., Moghaddass and Terekhov (2020)). At a given iteration k, we use $\boldsymbol{\alpha}^{(k)}$ to solve the FOP and obtain a solution $\mathbf{y}^{(k)}$ that is optimal for objective weighting $\boldsymbol{\alpha}^{(k)}$. Recall that $\mathcal{C}(\mathbf{y}^{(k)}, \mathcal{S})$ denotes the set of $\boldsymbol{\alpha}$ values that render $\mathbf{y}^{(k)}$ optimal for the FOP.

Proposition 2.3. *If* $\boldsymbol{\alpha}^{(k)} \in \mathcal{C}(\mathbf{y}^{(k)}, \mathcal{S})$, *then* $\xi_{ABS}(\boldsymbol{\alpha}^{(k)}) = \boldsymbol{\alpha}^{(k)\intercal}\mathbf{C}\hat{\mathbf{y}} - \boldsymbol{\alpha}^{(k)\intercal}\mathbf{C}\mathbf{y}^{(k)}$.

Proof. Proof of Proposition 2.3: By the definition of the absolute sub-optimality loss function, $\xi_{\text{ABS}}(\boldsymbol{\alpha}^{(k)}) = \max_{\mathbf{y}\in\mathcal{S}} \boldsymbol{\alpha}^{(k)\intercal}\mathbf{C}\hat{\mathbf{y}} - \boldsymbol{\alpha}^{(k)\intercal}\mathbf{C}\mathbf{y}$. Since $\mathbf{y}^{(k)} = \arg\min_{\mathbf{y}\in\mathcal{S}} \boldsymbol{\alpha}^{(k)\intercal}\mathbf{C}\mathbf{y}$ by the definition of $\mathbf{y}^{(k)}$, it must be true that $\xi_{\text{ABS}}(\boldsymbol{\alpha}^{(k)}) = \max_{\mathbf{y}\in\mathcal{S}} \boldsymbol{\alpha}^{(k)\intercal}C\hat{\mathbf{y}} - \boldsymbol{\alpha}^{(k)\intercal}C\mathbf{y}^{(k)}$. □

The implication of Proposition 2.3 is that we can efficiently query subgradients of the gap function, $\xi_{\text{ABS}}(\boldsymbol{\alpha}^{(k)})$. We demonstrate this via two cases. First, suppose that $\mathbf{y}^{(k)}$ is a vertex of conv($\mathcal{B}$), then $\mathcal{C}(\mathbf{y}^{(k)}, \mathcal{S})$ is a closed convex subset of $\mathcal{A}$ with dimension $\mathcal{K} - 1$ (by

the definition of a vertex). For any $\boldsymbol{\alpha}$ in the interior of $\mathcal{C}(\mathbf{y}^{(k)}, \mathcal{S})$, the *gradient* of $\xi_{\text{ABS}}(\boldsymbol{\alpha})$ can be calculated as $\nabla(\boldsymbol{\alpha}^\intercal C\hat{\mathbf{y}} - \boldsymbol{\alpha}^\intercal C\mathbf{y}^{(k)}) = C(\hat{\mathbf{y}} - \mathbf{y}^{(k)})$. Second, suppose that $\mathbf{y}^{(k)}$ is *not* a vertex of $\text{conv}(\mathcal{B})$. Then we note that $\xi_{\text{ABS}}(\boldsymbol{\alpha}^{(k)}) = \boldsymbol{\alpha}^{(k)\intercal} C\hat{\mathbf{y}} - \boldsymbol{\alpha}^{(k)\intercal} C\mathbf{y}^{(k)}$, and $\forall \boldsymbol{\alpha} \in \mathcal{A}$, $\xi_{\text{ABS}}(\boldsymbol{\alpha}) \geq \boldsymbol{\alpha}^\intercal C\hat{\mathbf{y}} - \boldsymbol{\alpha}^\intercal C\mathbf{y}^{(k)}$, since $\max_{\mathbf{y}\in\mathcal{S}} \boldsymbol{\alpha}^\intercal C\hat{\mathbf{y}} - \boldsymbol{\alpha}^\intercal C\mathbf{y} \geq \boldsymbol{\alpha}^\intercal C\hat{\mathbf{y}} - \boldsymbol{\alpha}^\intercal C\mathbf{y}^{(k)}$. Thus, the linear expression $\boldsymbol{\alpha}^\intercal C\hat{\mathbf{y}} - \boldsymbol{\alpha}^\intercal C\mathbf{y}^{(k)}$ defines a *subtangent plane* of $\xi_{\text{ABS}}(\boldsymbol{\alpha})$, and as such the gradient of the subtangent plane, $\nabla(\boldsymbol{\alpha}^\intercal C\hat{\mathbf{y}} - \boldsymbol{\alpha}^\intercal C\mathbf{y}^{(k)}) = C(\hat{\mathbf{y}} - \mathbf{y}^{(k)})$ constitutes a *subgradient* of $\xi_{\text{ABS}}(\boldsymbol{\alpha})$.

Remark 2.4. *Suppose a generalized inverse optimization problem has a gap function $\xi(\boldsymbol{\alpha})$ that is convex, a gap function domain $\mathcal{A}$ that is convex, and that the gap function is lower bounded over its domain. If it is possible to query a subgradient of $\xi(\boldsymbol{\alpha})$ for any $\boldsymbol{\alpha} \in \mathcal{A}$, then any subgradient method is guaranteed to converge towards the gap function minimizer as the number of iterations approaches infinity. Since we can query subgradients of $\xi_{ABS}(\boldsymbol{\alpha})$ at any $\boldsymbol{\alpha}^{(k)}$, we thus have the possibility of finding the gap function minimizer with any first-order method of minimizing convex functions over bounded domains.*

Additional generalized inverse optimization scenarios that satisfy the conditions stated in Remark 2.4 include a relative sub-optimality loss function where $\mathcal{A}$ is the unit simplex (Chan et al. 2023). See A.3 for details on how our method applies to the relative sub-optimality loss function.

In general, subgradient methods do not necessarily guarantee finite-time convergence like the cutting plane method presented by Moghaddass and Terekhov (2020), which alternates solving relaxed master problems and generating cutting planes. For example, the descent-based methods developed by Scroccaro et al. (2023) are not guaranteed to converge upon loss function minimizers in finite time. However, in the case of a linear piece-wise gap function, each iteration of a subgradient method is capable of producing a linear cutting plane that can be added to the master problem and, when a sufficient set of cutting planes has been found, return the exact minimizer. By occasionally solving the master problem between subgradient method steps, finite-time convergence can be guaranteed. Moreover, because subgradient methods do not need to solve the master problem at every iteration, it is possible that they may converge towards the function minimizer at a faster rate than traditional cutting plane algorithms. Subgradient methods also have the ability to query subgradients with respect to non-optimal FOP solutions for heuristic approximations and stochastic descent methods, which we explore in Section 2.6. With these ideas in mind, we implement solution methods that make use of two major methods from the literature; projected gradient descent (Boyd et al. 2003) and the Frank-Wolfe method (Frank and Wolfe 1956).

2.4.2.1 Projected gradient descent (PGD).

To execute an iteration of projected gradient descent, given some step-size $t_k > 0$, we choose our next $\boldsymbol{\alpha}$ with $\boldsymbol{\alpha}^{(k)} \leftarrow \boldsymbol{\alpha}^{(k-1)} - t_k(C(\hat{\mathbf{y}} - \mathbf{y}^{(k-1)}))$, and then project $\boldsymbol{\alpha}^{(k)}$ onto its domain $\mathcal{A}$ (Boyd et al. 2003). For a sufficiently small step size t_k, if $\boldsymbol{\alpha}^{(k)}$ is not the minimizer of ξ_{ABS}, then $\boldsymbol{\alpha}^{(k+1)}$ will yield a lower optimality gap. Further, there exist multiple choices of the sequence t_k such that the method will converge upon an optimal solution as k approaches infinity (Boyd et al. 2003).

Remark 2.5. *Once all facet-defining tangent planes that are tangent at the gap function minimizer are known, then a single run of the master problem will yield the exactly optimal solution. We can use this knowledge to create a descent method that will be guaranteed to terminate at an optimal solution rather than infinitely converge on it.*

We formalize this idea as follows. Let $\xi_{\text{ABS}}(\boldsymbol{\alpha})$ be the absolute gap function generated by a polyhedral forward feasible region $\mathcal{S}$ and an inverse input $\hat{\mathbf{y}}$. Let $\boldsymbol{\alpha}^*$ be a minimizer of $\xi_{\text{ABS}}(\boldsymbol{\alpha})$. Let $\boldsymbol{\alpha}^{(k)}$ be generated by a projected subgradient descent method that uses step-size sequence t_k, and let $\mathcal{S}^k$ denote a set of FOP solutions $\mathbf{y}^{(0)} \dots \mathbf{y}^{(k)}$ found in the descent process. At every step k, a subtangent plane to $\xi_{\text{ABS}}(\boldsymbol{\alpha})$ at $\boldsymbol{\alpha}^{(k)}$ is generated from $\mathbf{y}^{(k)} \in \mathcal{S}^k \supseteq \mathcal{S}^{k-1}$. If $\boldsymbol{\alpha}^*$ is a unique minimizer of $\xi_{\text{ABS}}(\boldsymbol{\alpha})$, then it corresponds to a facet of $\text{conv}(\mathcal{B})$ that contains $\mathbf{C}\hat{\mathbf{y}} - \xi_{\text{ABS}}(\boldsymbol{\alpha}^*)\mathbf{1}$ in its interior (proof that $\mathbf{C}\hat{\mathbf{y}} - \xi_{\text{ABS}}(\boldsymbol{\alpha}^*)\mathbf{1}$ is contained on the boundary of $\text{conv}(\mathcal{B})$ follows from Theorem 1 (b) of Chan et al. (2014)). Similarly, each facet-defining tangent plane of $\xi_{\text{ABS}}(\boldsymbol{\alpha})$ corresponds to a vertex of $\text{conv}(\mathcal{B})$. Let B^k denote a set of points on the boundary of $\text{conv}(\mathcal{B})$ corresponding to $\mathcal{S}^k$, i.e. the set $\{\mathbf{C}\mathbf{y}^{(0)} \dots \mathbf{C}\mathbf{y}^{(k)}\}$. Finally, note that if $\boldsymbol{\alpha}^*$ is not a unique minimizer, then the above is true for a lower dimensional face of $\text{conv}(\mathcal{B})$ rather than a facet.

Proposition 2.6. *Suppose $\boldsymbol{\alpha}^*$ is a unique minimizer of $\xi_{ABS}(\boldsymbol{\alpha})$. If the convex hull of B^k contains $\mathbf{C}\hat{\mathbf{y}} - \xi_{ABS}(\boldsymbol{\alpha}^*)\mathbf{1}$ in the interior of one of its facets, then solving the master problem $GIO_{ABS}(\hat{\mathbf{y}}, \mathcal{S}^k)$ will yield $\boldsymbol{\alpha}^*$.*

We provide proof of this proposition in A.1.1. As stated in the proof, if $\text{conv}(\mathcal{B})$ is a set of dimension $|\mathcal{K}|$, then there exists at least one set of only $|\mathcal{K}|$ elements of $\text{conv}(\mathcal{B})$ that need to be discovered for this condition to be met. We henceforth refer to the case where solving the master problem yields the optimal solution as the *MP solve condition*.

Definition 2.7 (MP Solve Condition)**.** *For a sequence $\boldsymbol{\alpha}^{(k)}$ generated by an inverse optimization solution method, the MP Solve Condition is achieved at an iteration k when solving $GIO_{ABS}(\hat{\mathbf{y}}, \mathcal{S}^k)$ yields $\boldsymbol{\alpha}^*$. As stated in Proposition 2.6, this occurs when conv(B^k) contains $\mathbf{C}\hat{\mathbf{y}} - \xi_{ABS}(\boldsymbol{\alpha}^*)\mathbf{1}$ in the interior of one of its facets.*

Our implementation of a terminating projected gradient descent solution algorithm is presented in Algorithm 1. Our algorithm iterates through steps of projected gradient descent, and then solves the master problem whenever a descent step neither decreases the gap function by the maximum expected amount nor returns a new FOP extreme point. This occurs when $\boldsymbol{\alpha}^{(k)\intercal} C\hat{\mathbf{y}} - \boldsymbol{\alpha}^{(k)\intercal}\mathbf{C}\mathbf{y}^{(k)} > \boldsymbol{\alpha}^{(k-1)\intercal}\mathbf{C}\hat{\mathbf{y}} - \boldsymbol{\alpha}^{(k-1)\intercal} C\mathbf{y}^{(k-1)} + (\boldsymbol{\alpha}^{(k)} - \boldsymbol{\alpha}^{(k-1)})^\intercal (C(\hat{\mathbf{y}} - \mathbf{y}^{(k-1)}))$ and $\mathcal{S}^k = \mathcal{S}^{k-1}$. One possible case when this occurs is when a sufficient number of facets of $\xi_{ABS}(\boldsymbol{\alpha})$ surrounding the minimizer have been found and the descent method is repeatedly travelling between them. In this case, the MP solve condition has been reached, and the final output of $\text{GIO}_{\text{ABS}}(\hat{\mathbf{y}}, \mathcal{S}^k)$, denoted $\boldsymbol{\alpha}^{\text{final}}$ will be the gap function minimizer $\boldsymbol{\alpha}^*$, which can be verified by solving the FOP with $\boldsymbol{\alpha}^{\text{final}}$. If the resulting gap function value is equal to the objective value of $\text{GIO}_{\text{ABS}}(\hat{\mathbf{y}}, \mathcal{S}^k)$, then $\boldsymbol{\alpha}^{\text{final}} = \boldsymbol{\alpha}^*$ and the algorithm terminates. If this FOP solve yields a different gap function value than the objective value returned by $\text{GIO}_{\text{ABS}}(\hat{\mathbf{y}}, \mathcal{S}^k)$, then the MP solve condition had not yet been reached. The termination criterion may have been reached by descent steps travelling between facets of $\xi_{ABS}(\boldsymbol{\alpha})$ that are not adjacent to the minimizer $\boldsymbol{\alpha}^*$, and thus the descent step size is too large. In this case, the step size is divided by two and iterations of projected gradient descent proceed until the termination criterion is reached again.

Proposition 2.8. *Algorithm 1 (i) terminates in a finite number of iterations, and (ii) returns the gap function minimizer.*

Proof of Proposition 2.8 is provided in Appendix A.1.2.

2.4.2.2 Accelerated projected gradient descent (PGD-A).

We implement Polyak's Heavy Ball Method as part of Algorithm 1 by adding a momentum term with a coefficient $\beta \in (0, 1]$ to the step formulation before projecting onto the domain (Polyak 1964). We use the same termination method as in the projected gradient descent method. See Algorithm 4 in A.7 for pseudocode of the algorithm structure with the addition of a momentum term, utilizing the same termination technique as projected gradient descent.

2.4.2.3 Frank-Wolfe method (FW).

The Frank-Wolfe method of optimization over a convex bounded function entails querying the loss function gradient at a given point, finding the minimizer of said gradient in the function domain, and moving in the direction towards said point with a step size that decreases at a a rate of $O(\frac{1}{k})$. In our case, the domain of the function we are minimizing $(\mathcal{A})$ is a unit simplex, so the task of finding the gradient minimizer is trivial; we find the

Algorithm 1: Gap-gradient projected gradient descent method.

input : C, $\hat{\mathbf{y}}$, FOP, t
output: $\boldsymbol{\alpha}^{\text{best}}$, ξ^{final}
$k = 0$, $\mathcal{S}^k \leftarrow \emptyset$, $\boldsymbol{\alpha}^{(k)} \leftarrow \frac{1}{\mathcal{K}}\mathbf{1}$;
$\mathbf{y}^{(k)} \leftarrow \text{FP}(\boldsymbol{\alpha}^{(k)})$;
$\xi_{\text{ABS}} \leftarrow \boldsymbol{\alpha}^{(k)\intercal} C\hat{\mathbf{y}} - \boldsymbol{\alpha}^{(k)\intercal} C\mathbf{y}^{(k)}$;
while $\boldsymbol{\alpha}^{(k)\intercal} C\hat{\mathbf{y}} > \boldsymbol{\alpha}^{(k)\intercal} C\mathbf{y}^{(k)}$ **do**
  $k \leftarrow k + 1$;
  $\mathcal{S}^k \leftarrow \mathcal{S}^{k-1} \cup \mathbf{y}^{(k-1)}$;
  $\boldsymbol{\alpha}^{(k)} \leftarrow \boldsymbol{\alpha}^{(k-1)} - t(C(\hat{\mathbf{y}} - \mathbf{y}^{(k-1)}))$;
  $\boldsymbol{\alpha}^{(k)} \leftarrow \text{proj}_{\Delta^{\mathcal{K}}}(\boldsymbol{\alpha}^{(k)})$;
  $\mathbf{y}^{(k)} \leftarrow \text{FOP}(\boldsymbol{\alpha}^{(k)})$;
  if $\boldsymbol{\alpha}^{(k)} = \boldsymbol{\alpha}^{(k-1)}$ ***or*** $\boldsymbol{\alpha}^{(k)\intercal} C\hat{\mathbf{y}} - \boldsymbol{\alpha}^{(k)\intercal} C\mathbf{y}^{(k)} > \xi_{ABS} + (\boldsymbol{\alpha}^{(k)} - \boldsymbol{\alpha}^{(k-1)})^\intercal (C(\hat{\mathbf{y}} - \mathbf{y}^{(k-1)}))$
   and $\mathbf{y}^{(k)} \in \mathcal{S}^k$ **then**
    $\boldsymbol{\alpha}^{\text{final}}, \xi^{\text{final}} \leftarrow \text{GIO}_{\text{ABS}}(\hat{\mathbf{y}}, \mathcal{S}^k)$;
    $k \leftarrow k + 1$;
    $\mathbf{y}^{(k)} \leftarrow \text{FOP}(\boldsymbol{\alpha}^{\text{final}})$;
    if $\boldsymbol{\alpha}^{final\intercal} C\hat{\mathbf{y}} - \boldsymbol{\alpha}^{final\intercal} C\mathbf{y}^{(k)} = \xi^{final}$ **then**
      stop
    else
      $\boldsymbol{\alpha}^{(k)} \leftarrow \boldsymbol{\alpha}^{\text{final}}$;
      $\mathcal{S}^k \leftarrow \mathcal{S}^{k-1} \cup \mathbf{y}^{(k-1)}$;
      $t \leftarrow \frac{t}{2}$;
    end
  else
    $\xi_{\text{ABS}} \leftarrow \boldsymbol{\alpha}^{(k)\intercal} C\hat{\mathbf{y}} - \boldsymbol{\alpha}^{(k)\intercal} C\mathbf{y}^{(k)}$;
  end
end

smallest component i of the gradient, and the minimizer is the vector with 1 at component i and 0 for all others. For a Lipschitz-continuous convex function (e.g., the absolute gap function, ξ_{ABS}), the algorithm converges on an optimal solution at a rate of $O(\frac{1}{k})$ (Frank and Wolfe 1956). See Algorithm 5 in A.7 for our implementation of the Frank-Wolfe method in the context of generalized inverse optimization, using the same termination method as projected gradient descent.

Remark 2.9. *We note that it is actually possible to converge on the optimal solution using the Frank-Wolfe method without necessarily being able to query the gradient accurately, so long as one can find the point in the gap space that minimizes said gradient. For the case where the loss function is absolute sub-optimality and $\mathcal{A}$ is the unit simplex, we only need to consider $|\mathcal{K}|$ such points in*

our domain, so it may be possible to not solve the forward problem to completion, as long as we do enough work to be certain which of these points would be the gradient minimizer (i.e., the optimal solution will produce a gradient with a specific component being the smallest). As soon as we know this, we can abort solving the forward problem and take the next Frank-Wolfe step. Since solving the MIP forward problem is generally the most computationally complex step of all algorithms discussed so far, this has the potential to significantly improve time-per-iteration in the Frank-Wolfe algorithm (and thus potentially in overall time). However, if our method of executing early cutoffs is not able to accurately query the gap function value and yield tangent planes, we will not be able to only conduct partial FOP solves and then terminate the algorithm with an MP solve condition. However, one may choose to run several partial solve iterations before changing the approach and using fewer full solve iterations in a close neighborhood of the solution before terminating. Note that the gap function tangent planes generated when far from the solution are not necessarily useful for terminating the master problem, and only those planes tangent to facets that include a function minimizer need to be known.

We provide a formal method for operationalizing this idea in A.4, but we leave the practical implementation of this approach for future work.

2.5 Experimental Evaluation of Solution Methods

In this section, we evaluate our proposed methods and compare them to an existing method using a set of multi-objective inverse mixed integer optimization problems derived from FOPs obtained from the MIPLIB 2017 mixed integer programming library (Gleixner et al. 2021).

2.5.1 Experimental Setup

We implement and compare our three proposed solution methods: 1) projected gradient descent (PGD), 2) accelerated projected gradient descent (PGD-A), and 3) Frank-Wolfe method (FW). We compare our approaches with the only applicable solution method from the literature: the Moghaddass-Terekhov cutting plane method (CP) (Moghaddass and Terekhov 2020). For PGD and PGD-A, the initial step size is chosen so that the Euclidean norm of the first descent step is 0.1, i.e. $||\boldsymbol{\alpha}^{(1)} - \boldsymbol{\alpha}^{(0)}||_2 = 0.1$. For PGD-A, we use a momentum coefficient of 0.5. We note that our approach produces a conservative estimate for the performance of these methods because a more rigorous search for optimal step sizes and momentum coefficients may substantially improve solution times.

The list of selected MIPLIB instances (i.e., FOPs) is included in A.6.1. These nine instances were chosen because (1) they were flagged as "benchmark-suitable" and rated as "easy" solution difficulty, allowing for relatively tractable run times for the inverse solution algorithms, (2) they contained at least 128 continuous variables which can serve as reasonable sub-objectives, in addition to containing integer or binary variables, (3) they were neither infeasible nor unbounded for any sampled objective weights $\boldsymbol{\alpha}$ in our computational trials, and (4) they span a variety FOP structures that are suited for different application settings.

To adapt each of the nine FOPs into a multi-objective generalized inverse optimization problem, we conduct the following process. For each value of k in the sequence $[4, 8, 16, 32, 64, 128]$:

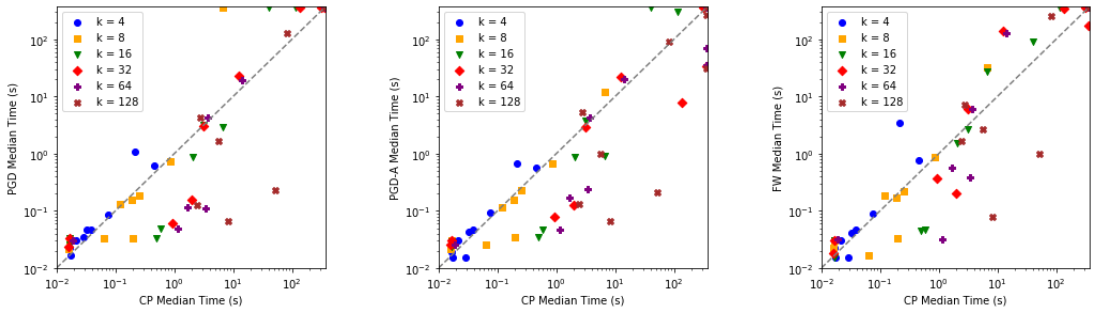
1. We randomly sample k continuous variables from the FOP formulation to be the individual sub-objectives. Each sub-objective function in our multi-objective formulation is the value of one of these sampled variables. Thus each row in the matrix $\mathbf{C}$ contains all entries equalling 0 except for one sampled variable, which has coefficient 1. We execute this sampling process 3 times per FOP.

2. For each sampled matrix $\mathbf{C}$, we sample 3 different coefficient vectors $\boldsymbol{\alpha}$ from the unit simplex in k dimensions.

3. For each sampled combination of $\mathbf{C}$ and $\boldsymbol{\alpha}$, we solve the FOP with the objective function $\boldsymbol{\alpha}^\intercal \mathbf{C}\mathbf{y}$ to obtain our inverse input solution $\hat{\mathbf{y}}$. In total, this gives us 9 inverse inputs per FOP formulation (per k).

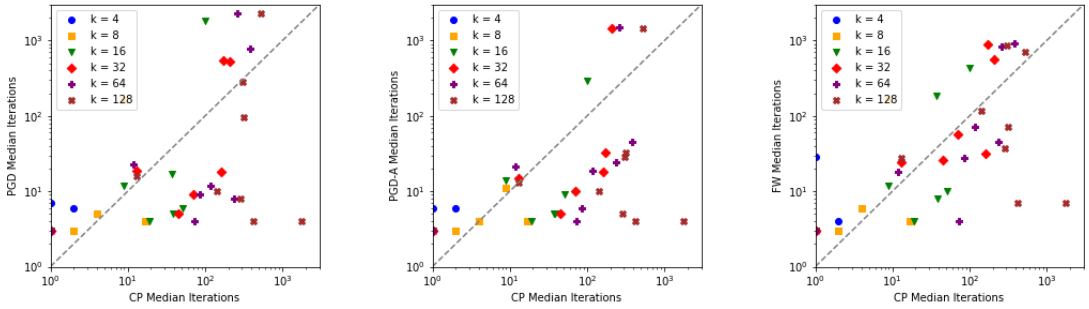
4. We used each of the four inverse solution methods to solve the inverse optimization problem for each of the nine input solutions. We use the absolute sub-optimality loss function and the unit simplex for the set of allowable objectives. For each solve, we record the total run time and the total number of iterations.

All computational experiments were executed on an HP EliteDesk 800 G4 TWR with an Intel(R) Core(TM) i5-8500 CPU running at 3.00GHz with 6 Cores and 6 Logical Processors and 16.0 GB of RAM. All code was written and executed in Python version 3.7.3 and all FOPs were solved using Gurobi version 9.0.2. A maximum running time of 360 seconds was used in each trial before cutting off the solution algorithms.



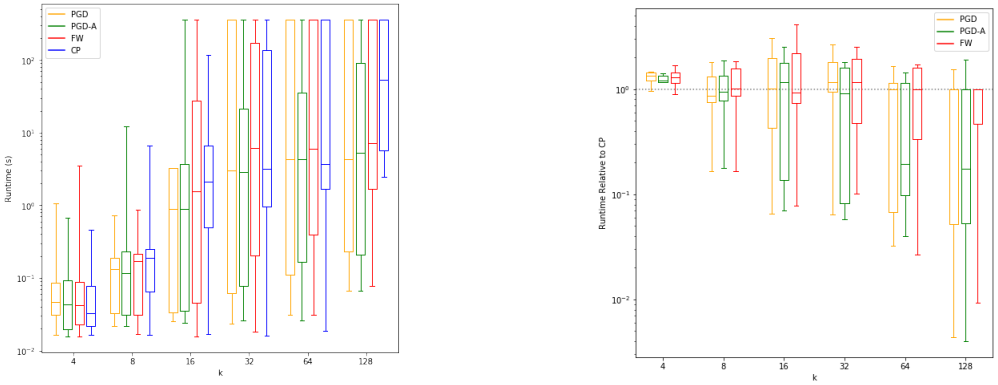
(a) (b) (c)

Figure 2.2: A comparison of median solution times for each value of k (marker type): (a) PGD vs. CP, (b) PGD-A vs. CP, and (c) FW vs. CP. For each marker type (value of k), there are nine markers (one for each FOP instance) that denote the median solution time across all samples for that FOP instance.



(a) (b) (c)

Figure 2.3: A comparison of the median number of iterations for each value of k (marker type): (a) PGD vs. CP, (b) PGD-A vs. CP, and (c) FW vs. CP.



(a) (b)

Figure 2.4: Boxplots for each value of k of: (a) the median runtimes across FOP instances, and (b) the ratio of median runtimes compared to CP median runtime for the 3 gap-gradient methods.

2.5.2 Results

Figure 2.2 displays the median solution times of each novel algorithm in comparison to CP. PGD, PGD-A, and FW improved upon CP in 27, 30, and 26 of the 54 instances, respectively. For those instances where our methods did not improve upon CP, the performance was similar. Figure 2.3 displays the median iteration count of each novel algorithm in comparison to CP. By design of the termination criterion, our gap-gradient methods take a minimum of 3 iterations before terminating, and for many instances where $k = 4$, the initialized value of $\boldsymbol{\alpha}^{(0)} = \frac{1}{\mathcal{K}}\mathbf{1}$ is a loss function minimizer, and thus can be completed in one iteration by the CP method.

Figure 2.4 displays boxplots of the median solution time for each of the four methods and the improvement in median solution time over CP. The median solution time ($\pm$ standard deviation) as a function of k for CP was 0.03s ($\pm$0.14), 0.19s ($\pm$2.01), 2.10s ($\pm$37.0), 3.17s ($\pm$137.2), 3.75s ($\pm$168.5), 53.1s ($\pm$160.0). The median solution time ($\pm$ standard deviation) for our best performing implementation (PGD-A) was 0.04s ($\pm$0.24), 0.12s ($\pm$3.82), 0.88s ($\pm$137.5), 2.91s ($\pm$111.1), 4.29s ($\pm$110.3), 5.29s ($\pm$128.1). The median improvement in median solution time as a function of k was -0.01s (-33.3%), 0.07s (36.8%), 1.22s (58.1%), 0.26s (8.2%), -0.54s (-14.4%), and 47.81s (90.0%).

2.6 Application to Political Gerrymandering

In this section, we present the FOP used for political districting (Section 2.6.1), present custom solution approaches based on graph coarsening and ensemble methods (Section 2.6.2), and then evaluate our solution approaches using randomly generated problem instances (Section 2.6.3). In the following section (Section 2.7), we present a case study using real data from the State of Iowa.

2.6.1 Forward optimization problem (FOP)

We use the political districting model from Swamy et al. (2022) as our FOP. The model is a mixed-integer linear program that determines the political districting for a given state. Let $G = \{V, E\}$ be a graph where each vertex $v \in V$ represents a census block or some larger tract of land within the state and edges $e \in E$ represent the adjacency of such areas. Each vertex v is associated with a population $p_v \in V$, the number of democratic (republican) voters $p_v^D(p_v^R), v \in V$, and an area (in square feet) $a_v, v \in V$. We let $d_{i,j}, i, j \in V$ represent the Euclidean distance between vertex i and j (not necessarily adjacent).

Let the decision variables x_{ij} be a binary indicator of whether the vertex $i \in V$ is the center of a district that contains the vertex $j \in V$. The decision variable f_{ijv} indicates a network flow between vertices j and $v \in V$ where they both have district center i, which is used to maintain contiguity of modeled districts. The binary decision variable z_i^D indicates if the district with center i is won by the Democratic party, and v_{ij}^D indicates whether or not j is in a district with center i which is won by the Democratic party. Finally, the decision variable w_i indicates the number of wasted Democratic votes minus the number of wasted Republican votes in the election in the district with center i.

Intuitively, the model solution determines a partition of the graph G into L disjoint connected subgraphs, which denote the legislative districts. We provide the full FOP formulation in Appendix A.5. We define the objective of the FOP as a weighted sum of three components: a measure of population imbalance, a measure of compactness, and a measure of efficiency gap.

2.6.1.1 Population imbalance (ρ).

We let ρ represent the population imbalance measured as the largest relative deviation from the average district population exhibited by any district.

$$\rho = \frac{\max_{i \in V} |\sum_{j \in V} p_i x_{ij} - \frac{\sum_{j \in V} p_j}{L}|}{\frac{\sum_{j \in V} p_j}{L}}$$

For example, in a state of population 1000, ten districts of population 100 would yield $\rho = 0$, while nine districts with population 101 and one of population 91 would yield $\rho = \frac{|91-100|}{100} = 0.09$. Unlike Swamy et al. (2022), we include population imbalance in the objective rather than a constraint because the current legal text and judicial practice indicates that district populations should be *as close to equal as possible*, clearly indicating it as an objective to be minimized.

2.6.1.2 Compactness (σ_A).

We let σ_A represent the compactness of a districting. Compactness is formulated as a measure of the p-median distance weighted by the area of each tract of land, divided by the area-weighted 1-median distance of the entire state (denoted by M). Using our notation, this can be written as:

$$\sigma_A = \frac{\sum_{i,j \in V} d_{ij} a_j x_{ij}}{M}.$$

2.6.1.3 Efficiency gap (ϕ_{EG}).

We let ϕ_{EG} denote the efficiency gap, measured as defined by Stephanopoulos and McGhee (2015) using a set of constraints formulated by Swamy et al. (2022). The measurement of the efficiency gap is motivated by the construction and counting of *wasted votes*, which contain both votes cast for a losing candidate and votes cast for a winning candidate beyond the necessary majority. The relative disparity in wasted votes distributed across two major parties as the result of district-based elections is measured as the efficiency gap, which can be calculated as:

$$\phi_{EG} = \frac{|\sum_{i \in V} w_i|}{\sum_{i \in V} p_i^D + p_i^R}$$

The construction of this metric directly targets districting tactics known as *cracking* and *packing*, which are alleged to be a major strategy of partisan gerrymandering.

Since this problem is very computationally taxing to solve, Swamy et al. (2022) propose a graph coarsening approach to reduce the size of the problem formulation.

2.6.2 Graph coarsening

In this section, we present custom solution approaches for graph based inverse mixed integer optimization problems. Our approach uses a combination of graph coarsening with ideas from machine learning and data-driven inverse optimization. Although these methods were inspired by our application of political districting, they can be applied to any inverse optimization problem where the FOP can be represented as a graph-based problem.

2.6.2.1 Maximal matching.

Basic graph coarsening is achieved by producing a random maximal matching of the edges of the graph and contracting all edges in the produced matching into vertices. A random maximal matching is produced by iterating through a random ordering of the edges, and selecting an edge to be contracted if both of its endpoints are not adjacent to any edges that have already been selected for contraction. All population and area data at the endpoints of a contracted edge is summed together to produce the data for the newly formed vertex, and the distances associated with the edges of newly formed vertices are calculated from landmass centroids derived from area-weighted averages of the two contracted endpoints. This process can be iteratively repeated as many times as desired to shrink the graph to a needed size. Overall, this process produces a smaller graph with the same underlying spatial structure but at a lower data resolution. This allows for a problem that is less

computationally taxing, at a tradeoff of reduced accuracy of the final solution due to lower data resolution.

Swamy et al. (2022) suggest that coarsening methods that choosing matchings that prioritize merging vertices with a low combined population preserve solutions with lower population imbalances, and lower objectives values for many subobjectives, with the same improvement on computational efficiency as random maximal matchings. As such, their method is deterministic, producing an ordering of edges by directly ranking the combined populations.

We propose a variation that still involves random choices of edges (which we exploit in the next section), while gaining the advantage of coarsenings that preserve better solutions. As such, our method creates an ordering of edges by sampling an exponential random variable for each edge with a mean of the combined population divided by twice the mean vertex population, and ordering by the edges by the values of the random samples. We note that if all edges have identical combined populations, then this method produces the same distribution of outputs as a random maximal matching method.

2.6.2.2 Ensemble-based coarsening.

We leverage the idea of *ensembles* from machine learning to produce and exploit multiple coarsenings of the same graph (Breiman 1996). Since both the basic maximial matching and our proposed variation are non-deterministic, multiple (different) coarsenings of the same graph can be produced with the same method. While each coarsening loses some resolution, elements lost in one coarsening may be preserved in another. An ensemble-based approach may allow for the risks of coarsening to be reduced in the aggregate, while potentially preserving the computational savings of coarsening. Let FOP_1, FOP_2 ... FOP_n denote the forward optimization problems generated for an ensemble of n different coarsenings of a political districting FOP, and let $\mathcal{S}_1, \mathcal{S}_2 \ldots \mathcal{S}_n$ denote their respective feasible regions. We present two options for how an ensemble of coarsenings can be used to produce heuristic solutions to the inverse problem:

1. **Solve each instance independently.** In this approach, we solve an inverse model independently on each coarsened graph, generating the inverse solution $\boldsymbol{\alpha}_i^*$ from each individual inverse formulation $\text{GIO}_{\text{ABS}}(\hat{\mathbf{y}}, \mathcal{S}_i)$. The average of the outputs, $\frac{1}{n} \sum_{i \in 1 \ldots n} \boldsymbol{\alpha}_i^*$, can then be used as an approximation of the true $\boldsymbol{\alpha}^*$. Alternatively, the set of inverse solutions can be analyzed as a distribution. For example, the convex hull of the set of outputs can be interpreted as a polytope of potential inverse solutions.

2. **Solve a multi-point formulation.** In this approach, we solve a single multi-point inverse optimization formulation produced by creating a master problem with constraints that are derived from multiple coarsened graphs at once. A single slack variable describes the optimality gap across all constraints, thus minimizing the maximum optimality gap across all coarsenings.

For the remainder of this section, we focus on the second approach, which can be formalized as the following optimization problem:

$$\begin{aligned} \underset{\boldsymbol{\alpha},\, \xi_{\text{ENS}}}{\text{minimize}} \quad & \xi_{\text{ENS}} && \text{(MultiGIO}_{\text{ABS}}\text{MinMax)} \\ \text{subject to} \quad & \boldsymbol{\alpha}^\intercal \mathbf{C}\hat{\mathbf{y}} \leq \boldsymbol{\alpha}^\intercal \mathbf{C}\mathbf{y} + \xi_{\text{ENS}}, \quad \forall\, \mathbf{y} \in \mathcal{S}_1, \mathcal{S}_2, \dots \mathcal{S}_n, \\ & ||\boldsymbol{\alpha}||_1 = 1, \\ & \boldsymbol{\alpha} \geq \mathbf{0},\ \ \xi_{\text{ENS}} \geq 0. \end{aligned}$$

For MultiGIO$_{\text{ABS}}$MinMax, we can show that as as the size of the ensemble approaches infinity, the likelihood of our method finding the same solution as the full resolution graph approaches 1.

Theorem 2.10. *Given an ensemble of n independently sampled coarsenings of G down to $v \geq L$ vertices, such that any coarsened graph with v vertices has a non-zero probability of being sampled, as $n \to \infty$, the probability of MultiGIO$_{ABS}$MinMax yielding $\boldsymbol{\alpha}^* =$ approaches 1.*

Proof of Theorem 2.10 is provided in Appendix A.1.3. In the context of implementing gap-gradient methods for the multipoint inverse formulation, the coarsening that returns the highest optimality gap at the hypothesis $\boldsymbol{\alpha}^{(k)}$ determines the next step in the descent process.

Proposition 2.11. *Let FOP$_1$, FOP$_2$ ... FOP$_n$ denote the forward optimization problems generated for an ensemble of n different coarsenings of a political districting FOP, and let $\xi_{ENS}(\boldsymbol{\alpha})$ denote the gap function corresponding to the multipoint inverse formulation MultiGIO$_{ABS}$MinMax generated by this ensemble of coarsenings. Let $\boldsymbol{\alpha}^{(k)}$ be the hypothesis cost vector at iteration k of a solution method for MultiGIO$_{ABS}$MinMax, and let $y_i^{(k)}$ denote the optimal solution of FOP$_i(\boldsymbol{\alpha}^{(k)})$. At any $\boldsymbol{\alpha}^{(k)}$, the subgradient of $\xi_{ENS}(\boldsymbol{\alpha})$ is equal to $\mathbf{C}(\hat{\mathbf{y}} - \underset{\mathbf{y} \in \mathbf{y}_1^{(k)} \dots \mathbf{y}_n^{(k)}}{\arg\min}\ (\boldsymbol{\alpha}^{(k)\intercal}\mathbf{C}\mathbf{y}))$.*

Proof of Proposition 2.11 is provided in Appendix A.1.4.

Since the loss function $\xi_{\text{ENS}}(\boldsymbol{\alpha})$ satisfies all the properties listed in Remark 2.4, the gap-gradient methods described in Section 2.4.2 may be used to find solutions to MultiGIO$_{\text{ABS}}$MinMax.

2.6.2.3 Stochastic descent with coarsened ensembles.

In this section, we demonstrate how stochastic gradient descent can be used with coarsening to solve generalized inverse optimization problems with graph-based FOPs. Instead of generating an ensemble of coarsenings to create a lower approximation ξ_{ENS} of ξ_{ABS}, and then deterministically minimizing the approximation, one can instead use ensembles of coarsenings to produce estimators of subgradients of ξ_{ABS} in the application of a stochastic gap-gradient method to an inverse model formulated for the full sized graph. Similar to the multipoint formulation method, our gradient approximation is calculated as $\nabla\xi_{\text{ABS}}(\boldsymbol{\alpha}^{(k)}) \approx \mathbf{C}(\hat{\mathbf{y}} - \underset{\mathbf{y}\in\mathbf{y}_1^{(k)}\dots\mathbf{y}_n^{(k)}}{\arg\min}\ (\boldsymbol{\alpha}^{(k)\intercal}\mathbf{C}\mathbf{y}))$. However, in this application, an ensemble of coarsenings $\text{FOP}_1^{(k)}$, $\text{FOP}_2^{(k)}\dots\text{FOP}_n^{(k)}$ is independently randomly sampled during each iteration k of the subgradient method. Here the use of coarsenings and ensembles is analogous to the process of mini-batching in stochastic gradient descent for training neural networks, where the true loss function gradient is estimated by calculating the loss function with respect to a randomly sampled subset of the the full set of available samples (Schmidt 2019). In neural networks, the samples come from the set of training observations; in the inverse graph partitioning case, 'samples' come from the set of possible partitions $\mathcal{S}$, and each coarsened graph contains contains a subset of such samples.

It is not obvious that solving the FOP on coarsened graphs or ensembles thereof yields an unbiased estimator of $\nabla\xi_{\text{ABS}}$. However, Theorem 2.10 does show that approximating the gradient of ξ_{ABS} with $\mathbf{C}(\hat{\mathbf{y}} - \underset{\mathbf{y}\in\mathbf{y}_1^{(k)}\dots\mathbf{y}_n^{(k)}}{\arg\min}\ (\boldsymbol{\alpha}^{(k)\intercal}\mathbf{C}\mathbf{y}))$ constitutes a *consistent and asymptotically unbiased estimator* of $\nabla\xi_{\text{ABS}}$, as defined by Chen and Luss (2019). Under the conditions of a convex but not strongly convex gap function (such as absolute sub-optimality), when using a subgradient estimation method where expected estimation error is inversely proportional to $\sqrt{k}$, stochastic projected gradient descent observes a convergence rate of $O(\frac{1}{\sqrt{k}})$ towards the gap function minimum value. To obtain this bound for the coarsening application, the subgradient estimation must involve progressively larger ensembles with each iteration. Without progressively increasing the ensemble size, we can expect our method to converge to the neighborhood of the minimizer, but is not guaranteed to find the exact minimizer.

Algorithm 9 presents the structure of an implementation of stochastic subgradient estimation within a projected gradient descent approach to loss function minimization, where n_k indicates a rule for selecting a coarsening ensemble size at a given iteration k, and K denotes a maximum number of iterations. At the end of K iterations, a weighted averaging of the values of $\boldsymbol{\alpha}^{(k)}$ is returned. We note that the termination criterion utilized in Section 2.4.2 are much less likely to be triggered at any given iteration. As such, in

our evaluation of stochastic subgradient methods, we examine how closely the descent methods approach the true loss function minimum and minimizer, rather then comparing the time until termination yielding an exact minimizer, as in the previous computational evaluations.

2.6.2.4 Boosted ensembles of coarsenings.

To accentuate the diversity of the ensemble of coarsened graphs, we implement a method of sequential coarsening inspired by boosting ensemble methods from machine learning (Schapire 1999). In an unboosted coarsening implementation, the random maximal matching of a graph is determined by creating a uniformly randomly ordered list of the edges, proceeding through the list in order, and contracting each edge if and only if both endpoints are not adjacent to an edge that has already been contracted. The random shuffling that determines each coarsening is independent for each element of the ensemble. In a *boosted implementation*, the randomly ordered shuffling of the edges is done in a weighted fashion such that an edge with a higher weight is more likely to be later in the ordering. In coarsening the first element of the ensemble, either the initial edge weights are uniform, or weighted by population as detailed in Section 2.6.2.1. If an edge $e \in E$ is contracted when creating the i^{th} element of the ensemble, then for the $(i+1)^{\text{th}}$ element, the weight of edge e is multiplied by a factor $\eta > 1$, and its weight is multiplied by $\frac{1}{\eta}$ otherwise. This ensures that every ensemble member is less likely to have certain edges contracted that were frequently contracted in previous coarsenings, discouraging highly correlated coarsenings in the ensemble. This process is detailed in Appendix A.7 (Algorithm 8). In the case of ensembles of graphs coarsened multiple times over, this process can be repeated at each tier of coarsening in a tiered tree-like fashion until the desired depth of coarsening is reached.

2.6.3 Computational Experiments

In this section, we evaluate our solution approaches using randomly generated problem instances representative of political districting.

2.6.3.1 Experimental setup.

To evaluate our models, we use 8 simulated graphs/states of size $|G| = 20$. See A.6.2 for details on how we simulate a state. For each graph/state, we create 5 uniformly randomly sampled multi-objectives weightings, and solve the corresponding FOP to create 5 inverse inputs for each simulated state. In total, we have 40 simulated IO instances. For each instance, we perturb the value of each subobjective by adding a random value in the range

[0.0375, 0.0625]; this allows us to find an interior point that is likely to have a unique gap function minimizer. We solve each instance to optimality without any coarsening as a benchmark.

We conduct three experiments. First, we evaluate the impact of coarsening on the accuracy of inverse optimization solutions. We evaluate two coarsening methods: contracting edges selected from a random maximal matching of vertices and contracting edges selected from a population-weighted maximal matching. For each method, we conduct both one and two rounds of coarsening for each of the 40 problem instances. We then solve an IO problem for each instance using the PGD-A method to obtain a heuristic solution $\boldsymbol{\alpha}^h$.

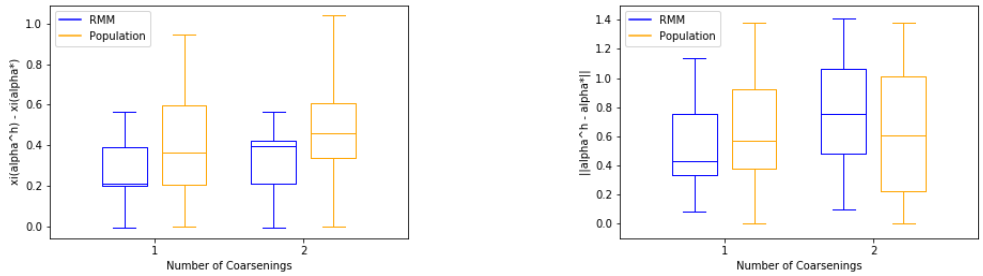
Second, we evaluate the efficacy of the multi-point ensemble formulation as a heuristic for minimizing the gap function. We generate coarsened graphs with one round of random maximal matching, and we create ensembles of size 1, 4, 16, and 64, using both the independently random coarsening and the boosted coarsening techniques described in Section 2.6.2.4 with learning rate $\eta = 1.5$. For each ensemble, we apply the multi-point inverse formulation MultiGIO$_{\text{abs}}$MinMax on the ensemble to achieve the heuristic solution $\boldsymbol{\alpha}^h$.

Third, we evaluate the performance of stochastic descent methods on estimating the gap function minimizer. We apply stochastic descent methods with unboosted ensembles of sizes 16 and 64, and an increasing ensemble rule $n = k$. For the descent methods, a step size is automatically chosen such that the first step has length 0.1, with following step sizes decreasing at a rate of $\frac{1}{\sqrt{k}}$, and a momentum coefficient of 0.1. We run each descent algorithm for a total of 12 iterations and return a heuristic solution $\boldsymbol{\alpha}^h$ at iterations 0, 4, 8, and 12.

For each experiment, we record the total solution time, the gap function value of the returned output cost vector minus the true gap function minimum value (i.e., $\xi_{\text{ABS}}(\boldsymbol{\alpha}^h) - \xi_{\text{ABS}}(\boldsymbol{\alpha}^*)$), and the Euclidean distance between the returned cost vector and the optimal cost vector (i.e., $||\boldsymbol{\alpha}^h - \boldsymbol{\alpha}^*||_2$).

2.6.3.2 Experimental results.

Figure 2.5a displays the gap function value of the returned output cost vector minus the true gap function minimum value (i.e., $\xi_{\text{ABS}}(\boldsymbol{\alpha}^h) - \xi_{\text{ABS}}(\boldsymbol{\alpha}^*)$). For coarsening by random maximal matching, the median (standard deviation) difference was 0.21 (0.14) and 0.39 (0.16) for 1 and 2 rounds of coarsening, respectively. For coarsening by population-weighted matching, the median (standard deviation) difference was 0.37 (0.26) and 0.46 (0.27) for 1 and 2 rounds of coarsening, respectively. Figure 2.5b displays the Euclidean distance between the returned cost vector and the optimal cost vector (i.e., $||\boldsymbol{\alpha}^h - \boldsymbol{\alpha}^*||_2$). For coarsening by random



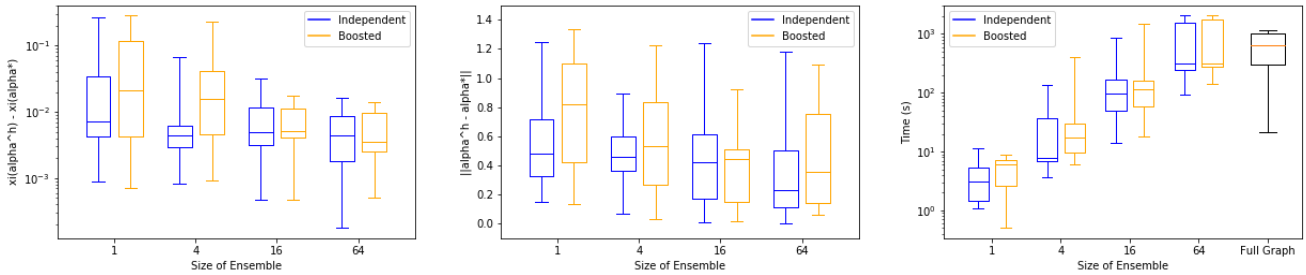
(a) (b)

Figure 2.5: A comparison of (a) the difference between retrieved and true optimality gap, and (b) the distance from the original objective for different numbers of coarsenings.

maximal matching, median (standard deviation) distance was 0.43 (0.31) and 0.75 (0.33) for 1 and 2 rounds of coarsening, respectively. For coarsening by population-weighted matching, median (standard deviation) distance was 0.57 (0.33) and 0.60 (0.43) for 1 and 2 rounds of coarsening, respectively. Overall, random maximal matching produced solutions that were closer to the optimal solution as compared to population-weighted matching.

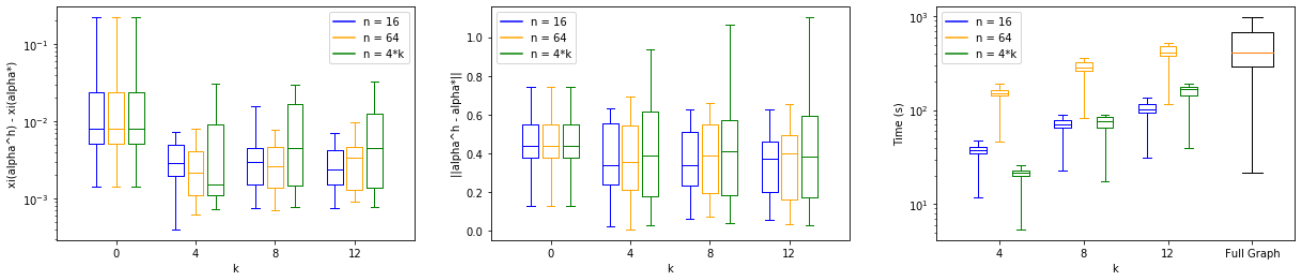
Figure 2.6a displays $\xi_{\text{ABS}}(\boldsymbol{\alpha}^h) - \xi_{\text{ABS}}(\boldsymbol{\alpha}^*)$. For unboosted ensembles, the median (standard deviation) difference was 0.0071 (0.075), 0.0045 (0.014), 0.0050 (0.0079), and 0.0045 (0.0046), for ensembles of size 1, 4, 16, and 64, respectively. For boosted ensembles, the median (standard deviation) difference was 0.021 (0.092), 0.016 (0.048), 0.0051 (0.0053), and 0.0036 (0.0043), for ensembles of size 1, 4, 16, and 64, respectively. Overall, an unboosted ensemble of size 64 was able to reduce the median and standard deviation of $\xi_{\text{ABS}}(\boldsymbol{\alpha}^h) - \xi_{\text{ABS}}(\boldsymbol{\alpha}^*)$ by 0.0026 (37%) and 0.704 (93%) over an ensemble of size 1, respectively. Figure 2.6b displays $||\boldsymbol{\alpha}^h - \boldsymbol{\alpha}^*||_2$. For unboosted ensembles, the median (standard deviation) distance was 0.48 (0.28), 0.46 (0.20), 0.42 (0.31), and 0.23(0.30), for ensembles of size 1, 4, 16, and 64, respectively. For boosted ensembles, the median (standard deviation) distance was 0.82 (0.36), 0.53 (0.36), 0.45 (0.26), and 0.35 (0.33), for ensembles of size 1, 4, 16, and 64, respectively. Overall, an unboosted ensemble of size 64 was able to reduce the median distance by 0.25 (52%) over an ensemble of size 1. Figure 2.6c displays the solution time; an unboosted ensemble of size 64 was able to reduce the median and standard deviation of the solution time of the full graph by 341s(52%) and -396s (-10%), respectively

Figure 2.7a displays $\xi_{\text{ABS}}(\boldsymbol{\alpha}^h) - \xi_{\text{ABS}}(\boldsymbol{\alpha}^*)$. For stochastic descent, the median (standard deviation) difference after 12 iterations was 0.0031 (0.0047), 0.0037 (0.0036), and 0.0056 (0.0083), for ensembles of size 16, 64, and increasing size, respectively. Figure 2.7b displays $||\boldsymbol{\alpha}^h - \boldsymbol{\alpha}^*||_2$. The median (standard deviation) difference after 12 iterations was 0.37 (0.18),



(a) (b) (c)

Figure 2.6: A comparison of (a) the difference between retrieved and true optimality gap, (b) the distance from the original objective, an (c) the solution time for various ensemble sizes.



(a) (b) (c)

Figure 2.7: A comparison of (a) the difference between retrieved and true optimality gap, (b) the distance from the original objective, (c) the running time.

0.40 (0.19), and 0.38 (0.31), for ensembles of size 16, 64, and increasing size, respectively. Figure 2.7c displays the solution time; 12 iterations of stochastic descent with an ensemble size of 64 was able to reduce the median and standard deviation solution time of the full graph by 55s (11%) and 211s (71%), respectively.

Overall, these results suggest that a multi-point formulation with an ensemble of 64 unboosted graphs coarsened by random maximal matching performs best in providing accurate approximations, while still providing decreased solution times.

2.7 Case Study: the State of Iowa

To demonstrate the application of inverse optimization to political gerrymandering, we applied our methods to the current congressional districts for the state of Iowa.

2.7.1 Context

Iowa's 2022 congressional districts were designed by an independent districting commission. The first set of districts submitted by the commission were rejected by the state legislature. A second set of districts returned by the commission was subsequently approved by the state legislature.

Congressional districting in Iowa is unique in that the state constitution requires that counties are not split by district borders. The Iowa constitution also provides specific guidelines on how district compactness is to be measured (Iowa Code section 42.2). Two possible metrics are defined; *length-width compactness*, which is calculated as the absolute value of the difference in a district's east-west distance and its north-south distance, and *perimeter compactness*, which is calculated as the perimeter of a district. We modify our FOP to measure compactness as the sum of the perimeter distance of each district, divided by the state perimeter times the number of districts (to scale the metric to the same range as the other sub-objectives). Perimeter compactness was chosen over length-width compactness as there exist numerous examples of possible districts with near-0 length-width compactness that by most common-sense approaches may be considered very non-compact, and while perimeter compactness may be susceptible to 'coastline paradoxes' that can make perimeter measurements not correspond to the broader shape of a district, the majority of Iowa's counties are rectilinear in their boundaries, making this issue mostly irrelevant.

Let σ_P denote the perimeter compactness, let q_{ij} be binary variable indicating if adjacent vertices i and j are in different districts, let b_{ij} represent the length of the border shared by counties i and j, and let M_p represent the perimeter distance of the state of Iowa. We represent the metric using the following constraints:

$$q_{ij} \geq x_{ki} - x_{kj}, \qquad \forall i, j, k \in V, b_{ij} \neq 0, \tag{2t}$$

$$\sigma_P = \frac{(\sum_{i,j \in V} q_{ij} b_{ij}) + M_p}{L M_p}, \tag{2u}$$

$$q_{ij} \in \{0, 1\}, \qquad \forall i, j \in V, b_{ij} \neq 0. \tag{2v}$$

2.7.2 Data

State data for Iowa districts and electoral data were obtained from the Metric Geometry and Gerrymandering Group and the ALARM Project (McCartan et al. 2022). The full-sized graph of Iowa comprises 99 counties, partitioned into 4 districts. Figure 2.8 shows the full-sized graph of the state. The first subfigure displays the current assignment of counties into 4 districts (4 colours), the second subfigure displays the initially proposed 4 distrcts,

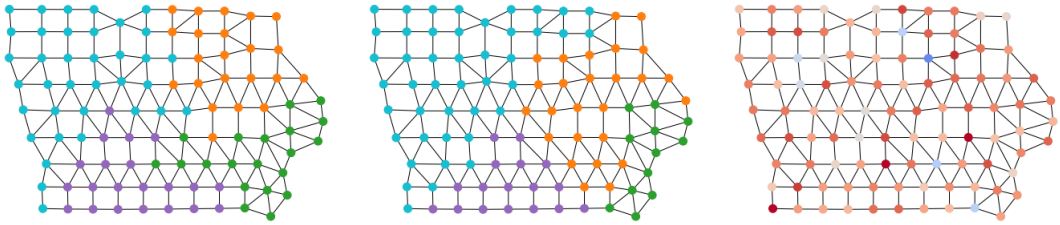


Figure 2.8: The full-sized graph of Iowa Counties, with color and shape denoting (a) the enacted 2022 districts, (b) the initially proposed and rejected 2022 districts, and (c) by relative partisan slant of the population in 2020 statewide elections

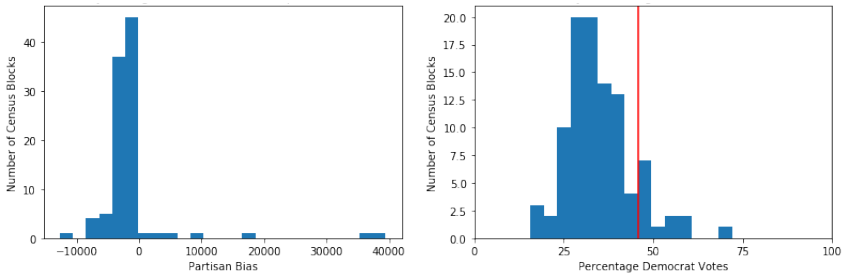


Figure 2.9: Distribution of (a) number of Democrat votes minus Republican votes and (b) percentage of votes towards Democrats by county

and the final subfigure displays the relative partisan slant of voters in 2020 statewide elections, with blue indicating more Democrats and red indicating more Republicans.

Figure 2.9 (a) and (b) display the distribution of the difference between Democrat and Republican votes across the Iowa counties in the 2020 presidential election (the positive side of the x-axis represents more Democrat votes) by absolute difference and per capita percentages, respectively. The vertical line in subfigure (b) indicates the partisan slant of the entire whole. Democratic votes make up 46% of all votes cast, but 93 out of the 99 of counties have republican majorities, with the democratic majorities being by larger margins in fewer counties. As a result the counties themselves display a noticeable efficiency gap, but rather than attributing this to gerrymandering of county borders, it may be possible that socio-geographical phenomena (e.g., urban/rural divides) make it such that Democratic and Republican populations are generally distributed as such in the Iowa political geography, regardless of borders. This illustrates the importance of understanding partisan fairness metrics such as efficiency gap in terms of what is actually feasible in the given political geography, and in relation to the existence of other competing metrics.

The sub-objective values of our inverse inputs are detailed in Table 2.1. Population data is drawn from the 2020 Census and political data from the 2020 presidential election. We note that in both districtings, the evaluation of the efficiency gap is in favor of Republican over-representation. We also supply district-by-district measurements of the metrics that

Metric	Rejected Value	Accepted Value
Perimeter Compactness (σ_P)	0.5773	0.6116
Population Imbalance (ρ)	$7.8674 * 10^{-5}$	$6.6137 * 10^{-5}$
Efficiency Gap (ϕ_{EG})	0.0882	0.4163

Table 2.1: Iowa Case Study Inverse Inputs

District	Population	Population Deviation	Perimeter	% Democrat
1	797,655	0.008%	503.03	54.6%
2	797,556	0.005%	786.27	44.9%
3	797,584	0.001%	515.54	49.9%
4	797,574	0.002%	967.18	34.9%

Table 2.2: Iowa Rejected Individual District Metrics

District	Population	Population Deviation	Perimeter	% Democrat
1	797,584	0.001%	696.48	49.0%
2	797,589	0.00037%	624.17	48.1%
3	797,551	0.0051%	619.72	49.8%
4	797,645	0.0066%	997.50	37.4%

Table 2.3: Iowa Accepted Individual District Metrics

are used to calculate inverse inputs in Tables 2.2 and 2.3 for the rejected and accepted districting plans, respectively. For context we note that the ideal district population is 797,592 and the Iowa state perimeter is 1151 miles.

2.7.3 Inverse Optimization Analysis

Although Iowa is computationally easier than other larger states, coarsening and ensemble methods are still needed for tractability. We use our heavy ball descent method, in conjunction with an ensemble of 64 graphs, each coarsened from the original state data three times. For our preliminary inverse analysis of the state, we produce an ensemble of 64 coarsenings of the state using 3 rounds of random maximal matching. Over this ensemble of coarsenings, we solve the MultiGIO$_{\text{abs}}$MinMax multipoint formulation, using PGD-A for finding the optimality gap minimizer with the same parameters described in Section 2.6.3.1.

We conduct two inverse optimization analyses for the accepted and rejected districtings. In Analysis 1, we minimize the optimality gap for a objective weighting that is a combination of minimizing perimeter compactness (σ_P), population imbalance (ρ), and efficiency gap (ϕ_{EG}). In Analysis 2, we reformuate the efficiency gap metric so that the objective component represents *maximizing* efficiency gap in favor of the Republican party (which

Metric	Analysis 1 Objectives	Analysis 2 Objectives
Perimeter Compactness	0.055	0
Population Imbalance	0.933	1
Efficiency Gap	0.011	0
Reported Optimality Gap	-0.0014	-0.0014

Table 2.4: Iowa 2022 Rejected Districts Inverse Output

Metric	Analysis 1	Analysis 2
Perimeter Compactness	0	0
Population Imbalance	1	1
Efficiency Gap	0	0
Reported Optimality Gap	0.0772	0.0042

Table 2.5: Iowa 2022 Accepted Districts Inverse Output

both districtings favor). Since the original efficiency gap objective is formulated as an absolute value of a quantity $\frac{\sum_{i\in V} w_i}{\sum_{i\in V}(p_i^D+p_i^R)}$ that is positive when it benefits Democrats and negative when it benefits Republicans (see equations (2p) and (2q)), we substitute ϕ_{EG} in the objective with this 'directional' quantity, and for our inverse input, the value of this quantity is equal to the negative efficiency gap. Minimizing this quantity is equivalent to maximizing the amount that the efficiency gap favors Republicans. We note that the linear dependence of the minimizing and maximizing ϕ_{EG} objectives makes it impossible for both subobjectives to be included in the same inverse analysis as the FOP objective feasible space might not have a fully dimensional feasible region. Thus by doing side-by-side analyses with both objectives and comparing the found optimality gaps, we can assess these objectives in comparison.

The results from both analyses are detailed in Tables 2.4 and 2.5 for the rejected and accepted districting plans, respectively. All analyses indicate that both districting plans place nearly all emphasis on minimizing population imbalance, with only the rejected districting placing some objective weight on minimizing perimeter compactness and partisan efficiency gap.

The interpretation of our inverse analyses of these two districtings is that the accepted districting prioritizes minimizing population imbalance with. This supports the hypothesis that the currently enacted Iowa districts are not intentionally gerrymandered to benefit the Republican party. However, the extra priority placed on minimizing the population imbalance between the rejected and accepted plans in order to reduce imbalance by 16% also increases to the perimeter and efficiency gap metrics by 6% and 372% respectively. Thus, this may still raise questions and discussion as to whether sole focus on reducing population imbalance is necessary when it entails such costs to other democratic values,

even if there is not explicit intent to gerrymander districts.

We note that alternative quantitative analysis methods in the literature often model equal population considerations in districtings through constraints that put an upper limit on the population imbalance in modeled districts; the ALARM Project (McCartan et al. 2022) simulates possible districts with an upper limit of 0.5% population imbalance, and Swamy et al. (2022) utilize an upper bound of 4.8% in analyzing possible district plans for Wisconsin. We note that our application of generalized inverse optimization methods allows for the addition of population imbalance as a metric to be considered as optimized in itself (whereas Swamy et al. (2022)'s method of plotting Pareto frontiers is not practical for considering more than two objectives at once). As such, our methods are able to bring new considerations that indicate that with this objective considered, Iowa's districts may not be the result of prioritizing Republican advantage, whereas the ALARM Project's simulation-based analysis that permits larger population imbalances indicates that the efficiency gap of the employed districting is anomalous to the point of being indicative of gerrymandering.

We note that the analysis of the rejected districting reports optimality gaps that are negative, which indicates that our inverse input objective values cannot be Pareto-dominated by any solution that is feasible for some element in our ensemble of coarsened graphs. Since the Iowa districting input is in fact a feasible districting, it should be possible that a more comprehensive inverse modeling analysis, either utilizing the full-sized graph of Iowa without any heuristic methods, or an analysis with larger ensembles and/or graphs that are coarsened to a lesser extent, could retrieve a solution with a non-negative reported optimality gap. However, if the Pareto frontier of the feasible region covered by our ensemble is reasonably similar to the geometry of the true forward problem Pareto frontier, then the results are still interpretable.

2.8 Conclusion

In summary, this paper makes three contributions. First, we propose a new approach for solving generalized mixed-integer IO problems based on sub-gradient methods. Second, we develop custom heuristic methods for graph-based inverse problems using a combination of graph coarsening and ensemble methods. Third, we propose a new application domain – quantitatively identifying gerrymandering – for generalized inverse integer optimization. We argue that IO can produce more nuanced data-driven arguments that proposed districtings should be considered gerrymandered.

Chapter 3

Using Inverse Optimization to Detect Biased Training Sets in Machine Learning Predictors

3.1 Abstract

Training dataset imbalance is a known issue that can cause unfair outcomes for machine learning predictors when applied across social groups, and can be a particularly difficult problem to address when proprietary models keep their datasets private. We apply generalized inverse optimization to the problem of inferring relative weighting and representation of social classes in the training set of optimization-based machine learning predictors that can be queried as black boxes. We apply a constrained generalized inverse optimization model such that inverse-optimal observations weights on a representative training set are explained by a linear expression of select observation attributes, yielding explanatory coefficients of how certain attributes impact representation, and outline a permutation test for evaluating the statistical significance of the inferred coefficient's difference from zero. We apply our methods in cases of simulated imbalance via undersampling on training sets for ridge regression, optimal trees, and support vector machines. When estimating the degree of underrepresentation, inverse optimization-inferred coefficients observe a correlation with the ground truth amount of bias, with a Pearson ρ coefficient of 0.35, and a corresponding p-value of 0.0002, with the best performing model type, Support Vector Classifiers, having $\rho = 0.68$ and a holm-adjusted $p = 0.003$. When utilizing hypothesis tests on inferred model parameters to diagnose the existence of bias, we find that classifying biased vs. unbiased cases outperforms baseline methods from the literature using

hypothesis tests on prediction residuals, achieving an improvement in AUC of 0.02 across all cases and an improvement of 0.13 in classification models. When utilizing a threshold of $p = 0.05$ to determine biased vs. unbiased cases, baseline methods achieve a negative Matthews Correlation Coefficient (MCC), while inverse optimization methods achieve an MCC of 0.28.

3.2 Introduction

There is much discussion in algorithmic fairness research about the potential for significant biases to be present in machine learning (ML) models trained on large datasets, in part due to biased training sets that misrepresent or under-represent certain features of the broad sample space to which the model may be applied (Crawford and Paglen 2021, Benjamin 2019, Noble 2018). This is a crucial area of further investigation for three key reasons: 1) models at risk of these biases are applied in settings that have substantial material impact on people's lives (Engel et al. 2024), 2) models are being applied to populations across the world (through broad application over the internet) despite being possibly trained on one local population not representative of the world, and 3) models are produced with little transparency regarding training data (either for data privacy reasons, protection of proprietary trade secrets, or maintaining a competitive edge in markets). There has been significant research on how notable large datasets that are used for image and facial recognition underrepresent various social groups, particularly racial groups, which can lead to inaccurate and biased predictions on those populations that cause negative material consequences. However, there remain algorithmic technologies in this field and others that maintain proprietary datasets where this cannot be directly verified (e.g., Northpointe COMPAS (Rudin et al. 2020)). We ask the question: what methods can be used for determining if a model was trained on a biased training dataset, without being able to directly observe the dataset, but instead by being able to query the model as a black-box?

Relevant areas where dataset bias evaluation is necessary include criminal justice applications where policing practices may bias crime data collection (Rudin et al. 2020), healthcare models that are transferred across facility locations (see Smith et al. (2023) for details on the spatial limits of transfer learning in facility-level healthcare operations), and models trained on web-scraped datasets where search engines introduce their own biases or online accessibility of relevant data from different demographics is imbalanced (Noble 2018). We also note that not all cases of training dataset bias are the result of prejudiced intent or negligence, but may also be the result of data drift. In these settings, our tools may be used to assess at what point a model may need to be retrained on a more recent

dataset when data drift has reached a significant extent.

When auditing black box models, most methods in the literature focus on identifying unfairness across groups in the outcomes of predictions. We note that in Pessach and Shmueli (2022)'s survey of fairness in machine learning, the listed measures of algorithmic bias do not correspond to measuring any of their listed causes of unfairness, which highlights the potential for imbalanced datasets to favor majority groups of underrepresented groups. While giving much insight into the disparate impact that an unfair model could inflict, these listed measurement methods do not reveal how the propagation of unfairness could be rectified at the (potential) root source by correcting imbalanced distributions of training data.

In contrast to existing measurement methods, our approach measures imbalance across groups in their representation in models' training data, rather than in their resulting predictions. This takes aim at identifying one potential root cause of unfairness. We present a method based on inverse optimization (IO) for estimating how a training dataset may be differently distributed than a known dataset, for the purpose of auditing bias in training data without direct access to the data itself. By taking a known dataset that has a socially accurate distribution with respect to minority classes subject to unfairness, generating predictions over it by the audited model, and using an inverse optimization model to determine observation weights in an inferred prediction loss function, one can estimate the relative weighting of various minority classes within the loss function, which indicates their relative prevalence in the training dataset. Our method can be used as a diagnostic tool for training bias, but also as a calibration tool for future model training. Given a model that is known to address class imbalance issues properly and avoid unfair predictions, our method of inverse analysis can be used to indicate how datasets can be properly reweighted for future model development.

3.2.1 Contributions

Our work has three main contributions.

1. We introduce a new application domain for inverse optimization; analyzing the demographic balance of training sets for machine learning predictors.

2. We introduce non-parametric uncertainty quantification techniques for parameters estimated by inverse optimization in machine learning bias assessment applications.

3. We contribute a new methodological approach to the literature of measuring unfairness in opaque machine learning models, which takes a very different perspective

than the existing literature. Our approach seeks to identify the root source of unfairness in a predictive model by way of training set imbalances, rather than identifying symptoms of unfair models by way of differential impact of model predictions. On experiments applying both approaches to models trained on datasets with simulated minority group undersampling, we find that inverse optimization approaches improve on existing approaches when identifying undersampled datasets. Furthermore, the estimated degree of imbalance has a significant positive correlation with the ground truth degree of imbalance, indicating how existing imbalances that lead to unfair performance can be rectified with reweighting adjustments.

3.3 Literature Review

Our work engages with two major streams of literature: inverse optimization (Section 3.3.1) and auditing machine learning models (Section 3.3.2).

3.3.1 Inverse Optimization

Inverse optimization is a modeling method that seeks to infer input parameters of a forward optimization problem (FOP) that render an observed solution optimal or close to optimal. Inverse optimization was originally proposed by Ahuja and Orlin (2001), who applied the method to linear FOP's. It has since been applied to a number of additional modeling contexts, including for conic FOPs (Iyengar and Kang 2005), general convex FOPs (Zhang and Xu 2010), purely integer FOPs (Schaefer 2009), and mixed-integer FOPs (Wang 2009, Bulut and Ralphs 2021, Smith and Boutilier 2024). See Chan et al. (2023) for a recent review.

In recent years, there have been several extensions to IO, often inspired by machine learning and data-driven optimization. Importantly, Chan et al. (2014) developed the generalized IO paradigm, where no assumptions are made about the optimality or feasibility of the observed solutions. Other extensions include IO with multiple observations (Keshavarz et al. 2011, Babier et al. 2021), IO with uncertain data (Ghobadi et al. 2018, Aswani et al. 2018), and goodness of fit measures for IO (Chan et al. 2019). Yousefi et al. (2024) present the first study into quantifying uncertainty in inverse optimization solutions with noisy data, producing confidence regions of inferred model parameters, utilizing a parametric model for the source of measurement noise.

We apply generalized inverse optimization approaches to FOP's that are used to train machine learning models, using subgradient methods to minimize an optimality gap function, introduced by Smith and Boutilier (2024). We additionally produce a method

for uncertainty quantification in our application domain that is non-parametric, utilizing permutation tests to test hypotheses that an inferred model parameter is significantly different from 0.

3.3.2 Auditing ML Models for Fairness

Addressing issues of fairness in machine learning is a topic of much study (Barocas et al. 2023, Mehrabi et al. 2021, Pessach and Shmueli 2022, Chouldechova and Roth 2018). In addition to study of methods for creating models with increased fairness, one area of study is methods for auditing existing models to detect unfairness.

Auditing fairness of ML models first relies on defining quantitative measures of *fairness*. Fairness as a concept has a large variety of potential understandings, quantifications, and operationalizations. Mulligan et al. (2019) present a survey of the diverse ways in which fairness is conceived and quantified in technical projects, including but not limited to machine learning contexts. Giovanola and Tiribelli (2023) investigate how fairness should be conceptualized in clinical machine learning contexts where medical ethics frameworks are also present.

To our knowledge, most existing methods in the literature for auditing opaque machine learning models in use for unfairness utilize a 'consequentialist' framework of identifying disparities in outcomes from model predictions. The Fairlearn Project provides comprehensive guidance in a number of ways that distributions of prediction outcomes may be analyzed to measure disparities in outcome across social groups that interface with a potentially unfair model (Weerts et al. 2023). Many metrics have been proposed as ways of measuring differential outcome across social groups for different contexts, including demographic parity (Dwork et al. 2012), equalized odds and equal opportunity (Hardt et al. 2016), and bounded group loss (Agarwal et al. 2019).

There has been much recent interest in the use of Shapley values as a means for increasing the explainability of machine learning predictors (Lundberg and Lee 2017). Shapley values provide a way to understand how the feature values of an observation contribute to a model's resulting prediction. In contexts of auditing fairness, this can be used to show how features susceptible to discrimination may be contributing to decisions made by the model, and particularly how features influence decisions differently for different groups of observations (Hickey et al. 2021).

While we do not necessarily disagree with applied consequentialist philosophies of unfairness on principle, we note that this framework diagnoses the symptoms of an unfair machine learning model rather than the causes, and as such, these diagnosis methods

may not provide much insight as to how a training set could be amended as to prevent the future propagation of unfair models. In contrast, our approach provides a direct method for diagnosing a root cause of unfairness with a quantitative measurement of underrepresentation of a given group in a training dataset, which allows for ways to fix a model that is determined to be unfair. Additionally, our method is capable of inferring biases along categories that are not directly used as features for model predictions, unlike Shapley value methods.

One additional relevant thread in the ML model auditing literature is research on dataset inference. Dataset inference research seeks to determine methods for auditing a black-box ML model in order to infer properties of its training dataset (Maini et al. 2021, Dziedzic et al. 2022, Maini et al. 2024). However, most academic literature aimed towards application in protecting intellectual property of datasets and detecting data theft, seeking to determine if a model developed by a competitor is trained on proprietary data owned by another party. Although our approach also seeks to infer properties of the training data for a black-box model, we are not interested in discerning whether specific observations are present in the model's training set, but rather aggregated properties about the distributions of demographic groups in the data.

3.4 Preliminaries

Our proposed approach to applying inverse optimization to a black-box model has the following framework: for a black-box model that we wish to audit, we query predictions from that model over a known dataset, and we utilize generalized inverse optimization to find a set of training weights for observations from our known dataset such that a model trained with such weights will yield predictions as close as possible to the predictions the black-box model created. We add constraints to our inverse optimization formulation such that inferred training observation weights must be expressed as a linear function of membership to some demographic class (as well as any other desired features of the dataset). Then, the linear function coefficient associated with any given class can be interpreted as the degree of over/under-representation of that class in the black-box model's training data relative to the known dataset it is applied to.

In order to solve the inverse optimization problem, we present calculations of sub-gradients for descent method approaches for three model types; ridge regression (Hoerl and Kennard 1970), optimal decision trees (Bertsimas and Dunn 2017), and soft margin support vector machines (Cortes and Vapnik 1995). We additionally propose methods for hypothesis tests of statistical significance of model fits and found coefficients utilizing

permutation methods. This allows us to make diagnoses of biased black box models with degrees of confidence. We will evaluate these methods on simulated examples of bias in a dataset through undersampling a minority class, and then compare our methods against a 'consequentialist' approach from the literature.

3.5 Methods

Let $\phi(\mathbf{x}, \mathcal{L})$ denote a machine learning predictor that predicts a target value $\hat{y}$ based on a feature vector $\mathbf{x}$ and a training dataset with n observations, $\mathcal{L} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2) \dots (\mathbf{x}_n, y_n)\}$. Suppose we are able to sample a probability distribution P that represents an unbiased depiction of the population of observations to which the model ϕ will be applied. We seek to evaluate the null hypothesis that $\mathcal{L}$ is drawn from P, without being able to directly observe $\mathcal{L}$, but instead by querying predictions from $\phi(\mathbf{x}, \mathcal{L})$ with our choice of $\mathbf{x}$. Furthermore, we seek a method that can indicate if the distribution from which $\mathcal{L}$ was drawn under-represents certain demographic groups with respect to P, and to what extent.

We propose a method of analysis in the case where ϕ is of a known model type (e.g. linear regression, decision tree, support vector machine), and the process of fitting ϕ to the training set $\mathcal{L}$ is done by the solution of a (potentially constrained) optimization problem where the objective function contains a linear expression of error terms for predictions on observations of $\mathcal{L}$. This category includes linear regression, soft-margin support vector machines, and optimal decision trees. Our proposed method is to sample a test set $\mathcal{R}$ with m observations from our unbiased distribution P, generate predictions using ϕ for the observations of $\mathcal{R}$, yielding the set of predictions $\hat{Y}_\mathcal{R} = \{\phi(\mathbf{x}_i, \mathcal{L}) \,|\, (\mathbf{x}_i, y_i) \in \mathcal{R}\}$, and utilize a generalized multi-objective inverse optimization model with input $\hat{Y}_\mathcal{R}$ to discern at which objective coefficients a model of the given type trained on $\mathcal{R}$ would most closely reproduce the predictions generated by $\phi(\mathbf{x}, \mathcal{L})$.

Let $\theta_\mathbf{c}(\mathbf{x}, \mathcal{R})$ denote an optimal machine learning predictor of the same type as ϕ that is fit by minimizing the objective function $\min \sum_{i \in \mathcal{R}} c_i f(\tilde{y}_i(\mathbf{c}), y_i)$ where $f(\tilde{y}(\mathbf{c}), y_i)$ denotes a some measurement of prediction error by the model's predicted target $\tilde{y}(\mathbf{c})$ with respect to the true target value y. Then let $\tilde{Y}_\mathcal{R}(\mathbf{c}) = \{\theta_\mathbf{c}(\mathbf{x}_i, \mathcal{R}) \,|\, (\mathbf{x}_i, y_i) \in \mathcal{R}\}$ denote the predictions of the observations of $\mathcal{R}$ made by said model θ. Finally, let $Y_\mathcal{R} = \{y_i \,|\, (\mathbf{x}_i, y_i) \in \mathcal{R}\}$ denote the true target values of the observations in $\mathcal{R}$. Our inverse optimization model with an input of $\hat{Y}_\mathcal{R}$ would then be formulated as such:

$$\min_{\mathbf{c}} \quad \ell(\hat{Y}_\mathcal{R}, \tilde{Y}_\mathcal{R}(\mathbf{c})) \qquad \text{(IO}(\hat{Y}_\mathcal{R}, \theta)\text{)}$$

$$\begin{aligned} \text{subject to} \quad & \mathbf{c}^\intercal \mathbf{1} = m, \\ & \mathbf{c} \geq \mathbf{0}. \end{aligned}$$

Where ℓ denotes some loss function that measures optimality loss between the predictions generated by $\theta_{\mathbf{c}}(\mathbf{x}, \mathcal{R})$ at objective $\mathbf{c}$ and the predictions generated by $\phi(\mathbf{x}, \mathcal{L})$. Popular loss functions in the generalized inverse optimization literature include; absolute sub-optimality ($\ell_{\text{abs}} = \sum_{i \in \mathcal{R}} c_i (f(\hat{y}_i, y_i) - f(\tilde{y}_i(\mathbf{c}), y_i))$), relative sub-optimality ($\ell_{\text{rel}} = \frac{\sum_{i \in \mathcal{R}} c_i f(\hat{y}_i, y_i)}{\sum_{i \in \mathcal{R}} c_i f(\tilde{y}_i(\mathbf{c}), y_i)}$), and distance between the optimal and input solutions ($\ell_{\text{dist}} = ||\hat{Y}_{\mathcal{R}} - \tilde{Y}_{\mathcal{R}}(\mathbf{c})||$) (Chan et al. 2023).

Normally when training a machine learning predictor on $\mathcal{R}$, a practitioner might attribute equal weight to each observation, but if a model "normally" trained on $\mathcal{R}$ performs substantially differently than ϕ then there is the possibility that ϕ places different degrees of importance on predictive error across certain demographics of the distribution, which could be caused by an imbalance in the training dataset $\mathcal{L}$. The inverse optimization analysis may yield specific insights here.

Specifically, if we suppose that $\mathcal{L}$ is in fact drawn from P (and the model type at hand does not overfit ϕ on $\mathcal{L}$), then we may expect the resulting inverse-optimal objective coefficients to be evenly distributed across the observations of $\mathcal{R}$, particularly with respect to any stratifications of the observations where one might expect to find bias. However, if $\mathcal{L}$ is drawn from a different distribution that under-respresents a certain demographic of observations with respect to their representation in P, then we would expect that observations in the underrepresented demographic would have smaller objective coefficients than the group average for observations. Additionally, we can use inverse optimization to learn how demographic features of an observation directly contribute to its relative training set weight.

We investigate demographic imbalances by way of a constrained inverse optimization formulation. Let $\mathcal{R}_M \subset \mathcal{R}$ denote a demographic subset of $\mathcal{R}$ that we suspect might be underrepresented in $\mathcal{L}$. Then, we may solve a constrained formulation of the inverse optimization problem where observation weights are explained by a linear function of various explanatory features, and a coefficient for a feature indicating membership to $\mathcal{R}_M$ that is significantly different from 0 (as evaluated by a hypothesis test) can indicate that there is under- or over- representation of class M in $\mathcal{L}$ with respect to P.

We present constraint structure that still allows for easy solution of the inverse model, and is particularly useful in practice for the context at hand. This constraint comes in the general form that $\mathbf{c}$ must be expressed as a linear combination of some set of basis

vectors. If, for example, these vectors describe the feature values of the observations in the analysis set, then the resulting inverse optimal weights may be explained by a linear combination a observations' features. For a simpler inverse model, if one is primarily curious in understanding the overall weighting of demographic classes that partition the data, one may use binary vectors that indicate membership to each class as the vector basis, resulting in the constraint that observations in the same demographic class must all have the same weight. We note that when two or more such basis vectors are used, the constrained inverse model can be formulated as a *multi-objective* generalized inverse optimization problem (Chan et al. 2014). Under these circumstances the set of feasible $\mathbf{c}$ is constrained to a linear subspace spanned by the rows of a matrix C. The constrained formulation of the generalized inverse optimization problem is presented below:

$$\begin{aligned} \min_{\mathbf{c},\boldsymbol{\alpha}} \quad & \ell(\hat{Y}_{\mathcal{R}}, \tilde{Y}_{\mathcal{R}}(\mathbf{c})) && (\text{CIO}(\hat{Y}_{\mathcal{R}}, \theta, C)) \\ \text{subject to} \quad & \mathbf{c}^\intercal \mathbf{1} = m, \\ & \boldsymbol{\alpha}^\intercal C = \mathbf{c} \\ & \mathbf{c} \geq \mathbf{0}. \end{aligned}$$

We note that if $C = I$, this formulation reduces to IO$(\hat{Y}_{\mathcal{R}}, \theta)$. If one desires to formulate an inverse problem such that observation weights are explained by some linear combination of their feature values, then one would choose $C = X^\intercal$. The resulting values of $\boldsymbol{\alpha}$ then indicate the coefficient of the linear relationship by which the features affect observation weights, allowing for easy interpretation similar to a conventional linear regression. We note that the above formulation differs from the multiobjective generalized inverse optimization formulation used in Smith and Boutilier (2024) because in this context of fitting observation weights for a machine learning predictor, it is more relevant to constrain the individual observation weights to be non-negative and have an average value of 1, rather than constraining the broader subobjectives as such. The reasoning for preserving the mean observation weight is so any additional penalty terms in the forward problem formulation (such as the L2 term in Ridge regression) maintain their scale relative to the error-minimizing terms of the objective.

We note that so long as we can partition or stratify $\mathcal{R}$ by demographic class membership, the inverse optimization analysis could potentially infer such imbalances in $\mathcal{L}$ regardless of such status being a feature used by the model ϕ or if the producers of the model ϕ have any such knowledge of such demographic information for their training dataset $\mathcal{L}$. We also note that this formulation allows for inferring representational imbalances in training datasets

that are proportional to continuous numerical attributes of observations, in addition to imbalances along categorical attributes.

3.6 Solution Methods

We first discuss solution methods for the unconstrained inverse optimization formulation IO($\hat{Y}_{\mathcal{R}}, \theta$), and then show how these methods may also be used to solve the constrained inverse optimization formulation CIO($\hat{Y}_{\mathcal{R}}, \theta, C$).

We note that for convex loss functions, so long as we can query the loss function gradient, we can converge towards loss function minimizers using gap-gradient methods (Smith and Boutilier 2024). In this section, we present the gradient calculations for the absolute sub-optimality loss function for various model types. We first show that for any machine learning model whose objective consists *solely* of minimizing weighted error terms and which can feasibly allow the same model parameters for any set of sample weights, that absolute suboptimality as a function of $\mathbf{c}$ is convex, which we show by providing a lower bounding plane of that is tangent at any value of $\mathbf{c}$.

Proposition 3.1. *Let $\theta_{\mathbf{c}}(\mathbf{x}, \mathcal{R})$ be a machine learning predictor which is fit with a forward optimization problem that minimizes a linear combination of error terms on training set observations, $\min \sum_{i \in \mathcal{R}} c_i f(\tilde{y}_i(\mathbf{c}), y_i)$. Furthermore, let the feasible region of the forward optimization problem be independent of $\mathbf{c}$. Then the absolute suboptimality loss function $\ell_{abs}(\mathbf{c}) = \sum_{i \in \mathcal{R}} c_i(f(\hat{y}_i, y_i) - f(\tilde{y}_i(\mathbf{c}), y_i))$ for the inverse optimization problem IO($\hat{Y}_{\mathcal{R}}, \theta$) is convex.*

Proof. We show convexity by showing that for any $\mathbf{c}^*$ we can derive a linear function $p(\mathbf{c})$ that lower bounds $\ell_{\text{abs}}(\mathbf{c})$ and is equal to $\ell_{\text{abs}}(\mathbf{c})$ at $\mathbf{c}^*$. Let the linear function in question be:

$$p(\mathbf{c}) = \sum_{i \in \mathcal{R}} c_i(f(\hat{y}_i, y_i) - f(\tilde{y}_i(\mathbf{c}^*), y_i))$$

We note that $p(\mathbf{c}^*) = \ell_{\text{abs}}(\mathbf{c}^*)$ by definition of p and ℓ_{abs}. Furthermore, because $\tilde{Y}_{\mathcal{R}}(\mathbf{c})$ is the set of feasible model predictions that minimizes $\sum_{i \in \mathcal{R}} c_i f(\tilde{y}_i, y_i)$, for any $\mathbf{c}$ we have that:

$$\sum_{i \in \mathcal{R}} c_i f(\tilde{y}_i(\mathbf{c}), y_i)) \leq \sum_{i \in \mathcal{R}} c_i f(\tilde{y}_i(\mathbf{c}^*), y_i))$$

$$\sum_{i \in \mathcal{R}} c_i(f(\hat{y}_i, y_i) - f(\tilde{y}_i(\mathbf{c}), y_i)) \geq \sum_{i \in \mathcal{R}} c_i(f(\hat{y}_i, y_i) - f(\tilde{y}_i(\mathbf{c}^*), y_i))$$

$$\ell_{\text{abs}}(\mathbf{c}) \geq p(\mathbf{c})$$

□

Such a lower bounding plane provides a subgradient for the loss function $\nabla \ell = (f(\hat{y}_i, y_i) - f(\tilde{y}_i(\mathbf{c}), y_i))$, which we can use to minimize the loss function with descent methods. From here, we focus on absolute suboptimality loss as our loss function of choice to make use of this results.

We note that while a similar result in Smith and Boutilier (2024) allows for finite time exact solution methods for inverse optimization over mixed integer linear programs, this is not generally the case for model training forward problems that have non-linear objectives, because there is not necessarily a finite number of optimal solutions to the model training forward problem, and thus not a finite maximum number of tangent planes to be discovered before the inverse optimization problem can be solved to optimality by solving a master problem.

We note that many machine learning models are trained to minimize a function not only of weighted predictions errors on the training set, but additionally some other term that is impacted by $\mathbf{c}$, such as a regularization term. Let $r(\mathbf{c})$ denote such a penalty term, so that the forward problem of training such a model $\theta_{\mathbf{c}}(\mathbf{x}, \mathcal{R})$ optimizes the objective $\min \sum_{i \in \mathcal{R}} (c_i f(\tilde{y}_i(\mathbf{c}), y_i)) + r(\mathbf{c})$. In these cases, while the true suboptimality loss function contains a constant penalty term for the black-box model ϕ which may not be known, we note the value of $\mathbf{c}$ that minimizes the absolute suboptimality loss function also minimizes the function $\ell(\mathbf{c}) = \sum_{i \in \mathcal{R}} (c_i (f(\hat{y}_i, y_i) - f(\tilde{y}_i(\mathbf{c}), y_i))) - r(\mathbf{c})$. In these cases, the above proof of convexity does not necessarily hold since the penalty term's impact on model fitting does not guarantee that $\sum_{i \in \mathcal{R}} c_i f(\tilde{y}_i(\mathbf{c}), y_i) + r(\mathbf{c}) \leq \sum_{i \in \mathcal{R}} c_i f(\tilde{y}_i(\mathbf{c}^*), y_i) + r(\mathbf{c}^*)$ for any given $\mathbf{c}$. However, at any $\mathbf{c}^*$, ℓ is still lower bounded by the potentially nonlinear function $p(c) = \sum_{i \in \mathcal{R}} c_i f(\tilde{y}_i(\mathbf{c}^*), y_i) + r(\mathbf{c})$, so if a gradient of $r(\mathbf{c})$ can be calculated, this still allows for calculation of descent directions of the loss function. In the following subsections we provide calculations for gradients of the absolute suboptimality loss function for machine learning models in this category where relevant. Some gradient expressions below are calculated courtesy of the MatrixCalculus project (Laue et al. 2018, 2020).

3.6.1 Linear Regression with Ridge Regularization

Linear regression models that minimize mean-squared error have a closed form for representing the optimal regression coefficients, and thus also the model predictions, given a weight vector $\mathbf{c}$. Let D denote the diagonal matrix with $\mathbf{c}$ as the diagonal. Then:

$$\tilde{\mathbf{y}}(\mathbf{c}) = X(X^\intercal DX)^{-1}X^\intercal D\mathbf{y}$$

The absolute sub-optimality loss function is then calculated as:

$$\ell = \mathbf{c}^\intercal((\hat{\mathbf{y}} - \mathbf{y})^2 - (X(X^\intercal DX)^{-1}X^\intercal D\mathbf{y} - \mathbf{y})^2)$$

Where the exponent ^{2} denotes a component-wise square of the elements of a vector. Taking the derivative of this with respect to c yields:

$$\begin{aligned}\nabla\ell =&(\hat{\mathbf{y}} - \mathbf{y})^2 - (X(X^\intercal DX)^{-1}X^\intercal D\mathbf{y} - \mathbf{y})^2 \\ &+ 2X(X^\intercal DX)^{-1}X^\intercal D(X(X^\intercal DX)^{-1}X^\intercal D\mathbf{y} - \mathbf{y})^2\end{aligned}$$

Adding L2 regularization with coefficient λ adds a penalty term to the absolute sub-optimality loss function (Hoerl and Kennard 1970). We note that for a given regularization coefficient λ, the optimal regression coefficients still have a closed form, $\beta = (X^\intercal DX + m\lambda\mathbf{I})^{-1}X^\intercal D\mathbf{y}$. This modifies our formulation of $\tilde{\mathbf{y}}(\mathbf{c})$, and introduces a penalty function $r(\mathbf{c}) = \lambda||(X^\intercal DX + m\lambda\mathbf{I})^{-1}X^\intercal D\mathbf{y}||_2^2$. This gives us the following loss function and loss function gradient:

$$\tilde{\mathbf{y}}(\mathbf{c}) = X(X^\intercal DX + m\lambda\mathbf{I})^{-1}X^\intercal D\mathbf{y}$$

$$\ell = \mathbf{c}^\intercal((\hat{\mathbf{y}} - \mathbf{y})^2 - (X(X^\intercal DX + m\lambda\mathbf{I})^{-1}X^\intercal D\mathbf{y} - \mathbf{y})^2) - \lambda||(X^\intercal DX + m\lambda\mathbf{I})^{-1}X^\intercal D\mathbf{y}||_2^2$$

$$\begin{aligned}\nabla\ell =&(\hat{\mathbf{y}} - \mathbf{y})^2 - (X(X^\intercal DX + m\lambda\mathbf{I})^{-1}X^\intercal D\mathbf{y} - \mathbf{y})^2 \\ &+ 2X(X^\intercal DX + m\lambda\mathbf{I})^{-1}X^\intercal D(X(X^\intercal DX + m\lambda\mathbf{I})^{-1}X^\intercal D\mathbf{y} - \mathbf{y})^2 \\ &- (2\lambda X(X^\intercal DX + m\lambda\mathbf{I})^{-1}(X^\intercal DX + m\lambda\mathbf{I})^{-1}X^\intercal D\mathbf{y}) \\ &\odot (\mathbf{y} - X(X^\intercal DX + m\lambda\mathbf{I})^{-1}X^\intercal D\mathbf{y})\end{aligned}$$

3.6.2 Optimal Trees

One common setup for optimal decision trees is classification trees that are trained to minimize the total proportion of incorrect classification on the training set. In this case, the FOP for training these trees is a Mixed Integer Linear Program (Bertsimas and Dunn 2017), and as such the subgradient of the absolute suboptimality loss function can be calculated as $\nabla\ell = (f(\hat{y}_i, y_i) - f(\tilde{y}_i, y_i))$.

There are other cases of optimal decision trees, such as regression trees that minimize the mean squared error of predictions, and classification trees that minimize Gini impurity of predicted probabilities of classification. We note that in these instances of optimal decision trees, while infinitesimal perturbations to $\mathbf{c}$ impact predictions made on observations, they do not alter the overall structure of the decision tree for some compact set of perturbations including $\mathbf{0}$; predictions are only altered by the re-weighting of observations within the leaf any one observation occupies. This simplifies the gradient calculation as we can assume that the attribution of observations to leaves is fixed at the point of calculating the gradient of the loss function for any given $\mathbf{c}$. Let A denote a binary $m \times m$ "leaf-adjacency" matrix that indicates if two observations are contained in the same leaf. Then, the predictions of an optimal tree trained with weights $\mathbf{c}$ can be expressed as:

$$\tilde{\mathbf{y}} = (\mathbf{c} \odot A\mathbf{y}) \oslash (A\mathbf{c})$$

For optimal regression trees that minimize mean-squared error, the absolute sub-optimality loss function is then calculated as:

$$\ell = \mathbf{c}^\intercal((\hat{\mathbf{y}} - \mathbf{y})^2 - ((\mathbf{c} \odot A\mathbf{y}) \oslash (A\mathbf{c}) - \mathbf{y})^2)$$

Where ^{2} denotes a component-wise square of the elements of a vector. Taking the derivative of this with respect to c yields:

$$\begin{aligned}\nabla\ell =&((\mathbf{c} \odot A\mathbf{y}) \oslash (A\mathbf{c}) - \mathbf{y})^2 - (\hat{\mathbf{y}} - \mathbf{y})^2 \\ &+ 2A(\mathbf{c}^2 \odot ((\mathbf{c} \odot A\mathbf{y}) \oslash (A\mathbf{c}) - \mathbf{y})) \odot (A\mathbf{y}) \oslash (A\mathbf{c})^2) \\ &- 2\mathbf{c} \odot ((\mathbf{c} \odot A\mathbf{y}) \oslash (A\mathbf{c}) - \mathbf{y})) \odot (A\mathbf{y}) \oslash (A\mathbf{c})\end{aligned}$$

Gini impurity for optimal classification trees can be equivalently calculated as the mean squared error of predicted probability if the classification label is interpreted as a regression target, so the above gradient calculation holds exactly for optimal classification trees that minimize Gini impurity as well.

We note that an optimal tree formulation is required in this method as the greedy splitting method found in more rudimentary implementations of CART can potentially render the loss function discontinuous with respect to $\mathbf{c}$. We also note that the gradient calculations presented above make no assumptions about the structure of the trees and leaves, beyond the optimality of the partitioning of observations into leaves. As such, the stated solution methods apply to optimal tree formulations that allow hyperplane splits in addition to conventional single-feature splits.

3.6.3 Support Vector Machines

Soft margin support vector machines (SVMs) are fit by minimizing a sum of prediction error on training set observations (with error functions including hinge loss, ramp loss, and epsilon-insensitive loss) and a term that maximizes the size of the margin. While most formulations of support vector machine models in the literature make use of a quadratic programming forward problem in order to maximize margin width by minimizing the 2-norm of a weight vector $\mathbf{w}$ in conjunction with a prediction error expression, there has also been research into alternative methods that instead minimize alternative norms of $\mathbf{w}$, which can result in linear program (LP) or mixed integer linear program (MILP) formulations (Rivas-Perea et al. 2012). This offers a substantial advantage in computing loss function gradients, seeing as infinitesimal perturbations in $\mathbf{c}$ do not impact the predictions made by an LP- or MILP-formulated model, due to the extreme point nature of their solutions.

In such a case we can simply compute the loss function subgradient as $\nabla\ell = (f(\hat{y}_i, y_i) - f(\tilde{y}_i(\mathbf{c}), y_i))$, where f denotes the prediction error function used in model fitting. Furthermore, in this case, gap-gradient methods from Smith and Boutilier (2024) may be used to find an exact loss function minimizer in finite iterations since the loss function is piecewise linear. Minimizing the loss function then requires the ability to query the value of this prediction error function from the black-box model under scrutiny. This would require stipulating some ϵ for epsilon-insenstive loss, or in hinge loss formulations for classifiers, reversing the Platt scaling of a predicted probability to get a position relative to the margin.

When the support vector machine has a quadratic objective (i.e. containing the 2-norm of $\mathbf{w}$ in its objective), the behavior of the objective penalty function $r(\mathbf{c})$ for the model $\theta_{\mathbf{c}}$, complicated the computation of the loss function gradient. However, in this case, the dual formulation of the support vector machine provides an alternative expression for the objective function of $\theta_{\mathbf{c}}$ which is more easily differentiable with respect to $\mathbf{c}$. In the case of a classification SVM with hinge loss function with penalty coefficient M, and an arbitrary kernel $K(\mathbf{x}_i, \mathbf{x}_k)$, the dual formulation is stated as follows:

$$\begin{aligned}
&\max_{\lambda} \sum_{i\in\mathcal{R}} \lambda_i - \frac{1}{2}\sum_{i\in\mathcal{R}}\sum_{k\in\mathcal{R}} \lambda_i\lambda_k y_i y_k K(\mathbf{x}_i, \mathbf{x}_k) \\
&\text{s.t.} \sum_{i\in\mathcal{R}} \lambda_i y_i = 0 \\
&\quad Mc_i \geq \lambda_i \geq 0 \qquad\qquad \forall i \in \mathcal{R}
\end{aligned}$$

We note that for an SVM fit by optimizing this model, infinitesimal perturbations to c_i impact the objective and decision variable values *only if* at its current solution $\lambda_i = Mc_i$. Thus for observations where $\lambda_i < Mc_i$, the partial derivative of the objective function with respect to c_i is 0. When $\lambda_i = Mc_i$ however, the partial derivative of the objective function for $\theta_{\mathbf{c}}$ with respect to c_i is expressed as:

$$\begin{aligned}
\frac{\partial \text{objval}}{\partial c_i} &= \frac{\partial}{\partial c_i}\sum_{j\in\mathcal{R}} \lambda_j - \frac{1}{2}\sum_{j\in\mathcal{R}}\sum_{k\in\mathcal{R}} \lambda_j\lambda_k y_j y_k K(x_j, x_k) \\
&= \frac{\partial}{\partial c_i}\lambda_i - \frac{1}{2}\sum_{k\in\mathcal{R}} \lambda_i\lambda_k y_i y_k K(x_i, x_k) \\
&= M - \frac{1}{2}M\sum_{k\in\mathcal{R}} \lambda_k y_i y_k K(x_i, x_k)
\end{aligned}$$

3.6.4 Getting Stable Solutions

We note that solving the inverse optimization problem of assigning weights to every observation in $\mathcal{R}$ to minimize the loss function likely entails fitting weight variables in a higher dimensional space than the space of model parameters, and as such there can be potentially infinitely many possible values of $\mathbf{c}$ that fit a model that minimizes the loss function. This can yield instability to the retrieved coefficients $\mathbf{c}$ and can encourage high variance and extreme values, when a less extreme and more interpretable solution to the inverse optimization problem also exists. As such one may run a risk of 'inverse overfitting' (Chan et al. 2019).

Using a constrained formulation such as CIO$(\hat{Y}_{\mathcal{R}}, \theta, C)$ is one way to reduce the likelihood of instability and 'inverse overfitting', as it directly reduces the dimensionality of the set of feasible observation weights. Since all of the newly imposed constraints in this scenario are linear equality constraints, the only required modification for solving this formulation with a gradient descent-based method is to project the $\mathbf{c}^{(k)}$ onto the intersection of the linear subspace spanned by the rows of C and the pre-existing simplex that

constitutes the domain of $\mathbf{c}$. In the implementation of solution methods utilizing projected subgradient methods, the projection onto the constrained space is achieved using Dykstra's projection algorithm (Boyle and Dykstra 1986), utilizing both a projection matrix to project onto the linear subspace spanned by the rows of C, and an algorithm for projection onto a scaled unit simplex developed by Blondel et al. (2014).

We propose two additional potential methods for adding stability to solutions to mitigate these risks for an unconstrained formulation:

Adding a secondary objective: In contrast to the generalized inverse optimization approach that seeks to find a loss function minimizer, *classical* inverse optimization usually seeks to find a weight vector $\mathbf{c}$ for which the optimality of the input $\hat{\mathbf{y}}$ with respect to the FOP is modeled as a constraint, and some auxiliary objective function is also optimized, often distance from some ideal weight vector. To borrow from this approach, we may solve a generalized inverse formulation to find the loss function minimum value, and then solve a second optimization problem that finds a value for $\mathbf{c}$ that yields the minimum loss function value, as a constraint, and optimizes an additional objective. Our proposed objective is $||\mathbf{c} - \mathbf{1}||_p$ for some p-norm. When $p = 2$, we specifically note that this objective minimizes the variance of entries of $\mathbf{c}$, and as such recommend this choice of p. This would likely require solution methods more sophisticated than gradient descent based methods.

Regularization: Instead of creating a primary and secondary objective as described above, we could instead create an objective that is the sum of these two separate objectives. Altering the descent process of optimization in this scenario is trivial, as all that needs to be done is to add the gradient of the regularization term to the overall gradient calculation at each step. This type of inverse optimization loss function additionally objective falls under the "Augmented Suboptimality Loss" function category defined by Scroccaro et al. (2023), and as such may allow for more sophisticated descent algorithms to minimize the loss function at a quicker convergence rate.

3.7 Model Selection, and Applications of Inverse Optimal Weights

When faced with possibilities of multiple formulations of the generalized inverse optimization model (whether from a variety of options for regularization or constraints on $\mathbf{c}$) we are tasked with a process of model selection and validation in order to arrive at a set of inverse observation weights that is most useful for a given purpose. We note that the criteria of selection may vary for different purposes of inverse optimization which we group under

two categories; (1) diagnoses of imbalances in the training set of the model being audited, and (2) calibrating objective weights for training further models. We suggest that the two categories listed below have model selection concerns that exhibit parallels with model selection concerns in explanatory and predictive modeling, respectively (Shmueli 2010). The fundamental concerns of model selection in general comprise of selecting a model that has a good fit of the observed data while also providing a parsimonious explanation.

To define the quality of an inverse model's fit for this model selection process, we develop an application-specific goodness-of-fit score that is based on Chan et al. (2019)'s *coefficient of complementarity* score ρ for generalized inverse optimization over linear programs, with our adaptation being specifically designed for the context of fitting sample weights for machine learning models. In addition to allowing for comparison of tradeoffs of closeness of fit of an inverse model versus the degree of parsimony/constraint, this also forms a test statistic for hypothesis tests evaluating statistical significance against a null hypothesis that the goodness of fit is achieved by an inverse model fit onto black-box prediction with errors distributed independently without biases.

Given a model that predicts a set of values $\hat{\mathbf{y}}$ and a set of true target values $\mathbf{y}$, an R^2 score measures the proportion of the variance in $\mathbf{y}$ that is explained by the model predictions $\hat{\mathbf{y}}$. For our inverse application to regression models, since we are fitting parameters for optimization model objectives that minimize a prediction loss function that is specific to the given model type, our score is stated in terms of this particular prediction loss function. Recall that $f(\hat{y}, y)$ describes the function of prediction loss between predicted target $\hat{y}$ and true target value y that is optimized as part of the objective for the machine learning model being analyzed. We assume that $f(\hat{y}, y) \geq 0$, where $f(\hat{y}, y) = 0$ if and only if $\hat{y} = y$. Then, our goodness-of-fit score is formulated as:

$$\rho_{\mathbf{c}} = 1 - \frac{\mathbf{c}^\intercal (f(\phi(X), \mathbf{y}) - f(\theta_{\mathbf{c}}(X), \mathbf{y}))}{\mathbf{1}^\intercal f(\phi(X), \mathbf{y})}$$

Thus, $\rho_{\mathbf{c}}$ measures the weighted proportion of the total prediction loss by ϕ over X that is explained by the prediction loss from a model $\theta_{\mathbf{c}}$ trained with sample weights $\mathbf{c}$. This metric is a direct adaptation of Chan et al. (2019)'s goodness-of-fit metric for the absolute suboptimality loss function, where the linear statement of the forward problem objective function in the numerator is substituted with the nonlinear loss function $f(\hat{y}, y)$, and the denominator is an equivalent formulation of the corresponding denominator expression, when the vectors $\mathbf{c}^i$ in their original formulation are taken to be the indicator vector for the i^{th} dimension (the denominator simplifies as such because for each i, $\mathbf{c}^{i\intercal} f(\theta_{\mathbf{c}^i}(X), \mathbf{y}) = 0$, as it is training and evaluating accuracy on a single observation, thus achieving zero loss).

Proof of optimality for inverse-optimal $\mathbf{c}$ is trivial since the numerator is the value optimized by the inverse optimization problem and the denominator is constant. We note that it is possible that $\rho_{\mathbf{c}} > 1$ in the case that $\mathbf{c}^\intercal f(\theta_{\mathbf{c}}(X), \mathbf{y}) > \mathbf{c}^\intercal f(\phi(X), \mathbf{y})$. This would suggest that the exact model parameters fit in ϕ are not feasibly attained by any model θ trained on X, which is possible if the model ϕ overfits onto specific observations of its training set $\mathcal{L}$ that are not present in the analysis set $\mathcal{R}$. We note that Chan et al.'s coefficient of complementarity measure for the absolute suboptimality case is also capable of exceeding 1 when the input solution is infeasible for the forward problem being modeled.

The availability of a goodness-of-fit measure such as $\rho_{\mathbf{c}}$, allows for the comparison of multiple models and their resultant optimal observation weights for their efficacy of explaining the behavior of the model ϕ. Of particular interest for diagnostic settings is comparison against the 'null hypothesis weights' $\mathbf{c} = \mathbf{1}$, which assumes no relative over- or under-representation of any groups of observations in the training set $\mathcal{L}$.

3.7.1 As a Diagnostic Tool (Auditing Existing Models)

When using generalized inverse optimization in the context of determining biases in training sets, the primary goal is to find a set of optimal weights for some inverse model such that a null hypothesis of an unbiased training set can be rejected with statistical significance and that the alternative hypothesis can be interpreted in a way that explains the degree to which a certain demographic group or group attribute is underrepresented in the training data.

Lacking knowledge of a parametric distribution of expected test statistics such as $\rho_{\mathbf{c}}$ or particular coefficients α_i under a null hypothesis, we can instead utilize non-parametric tests to estimate the statistical significance of models and model parameters. We propose two separate permutation test setups; one for measuring significance of an inverse model's fit as a whole, and one for estimating the significance of the coefficient α_i of a certain basis vector of C in explaining inverse optimal sample weights. Both tests are implementations of Fisher-Pitman Monte Carlo Permutation tests, using different test statistics to evaluate different null hypotheses (Berry et al. 2019).

Test 1 - Assessing overall model significance: Let $\mathbf{c}^*, \boldsymbol{\alpha}^*$ denote the optimal solution to CIO$(\hat{Y}_{\mathcal{R}}, \theta, C)$. We note that when $C = I$, this formulation is equivalent to IO$(\hat{Y}_{\mathcal{R}}, \theta)$, so the following test is relevant in unconstrained cases as well. Suppose that our loss function ℓ is defined to only involve $\hat{Y}_{\mathcal{R}}$ in the context of $f(\hat{Y}_{\mathcal{R}}, \mathbf{y})$ (we note that this includes absolute and relative sub-optimality loss, but not the distance between optimal and input solutions).

1. For each iteration in $1 \dots k$, we randomly permute $f(\phi(X), \mathbf{y})$

2. We solve an inverse formulation CIO($\bar{Y}_{\mathcal{R}}, \theta, C$) where $\bar{Y}_{\mathcal{R}}$ reproduces the permuted loss function values of $f(\phi(X), \mathbf{y})$, deriving an optimal solution $\bar{\mathbf{c}}^*, \bar{\boldsymbol{\alpha}}^*$. Let $\rho_{\bar{\mathbf{c}}}$ denote the goodness-of-fit measure for $\bar{\mathbf{c}}^*$, where $f(\phi(X), \mathbf{y})$ is permuted in the same way that derived $\bar{\mathbf{c}}^*$ when calculating $\rho_{\bar{\mathbf{c}}}$.

3. Let s denote the number of permutations in which $\rho_{\bar{\mathbf{c}}} \geq \rho_{\mathbf{c}^*}$

4. We return $\frac{s}{k}$ as our estimated p-value of the significance of fit of the inverse model CIO($\hat{Y}_{\mathcal{R}}, \theta, C$), with respect to rejecting the null hypothesis that the goodness-of-fit is achieved by the inverse model when there is no relationship between the basis vectors of C and the prediction loss of ϕ

Test 2 - Assessing significance of a coefficient for a basis vector of C: Let j indicate the index of the basis vector that is to be examined. Let $\mathbf{c}^*, \boldsymbol{\alpha}^*$ denote the optimal solution to CIO($\hat{Y}_{\mathcal{R}}, \theta, C$).

1. For each iteration in $1 \ldots k$, we randomly permute row j of C, to produce a modified constraint matrix $\bar{C}$

2. We fit the inverse model CIO($\hat{Y}_{\mathcal{R}}, \theta, \bar{C}$), deriving an optimal solution $\bar{\mathbf{c}}^*, \bar{\boldsymbol{\alpha}}^*$

3. Let s denote the number of permutations in which $|\bar{\boldsymbol{\alpha}}_j^*| \geq |\boldsymbol{\alpha}_j^*|$ for a two-tailed hypothesis test. For a one-tailed hypothesis test, we let s be the number of permutations where $\bar{\boldsymbol{\alpha}}_j^* \geq \boldsymbol{\alpha}_j^*$ or $\bar{\boldsymbol{\alpha}}_j^* \leq \boldsymbol{\alpha}_j^*$ for an upper- or lower-tailed test respectively.

4. We return $\frac{s}{k}$ as our estimated p-value of the significance of $\boldsymbol{\alpha}_j^*$, with respect to rejecting the null hypothesis that $\boldsymbol{\alpha}_j^* = 0$

Once an appropriate model has been selected that provides the best fit, depending on the model structure, there may exist various ways for interpreting the resultant model parameters/observation weights as indicators of training set under-representation. In the context of a model constrained by a basis matrix C, the resulting coefficients $\boldsymbol{\alpha}$ of each vector can be interpreted like the coefficients of an explanatory linear regression in terms of explaining how characteristics of observations contribute to their representation within the inverse-optimally fit training set weighting. We note that when comparing such a constrained inverse model to a null hypothesis $\mathbf{c} = \mathbf{1}$, it is recommended for one to ensure that $\mathbf{c} = \mathbf{1}$ is a feasible solution with respect to the basis matrix C so that the constrained inverse model will find a set of inverse weights that produces a goodness-of-fit at least as good as the null hypothesis. The easiest way to do this is to add a constant row to C to act as an "intercept term" in the explanatory linear regression interpretation.

3.7.2 As a Calibration Tool (Training Future Models)

In addition to diagnostic settings, inverse optimization also has a history of use in calibration settings where discerning how to weigh a tradeoff in a multi-objective setting is guided by observing inverse-optimal tradeoffs for previous decisions that are deemed successful. This is seen in the domain of radiation therapy treatment planning (Chan et al. 2014). In the setting of assessing dataset (im)balance for machine learning models, a similar use of inverse optimization could take the form of deciding how to reweight/undersample/oversample training data in a training set for a model that seeks to replicate fairness properties of an existing predictive model (training a new model with an expanded/updated/altogether new dataset). This application may be of particular interest in the domain of federated learning, where the balancing of heterogeneous local datasets' impact on the global model faces difficulty due to privacy of local datasets (Li et al. 2020).

We also posit that inverse optimization can be used in a similar sense for the purposes of dataset reduction, to find a sparse set of observation weights that reproduces predictions similar to the initial model, reducing the number of observations in the training set for potentially enhanced portability/computational ease. We note that absent any additional constraints or L2 regularization to the inverse optimization formulation, the scaled-unit-simplex structure of the feasible region of IO($\hat{Y}_{\mathcal{R}}, \theta$) makes sparse weights for the inverse optimal $\mathbf{c}$ likely, lending itself to this purpose.

We leave these calibration-oriented use cases for further development in future work.

3.8 Computational Tests

3.8.1 Experimental Setup

We apply these methods to a set of simulated instances of biased datasets to measure the efficacy of our methods, with the knowledge of a ground truth that is typically unobservable. We additionally compare the results of our method to a baseline method for auditing unfairness in the algorithmic fairness literature. The base dataset upon which we create simulated instance of bias is the UC Irvine Machine Learning Repository "Communities and Crime" Dataset (Redmond 2002). We create both regression and classifiation models to be trained and audited with this dataset; for regression models, the target is the $[0, 1]$-normalized violent crime rate of a given community, and for classification models the target is a binary value of whether or not the normalized crime rate is greater than or equal to 0.25, the 66^{th} percentile value for the crime rate variable. We simulate imbalance in the training data by undersampling a minority group to create datasets with differing degrees

of imbalance. Our minority group is observations of communities with greater than or equal to 20% black population.

See Appendix B.1 for further details on features used in the dataset. For the classification problems, we conduct the experiments denoted below with both optimal tree classifiers (OTC) and support vector classifiers (SVC), and for regression problems we utilize ridge regression, optimal tree regressors (OTR), and support vector regressors (SVR).
For each of these datasets, we test our inverse optimization-based bias detection method using the following experimental setup:

1. We partition a dataset into a learning set $\mathcal{L}$ and an analysis set $\mathcal{R}$ with a random 50-50 train-test split.

2. We create various copies of $\mathcal{L}$ that randomly undersample the minority group at a rate $b \in \{0\%, 10\%, 25\%, 50\%, 75\%, 90\%, 99\%\}$. These are notated by $\mathcal{L}_0 \dots \mathcal{L}_{99}$ respectively. The degree of undersampling done here represents a ground truth that we aim to recover in our simulated experiments, which is often not directly observable in real application.

3. For each manipulated learning set $\mathcal{L}_b$, we train a model $\phi(\mathbf{x}, \mathcal{L}_b)$. Hyperparameter values are determined by a grid search over $\mathcal{R}$ for $\phi(\mathbf{x}, \mathcal{L}_0)$. The specific hyperparameter values searched over for each model type are detailed in Appendix B.2.

4. We run a constrained inverse optimization model on $\phi(\mathbf{x}, \mathcal{L}_b)$ utilizing the analysis set $\mathcal{R}$, with the following model constraints:

   $C_1 = [1 \dots 1]$.

   $C_2 =$ the vector that details member of observations of $\mathcal{R}$ to the minority group prone to undersampling.

   $C_3 =$ the vector of crime rates for observations of $\mathcal{R}$ (n.b. this is the target for regression models, and determines the target for classification models).

   This models an explanatory linear regression for weights explained along the axis of injected bias, controlling for the impact of the observation crime rates. This yields a vector of inverse optimal coefficients α^*.

5. For each model, we use α_2^* as a measurement of the degree of underrepresentation of the minority class, and compare it to the relative difference in representation of the minority class in $\mathcal{L}_b$.

6. We evaluate the accuracy of the estimated imbalance amount by R^2 score. We note that there is no existing baseline technique in the literature for assessing this value to which we can compare our methods (the method of using a model to explain prediction error does not assess a degree of training set imbalance).

7. We additionally evaluate the estimated coefficients with Pearson's correlation coefficient ρ to determine correlations with the true amount of imbalance, and we use a matched-pair Friedman test to assess if the inferred coefficients belong to the same distribution across groupings according to the amount of sampling bias, pairing inverse optimization results within the same initial split (Friedman 1937). We follow the Friedman test with a post-hoc analysis using paired Wilcoxon signed-rank tests to assess the pairwise differences between groups, with the one tailed alternative that the group with larger sampling bias has more negative coefficients (Church and Wike 1979). We utilize a Holm adjustment for the pairwise tests to correct for bias introduced with multiple tests.

In addition to evaluating the ability of the inverse analysis method to accurately assess the *amount* of imbalance, we also assess the ability of permutation methods to reject the null hypothesis of no biasing where relevant, using following experimental procedure:

1. - 4. We repeat steps 1 - 4 from the above experiment.

5. For each model, we utilize a Monte-Carlo permutation test to evaluate the null hypothesis that $\alpha_2^* = 0$ (the coefficient associated with imbalanced representation along the discriminated axis), with a one-tailed alternative that $\alpha_2^* < 0$. We utilise a permutation test with 50 permutations to estimate a p-value p. We additionally estimate a second p-value p_M using a moment-approximation method (Berry et al. 2019), fitting a Pearson type III distribution to the mean, standard deviation, and skew of the Monte-Carlo distribution of test statistics, and calculating a lower-tailed p-value of the observed test statistic α_2^* relative to the fit distribution.

6. We utilize $1 - p$ (and $1 - p_M$) as an estimator of assessing that there is an underlying imbalance in the training data representation. We then evaluate our resulting predictions based on the ground truth of whether $\mathcal{R}$ and $\mathcal{L}_b$ have significantly different distributions of the discriminated feature. The ground truth labels for existing bias are determined by a χ^2 test of a contingency table of the discriminated feature between $\mathcal{L}_b$ and $\mathcal{R}$, tested at a significance level of 0.05. We evaluate predictions using an ROC curve and measuring the area under the curve (AUC), as well as producing

a confusion matrix at a classification threshold corresponding to a p-value of 0.05, and evaluating the Matthews Correlation Coefficient (MCC).

7. We compare the success of this method with existing methods in the literature that analyze prediction residuals as a means of determining bias. We produce two baseline methods for comparison.

   Baseline 1: We fit a linear regression explaining prediction square error (or Brier score for classification models) for the prediction of ϕ over $\mathcal{R}$, regressing over the same features as the constraint vectors for our inverse optimization approach, and calculating the upper-tailed p-value p_b for the coefficient of C_2, evaluating the significance that prediction error is greater for the minority group than the other observations.

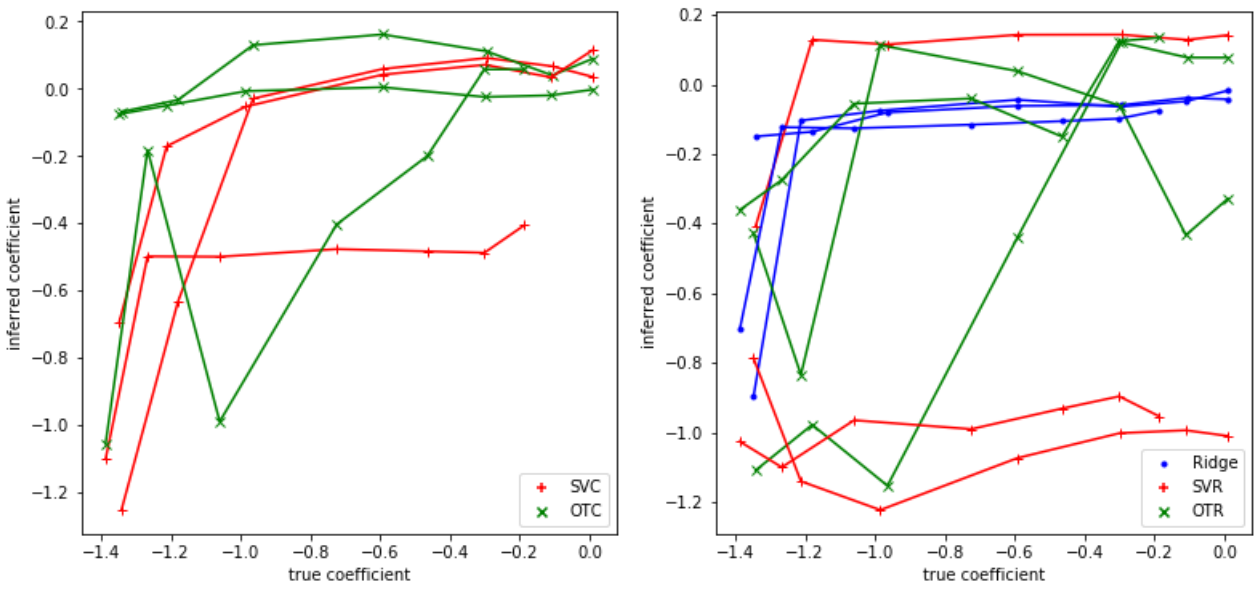
   Baseline 2: We conduct a t-test on the distribution of prediction square errors (or Brier score for classification models) across the undersampled and non-undersampled groups, utilizing a one-tailed test that the undersampled group has a higher mean square error.

   We note that for both methods the p-value assesses the strength of rejecting the null hypothesis that the *bounded group loss* definition of fairness is not violated for any value of ζ greater than the average prediction loss, with and without controlling for the impact of other variables (Agarwal et al. 2019). We use $1 - p_b$ as our baseline predictor of training data imbalance.

We perform the above experiments for 3 different random partitions into $\mathcal{L}$ and $\mathcal{R}$. For both of the above experiments, we implement the solution of the constrained inverse optimization problem using projected gradient descent with Nesterov acceleration (Nesterov 1983), with a maximum of 200 descent iterations. For optimal trees, we run 50 iterations of descent, due to the higher computational burden of training optimal trees. See Appendix B.3 for an outline of the solution algorithm.

3.8.2 Results

Figure 3.1 displays line plots of inferred magnitude of bias versus the ground truth value of undersampling bias, with separate lines for each random split and model type for classification models (a) and regression models (b). The R^2 score of the inferred magnitude of bias are -0.69 and -0.72 for classification and regression contexts respectively. The Pearson ρ coefficient between inferred coefficient and ground truth coefficient value is 0.396 across all models, with a corresponding p-value of $3 * 10^{-5}$. Table 3.1 shows Pearson ρ correlations for each model type with corresponding p-values, with and without Holm corrections.



(a) (b)

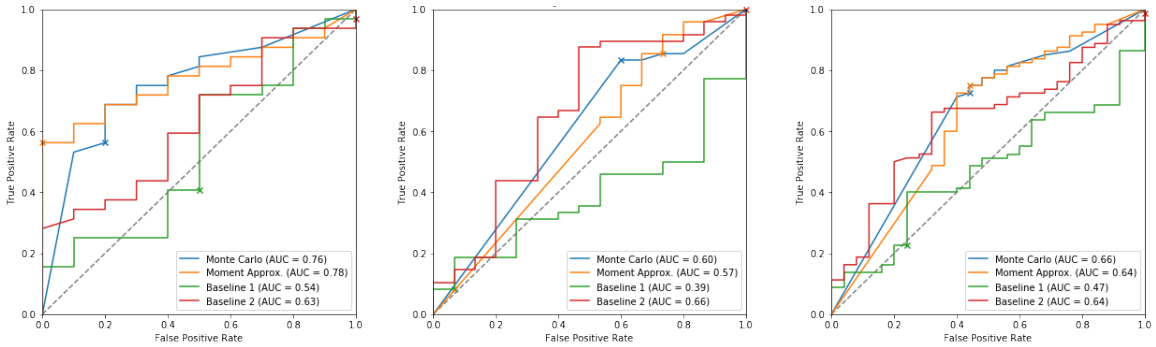
Figure 3.1: Ground truth imbalance coefficients vs. inferred coefficients for (a) classification models, and (b) regression models. Each line indicates experimental results from the same $\mathcal{L} - \mathcal{R}$ split, across different amounts of injected biasing.

Model	**Pearson ρ**	**Uncorrected p-value**	**Holm Corrected p-value**
SVC	0.68	*0.00068*	*0.0034*
OTC	0.45	*0.039*	0.118
Ridge	0.56	*0.0084*	*0.031*
SVR	0.16	0.48	0.48
OTR	0.56	*0.0077*	*0.031*

Table 3.1: Pearson correlation coefficients and corresponding p-values between inferred coefficients of imbalance and ground truth coefficients. Italicized entries indicate results significant to a level of $p < 0.05$

$b_i b_j$	**10%**	**25%**	**50%**	**75%**	**90%**	**99%**
0%	0.27	0.68	0.32	0.13	*0.0006*	*0.0099*
10%		0.68	0.68	0.29	*0.0012*	*0.012*
25%			0.42	0.16	*0.0041*	*0.0076*
50%				0.29	*0.040*	*0.040*
75%					0.42	*0.044*
90%						0.28

Table 3.2: p-values of Wilcoxon signed-rank tests for pairwise comparisons of inferred coefficients grouped by b, with the alternative hypothesis that the greater value of b has a more negative distribution. A Holm adjustment controls the family-wise error rate at 0.05. Italicized entries indicate results significant to a level of $p < 0.05$



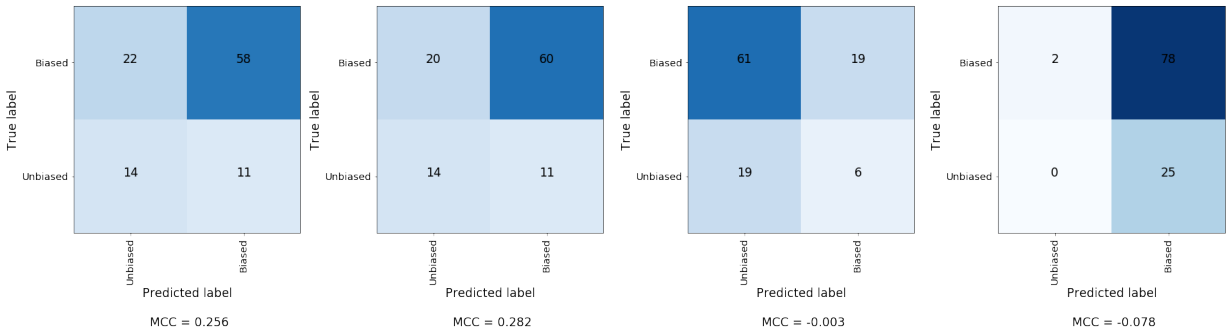
(a) (b) (c)

Figure 3.2: ROC curves for classifying imbalanced training sets using inferred coefficient p-values, for (a) classification models, (b) regression models, and (c) across all models. The x-mark on each curve corresponds to using a p-value of 0.05 as a classification threshold.

A matched-pair Friedman test at a significance level of 0.05 rejects the null hypothesis that inferred coefficient is independent of the group of undersampling treatment to $\mathcal{L}$, with a p-value of $2 * 10^{-8}$. Table 3.2 shows the p-values of paired Wilcoxon signed-rank tests of inferred coefficients for each pair of groups by value of b, with the alternative hypothesis that the group with a greater undersampling rate has a stochastically more negative coefficient, with a Holm adjustment applied to correct for bias introduced by multiple tests. We find that all pairs with $b_i \leq 50\%$ and $b_j \geq 90\%$ have significant p-values at a significance level of 0.05, as well as $b_i = 75\%$ and $b_j = 99\%$.

Figure 3.2 displays ROC curves when $1-p$ is used as a predictor of discernible undersampling bias in $\mathcal{L}_b$, comparing both the inverse optimization method with both Monte-Carlo estimations and moment approximation methods, as well as the baseline method. Diagnosing biased models using p-values produced by Monte Carlo methods achieve an area under the curve (AUC) of 0.66 overall, and 0.76 on classification models. Diagnosing by Monte Carlo p-values improves upon performance of the best baseline's AUC by 0.02 across all models, and by 0.13 across classification models. When using a p-value of 0.05 to discern imbalanced datasets, the both baseline methods have a higher false positive rate than its true positive rate, indicating performance worse than random guessing, and for diagnosing classification models, both the Monte Carlo and moment approximation approach produce classifiers that strictly dominate the first baseline method at the same threshold, with higher true positive rates and lower false positive rates.

Figure 3.3 displays confusion matrices when a threshold corresponding to a p-value of 0.05 is used for each method of classifying biased vs. unbiased model training sets, and the corresponding Matthews Correlation Coefficient (MCC) of classifications. Both baseline methods observe negative MCC's, while inverse optimization methods achieve MCC's



(a) (b) (c) (d)

Figure 3.3: Confusion matrices when classifying biased vs. unbiased models according to a threshold corresponding to a p-value of 0.05, and the corresponding Matthews Correlation Coefficient for (a) inverse optimization with Monte Carlo permutation methods, (b) inverse optimization with moment approximation methods, (c) baseline method 1, and (d) baseline method 2.

of 0.256 and 0.282 for Monte Carlo and moment approximation methods of estimating p-values respectively. We note that the second baseline method predicts low p-values across all test cases, with only 2 out of 105 cases having values above 0.05, both of which yield false negatives.

3.9 Discussion

In preliminary experiments, direct estimation of the degree of imbalance does not predict accurately. However, there is a significant positive correlation between inferred coefficient and ground truth coefficient, such that models with more training set bias are likely to have more negative coefficients. Additionally, when two models of the same type are compared using the same analysis set, the model finds significantly more negative coefficients for the models with more extreme undersampling. We know of no other methods in the literature for estimating these values that we can compare our methods against.

Classification of models with undersampled training sets outperforms consequentialist baseline methods using regression and hypothesis testing on prediction residuals. This is particularly effective for audited models that are classification models. We note that the best performing baseline method in terms of AUC, which utilizes a direct hypothesis test of prediction residuals across groups, requires thresholds on very low p-values to determine valuable classifications, and determining an appropriate threshold for the use of baseline 2's method without ground truth knowledge would likely be very difficult when applied in other model settings, whereas a threshold corresponding to the p-value

that actually determines the target performs very well for inverse optimization methods, yielding thresholds that are very effective points along the ROC curve.

3.9.1 Limitations

One potential limitation is that the current application of inverse optimization methods compares the out-of-sample prediction of ϕ to the in-sample predictions of θ to evaluate loss function. Thus the loss function being minimized contains an implied additional term for "model optimism" that could potentially vary with $\mathbf{c}$ and distract from the intended impact of minimizing the objective; finding observation weights such that the out-of-sample prediction behavior is (as close as possible to) optimal for the model training FOP. This could skew results of inverse optimization analysis, particularly for model types that are prone to overfitting.

One additional limitation is that the computational burden of minimizing the inverse optimization problem (and solving FOP's, especially for training optimal trees) restricts the feasible number of permutation iterations for hypothesis testing. This results in long line segment portions of the left end of the ROC curve for the Monte Carlo permutation methods, which could likely be improved with more permutation iterations to measure smaller non-zero p-value estimations. Designing and implementing more effective solution methods for the inverse optimization problem, as well as leveraging the parallelizability of permutation methods, can help with this issue.

3.9.2 Future Work Opportunities

Several opportunities for improvements and extensions of these methods are available for future work.

1. There is the opportunity to adapt the inverse optimization model to different loss functions. Different selections of loss function (e.g. relative suboptimality, distance to optimal solution) may result in better capabilities for accurately detecting imbalances and degrees of imbalance.

2. To address one point discussed in the Limitations section, the inverse optimization analysis method could potentially be modified in a way that compares test-set predictions to test-set predictions: this would entail splitting up the analysis set into training and testing subsets, and defining the loss function in terms of test set predictions, while fitting for the inverse optimal training set weights. This would require being

able to calculate the gradient of the loss function evaluated on test set as a function of weights on training set.

3. There is room for improvement in solution methods for the inverse problem. In diagnosing models for imbalanced trainings sets, improved solution time will allow more permutation iterations within a tractable time frame for more accurate hypothesis tests. Some machine learning learning methods that may be partially trained with online updates may be amenable to quicker solving if there is no need to fully retrain a model at every iteration of the solution method.

4. There are more machine learning models that are fit as the result of an optimization problem (e.g. logistic regression), so there remains work to study the implementation towards and efficacy of these methods on other such models.

5. In addition to estimating p-values for uncertainty quantification of inferred parameters, there is also the possibility of estimating confidence intervals of inferred parameters α. One possible method for doing so with non-parametric methods is to bootstrap the analysis set and obtain a distribution of coefficient values.

6. The current developed methods still require knowledge of model type and hyperparameter values to retrieve inverse optimal observation weights, meaning that the audited models are not completely "black boxes". There may be potential modifications to the inverse optimization model that allow inferring numerical hyperparameter values, thus not requiring prior knowledge.

3.10 Conclusion

We present a novel application of inverse optimization for detecting training dataset imbalances of machine learning predictors, when the datasets cannot be directly observed. Our methods contribute a new approach to the literature of auditing machine learning models for fairness.

Chapter 4

Preference Elicitation Algorithms, Power, and Sociotechnical Infrastructurings of the Preferring Subject

4.1 Abstract

As the deployment of preference elicitation technologies proliferates and enters innumerable realms of social life, there has been substantive development in the algorithmic and mathematical methods used. By bringing a critical eye to reading different structures within preference elicitation methods across deployment domains, informed by the frameworks of Karen Barad's *agential realism* and Susan Leigh Star's *sociotechnical infrastructuring*, I observe that different methods prefigure, necessitate, and enforce holding differing notions of agency, individuality, and identity in order to produce knowledge claims on observed subjects and/or the objects/fields over which they hold preference. I also observe that this that this progression is coextensive with the progression of power differentials that are also seen in Donna Haraway's theorized shift from 'Organics of Domination' to 'Informatics of Domination', and thus propose a new category, 'Latencies of Domination' that corresponds to the most recent methods in preference elicitation, matrix factorization methods. As the deployment of these various approaches to preference elicitation varies across fields, I pose that understanding the specific choice of methods deployed can be an important tool for understanding the underlying power relations that prefigure subjects as either individuals or members of masses, and that observation of this phenomenon in newly developing fields can open up critical opportunities for intervention in newly forming power structures.

4.2 Introduction: What is Meant by Preference Elicitation?

As sets and fields of decision encountered in one's life (such as those of consumption in markets) rapidly unfold, people are often posed as facing problems of navigating a large number of choices. The answer to how one ought to go about navigating them often becomes a scheme of defining what it is to have a preference. Thus, to be able to give shape and definition to preference would be to transform a set of options into a space, allowing for direction, orientation, and dimension. This provides one with ways of knowing both the field in which they are situated as well as ways of knowing themselves.

As the realm of choices, decision, actions, etc. grows beyond large to overwhelming, we have seen an increase in the deployments of tools that are designed to help one understand one's own preference, especially in a digital/algorithmic mode. Often conjoined with the delivery of decisions to be made (forming an ensemble known as a *recommender system*), the processes of preference elicitation can be as open to see as a solicited questionnaire with informative responses about how your choices impact the result, or they can be as covert as a sidebar of suggested content that acts on passively surveilled behavior.

4.3 Infrastructures of Intelligibility

Systems of production require infrastructures to stay alive. Not only is this true for infrastructure in the most common sense of the word in material production (a factory requires roads to receive raw material and ship out products), but social and discursive infrastructures are also required in production of knowledges and subjects. In the use of material and algorithmic apparatuses to render the world around us intelligible, conceptual schemas need to be applied in order to translate the material agencies that return to us through the technological apparatus into a human understanding. Infrastructuring thus includes the process by which discourses provide satisfactory alignments in these conceptual schema such that our actions, utterances, inscriptions, and constructions become temporarily useful enough (even if not perfectly translated) to each other, allowing agents to use them as boundary objects (see Star and Griesemer (1989)) to enroll each other in progressing on all sorts collaborative endeavors from business projects to basic communication. In this sense, Star and Ruhleder (1996) write,

> "We hold that infrastructure is a fundamentally relational concept. It becomes infrastructure in relation to organized practices. Within a given cultural context, the cook considers a water system a piece of working infrastructure integral to

> making dinner; for the city planner it becomes a variable in a complex equation. Thus we ask, when – not what – is an infrastructure."

Thus, the same technical object can experience different understandings as it traverses circles of differential expertise and lived experience. Additionally, the above quote suggests that infrastructuring is a never-ending, continuous process. This is reminiscent of Bruno Latour's tenet that there is "no group; only group formation" outlined in his reconstruction of Actor-Network Theory (Latour (2005): 27). Whereas Latour is referring to the creation of limning boundaries to construct a group of people as a coherent and distinct unit, we may also apply this mentality to the formation of a technical apparatus as an individual entity (that can have associative relationships with external entities), as opposed to an element or an ensemble. To elaborate upon this, I draw on Karen Barad's body of work described by the term "agential realism". Barad writes of technical apparatuses,

> "What precisely constitutes the limits of the apparatus that gives meaning to certain concepts at the exclusion of others?" (Barad 1998: 98)
>
> "The physical apparatus marks the conceptual subject-object distinction: the physical and conceptual apparatuses form a non-dualistic whole. That is, descriptive concepts obtain their meaning by reference to a particular physical apparatus which in turn marks the placement of a constructed cut between the "object" and the "agencies of observation"" (Barad 1998: 95)
>
> "Boundary-making practices, that is, discursive practices, are fully implicated in the dynamics of intra-activity through which phenomena come to matter." (Barad 2003: 822)
>
> "Phenomena are the effects of power-knowledge systems, of boundary drawing projects that make some identities/attributes intelligible, to the exclusion of others. The identities or attributes that are measured as part of knowledge projects do not represent inherent properties of subjects or objects. "Subjects" and "objects" do not preexist as such, but are constituted through and within particular practices. The objective referents for identities or attributes are the phenomena constituted through the intra-action of multiple regulatory apparatuses. Phenomena are inseparable from their apparatuses of bodily production." (Barad 1998: 106)

Taking these excerpts together, it is clear that for Barad the production of 'agential cuts' within phenomena into an apparatus and the entities it observes can be a result of

and a component of the infrastructuring that allows one to distinguish the apparatus as a whole, whose adopted 'usefulness' reinforces the created material-discursive practices and vice-versa. Component elements of such infrastructures can include making certain types of work to be more invisible than others, black-boxing, and producing grand narratives and imaginaries. (Star (1999): 384-385)

Currently there is increased interest in investigating infrastructuring in algorithmic information technology settings (including artificial intelligence and machine learning), particularly in circles centered around 'Fairness, Accountability, and Transparency' (FAccT), as a way of investigating how these specific values can be realized/embedded in the technologies they engage with, both through the material/digital agencies of the technical objects they engage with and through the agencies of the humans and non-human actants that create them, use them, describe them, and *are* described *by* them (Mulligan et al. (2019): 1-3).

I intend to investigate the literature surrounding the deployments of algorithmic technologies of preference elicitation to understand the ways that different parties see sets of values conceived of and promoted by the technology being deployed. These infrastructures are not only crucial to embedding such values in a technology, as is the concern of FAccT research collectives in information studies, but since preference elicitation apparatuses also require the production of agential cuts of entities of observation (both subjects from which it elicits, and objects about which the subject prefers), such infrastructuring practices also embed values in the people and fields subjected to preference elicitation. Thus to render such technologies intelligible is to allow them to give us knowledge of human subjects – both specific human subjects and humans more broadly – providing moments of subject formation consistent with Michel Foucault's notions of technology and subjectification. As such, the values I wish to examine are those of preference, individuality, and agency.

Particularly, I am interested in how these notions contribute to the infrastructuring of preference elicitation technologies as they move across settings that exhibit various power differentials. The two main settings I examine comparatively concern 1) the elicitation of the preferences of a public policy maker concerning ethical tradeoffs for use in aiding their decision making, and 2) the elicitation of music preferences by users on music streaming platforms, for the purposes of recommending further listening as well as targeting advertisements. As the setting of deployment moves from a highly exclusive setting (due to a scarcity of individuals at the sites of such positions of political power) towards one concerning masses of agents, the ways of seeing and knowing the preferring subject open up in a number of ways by allowing for algorithmic tools such as collaborative filtering. However, the presentations of and accesses to such knowledges are not necessarily spread

evenly among parties in the field being navigated.

As we trace the different methodologies of deployments seen, we note a correspondence to the chronological trend of technological innovation and coextensive social shifts that follows Haraway's theorized shift from "Organics of Domination" to "Informatics of Domination" in her *Cyborg Manifesto* (Haraway (2016): 28). However, emerging trends in preference elicitation technologies, such as latent factor methods, encourage the formulation of a new stage in the trend, which I notate the "Latencies of Domination", that is notable for the lack of internal components that are made to be humanly intelligible, thus not allowing for the creation of differential accesses to ways of knowing that can be productive of uneven power structures. However, I am quick to note that this progression does not deploy evenly across all domains. Notably, since each stage requires an increased scale in mass participation, the concentration of power within individuals requires restraining the scope of access and engagement, providing us with new ways to look at why certain domains experience certain deployments of preference elicitation technologies, and not others.

4.4 Content Based Recommendation: Concretizing the Individual as Site of Power via Ethical Substance

Vayanos et al.'s paper under review (as of Feb 2020) at Management Science, "Active Preference Elicitation via Adjustable Robust Optimization", details an algorithmic methodology for preference elicitation that is suitable for use by a single user. In addition to outlining a methodology, the paper also details a realm of application; namely, the production of policy by the Los Angeles Homeless Services Authority (LAHSA) concerning housing allocation programs designed to alleviate homelessness.

The use of preference elicitation here is motivated by the idea that once a preference is determined, optimization technologies can determine a policy choice that will maximize preference, making exploration of expansive Pareto frontiers (sets of outcomes that experience trade-offs of competing desirable qualities) simpler. In the deployment within the setting of policy formation for housing programs, policy choices describe sets of rules in the form of decision trees that determine for which programs a prospective applicant is eligible.

To make most effective use of its limited resources, LAHSA cannot accept every applicant into all of its programs. Thus a consistent set of rules is in need of creation to define eligibility for enrollment. These rules ultimately determine a specific population that enters each

housing program, and the broader impacts of these choices can be evaluated ethically by different yardsticks. In the paper, there are three such dimensions of comparison: *efficiency*, which describes the amount of the population that it is anticipated will successfully escape homelessness in the long term; *fairness*, which describes the amount to which access to the programs produces uneven outcomes across racial groups or other social stratifications of the homeless population; and *interpretability*, which describes how convoluted the rules tree is, such that the program is not too confusing to navigate and transparent to users. Each of these attributes is quantitatively measured (in a way such that a higher score is more conventionally desirable), and are not guaranteed to be simultaneously maximized by one policy choice. Thus in Vayanos et al.'s deployment, the preferring agent is assumed to possess a utility function that is a weighted sum of these constituent aspects, but with unknown weight coefficients that are to be discovered by the elicitation process.

With this general scheme in mind, the preference elicitation technologies deployed by Vayanos et al. pose the user with binary comparisons of options within the field of feasible policies; they must answer which of two choices they would prefer. With each choice the space of possible values for ethical weights gets smaller, as it outlaws possible combinations of weights that would have produced the alternative to the answer supplied by the user. Necessary to the legitimation of this process is the assumption that the preferring agent acts rationally within a scheme of preference described by the three properties observed, i.e. every preference of a comparison is only caused by the disparity in ethical properties that are given. While this is a component that allows the assumption that preferences are transitive, there is also another crucial factor that allows this assumption, which is an essential premise to knowing that the algorithm 'works'. On top of some basic assumptions of rationality of users, it must be assumed that the entire using entity is a rational individual. In other words, essential assumptions would be broken if the technology was interfacing with a group of policy-makers that answered every side-by-side comparison question with a majority vote, since transitivity is not guaranteed here even if individual actants are assumed to have rational preferences. Thus, in order for the actions of the user to be intelligible to the algorithm, and in turn for the outputs of the algorithm to be meaningful for the implementer, the using entity is made to be conceived as a rational individual.

Ultimately, the process does not end once the algorithm has decided it has elicited preference. Instead, the deduced preference (or range/space of credible preferences) is fed into a method to find the best choice of policy (within the quantitative space defined at the outset) for the user[1] . In fact, the authors note that part of the paper's key contribution is that they "study optimal approaches to preference elicitation that integrate the elicitation phase with the downstream recommendation" (Vayanos et al. (2020): 8). Here,

through engaging with the algorithmic technology, the user is presented with a reflection of themselves that encodes information about their stance on such ethical concerns as efficiency, fairness, and public interpretability, perhaps to be understood by the user either with specific reference to this application, but perhaps also as a grander statement about their outlook on ethics. It is unstated in the paper whether a user is also presented with the inferred weights (or range thereof) that define their preference, but there is no particular reason why this could not be the case.

We can see in this outcome the effects of a Foucauldian 'technology of the self', or technologies under the user's control by which they can achieve knowledge of the self, as a crucial form of the 'care of the self' (Foucault (1997): 228). Karakayali et al. summarize Foucault's notion of ethical self-formation as such,

> "Foucault (1990a, 1990b) qualifies practices of the self as ethical because they are utilized to give an 'ethos' – i.e. a 'character' or 'style' – to one's existence. More specifically, he conceptualizes technologies of the self as one of the four crucial dimensions of an ethical self-formation practice. Namely, a technology of the self is used by individuals: (i) to work on a particular aspect of their self ('ethical substance'); (ii) with a view to give this aspect an ultimate form ('telos'); and (iii) within the framework of historically variable 'modes of subjectivation', denoting whether such practices are carried out to conform to existing norms or to forge a 'new aesthetics of existence'." (Karakayali et al. (2017): 6)

Assuming that the LAHSA policy-maker is not powerless to change the policy once a recommendation (always framed as a recommendation and never as a resolution or an order) has been given, they have time to reflect on this assessment of themselves before proceeding to put a specific choice of policy into action. Perhaps the user will even go through the process of preference elicitation iteratively until they feel that they've displayed a preference that they are more satisfied with. This allows for a refinement of the self, producing the potential for the ethical drive of living one's life as a personal project (interestingly, this instance produces an ethics of developing one's ethics).

In a Baradian view, the material-discursive infrastructures required to render the technology intelligible enact an agential cut that fixes the objects of preference (policies) as agencies of observation in order to produce knowledges of a self that are indicative of the epistemic standpoint of a liberally constituted subject. To fix the objects of preference in order to do so also requires and enforces infrastructuring work that constructs definitions of efficiency, fairness, and interpretability from this standpoint. As Barad points out, every agential cut produces exclusions as it draws boundaries[2] , and notably excluded from a

technique in service of alleviating homelessness in LA, is the agencies of people experiencing homelessness themselves, reduced to simulated statistics of a policy outcome, and otherwise having no agency in the proposed policy-setting process.

4.5 Resonances and Dissonances in Musical Masses: Collaborative Filtering, Communities of Practice, and Redrawn Borderlands

One could imagine taking this general type of deployment described above and applying it to a broad range of domains, such as the setting of music consumption. Methods that show some approximate similarity to this method have been used for a long time in the field of music listening with little significant change, however predating quantitative algorithmic means. Most notably, this is seen in the process of using music genres as attributes of musical objects to identify a listener as having preference towards some genres over others. The discourses that infrastructure genre as a tool of organizing and knowing one's preference propagate across various sources from record stores, radio stations, music critics, to instrument manufacturers and music producers themselves.

This general ideology has been also imported into more algorithmic methods on music streaming services, notably Pandora Radio. Going even further than genre in descriptive level, Pandora radio employs its own system of tags that it denotes the "Music Genome Project" which, after fundamentally separating music into 7 fundamental music genotypes by genre[3] , uses 'expert' input to create granular tags on aspects of music, including lyrical themes, levels of guitar distortion, singer gender, etc. Songs can commonly possess a number of genes ranging from 150 to 500. Similarly, Spotify's corresponding music tagging system, called the "Echo Nest" utilizes computational acoustic analyses of sound files and scraped and processed text from online forum discussion about music to generate descriptive tags (Prey (2018): 1089-1090). Both of these companies claim that their methods' granular techniques allow them to capture more 'fundamental' and more meaningful attributes of a piece of music. Once again, one runs into issues of some epistemic standpoints overshadowing others, especially as ways of knowing a piece of music, particularly by genre, are contested by the parties that create, distribute and listen to such music[4] , often to the end of Baradian exclusions of the agencies of music producers.

Such methods as Pandora's method and the Vayanos et al. method, which rely on fixing essential attributes to the objects engaged with by a user (whether it be a policy plan or a song) such that the observed user can be defined in terms of its affinity for such attributes

in order to produce a recommendation, are generally known as *content-based filtering*. However, since music streaming platforms have access to a large mass of potential preference holders, other alternative ways to elicit preference reveal themselves. One such method is *collaborative filtering*. First coined in 1992, and originally deployed as a way to encounter the emerging problem of deciding what to prioritize reading in an overflowing email inbox, Goldberg et al. motivate their approach by claiming that "[c]ollaborative filtering simply means that people collaborate to help one another perform filtering by recording their reactions to documents they read" (Goldberg et al. (1992): 61). As such, collaborative filtering employs neighborhood-based methods, where variable values that describe the user's preference are not formulated in terms of observed interactions identifying essential attributes of the listener or of the songs, but rather the observations get interpreted as articulations of belonging to a group of users (hence, '*users like you* also purchased'). In such a spirit, a collaborative filtering system operates by calculating distance scores between pairs of users based on the similarity of their interactions with items they have both engaged with. From here, it constructs a neighborhood of the 'nearest neighbors' and recommends content that neighbors have overall had the most positive interactions (with which the user in question has not yet interacted) (Ning et al. (2015): 43-45).

By the nature of their construction, these methods need a multiplicity of users in order to generate recommendations. As such, it is perhaps not surprising that the beginning of their deployments in the 1990s coincides temporally with a trend Donna Haraway noted as an emerging trend in her 1985 *Cyborg Manifesto*, of a shift from "Organics of Domination" to "Informatics of Domination". Furthermore, for Haraway such a trend coincides with a shift towards constructions of the political individual embracing partiality and fracturedness in their identities (Haraway (2016): 21). This unsurprisingly aligns with additional possibilities of the formations of preferring subjects by these new methods.

In comparison and contrast to the content-based methods, a Baradian reading of the agential cuts of collaborative filtering methods fix content as an agency of observation, but now without fixing any essential categories onto them. As well, users (listeners) are still made to be observed entities, but rather than determining categorical dimensions of an individual's preference, the object of focus is the formations of groups of listeners, articulated by shared behaviors around specific tools of the apparatus, and individuals are described by their valences to various groups. Exclusions generated here may be on the side of the agencies exhibited by the objects fixed into the apparatus, or the agencies of association of the producers of said objects (there may be reasons to think that songs form groups just as much as listeners do, e.g. because they were recorded by the same artist).

How does this impact the experience of the music listener? Does the user still expe-

rience Foucauldian moments of subjectivation and accretions of ethical substance? Yes, as recommendations more or less take the same form from the perspective of the listener. ethnographic study of users of Last.FM (a music recommendation platform that uses collaborative filtering) by Karakayali et al. (2017) shows how engagement with such recommender systems transforms every moment of listening into an act of discovering and defining the self as a listener, producing drives to refine how one conceives of their preference of music (often with an implicit imperative to diversify their listening), elevating it to the realm of taste. In the unfolding of this scheme, one is not generally directed to other users that are identified as being in one's neighborhood, at most what one gets is a vague allusion to 'users like you' when presented with a content recommendation. Thus, for the music streamer, reflections of the self are returned from a black box, which enables their own readings of the ways to know the self from the encoded signals they receive back. The messages received back are in a structure that needs to be decoded if knowledge of the self is to be obtained, but without any institutionally prescribed key for interpretation (like one might get if their recommended songs are shown to have the tag "EDM").

However, the way that listeners receive knowledge of themselves is not the only way that preference elicitation algorithms of larger scale produce knowledges of subjects, especially if one has the ability to 'open the black box' of the internal mechanisms of the algorithm. With the advent of larger observable masses, it is no longer necessary to exclusively privilege the individual user/human as the unit/scale of holding preference. How this is possible becomes clear as we dig further into the implications of the neighborhoods that are constructed in the preference elicitation process.

Since the apparatus of collaborative filtering positions songs as agencies of observation through such an agential cut without fixing inherent properties onto them, this method can be deployed usefully with access to a larger population of subjects because a larger population can be recognized in how its constituents come to different ways of knowing the same objects, and thus indicating preference and taste in different ways that may in fact be incommensurable (especially with respect to a genome of universally prescribed attributes). In fact, awareness of this differential experiencing and coming to know the agencies of observation by subjects is necessary for identifying the discursive formation of user groups and neighborhoods as an instantiation of "communities of practice" in the sense deployed by Susan Leigh Star. For Star, the formation of communities of practice and the enrollment of individuals into them is facilitated by the naturalization of objects, or the paradigmatic adoption of ways of knowing them. Star writes:

> "I mean "object" to include [...] stuff and things, tools and technologies, and ideas, stories, and memories – those objects which are treated as things by

> community members. They are used in the service of an action, and mediate it in some way. Something actually becomes an object only in the context and action and use – it is then as well something that has force to mediate subsequent action.
>
> "A community of practice is defined in large part according to the co-use of such objects, since all practice is so mediated. The relationship of a newcomer to the community largely revolves around the nature of the relationship with the objects – and not, counterintuitively, directly with the people. Acceptance or legitimacy derives from the familiarity of action mediated by "member" objects.
>
> "But "familiarity" is a fairly sloppy word. I mean it here not instrumentally, as in proficiency, but relationally, as a measure of taken-for-grantedness. [...] A better way to describe the trajectory of an object in a community is as one of naturalization. By naturalization, I mean stripping away the contingencies of an object's creation and its situated nature. A naturalized object has lost its aura of anthropological strangeness, and is in a sense "de-situated" in that members have forgotten the local nature of the object's meaning.
>
> "Objects exist, with respect to a community of practice, along a trajectory of naturalization, which has elements of both ambiguity and duration. It is not predetermined whether an object will ever become naturalized or how long it will remain so – rather, practice/activity is required to make it so and keep it so. The more naturalized an object becomes, the more unquestioning the relationship of community to it; the more invisible the contingent and historical circumstances of its birth; the more it sinks into the community's infrastructure." (Star (2015): 153-154)

Here we can see that the continual activity of naturalizing objects, within a community that is also constantly reconstructing and re-outlining itself through the same activity, fully implicates the work of infrastructuring intelligibility in community formation and membership. With this formulation of communities of practice in mind, we can see how collaborative filtering technologies seek to detect such communities of practice within which a user's consumption choices are naturalized, without necessarily needing to know or encode what those naturalized knowledges are. The consequence of the resulting recommendation is to provide the user with more objects that are naturalized within the same community, and potentially pull them further into the center of such a community (although often without giving the user conscious connections to other people in the same community). Implicit in the notion that "you would like this object because these users

liked it" is that "you will experience/understand this object in the same way these users do".

However, objects can be differently naturalized by different communities of practice, as seen in the initial Star quote about the infrastructuring of water systems. At this point, for such communities to meaningfully interact and enroll each other, objects take on the role of *boundary objects*. Furthermore, Star is also quick to observe that people do not exist in a single community either, but are often members of several communities of practice at once, citing Gloria Anzaldua's theorizations of *borderlands*, and Donna Haraway's theorizations of *cyborgs* and *monsters* (Anzaldúa et al. (1987), Haraway (2016)).

Seeing as membership of a group is brought about by its continuing enactment, the lived experience of existing in multiple communities can result in a fluctuation of imbalanced performance of different community memberships across time and contexts. As such, recommender systems that utilize collaborative filtering can target such moments of temporarily visible identification as a way of targeting the communities of practice themselves through the *dividual* people that inhabit it, rather than *individual* subjects. Since a person's valence to certain communities and of practice can shift over time and place within the course of a day, this opens opportunities for preference elicitation algorithms to make observations and ensuing recommendations that are focused not at the site of a user, but at the site of a specific moment for the user, a process known as *context-based recommendation*.

Thus, whereas the application of preference elicitations in policy-setting contexts necessarily marks a sharp border around the individual as a unit, we see different leveraging of the conception of communities of practice and articulations of membership thereof marked by agential cuts on certain music streaming platforms such as Spotify. As Prey (2018) details, Spotify is not always interested in learning a better understanding of person X's listening preferences, but rather the listening preferences of a certain 'way of life' that people can move in and out of in their day-to-day lived experience, such as the "glow gal" or the "last train sprinter" (Prey (2018): 1094). Doing this allows such services to not only pinpoint individual listeners/accounts as the site of opportunity, but also listener-spatial-temporal intersections where/when a specific targeted ad is most likely to be received in an impactful way. Spotify as a platform leverages this to create what it calls "Branded Moments".

In other words, these deployments operate by conceptually dissolving the spatial-bodily (or online, account login) boundaries that divide the consuming mass into conventional individual subjects, and then recarving the larger body along an orthogonal set of lines that are intelligible as denoting 'ways of life' as desiring entities that can fluctuatingly enter

into and exit individual human bodies. As such, user accounts (and the humans behind them) not only become *dividuals* rather than *individuals* as discussed by Prey (2018) citing Deleuze (1992), but also *substrates* rather than *subjects*. For advertisers, an ad targeted at a conventional individual is often less effective at eliciting the desired response than an ad that is (accurately) targeted at a way of life. However, locating these opportunities for an ad to directly confront an instance of the desired way of life can be difficult because they can come in and out of human bodies in brief moments. Thus by identifying the patterns by which a 'way-of-life' that inhabits listeners (inside and outside of their virtual accounts) exerts agency in the momentary expression of music listening preference, Spotify can utilize the same usage data it uses to recommend track to partially surveil the transit of its constructed ways-of-life across masses and engage with it as it appears.

In one way, this allows the user to be recognized and observed as a site of malleable "becomings", but necessarily by way of temporarily fixing other entities (ways-of-life) as static "beings". Of course, the goal of fixing these ways in substance is to locate them in space, and by locating them, platforms seek to change them. For example, to give (sell) Bacardi access to speak directly to the 'glow gal' community of practice is to attempt to transform what it means to be a 'glow gal', presumably by (further) naturalizing Bacardi rum as an object to said community of practice. Of course the 'glow gal' does not only exist where the consuming mass touches the digital space, but also in the extra-digital world, e.g. in the dorm rooms of 'pre-gaming' college students, but critically to Spotify, in such rooms where bluetooth speaker is connected to a phone with Spotify open and playing.

4.6 Illustrative Example: Auto-theory Case Study and Google Ad Preferences

To further illustrate the points of the above section, I present an overview of some preference data that has been gathered about myself, the author, across parallel virtual locales. I have three different email addresses that I actively use that are connected to Google's G-suite. One for my current institution of my graduate work, one from my undergraduate days (that are still connected to certain online accounts and mailing lists so it's still useful to me), and one institutionally unaffiliated personal email that predates both. One feature of G-suite's algorithmic production of searcher/surfer/consumer subjects is that after some digging in the settings structure, one can find a list of the intelligible tags that the Google ad personalization has ascribed to their account. When looking at these for my Google accounts, the length of the lists of tags is roughly proportional to the amount of time I've

had the emails, indicating a continuous process of learning/classifying elicited by the algorithms in play.

Across my three accounts, the tags range and directly conflict each other. Two accounts assume I work for a "small employer (1-249 employees)" while one assumes I work for a "very large employer (10k+ employees)". Arguably both of these are 'correct' if they're variably tracking my career as a freelance musician and my employment as a graduate student assistant. My assumed household incomes include "lower middle," "upper middle," and "high". One account pins me as being a homeowner while two see me as a renter. All three identify me as "not a parent". All three think I speak English although one thinks that I also speak Norwegian (I wish I could). When I checked these in the summer of 2021, one had me pinned as a middle-aged woman, although as of winter 2022, all three list me as male, either in the 18-24 years range, or the broader 18-34 years range (generally I use he/they pronouns, although in some contexts I will drop the 'they' to avoid inconvenient encounters, demonstrating an acknowledged flexibility in my gendered ways-of-life). One thinks I work in hospitality, while the other two do not venture to guess.

Evidently my digital (and perhaps also extra-digital) ways-of-life somehow vary in correlation with which of three side-by-side Google Chrome buttons I click to enter the internet. At the same time, the ways in which the reflection back to me is more directly able to educate me about myself is frequently deflected – I rigorously employ ad-blocking tools on as much of my internet browsing activity as possible (which is still not perfect; no matter what advertisements will still somehow seep through the cracks). However, the surveillance and fitting of myself into these categories is not wasted in the eyes of Google; observations of my preference will still be helpful to them in producing trend-based understandings of what it means to exhibit a certain label. Evidently, from me, Google extracts information on what it means to be high income, upper middle income, and lower middle income all at nearly the same time.

As the multiplicity of email addresses propagates in the lives of many, and as the use of such addresses gets tied into aspects of one's daily life beyond their origin from work or school (this is certainly the case for the university email address, which for many becomes a main point of contact well beyond coursework, as universities intervene in people's housing, social groups, healthcare, food security, etc.), this process will produce less obvious and perhaps more interesting divisions into people's digital and extra-digital lives (or ways-of-life).

4.7 Latent Factor Methods, and Emergence of a Post-Intelligible Apparatus

Emerging methods in preference elicitation within the context of mass platforms of content consumption go beyond collaborative filtering methods. *Latent factor methods*, or *matrix factorization methods*[5] are broadly described as a method where, given a set of data about user-content interactions, and little other contextual information, and fit users and content optimally into a multidimensional 'latent space' formulated such that the inner product of the latent dimension vectors of two parties that have previously interacted is as close to estimating a quantified score of the the affinity of their interaction (roughly this can be lay-interpreted such that the co-members of an observed positive intra-action are situated closely in the space, while negative interactions yield positionings far apart in the space). From this, any recommendations to be produced are generated by identifying unexplored interactions between entities that would also have high-scoring inner products (and thus are nearby in the generated space) (Ning et al. (2015): 66-67). In fact, this allows us to consider the potential for preference or similarity relations between any two entities observed by the system, including two users that have never interacted with the same content, or two pieces of content. The totally symmetric nature of the algorithmic construction does not prioritize, or even necessarily distinguish users from content. The only given way to conceive the structure of preference beyond scoring a given pairing of entities is by examining each constituent dimension of the created space. However, the only information these yield is numerical values for users and content (which might be otherwise unlabeled) regarding their positions along each dimension; there is no deeper meaning in the sense that one could interpret dimensions in an earlier method like "efficiency" and "preference for efficiency" in the LAHSA example.

At this point, to find these algorithms to be useful, we are not obligated to take up infrastructural narratives that the grounds upon which associations and relations are based in inherent attributes with coherent meanings within either users or content, in the way that content-based recommendation does. Nor do we have to assume that engagement with an object articulates the belonging to a group that is given strict boundaries, in the way that collaborative filtering does. Rather the measures that get produced locate all entities, both human and non-human, as points in space that can potentially associate with any other entity. Perhaps the only assumptions we need to make are that the entities under the view of the algorithm have something to do with each other, and that the number of dimensions chosen for the generated spaces could be an adequate amount, and that the quantitative ways of defining 'closeness' between two spatial points are adequate[6] .

Thus, the way that users can engage with the actions that the algorithms reflect back to them to formulate a Foucauldian ethical substance and develop ways of knowing themselves and refining their lives by living it as a personal project, much like the other methods. However, unlike collaborative filtering methods, uneven access to 'opening the black box' of the algorithmic mechanisms does not privilege other parties with access to specific ways of knowing the user, since the internal components are unintelligible, beyond how they can be read through the act of personal reflection on the recommendation outputs.

A Baradian view of the agential cuts produced here is thus markedly different from those produced by a content-categorization based method. The deployment of latent factor methods brings awareness to the potential arbitrariness of distinctions in ascribing properties to users vs. content dualistically that are intentionally drawn by other methods, which is concurrent with the drawn boundary between objects and agencies of observation within a recommender system. In the latent factor model, all entities exhibit equally fluid agency in determining and reading each other's attributes relationally, posing them as concurrently assuming roles of observed object and agency of observation. As such a method is less tied to creating conceptual constraints, it also produces less exclusions in participation of knowledge production; every engaging entity has an impact on how the algorithmic structuring of the constitutive elements of preference get defined. In exchange for producing fewer exclusion by way of agential cuts, fewer concepts get delineated in terms of making the found preferences intelligible.

However, the ability to use these methods requires a scale of aggregation on a level above that required for collaborative filtering. "Cold start" problems mean that the implementation of these methods not only requires a mass of users and contents, but also a mass of interactions that spans entities to make the mass of users and content sufficiently well connected. If there are too few relations (e.g. consumption interactions) connecting nodes in our space, there becomes an issue with definitively establishing a space (Koren and Bell (2015): 83). At the technical level of the algorithmic realization, too few interactions means that a matrix may exhibit enough sparsity that it does not have an inverse matrix (a property known as *matrix singularity*), which makes finding an algebraically optimal fitting of the entities into a feature space impossible. Thus such methods not only need a mass of consumers, but more specifically they need a mass of *mass consumers*.

As such, the practical emergence of these technologies is coextensive with the emergence of even larger scales of consumption than was needed for the emergence of mass membership for collaborative filtering technologies to emerge. This is becoming increasingly accessible through platforms that enable and encourage 'infinite scrolling', 'binge watching', and similar practices. In fact, a historically motivating moment for the development of such

Organics of Domination	***Informatics of Domination***	**Latencies of Domination**
Content-Based Recommendation	Collaborative Filtering	Latent Space Mapping
Representation	*Simulation*	Factorization
Cooperation	*Communications*	Crowdsourcing
Functional Specialization	*Modular Construction*	User-Created Tagging (Folksonomy)
Depth, Integrity	*Surface, Boundary*	Locale in (Feature) Space, Motility
Heat	*Noise*	Data Sparsity/Matrix Singularity
Family Wage	*Comparable Worth*	Audience Impression/Impact
Microbiology, Tuberculosis	*Immunology, AIDS*	Histocompatibility, Transplant Failure

Table 4.1: Dominations of an even newer, more massive scale. Italicized entries are present in the original *Cyborg Manifesto* table.

methods was the publication of a very large dataset by Netflix of their user's film-watching data, with a competition (which ran from 2006 to 2009) awarding a monetary prize to those who could create the best recommendations on the data (evaluated by a second 'test set' of data that had 'right answers' to be predicted) (Koren and Bell (2015): 77). In addition to expansion of practices of consumption, we also see an expansion of ways that behaviors are monitored and interpreted as constituting interactions; for example, taking an extra second longer than average to scroll past a specific headline on a news website may be recorded as constituting a 'positive interaction' (especially when positive for the platform controllers constitute 'trapping' consumers within their platforms as long as possible).

If we conceive of the move from preference elicitation via mapping by essentialized product attributes to preference elicitation via collaborative filtering as part of a broader shift that Donna Haraway denotes as the shift from "Organics of Domination" to "Informatics of Domination", then perhaps a move towards post-intelligible latent factor methods is part of a third, newer category, which I will call the "Latencies of Domination". Latency in the sense that it permeates, and in fact constitutes an essential aspect of the constructed spaces in which we find ourselves situated, but also latent in the sense of tacit, unsaid, if not even unthought, like a reflex; as such mechanisms are fueled by regimes of actions that are not "thought" (at least humanly) before they are done, understandings need not be produced, because the appropriate responses are made by an algorithm itself without having to explain its reasonings to a human middle-person. As such, a tentative extension of relevant aspects of Haraway's table from the *Cyborg Manifesto* is shown in Table 4.1.

4.8 Tracking Trajectories of Deployment: Who Gets to Be Stuck in the Past?

We can trace these methods of preference elicitation as being developed in a chronological order of progression, as suggested by the chart above and the ways that Haraway deploys her subset of it. However, we can see that deployments of the oldest algorithmic methodologies are still seeing very active use (n.b. the Vayanos et al. paper is current research in the process of review). Why is this?

A technician's answer to this may be that the setting at hand itself has different requirements of that of the other fields examined of mass consumption. There are only so many policy-makers, to a point that collaborative filtering is not yet quite relevant for understanding their preference, much less latent factor methods. A question from another standpoint, however, might be why are there so few agents in policy making, or why can't we look at the preference of those not in political power? The framing of LAHSA's problem in Vayanos et al. clearly references that "there are an estimated 58,000 individuals experiencing homelessness in the L.A. area" (Vayanos et al. (2020): 5), who are primary stakeholders expected to possess preference on policy, which if able to be successfully enrolled (which would also require admitting their agency in the infrastructuring of deployed methods), could potentially provide more than enough mass of users to generate newer types of methodologies on creating knowledge claims about preference.

We also note that in the context of music recommendation, latent factor methods have been the subject of academic publications and have been shown to be effective (Jin and Han (2020), van den Oord et al. (2013)), but there is no discussion (publicly available that is, many details about preference elicitation in such commercial platforms is proprietary) indicating that they see commercial use on mainstream music streaming platforms. Since such methods are less immediately easy to mobilize in seeing consumer masses as segmented into ways of life that can translate to effective advertisement strategies, there is likely little incentive to apply such algorithms.

The move from organics to informatics to latencies of domination does not unfold evenly across the board of lived social life. Since each step in this progression requires an increased scale in participations and dispersed agencies, the concentration of power within individuals (rather than more dispersive systems), or to certain ways of seeing them (as advertisement segments), requires restraining the scope of access and engagement making certain technologies of preference elicitation closed off from access. This can provide us with new ways to look at why certain domains experience certain deployments of preference elicitation technologies, and not others, or track domains that (perhaps covertly) maintain

concentrations of agency within small groups of individuals.

Endnotes

[1] In the case that the preference is unambiguously identified, a best solution can be presented that maximizes utility. But under a constraint where a limited number of questions only reduces preferences to a range of solutions, the offered solution can either be one that maximizes minimum utility, or, as is recommended in the paper, one that minimizes regret (the maximum possible difference in utility between the made choice and what would be the optimal choice for any given credible preference).

[2] "Physical and conceptual constraints and exclusions are co-constitutive" (Barad (1998): 95)

[3] The 7 genotypes used by Pandora are; Pop, Rock, Jazz, Hip Hop, Electronica, World, and Classical (Gasser (2019): 303)

[4] To demonstrate the issues of conflicts of different standpoints at work, one can look to the contested boundaries of the label 'jazz', which often gets attributed to Black American musicians who wish to deny such labels, for example many members of the Association for the Advancement of Creative Musicians (AACM) who don't seek to classify their music beyond "Creative Music" (Lewis (2008): 343)

[5] These are often considered to be a subset of the world of collaborative filtering approaches, but as per my elaboration, possess certain qualities which could allow them to be identified as rather unique approaches compared to the neighborhood-based approaches that populate the rest of the domain.

[6] Interestingly, which way one ought to measure the space geometrically has been a subject of some study. Schmeier et al. (2019) suggest that a hyperbolic space rather than a conventional euclidean space may be better suited for capturing latent hierarchical relations, rather than latent independent categories, and claim that an applied algorithm in the realm of music recommendation sees higher user retention in an experimental setting.

Chapter 5

A recommender system for caregivers of individuals with Alzheimer's and related dementias

5.1 Abstract

The goal of this chapter is to use data collected by CareVirtue to develop and test a recommender system that recommends legal and financial resources for caregivers of individuals with ADRD. Our aim is to provide a curated set of helpful financial and legal planning topics for each caregiver – by providing a subset of topics, we avoid further overburdening the caregiver with irrelevant information. We evaluated four machine learning models for directly predicting caregiver confidence in 18 topics on a 1-5 Likert scale. We also evaluated a baseline model that simply predicts the average (across observations) for each topic. We found that an ElasticNet model performed best with an average RMSE of 1.061, which improved upon the baseline model by 0.217 (17%). The ElasticNet model outperformed the baseline for all topics; adding an additional feature improved the RMSE by 0.044 (4.1%); using five levels instead of three for the onboarding process improved the RMSE by 0.017 (1.6%); and using all 10 features as apposed to only five contextual features improved the RMSE by 0.194 (15.5%). We evaluated five recommendation models using simulation; we found that the ElasticNet model slightly outperformed the baseline approach. We conducted a series of robustness checks that confirm our primary results.

In a trial implementation of resource recommendations for online platform users, we found that the ElasticNet predicted resource usefulness with an average RMSE of 1.453, which improved upon the baseline model by 0.436 (26.7%). We also found that resource

categories have statistically different click-through rates based on the source of recommendation, with the personally recommended resources in the top predicted care topic having a click-through rate of 9.9%, more than doubling the click through rate of general recommendation resources (4.3%).

5.2 Introduction

There are more than 6 million individuals living with Alzheimer disease and related dementias (ADRD) in the United States (Better 2023). These individuals require prolonged and complex care that is primarily managed by informal caregivers, such as partners/spouses, children, etc. There are over 11 million informal caregivers in the United States and these caregivers provide informal care valued at more than $346 billion (Better 2023). Although caregivers can experience positive outcomes related to caregiving, they often report being under-supported and under-resourced.

One of the many tasks that caregivers typically manage is the financial and legal tasks for care recipients. Given that caregivers are already over-burdened, understanding, organizing, and completing the tasks associated with financial and legal planning can be very challenging (and time consuming). Platforms have been developed to assist caregivers with these tasks by providing education or connecting them to resources. For example, CareVirtue is a platform that was developed specifically to provide caregivers with learning modules and resources related to financial and legal planning topics.

The goal of this chapter is to use data collected by CareVirtue to develop and test a recommender system that recommends legal and financial resources for caregivers of individuals with ADRD. Our aim is to provide a curated set of helpful financial and legal planning topics for each caregiver – by providing a subset of topics, we avoid further overburdening the caregiver with irrelevant information. We will build the system using offline data that was previously collected. We will then implement the system and evaluate the performance in a pilot study with 66 participants.

5.3 Data

In this section, we describe the data used to train, test, and evaluate the recommender system.

5.3.1 Offline data

We obtained socio-demographic and survey data from 318 individuals who self-identified as caregivers of persons living with dementia. The socio-demographic data includes 12 features about the caregiver and care recipient. The survey data includes self-reported caregiver confidence (measured on a 1-5 Likert Scale) for 23 legal and financial topics. Of these 23 topics, 18 will form the set of potential recommendations (and thus our target variables for prediction), while the remaining five will be used as features (more details below). Table 1 displays the text of each of the 18 topics (survey questions) and Figure 1 displays bar plots of caregiver confidence for each of the 18 topics.

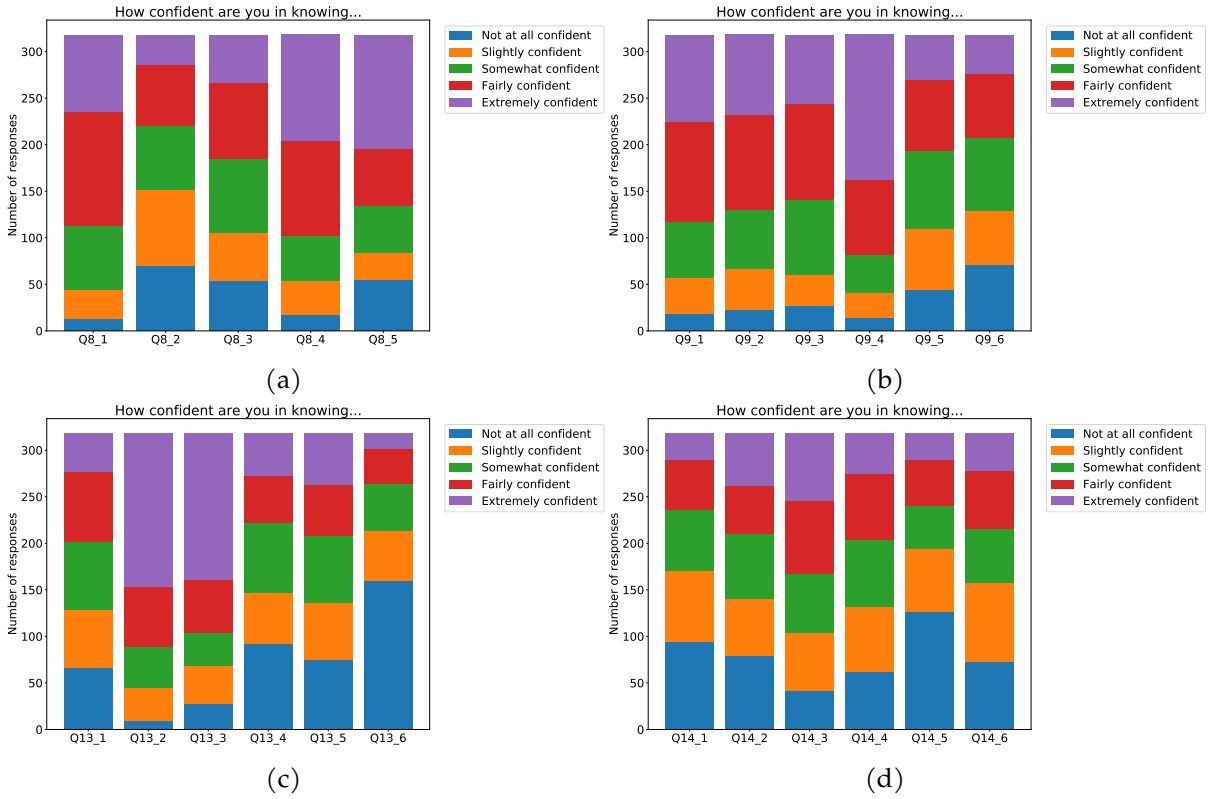


Figure 5.1: Confidence distributions for all 18 topics.

5.3.1.1 Data cleaning.

We drop one individual because they were missing confidence for one of the 18 topics (targets). We modified the relationship feature to include parent/mother/father, partner/wife/husband, and other (grandparent, relative, friend, etc.). For one individual, years of caregiving was reported as 365 years, so we replaced this value with the mean (as though it was missing). There were no other missing values.

5.3.1.2 Features.

The platform prompts caregivers to respond to 10 questions during the on-boarding process, so our models will have access to 10 features. There are five demographic / contextual features: age (in years), gender (male, female, other), stage of caregiving (first, second, third), relationship to care recipient (parent, partner, other), and years of caregiving experience. We will consider models trained using only these five socio-demographic features. There are also five questions about caregiver confidence (Not confident, somewhat confident, very confident), which are mapped to similar questions from the offline survey data:

- "How confident are you in using a Power of Attorney to help manage money for [insert name]?" We map this to Q9.3.
- "How confident are you in knowing where the financial accounts are for [insert name]?" We map this to Q13.2.
- "How confident are you in knowing how care will be paid for?" We map this to Q13.1.
- "How confident are you in carrying out the end of life wishes for [insert name]?" We map this to Q8.4.
- "How confident are you in having support from others for the legal and financial needs of [insert name]?" We map this to Q13.3.

For each of the mapped questions, we convert the survey confidence values of 1 and 2 to "Not confident", 3 and 4 to "Somewhat confident", and 5 to "very confident" so that they are aligned with the on-boarding questions. We evaluate the effect of this information loss on prediction accuracy in our experiments. These five questions are used as features in the models and not included in the set of 18 topics that we seek to recommend. We perform one-hot encoding for relationship and gender, and we scale all features to the $[0, 1]$ interval using min-max scaling

5.3.1.3 Targets.

Each of the 18 targets is represented on a 1-5 Likert scale and we use the raw targets in regression-based approaches.

Table 5.1: The text of each survey question (recommender topic), the question ID, and the average confidence rating.

ID	Avg. Rating	Question
Q8.1	3.73	how to have a conversation with a family member about their medical care choices
Q8.2	2.72	how to protect yourself legally as a caregiver
Q8.3	3.08	if you are legally responsible for care recipient expenses
Q8.4	2.19	how to honor end of life wishes for a family member
Q8.5	3.52	if funeral arrangements have been made and if they are prepaid
Q9.1	3.69	what legal documents are necessary to act on behalf of a family member for their medical care
Q9.2	3.58	what legal documents are necessary to manage a family member's finances
Q9.3	2.04	how to use the legal documents to act on behalf of a family member's medical care or manage finances
Q9.4	4.06	where a family member's current legal documents are located
Q9.5	3.06	the circumstances when legal documents should be updated or renewed
Q9.6	2.84	the options when legal documents are not in place and a family member is not legally competent
Q13.1	1.73	if there will be enough money to provide care
Q13.2	2.38	the location of a family member's financial accounts
Q13.3	2.28	if a family member has been working with a financial planner or other financial professional
Q13.4	2.69	if family members or friends can be paid for caregiving
Q13.5	2.85	how caregiving may impact your own finances
Q13.6	2.05	what tax deductions are available to cut care costs
Q14.1	2.51	about sources of money from government programs that a family member is eligible for such as Veterans Affairs and Military Benefits, Social Security Disability, etc.
Q14.2	2.82	about sources of money from life and long term care insurance policies
Q14.3	1.90	about sources of money from retirement accounts
Q14.4	2.89	about care budgeting and planning
Q14.5	2.32	the options when there is not enough money to provide care
Q14.6	2.72	the differences between Medicare, Medicare Advantage, and Medicaid programs (eligibility, benefits, providers, costs, etc.)

5.3.2 Online data

Following the development of a recommender system from offline data, the developed recommender system is implemented in a trial of the CareVirtue platform, used to recommend resources to trial users based on the predicted care topics that they need the most assistance with. For each trial user, we track which care topics get predicted and which resources get personally recommended. For each resource personally recommended by the recommender, as well as other resources available as "general recommendations," each user is able to provide feedback on the usefulness of the resource on a 1-5 Likert scale, with an optional open text field for comments. Additionally, we track which resources get viewed by trial users.

5.4 Recommender System

In this section, we describe and evaluate the recommender system. The goal of recommender system is to recommend a subset of the 18 financial and legal planning topics to caregivers.

5.4.1 Predictive models

We consider four regression models that predict the user rating directly: linear regression with elastic-net regularization (ElasticNet) (Zou and Hastie 2005), k-nearest neighbors (KNN) (Cover 1968), classification-and-regression trees (CART) Breiman et al. (2017), and random forest (Random Forest) (Breiman 2001). For all four models, we employ multitask learning where we simultaneously train the model(s) to predict all 18 targets. As a baseline, we use the "popular" method, where for a given target, we predict the average value of the target over all users.

We use root mean squared error (RMSE) and perform leave-one-out cross validation to evaluate our regression models. We repeat the cross-validation procedure to evaluate the following hyperparameters: linear regression with elastic-net regularization ('alpha':[0.0001,0.001,0.01,0.1,1], 'l1 ratio':[0,0.25,0.5,0.75,1]), k-nearest neighbors ('neighbors':[5,10,25,50,75,100,150], 'weights':['uniform','distance'], 'p':[1,2]), classification-and-regression tree ('max depth':[2,3,4,5,6,7,8,9,10]), and random forest ('max depth':[2,5,10,15], 'estimators':[10,100,250,500]). We perform 10-fold cross validation as a comparison in Section C.1.

Figure 5.2 displays boxplots (acrosss individuals) of the RMSE values for the best models of each type with all 10 features and with only 5 socio-demographic / contextual features. When using all features, the best KNN model has 25 neighbors, distance-based weighting, and minkowski distance; the best ElasticNet model had an alpha of 0.01 with l2 regularization; the best CART model had depth 3; and the best Random Forest model had 250 trees and depth 5. The overall best performing model was ElasticNet with an average RMSE of 1.061. This model improved upon the popular method by 0.217 (17.0%). The average RMSE for the popular, KNN, CART, and Random Forest methods was 1.278, 1.109, 1.117, and 1.067, respectively. The models that only had access to five features performed similarly to the popular approach. For example, the ElasticNet model with 10 features improved upon the five feature model by 0.194 (15.5%).

Figure 5.3 displays boxplots (across individuals) of RMSE values for each model with the 3-levels and 5-levels for the onboarding questions. The ElasticNet model with 5 levels for the onboarding features was able to improve upon the model with 3 levels by 0.017 (1.6%).

Figure 5.4 displays line plots of the average RMSE for each target individually for both the Popular and ElasticNet models. Both models performed the worst on topic Q8.5 with an average RMSE of 1.344 for ElasticNet and 1.500 for Popular. ElasticNet performed best on Q9.2 with an average RMSE of 0.847, while Popular performed best on Q8.1 with an average RMSE of 1.080.

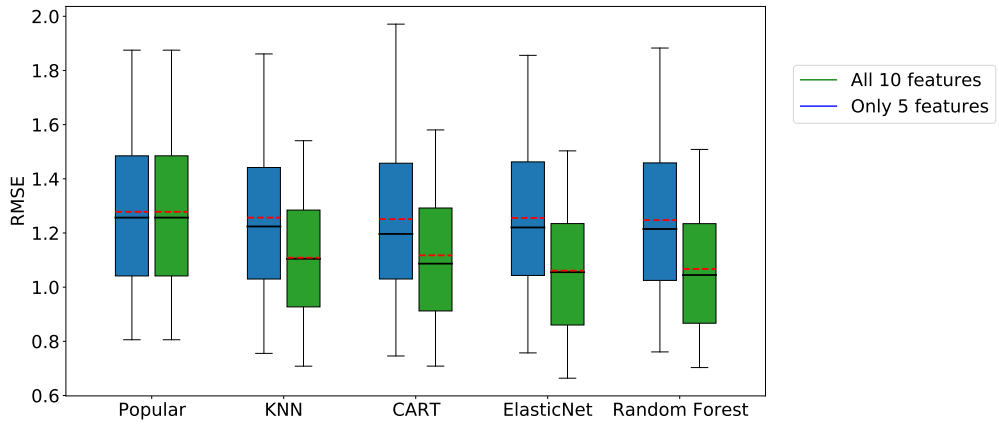


Figure 5.2: Boxplots (across individuals) of RMSE values for each model with all features and with only five socio-demographic features. The solid black line denotes the median and the dashed red line denotes the mean.

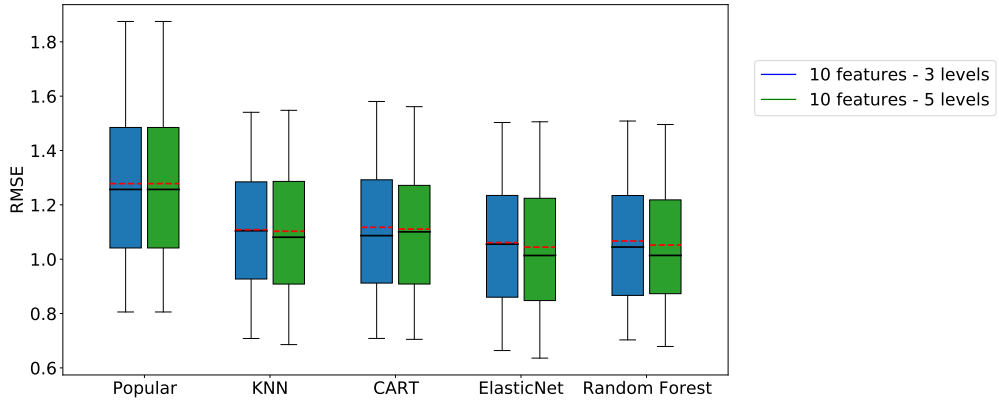


Figure 5.3: Boxplots (across individuals) of RMSE values for each model with the 3-levels and 5-levels for the onboarding questions. Note that the 5-level onboarding questions match the survey data. The solid black line denotes the median and the dashed red line denotes the mean.

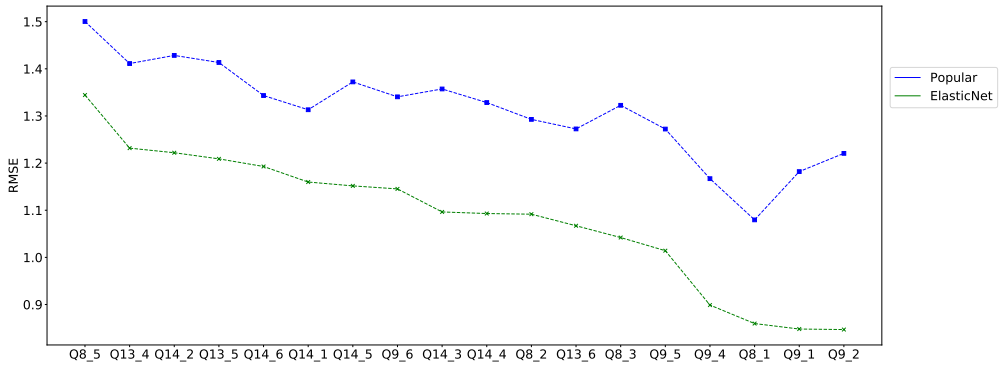


Figure 5.4: Line plots of the average RMSE for each individual question, sorted from largest to smallest according to the ElasticNet model.

Figure 5.5 displays boxplots (across individuals) of RMSE values for the ElasticNet model with an additional feature (noted on the x-axis) and the base model (shown on the far right). Note that the features are ordered according to mean RMSE, where the best additional feature is shown next to the base model. The best feature to add to the model was Q14.5, and this feature improved RMSE by 0.044 (4,1%) over the base model with 10 features. We evaluate the recommender system with an additional questions Q14.5 and Q14.4 in Sections C.2 and C.3, respectively.

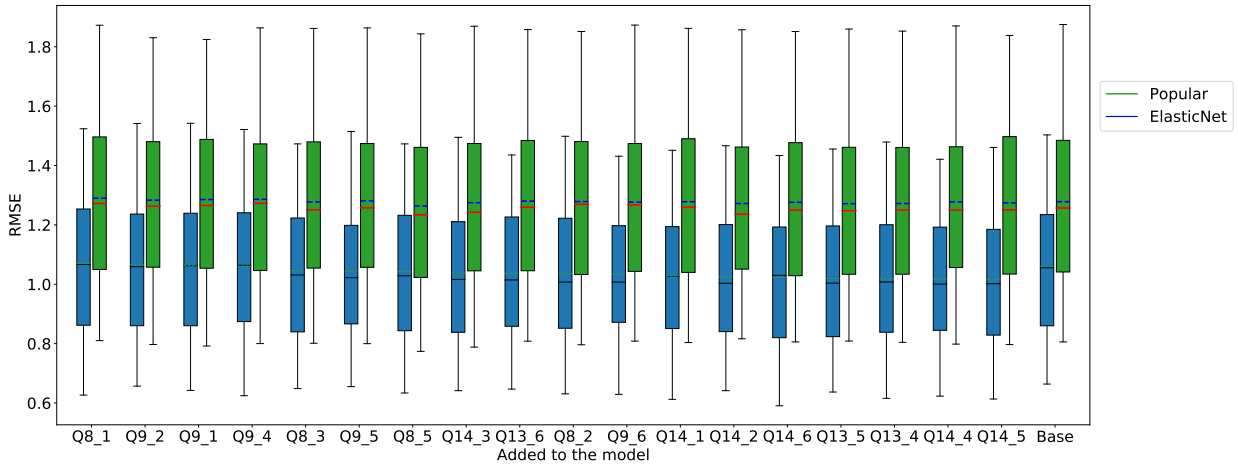


Figure 5.5: Boxplots (across individuals) of RMSE values for the ElasticNet model with an additional feature and the base model. The solid red line denotes the median and the dashed blue line denotes the mean.

5.4.2 Prescriptive models

In this section, we seek to evaluate via simulation how the recommender system might perform. We note that this is not typically done in the literature and simulated recommender systems may not correspond to reality – most recommender systems are simply evaluated on RMSE. However, this exercise does provide some insight and evidence into the performance.

The goal of each recommendation model is to provide a ranked list of the 18 topics (or targets from the predictive model perspective). We consider a recommendation as relevant if, for a given user, the target value of the topic is less than or equal to the average target value over all 18 topics for that user (i.e., it is one of the topics they rated below average). In this way, we have a set of relevant topics for each user. Figure 5.6 displays a histogram of the number of relevant topics for each individual. The mean number of relevant topics was 9.08. We consider two alternative approaches for determining the set of relevant recommendations in Sections C.5 and C.6.

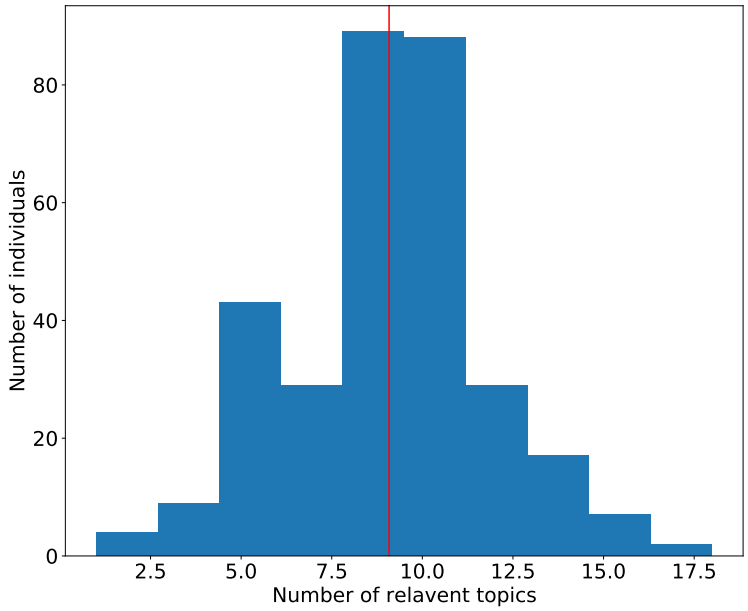


Figure 5.6: A histogram of the number of relevant topics for each individual. The red line denotes the mean (9.08).

5.4.2.1 Setup.

We employ the top two best performing models (ElasticNet and Random Forest). Each model outputs a predicted rating for each topic/target for a given user; we rank these predicted ratings (in increasing order) to provide our ranked recommendation list.

We also consider three additional baseline recommender models: the "popular" method from the predictive section; the "random" method, where we recommend randomly selected topics; and the "content-based" method, where for a given target, we recommend topics from the same group (Q8, Q9, Q13, Q14) ranked by mean confidence (over all users) and we rank the remaining topics using mean confidence (similar to "popular").

We perform leave-one-out cross validation to evaluate our recommendations. We compute the following metrics: precision at k (the number of relevant items our the k recommendations divided by k) , recall at k (the number of relevant items our the k recommendations divided by the number of relevant items for that user), f1 score at k (a combination of precision and recall), average precision at k (average the prevision values up until k), and diversity in recommendations at k (number of different recommendations given across all caregivers), where k is the number of recommendations given to a caregiver (over their entire time on the platform).

5.4.2.2 Evaluation.

Overall, the ElasticNet model performed best.

Figure 5.7 displays the precision and recall values as a function of the number of recommendations for each model. At five recommendations, the best performing model was ElasticNet with a precision of 0.717 and a recall of 0.409. This improved upon the popular method by 0.015 (2.2%) and 0.01 (2.4%), for precision and recall, respectively.

Figure 5.8 displays the f1 score as a function of the number of recommendations for each model. At five recommendations, the best performing model was ElasticNet with an f1-score of 0.507, which improved upon the popular method by 0.011 (2.3%).

Figure 5.9 displays the average precision as a function of the number of recommendations for each model. At five recommendations, the best performing model was ElasticNet with an average precision of 0.769, which improved upon the popular method by 0.01 (1.3%).

Figure 5.10 displays the number of unique recommendations as a function of the number of recommendations for each model. At five recommendations, the ElasticNet model recommended 17 different topics, while the popular model only recommended five topics.

5.4.3 Discussion

Most recommender systems are evaluated using RMSE, as we did in Section 5.4.1. For example, the Netflix Prize had RMSE values of 1.054 (popular), 0.951 (Cinematch), and 0.855 (winner) (Bell et al. 2010). Our problem appears slightly more difficult (likely due

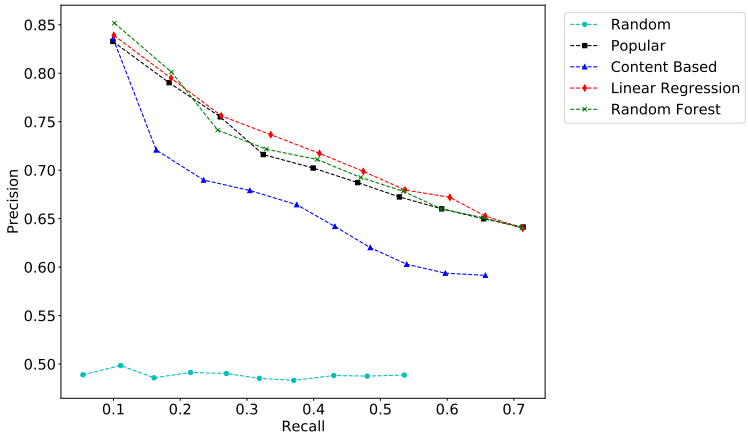


Figure 5.7: Precision and recall values as a function of the number of recommendations for each model.

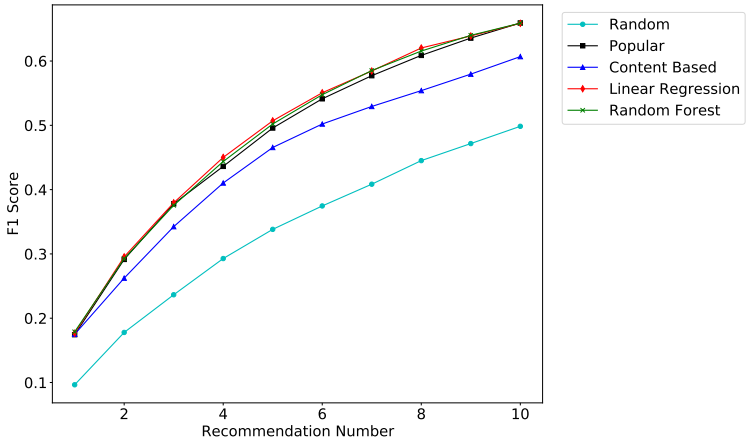


Figure 5.8: F1-score as a function of the number of recommendations for each model.

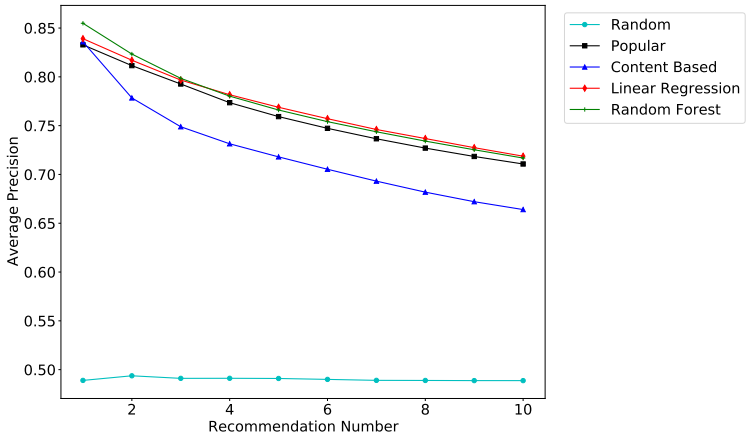


Figure 5.9: Average precision as a function of the number of recommendations for each model.

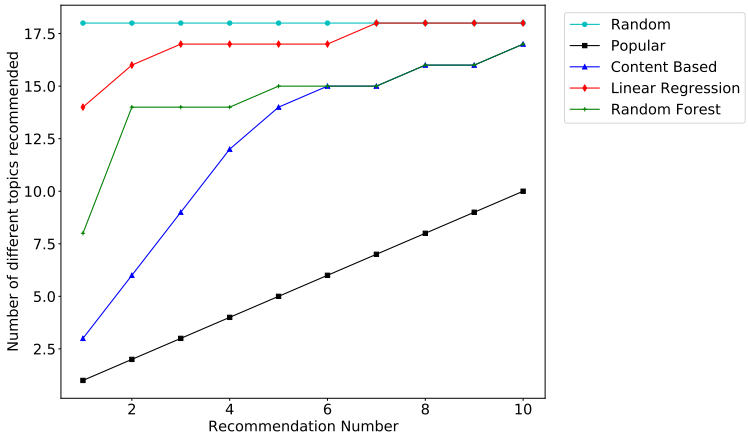


Figure 5.10: Number of unique recommendations as a function of the number of recommendations for each model.

to much less data) with popular scoring 1.278 and our best model (ElasticNet) scoring 1.061. ElasticNet is a type of linear regression with regularization that was trained to simultaneously predict all 18 target values. Overall, we improved upon popular by 17.0%, which is similar to the improvement found during the Netflix Prize (18.9%). See Figure 5.2 and Figure C.1. Below, we summarize the results in the context of practical implementation questions.

5.4.3.1 Do we need the five confidence questions during onboarding?

In Figure 5.2 and Figure C.1, we find that the addition of the five onboarding questions pertaining to confidence improves RMSE by 15.3-15.5% for our best model.

5.4.3.2 Do the five onboarding questions need to be on a 1-5 or 1-3 scale?

In Figure 5.3 and Figure C.3, we find that using three levels (instead of 5) for the onboarding confidence questions reduces RMSE by 1.6-1.7% for our best model.

5.4.3.3 Should we add a sixth onboarding question?

In Figure 5.4 and Figure C.4, we find that the addition of another onboarding confidence question increases RMSE by 2.0-4.1% for our best model. The next best questions to add are Q14.5 and Q14.4.

5.4.3.4 For which questions do we perform best?

Figure 5.4 shows the mean performance of ElasticNet for each of the 18 individual questions.

5.4.3.5 How does the model do in simulation?

We simulated the recommendation system in Section 5.4.2. We find that ElasticNet performs well across several metrics, and seems to improve as the number of recommendations increases. We note that while popular performed significantly worse in RMSE, it performs comparably to ElasticNet in simulation. However, ElasticNet is able to provide a diverse set of recommendations, indicating that it is attempting to personalize the recommendations to individuals. These results appear to hold across several different experimental setups, as shown in Sections EC2 - EC6.

5.5 Implementation and Evaluation in Platform Trial

Following the experimental results on the survey data, the prescriptive model based on the ElasticNet predictive model was implemented for a group of trial users of the CareVirtue planner.

5.5.1 Trial Setup

Upon onboarding of each trial member, the model predicted the three topics with lowest estimated confidence for each user. For each topic, users were recommended up to six resources that could aid them in care corresponding to that topic, denoted as "personal recommendations", located on the platform's Resources page. Personal resource recommendations were given at the beginning of the trial for the top recommendation, after two weeks for the second recommended topic, and after four weeks for the third topic. Additionally, each user was also given the same one 'universal recommendation' for an educational resource that covered many topics of caregiving at the beginning of the trial. Both personal and universal recommendations were located under a tab titles "Your Recommendations". An addition sixteen "general recommendation" resources could be found within larger resource banks sorted into four categories; "Honor End of Life Wishes," "Using a Power of Attorney," "Take Financial Inventory," and "Find Money and Use a Budget," which correspond to the overarching care topics Q8, Q9, Q13, and Q14, respectively. General recommendations were available from the beginning of the trial. We note that some resources in these general resource banks are directly referred to in the educational modules on the platform, however we did not consider them to be universal recommendations since they did not appear in the 'Your Recommendations' tab.

Users were encouraged, but not required, to give feedback on the usefulness of resources they used, including personal recommendations, general recommendations, or any 'non-

recommended' resources they found on the larger resource bank on the platform. Feedback was formatted as a 1-5 Likert score of the usefulness of the resource, with the option for a written comment.

To evaluate the efficacy of the recommender system, we compared the Likert score of resource feedback to five minus the user's predicted confidence in the care topic (we assume that resources for topics with low confidence are more useful than resources topics with high confidence). We evaluate accuracy of predictions with RMSE for both the enacted ElasticNet recommender and predictions made by the "popular" baseline model.

We also evaluate the efficacy of the recommender system not only by active feedback on resources, but also by users' viewing of resources. For all recommendations given, we use the number of unique resource views by a user and the total number of recommendations given to evaluate a "click-through rate", and compare click-through rates for the following groups; personal recommendations for the top care topic ("first wave" personal recommendations), personal recommendations for the second care topic ("second wave"), personal recommendations for the predicted third care topic ("third wave"), universal recommendation, and general recommendations. We use a χ^2 test at a significance level of $\alpha = 0.05$ to test the hypothesis that the click-through proportion is equivalent across all groups. As a post-hoc analysis, we compare each personal and universal recommendation group against the general recommendation group using χ^2 tests, with the alternative hypothesis that personal and universal recommendations have a higher click-through rate, with a Holm-Sidak adjustment to correct for bias introduced with multiple tests.

5.5.2 Results

There were a total of 66 users, who were given a total of 608 personal resource recommendations during the trial period. Over the course of the trial period, 84 resource ratings were received, with ratings given by 30 unique users. Out of the 84 ratings, 35 included a text comment. Table 5.2 shows the distribution of trial users by demographic characteristics.

Table 5.3 shows the counts of personal recommendations for the ten most recommended resources and their corresponding care topics. Across trial users, 50 different resources were given as personal recommendations, spanning 13 out of the 18 recommendable care topics. 83.3% of users received recommendations relating to topic Q13.6, and 50.0% received recommendations relating to topic Q14.5 which were the lowest and second lowest topics respectively in mean caregiver confidence in the initial survey data.

Figure 5.11 shows the distributions of Likert score feedback rating for (a) personally recommended resources, (b) universal recommendations, and (c) general recommendations.

User Demographic	
Caregiver Gender	
Female, n. (%)	57 (86.4%)
Male, n. (%)	9 (13.6%)
Other, n. (%)	0 (0%)
Relationship of Patient to Caregiver	
Parent, n. (%)	32 (48.5%)
Partner, n. (%)	31 (47.0%)
Other, n. (%)	3 (4.5%)
Patient Dementia Stage	
Early, n. (%)	12 (18.2%)
Middle, n. (%)	40 (60.6%)
Late, n. (%)	14 (21.2%)
Caregiver Age (years), mean (stdev)	58.4 (13.1)
Caregiver Years of Caregiving Experience, mean (stdev)	5.0 (3.7)

Table 5.2: Trial population details

Resource Title	Care Topic	Recommendations
IRS - Nursing Home Costs	Q13.6	55
Taxes, Deductions, Everything Caregivers Should Know	Q13.6	55
Alzheimers Association - Tax Deduction and Credits	Q13.6	55
AARP - Caregiver Tax Breaks	Q13.6	55
NerdWallet - What If I Can't Afford Care?	Q14.5	33
Alzheimers Association - Managing Money	Q14.5	33
AARP - Local Assistance Directory	Q14.5	33
AARP - Medicare vs. Medicare Advantage	Q14.6	30
DHHS - Medicare and Medicaid	Q14.6	30
AARP - How to Use Long Term Care Insurance	Q14.2	18

Table 5.3: Top ten personally recommended resources and their corresponding care topics

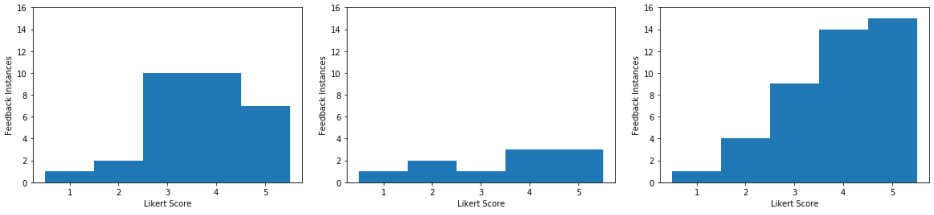


Figure 5.11: Histograms of Likert Score responses for (a) personally recommended resources, (b) universally recommended resources, and (c) generally recommended resources.

Resource Title	Care Topic	Recommendation Type	Number of Ratings	Mean Likert Score
Monthly Budget and Expense Form	Q14.4	General	15	3.47
Careblazers - Dementia Caregiving Education		Universal	10	3.5
Genworth - Cost of Care Estimator	Q14.4	General	9	4.22
FreeWill	Q8.4	General	4	3.5
AARP - Caregiver Tax Breaks	Q13.6	Personal	4	3.5
NAPFA		General	4	4.25
National Council on Aging - Benefits	Q14.1	General	4	4.5
ABA - Caregiver Must Do's	Q8.2	Personal	3	4.67
NerdWallet - What If I Can't Afford Care?	Q14.5	Personal	3	3.67
IRS - Nursing Home Costs	Q13.6	Personal	3	3.33

Table 5.4: Ten most rated resources

Resource Category	ElasticNet RMSE	Popular RMSE
Personal Recommendations	1.192	1.628
All Predictable Resources	1.453	1.804

Table 5.5: RMSE of predicted usefulness of trial resources

There were 30 instances of feedback for personally recommended resources, 10 instances for universal recommendations, and 44 instances for general recommendations. The mean (median) rating for resources in each category were 3.67 (4.0), 3.5 (4.0), and 3.88 (4.0), respectively.

Table 5.4 shows the ten most rated resources. Only three resources received more than four ratings, all of which were either general or universal recommendations.

Table 5.5 shows the RMSE of the predictions of resource usefulness, comparing prediction made by the implemented ElasticNet recommender as well as the baseline "popular" method. The ElasticNet recommender improved on popular recommendation's RMSE by 0.351 (19.5%) across all feedback for predictable resources, and by 0.436 (26.7%) for personally recommended resources. We note that the universally recommended resource, as well as some general recommendations, are not associated with a unique care topic, so neither the ElasticNet recommender not the popular baseline method were able to predict usefulness for these resources.

Table 5.6 shows the total resource views and click-through rates of recommended resources. A χ^2 test of resource views by these five categories yields a p-value of 0.00025, allowing us to reject the null hypothesis that the click-through rate for resources is independent of the category of recommendation. A post-hoc comparison of each group to the general recommendation group finds that first wave personal recommendations

Recommendation Category	Resource Views	Number of Recommendations	Click-Through Rate	Comparison to General χ^2 adjusted p-value
First Wave Personal	22	223	9.9%	0.0022
Second Wave Personal	13	208	6.3%	0.38
Third Wave Personal	6	177	3.4%	0.59
Universal	9	66	13.6%	0.0022
General	45	1056	4.3%	-

Table 5.6: Resource views and click-through rates of recommendations by recommendation category

and universal recommendations have significantly different click-through rates from the general recommendation click-through rate.

5.5.3 Discussion

The results of the trial implementation of the ElasticNet recommender, confirms many observed insights from the evaluation on survey data, as well as revealing new insights.

The ElasticNet recommender gives diverse recommendations: For the 66 trial users, when predicting the three most important care topics for each user, 13 care topics (out of a total 18) were predicted for at least one user.

The ElasticNet recommender improves upon the popularity-based method when predicting the Likert score usefulness of a resource: ElasticNet improves RMSE by 0.351 (19.5%) compared to the popular model for personally recommended resource, and it improves RMSE by 0.436 (26.7%) for all predictable resources.

All resources categories have overall high Likert score responses from trial users: 90%, 70%, and 88.3% of responses for personal, universal, and general recommendations respectively were scores of at least 3 out of 5. This indicates good general curation of a bank of resources that can support caregivers.

Users' likelihoods to view a resource is impacted by the source of the recommendation: The highest click-through rates for resources appear in the universal recommendation and first wave personal recommendations. This can be potentially explained by many factors: the "Your Recommendations" tab is the first that comes up when users enter the resources page, the relevance of the personal recommendations can encourage more resource views, and the delayed presence of second- and third-tier personal recommendations can reduce resource views as some users decrease platform engagement over time.

5.5.3.1 Implementation Considerations

In implementing the recommender developed in Section 5.4 for the trial study, there were many challenges and important considerations.

Reconciling different target definitions between the survey data and the platform: the goal of the recommender system on the platform was to recommend useful resources to caregivers, and collected data on the rated usefulness of specific resources, whereas the survey collected data on caregiver confidence with various legal and financial care topics. As such a model trained on survey data cannot directly predict the usefulness of a resource, as the survey respondents did not provide data on the usefulness of such resources. In order to predict resource usefulness, each resource that could be a personal recommendation was assigned to its most relevant care topic, and the predicted usefulness of the resource was calculated as five minus the predicted confidence in the care topic. We discuss potential drawbacks of this approach in Section 5.5.3.2.

5.5.3.2 Limitations

Text submissions accompanying user feedback indicates that many resources with Likert score ratings of 1-2 out of 5 received poor ratings not because assistance in the topic was unneeded, but rather because of quality issues with the resources themselves (e.g., contains broken links, asks for personal information that the user does not want to disclose). Written feedback that a resource covered an unneeded topic instead occurred most with ratings of 3 out of 5. This indicates one possible limitation of how understanding confidence in care topics is not solely sufficient for understanding the usefulness of resources. Furthermore, while the method of modeling resource usefulness based on the care topic covered offers advantages in that it allows newly encountered resources to be predicted and recommended without any prior observations of users rating the resource, it also prevents prediction and recommendation of resources that do not fit clearly into one care topic over others.

Regarding the distributions of Likert scores across personal and general recommendations, we note that multiple resources in the general recommendation categories are directly referred to in the educational modules of the CareVirtue platform, giving added context on how one can make best use of a specific resource in a specific instance, which is lacking for personal recommendations beyond a single sentence in the "resource card" on the platform.

5.6 Conclusion

In this paper we present the development and implementation of a recommender system for caregivers of individuals with Alzheimer's and related dementias. We show that such a recommender system is able to provide personalized recommendations that anticipate usefulness to users better than recommending popular resources to all caregivers.

Appendix A

Appendices to Chapter 2

A.1 Proofs

A.1.1 Proof of proposition 2.6

Proof. Proof of Proposition 2.6: Let $\xi^k_{ABS}(\boldsymbol{\alpha})$ denote the gap function corresponding to the inverse optimization model $\text{GIO}_{\text{ABS}}(\hat{\mathbf{y}}, \mathcal{S}^k)$. We note that that $\text{GIO}_{\text{ABS}}(\hat{\mathbf{y}}, \mathcal{S}^k)$ is a relaxation of $\text{GIO}_{\text{ABS}}(\hat{\mathbf{y}}, \mathcal{S})$, and as such $\xi^k_{ABS}(\boldsymbol{\alpha}) \leq \xi_{ABS}(\boldsymbol{\alpha})$. We prove the claim by showing that if $\mathbf{C}\hat{\mathbf{y}} - \xi_{\text{ABS}}(\boldsymbol{\alpha}^*)\mathbf{1} \in \text{conv}(B^k)$ then; (i) $\xi^k_{ABS}(\boldsymbol{\alpha}^*) = \xi_{ABS}(\boldsymbol{\alpha})$, and (ii) $\xi^k_{ABS}(\boldsymbol{\alpha}^*) < \xi^k_{ABS}(\boldsymbol{\alpha}),\ \forall \boldsymbol{\alpha} \in \mathcal{A} \setminus \boldsymbol{\alpha}^*$.

Proof of (i): Since $\mathbf{C}\hat{\mathbf{y}} - \xi_{\text{ABS}}(\boldsymbol{\alpha}^*)\mathbf{1} \in \text{conv}(B^k)$, it is true that $\mathbf{C}\hat{\mathbf{y}} - \xi_{\text{ABS}}(\boldsymbol{\alpha}^*)\mathbf{1} = \lambda_0\mathbf{C}\mathbf{y}^{(0)} + \lambda_1\mathbf{C}\mathbf{y}^{(1)} + \cdots + \lambda_k\mathbf{C}\mathbf{y}^{(k)}$ for some $\lambda \in [0,1]^k$, $||\lambda|| = 1$. Thus we have,

$$\begin{aligned}
\xi_{\text{ABS}}(\boldsymbol{\alpha}^*) &= \boldsymbol{\alpha}^{*\intercal}\mathbf{C}\hat{\mathbf{y}} - \min_{\mathbf{y}\in\mathcal{S}} \boldsymbol{\alpha}^{*\intercal}\mathbf{C}\mathbf{y} \\
\boldsymbol{\alpha}^{*\intercal}\mathbf{C}\hat{\mathbf{y}} - \boldsymbol{\alpha}^{*\intercal}(\mathbf{C}\hat{\mathbf{y}} - \xi_{\text{ABS}}(\boldsymbol{\alpha}^*)\mathbf{1}) &= \boldsymbol{\alpha}^{*\intercal}\mathbf{C}\hat{\mathbf{y}} - \min_{\mathbf{y}\in\mathcal{S}} \boldsymbol{\alpha}^{*\intercal}\mathbf{C}\mathbf{y} \\
\boldsymbol{\alpha}^{*\intercal}\mathbf{C}\hat{\mathbf{y}} - \boldsymbol{\alpha}^{*\intercal}(\lambda_0\mathbf{C}\mathbf{y}^{(0)} + \lambda_1\mathbf{C}\mathbf{y}^{(1)} + \cdots + \lambda_k\mathbf{C}\mathbf{y}^{(k)}) &= \boldsymbol{\alpha}^{*\intercal}\mathbf{C}\hat{\mathbf{y}} - \min_{\mathbf{y}\in\mathcal{S}} \boldsymbol{\alpha}^{*\intercal}\mathbf{C}\mathbf{y} \\
\boldsymbol{\alpha}^{*\intercal}(\lambda_0\mathbf{C}\mathbf{y}^{(0)} + \lambda_1\mathbf{C}\mathbf{y}^{(1)} + \cdots + \lambda_k\mathbf{C}\mathbf{y}^{(k)}) &= \min_{\mathbf{y}\in\mathcal{S}} \boldsymbol{\alpha}^{*\intercal}\mathbf{C}\mathbf{y}
\end{aligned}$$

This implies that each $\mathbf{y}^{(i)} \in S^k$ for which $\lambda_i > 0$ is a minimizer of $\min_{\mathbf{y}\in\mathcal{S}} \boldsymbol{\alpha}^{*\intercal}\mathbf{C}\mathbf{y}$, i.e. it is an optimal solution of the multiobjective FOP with weight vector $\boldsymbol{\alpha}^*$. Since $\mathcal{S}^k \subseteq \mathcal{S}$, we then have,

$$\begin{aligned}
\xi^k_{\text{ABS}}(\boldsymbol{\alpha}^*) &= \boldsymbol{\alpha}^{*\intercal}\mathbf{C}\hat{\mathbf{y}} - \min_{\mathbf{y}\in\mathcal{S}^k} \boldsymbol{\alpha}^{*\intercal}\mathbf{C}\mathbf{y} \\
\xi^k_{\text{ABS}}(\boldsymbol{\alpha}^*) &= \boldsymbol{\alpha}^{*\intercal}\mathbf{C}\hat{\mathbf{y}} - \boldsymbol{\alpha}^{*\intercal}(\lambda_0\mathbf{C}\mathbf{y}^{(0)} + \lambda_1\mathbf{C}\mathbf{y}^{(1)} + \cdots + \lambda_k\mathbf{C}\mathbf{y}^{(k)})
\end{aligned}$$

$$\xi^k_{\text{ABS}}(\boldsymbol{\alpha}^*) = \xi_{\text{ABS}}(\boldsymbol{\alpha}^*)$$

Proof of (*ii*): Let I denote the the set of indices $i \in 0 \dots k$ for which $\lambda_i > 0$. As stated above, this implies that each $\mathbf{y}^{(i)} \in \mathcal{S}^k,\ i \in I$ is a minimizer of $\min_{\mathbf{y} \in \mathcal{S}^k} \boldsymbol{\alpha}^{*\intercal}\mathbf{C}\mathbf{y}$. For any $\boldsymbol{\alpha} \in \mathcal{A} \setminus \boldsymbol{\alpha}^*$, we may express $\boldsymbol{\alpha}$ as $\boldsymbol{\alpha}^* + \epsilon$ for some $\epsilon \in \mathbb{R}^{|\mathcal{K}|},\ \epsilon^\intercal \mathbf{1} = 0, \epsilon \neq \mathbf{0}$. As such we have,

$$\xi^k_{\text{ABS}}(\boldsymbol{\alpha}) = \boldsymbol{\alpha}^\intercal \mathbf{C}\hat{\mathbf{y}} - \min_{\mathbf{y} \in \mathcal{S}^k} \boldsymbol{\alpha}^\intercal \mathbf{C}\mathbf{y}$$
$$\xi^k_{\text{ABS}}(\boldsymbol{\alpha}) = (\boldsymbol{\alpha}^* + \epsilon)^\intercal \mathbf{C}\hat{\mathbf{y}} - \min_{\mathbf{y} \in \mathcal{S}^k} (\boldsymbol{\alpha}^* + \epsilon)^\intercal \mathbf{C}\mathbf{y}$$

Let $\overline{\mathbf{y}}$ denote $\min_{\mathbf{y} \in \mathcal{S}^k, i \in I} (\epsilon)^\intercal \mathbf{C}\mathbf{y}$. Since $\mathbf{C}\hat{\mathbf{y}} - \xi_{\text{ABS}}(\boldsymbol{\alpha}^*)\mathbf{1}$ lies in the *interior* of a facet of $\text{conv}(B^k)$, then it must be true that there is a set of $|\mathcal{K}|$ linearly independent elements of $\mathbf{C}\mathbf{y}^{(i)} \in B^k$ such that $i \in I$. Let us denote this set $\overline{B}$. We note that $\text{conv}(\overline{B})$ is a subset of a facet of $\text{conv}(B^k)$, and that $\mathbf{C}\hat{\mathbf{y}} - \xi_{\text{ABS}}(\boldsymbol{\alpha}^*)\mathbf{1}$ is contained in the interior of this subset. The elements of $\overline{B}$ define a set of dimension $|\mathcal{K}| - 1$ contained in the plane $\boldsymbol{\alpha}^{*\intercal}\mathbf{C}\mathbf{y} = (\min_{\mathbf{y} \in \mathcal{S}} \boldsymbol{\alpha}^{*\intercal}\mathbf{C}\mathbf{y})$. Since the elements of $\overline{B}$ are linearly independent and $\boldsymbol{\alpha}^*$ and ϵ are not parallel vectors, it must be that there is at least one element $\mathbf{C}\tilde{\mathbf{y}} \in \overline{B}$ such that $\epsilon^\intercal \mathbf{C}\tilde{\mathbf{y}} > \epsilon^\intercal \mathbf{C}\overline{\mathbf{y}}$. Then it must also be true that $\epsilon^\intercal \mathbf{C}\overline{\mathbf{y}} < \epsilon^\intercal(\sum_{i \in I} \lambda_i \mathbf{C}\mathbf{y}^{(i)}) = \epsilon^\intercal(\mathbf{C}\hat{\mathbf{y}} - \xi_{\text{ABS}}(\boldsymbol{\alpha}^*)\mathbf{1})$. With this, we then have that

$$\xi^k_{\text{ABS}}(\boldsymbol{\alpha}) \geq \boldsymbol{\alpha}^{*\intercal}\mathbf{C}\hat{\mathbf{y}} - \boldsymbol{\alpha}^{*\intercal}\mathbf{C}\overline{\mathbf{y}} + \epsilon^\intercal \mathbf{C}\hat{\mathbf{y}} - \epsilon^\intercal \mathbf{C}\overline{\mathbf{y}}$$
$$\xi^k_{\text{ABS}}(\boldsymbol{\alpha}) \geq \xi^k_{\text{ABS}}(\boldsymbol{\alpha}^*) + \epsilon^\intercal \mathbf{C}\hat{\mathbf{y}} - \epsilon^\intercal \mathbf{C}\overline{\mathbf{y}}$$
$$\xi^k_{\text{ABS}}(\boldsymbol{\alpha}) > \xi^k_{\text{ABS}}(\boldsymbol{\alpha}^*) + \epsilon^\intercal \mathbf{C}\hat{\mathbf{y}} - \epsilon^\intercal(\mathbf{C}\hat{\mathbf{y}} - \xi_{\text{ABS}}(\boldsymbol{\alpha}^*)\mathbf{1})$$
$$\xi^k_{\text{ABS}}(\boldsymbol{\alpha}) > \xi^k_{\text{ABS}}(\boldsymbol{\alpha}^*) + (\epsilon^\intercal \mathbf{C}\hat{\mathbf{y}} - \epsilon^\intercal \mathbf{C}\hat{\mathbf{y}}) + \xi_{\text{ABS}}(\boldsymbol{\alpha}^*)\epsilon^\intercal \mathbf{1}$$
$$\xi^k_{\text{ABS}}(\boldsymbol{\alpha}) > \xi^k_{\text{ABS}}(\boldsymbol{\alpha}^*) + \xi_{\text{ABS}}(\boldsymbol{\alpha}^*) * 0$$
$$\xi^k_{\text{ABS}}(\boldsymbol{\alpha}) > \xi^k_{\text{ABS}}(\boldsymbol{\alpha}^*)$$

□

A.1.2 Proof of proposition 2.8

Proof. Proof of proposition 2.8: In order to show that the algorithm terminates, we first demonstrate that the algorithm will always reach a termination check criterion in a finite number of iterations, until one such check succeeds. We first show that for any algorithm step k, either $\nabla \xi_{\text{ABS}}(\boldsymbol{\alpha}^{(k)}) = 0$ and the termination check will occur, or there exists a step $k+n$ such that $\boldsymbol{\alpha}^{(k+n)\intercal} C\hat{\mathbf{y}} - \boldsymbol{\alpha}^{(k+n)\intercal} C\mathbf{y}^{(k+n)} > \xi_{\text{ABS}} + (\boldsymbol{\alpha}^{(k+n)} - \boldsymbol{\alpha}^{(k+n-1)})^\intercal (C(\hat{\mathbf{y}} - \mathbf{y}^{(k+n-1)}))$ and $\mathbf{y}^{(k+n)} \in \mathcal{S}^{(k+n-1)}$. At a given step k, $\boldsymbol{\alpha}^{(k)}$ is found on some facet of the gap function

ξ_{ABS} defined by $\mathbf{y}^{(k)}$. If $\nabla\xi_{\text{ABS}}(\boldsymbol{\alpha}^{(k)}) \neq 0$, then the descent method will proceed to take identical steps until some $\boldsymbol{\alpha}^{(k+n)}$ either leaves said facet, in which case $\mathbf{y}^{(k+n)} \neq \mathbf{y}^{(k+n-1)}$, or arrives at the boundary of $\mathcal{A}$ and $\boldsymbol{\alpha}^{(k+n)} = \boldsymbol{\alpha}^{(k+n-1)}$, at which point a termination check will occur. If $\boldsymbol{\alpha}^{(k+n)}$ leaves said facet, then $\mathbf{y}^{(k+n)} \neq \mathbf{y}^{(k+n-1)}$, and since the newfound facet has a different subgradient, the gap function will not decrease by the maximum amount, and thus $\boldsymbol{\alpha}^{(k+n)\intercal}C\hat{\mathbf{y}} - \boldsymbol{\alpha}^{(k+n)\intercal}C\mathbf{y}^{(k+n)} > \xi_{\text{ABS}} + (\boldsymbol{\alpha}^{(k+n)} - \boldsymbol{\alpha}^{(k+n-1)})^\intercal(C(\hat{\mathbf{y}} - \mathbf{y}^{(k+n-1)}))$. Since conv$(\mathcal{S})$ has a finite number of extreme points, there exists some step $k+n$ where $\boldsymbol{\alpha}^{(k+n)\intercal}C\hat{\mathbf{y}} - \boldsymbol{\alpha}^{(k+n)\intercal}C\mathbf{y}^{(k+n)} > \xi_{\text{ABS}} + (\boldsymbol{\alpha}^{(k+n)} - \boldsymbol{\alpha}^{(k+n-1)})^\intercal(C(\hat{\mathbf{y}} - \mathbf{y}^{(k+n-1)}))$ and $\mathbf{y}^{(k+n)} \in \mathcal{S}^{(k+n-1)}$. Thus, Algorithm 1 will always reach a termination check in a finite number of steps, and will continue to do so until one succeeds and terminates the algorithm.

Since each failed termination check must necessarily find a vertex of conv$(\mathcal{B})$ that is not in B^k and then adds it to B^{k+1}, and conv$(\mathcal{B})$ has a finite number of vertices, there must be a finite number of failed termination checks until one succeeds, and by Proposition 2.6, yields the correct solution. Thus, Algorithm 1 terminates in a finite number of steps. □

A.1.3 Proof of theorem 2.10

Proof. Proof of Theorem 2.10: Let $\mathcal{B}_i = \{\mathbf{Cy},\ \forall \mathbf{y} \in \mathcal{S}_i\}$. Since ξ_{ENS} lower bounds ξ_{ABS}, we have by Proposition 2.6 that if $\mathcal{B}_1 \cup \mathcal{B}_2 \cup \cdots \cup \mathcal{B}_n$ contains a set $\bar{B}$ of values of $\mathbf{Cy}$ whose convex hull contains $\mathbf{C}\hat{\mathbf{y}} - \xi_{\text{ABS}}(\boldsymbol{\alpha}^*)$, then ξ_{ENS} and ξ_{ABS} will have the same minimum value and minimizer. We also know from Proposition 2.6 that there exists some set $\bar{B}$ with a total of $|\mathcal{K}|$ elements. Each $\mathbf{Cy} \in \bar{B}$ is the objective vector of a given partitioning of G into k districts. We note that if the i^{th} coarsening of a graph $G = \{V, E\}$ does not contract any edges in the edge cut of said districting, then there is a feasible solution to FOP$_i$ that corresponds to the same districting. Since any such districting is a subgraph of G with L disjoint connected components that covers V, there are at least $|V| - L$ edges in E do not traverse the edge cut. Further any coarsening of G down to v vertices must contract exactly $|V| - v$ edges. Thus, if $v \geq L$ there exists at least one way of coarsening G such that no edges in the edge cut are contracted. Assuming that our ensemble generation method has a non-zero probability of generating any possible coarsening of G down to v vertices for any coarsening, then for any $\mathbf{Cy} \in \bar{B}$ and any coarsening i in our ensemble, Prob$(\mathbf{Cy} \notin \mathcal{B}_i) < 1$. If each coarsening in our ensemble is independently sampled, then as $n \to \infty$:

$$\text{Prob}(\mathbf{Cy} \notin \mathcal{B}_1 \cup \mathcal{B}_2 \cup \cdots \cup \mathcal{B}_n) = \prod_{i \in 1 \ldots n} \text{Prob}(\mathbf{Cy} \notin \mathcal{B}_i) \to 0$$

Since $\text{Prob}(\bar{B} \nsubseteq \mathcal{B}_1 \cup \mathcal{B}_2 \cup \cdots \cup \mathcal{B}_n) = \text{Prob}(\bigcup_{\mathbf{b} \in \bar{B}} (\mathbf{b} \notin \mathcal{B}_1 \cup \mathcal{B}_2 \cup \cdots \cup \mathcal{B}_n))$, we also have

that as $n \to \infty$, Prob$(\bar{B} \not\subseteq \mathcal{B}_1 \cup \mathcal{B}_2 \cup \cdots \cup \mathcal{B}_n) \to 0$.

A.1.4 Proof of proposition 2.11

Proof. Proof of Proposition 2.11: If we can find a subtangent plane of $\xi_{\text{ENS}}(\boldsymbol{\alpha})$ at $\boldsymbol{\alpha}^{(k)}$, then the gradient of said plane is a subgradient of $\xi_{\text{ENS}}(\boldsymbol{\alpha})$. Let us define our subtangent plane at $\boldsymbol{\alpha}^{(k)}$ as $P(\boldsymbol{\alpha}) = \boldsymbol{\alpha}^\intercal \mathbf{C}\hat{\mathbf{y}} - \boldsymbol{\alpha}^\intercal \mathbf{C}(\underset{\mathbf{y} \in \mathbf{y}_1^{(k)} \cdots \mathbf{y}_n^{(k)}}{\arg\min}\ \boldsymbol{\alpha}^{(k)\intercal}\mathbf{C}\mathbf{y})$. Then, $\nabla P(\boldsymbol{\alpha}) = \mathbf{C}(\hat{\mathbf{y}} - \underset{\mathbf{y} \in \mathbf{y}_1^{(k)} \cdots \mathbf{y}_n^{(k)}}{\arg\min}\ (\boldsymbol{\alpha}^{(k)\intercal}\mathbf{C}\mathbf{y}))$. To show that $P(\boldsymbol{\alpha})$ is subtangent at $\boldsymbol{\alpha}^{(k)}$, we show that (i) $P(\boldsymbol{\alpha}^{(k)}) = \xi_{\text{ENS}}(\boldsymbol{\alpha}^{(k)})$, and (ii) $P(\boldsymbol{\alpha}) \leq \xi_{\text{ENS}}(\boldsymbol{\alpha})$.

Proof of (i): we note by the formulation of MultiGIO$_{\text{ABS}}$MinMax that $\xi_{\text{ENS}}(\boldsymbol{\alpha}) = \max_{i \in 1 \ldots n} \xi_i(\boldsymbol{\alpha})$ where $\xi_i(\boldsymbol{\alpha})$ denotes the gap function generated by the inverse formulation GIO$_{\text{ABS}}$ applied to FOP$_i$. Then we have that:

$$\begin{aligned}
\xi_{\text{ENS}}(\boldsymbol{\alpha}) &= \max_{i \in 1 \ldots n} \xi_i(\boldsymbol{\alpha}) \\
\xi_{\text{ENS}}(\boldsymbol{\alpha}) &= \max_{i \in 1 \ldots n} (\boldsymbol{\alpha}^\intercal \mathbf{C}\hat{\mathbf{y}} - \min_{\mathbf{y} \in \mathcal{S}_i} \boldsymbol{\alpha}^\intercal \mathbf{C}\mathbf{y})
\end{aligned}$$

By Proposition 2.3, we have that:

$$\begin{aligned}
\xi_{\text{ENS}}(\boldsymbol{\alpha}^{(k)}) &= \max_{i \in 1 \ldots n} (\boldsymbol{\alpha}^{(k)\intercal} \mathbf{C}\hat{\mathbf{y}} - \boldsymbol{\alpha}^{(k)\intercal} \mathbf{C}\mathbf{y}_i^{(k)}) \\
\xi_{\text{ENS}}(\boldsymbol{\alpha}^{(k)}) &= \boldsymbol{\alpha}^{(k)\intercal} \mathbf{C}\hat{\mathbf{y}} - \boldsymbol{\alpha}^{(k)\intercal} \mathbf{C}(\underset{\mathbf{y} \in \mathbf{y}_1^{(k)} \cdots \mathbf{y}_n^{(k)}}{\arg\min}\ \boldsymbol{\alpha}^{(k)\intercal}\mathbf{C}\mathbf{y}) \\
\xi_{\text{ENS}}(\boldsymbol{\alpha}^{(k)}) &= P(\boldsymbol{\alpha}^{(k)})
\end{aligned}$$

Proof of (ii):

$$\begin{aligned}
\xi_{\text{ENS}}(\boldsymbol{\alpha}) &= \max_{i \in 1 \ldots n} (\boldsymbol{\alpha}^\intercal \mathbf{C}\hat{\mathbf{y}} - \min_{\mathbf{y} \in \mathcal{S}_i} \boldsymbol{\alpha}^\intercal \mathbf{C}\mathbf{y}) \\
\xi_{\text{ENS}}(\boldsymbol{\alpha}) &= \boldsymbol{\alpha}^\intercal \mathbf{C}\hat{\mathbf{y}} - \min_{\mathbf{y} \in \mathcal{S}_1 \ldots \mathcal{S}_n} \boldsymbol{\alpha}^\intercal \mathbf{C}\mathbf{y} \\
\xi_{\text{ENS}}(\boldsymbol{\alpha}) &\geq \boldsymbol{\alpha}^\intercal \mathbf{C}\hat{\mathbf{y}} - \boldsymbol{\alpha}^\intercal \mathbf{C}(\underset{\mathbf{y} \in \mathbf{y}_1^{(k)} \cdots \mathbf{y}_n^{(k)}}{\arg\min}\ \boldsymbol{\alpha}^{(k)\intercal}\mathbf{C}\mathbf{y}) \\
\xi_{\text{ENS}}(\boldsymbol{\alpha}) &\geq P(\boldsymbol{\alpha})
\end{aligned}$$

A.2 Relative sub-optimality loss function

The relative sub-optimality loss function, $\ell_{\text{rel}}(\hat{\mathbf{y}}, \mathcal{S}, \boldsymbol{\alpha}) = \frac{\boldsymbol{\alpha}^\intercal \mathbf{C}\hat{\mathbf{y}}}{\min_{\mathbf{y} \in \mathcal{S}} \boldsymbol{\alpha}^\intercal \mathbf{C}\mathbf{y}}$ measures the quotient of the input solution objective value and the optimal objective value. We note that this loss

function is well defined only if $\boldsymbol{\alpha} \in \mathbb{R}_+^{|\mathcal{K}|} \setminus \{\mathbf{0}\}$ and $\mathbf{C}\mathbf{y} > \mathbf{0}, \forall \mathbf{y} \in \mathcal{S}$. The corresponding data-driven inverse optimization problem with relative sub-optimality as a loss function can equivalently be written as:

$$\begin{aligned} \underset{\boldsymbol{\alpha},\, \xi_{\text{rel}}}{\text{minimize}} \quad & \xi_{\text{rel}} \\ \text{subject to} \quad & \boldsymbol{\alpha}^\intercal \mathbf{C}\hat{\mathbf{y}} \leq (\boldsymbol{\alpha}^\intercal \mathbf{C}\mathbf{y})\xi_{\text{rel}}, \quad \forall\, \mathbf{y} \in \mathcal{S}, \\ & \boldsymbol{\alpha} \in \mathcal{A}, \\ & \boldsymbol{\alpha} \geq 0. \end{aligned} \tag{GIO$_{\text{rel}}$}$$

Note that GIO$_{\text{rel}}$ is not a linear program because two continuous decision variables are multiplied ($\boldsymbol{\alpha}$ and ξ_{rel}) in the first set of constraints. Chan et al. (2014) solve a similar formulation by varying ξ_{rel} with a univariate search technique until the smallest value for ξ is found such that GIO$_{\text{rel}}$ is feasible. However in the case that $\mathcal{A}$ is defined as the unit simplex, using a method similar to Chan et al. (2019), the exact minimum relative gap and corresponding objective weighting can be derived quickly from the solution of a single linear program.

Theorem A.1. *$\{\bar{\boldsymbol{\alpha}}, \bar{\xi}_{rel}\}$ is an optimal solution to* (GIO$_{\text{rel}}$) *if $\bar{\boldsymbol{\alpha}} = \frac{\hat{\boldsymbol{\alpha}}}{||\hat{\boldsymbol{\alpha}}||_1}$ and $\bar{\xi}_{rel} = \hat{\boldsymbol{\alpha}}^\intercal \mathbf{C}\hat{\mathbf{y}}$, where $\hat{\boldsymbol{\alpha}}$ is an optimal solution to the following linear program* (GIO$^*_{\text{rel}}$)*:*

$$\begin{aligned} \underset{\boldsymbol{\alpha}}{\text{minimize}} \quad & \boldsymbol{\alpha}^\intercal \mathbf{C}\hat{\mathbf{y}} \\ \textit{subject to} \quad & \boldsymbol{\alpha}^\intercal \mathbf{C}\mathbf{y} \geq 1, \quad \forall\, \mathbf{y} \in \mathcal{S}, \\ & \boldsymbol{\alpha} \geq 0. \end{aligned} \tag{GIO$^*_{\text{rel}}$}$$

Proof. Proof of Theorem A.1: First, we show that (i) $\{\bar{\boldsymbol{\alpha}}, \bar{\xi}_{\text{rel}}\}$ is a feasible solution to (GIO$_{\text{rel}}$). Next, we show that (ii) if $\{\bar{\boldsymbol{\alpha}}, \bar{\xi}_{\text{rel}}\}$ is not optimal for (GIO$_{\text{rel}}$), then $\hat{\boldsymbol{\alpha}}$ is not optimal for for (GIO$^*_{\text{rel}}$), thus proving the claim by contrapositive.

(i) We observe that $||\bar{\boldsymbol{\alpha}}||_1 = 1$ and $\bar{\boldsymbol{\alpha}} \geq \mathbf{0}$ by the construction of $\bar{\boldsymbol{\alpha}}$ and $\bar{\xi}_{\text{rel}} = \hat{\boldsymbol{\alpha}}^\intercal C\hat{\mathbf{y}} \geq 0$ because $C\hat{\mathbf{y}} \in \mathbb{R}_+^{|\mathcal{K}|}$. The remaining constraints are satisfied because $\forall \mathbf{y} \in \mathcal{S}$:

$$\bar{\boldsymbol{\alpha}}^\intercal C\hat{\mathbf{y}} = \frac{\hat{\boldsymbol{\alpha}}^\intercal C\hat{\mathbf{y}}}{||\hat{\boldsymbol{\alpha}}||_1} = \frac{\bar{\xi}_{\text{rel}}}{||\hat{\boldsymbol{\alpha}}||_1} \leq (\hat{\boldsymbol{\alpha}}^\intercal C\mathbf{y})\frac{\bar{\xi}_{\text{rel}}}{||\hat{\boldsymbol{\alpha}}||_1} = \frac{\hat{\boldsymbol{\alpha}}^\intercal C\mathbf{y}}{||\hat{\boldsymbol{\alpha}}||_1}\bar{\xi}_{\text{rel}} = (\bar{\boldsymbol{\alpha}}^\intercal C\mathbf{y})\bar{\xi}_{\text{rel}}.$$

(ii) Suppose there exists a feasible solution $\{\boldsymbol{\alpha}^*, \xi^*_{\text{rel}}\}$ to (GIO$_{\text{rel}}$) such that $\xi^*_{\text{rel}} < \bar{\xi}_{\text{rel}}$, i.e. $\{\bar{\boldsymbol{\alpha}}, \bar{\xi}_{\text{rel}}\}$ is not an optimal solution. Let $\tilde{\boldsymbol{\alpha}} = \frac{\xi^*_{\text{rel}}\boldsymbol{\alpha}^*}{\boldsymbol{\alpha}^{*\intercal} C\hat{\mathbf{y}}}$. Then, $\tilde{\boldsymbol{\alpha}}$ is a feasible solution to (GIO$^*_{\text{rel}}$)

because $\tilde{\boldsymbol{\alpha}} \geq \mathbf{0}$ and $\forall \mathbf{y} \in \mathcal{S}$:

$$\tilde{\boldsymbol{\alpha}}^{\intercal} C\mathbf{y} = \frac{\xi^*_{\text{rel}}}{\boldsymbol{\alpha}^{*\intercal} C\hat{\mathbf{y}}}(\boldsymbol{\alpha}^{*\intercal} C\mathbf{y}) \geq \frac{\xi^*_{\text{rel}}}{\boldsymbol{\alpha}^{*\intercal} C\hat{\mathbf{y}}} \frac{\boldsymbol{\alpha}^{*\intercal} C\hat{\mathbf{y}}}{\xi^*_{\text{rel}}} = 1.$$

Since $\tilde{\boldsymbol{\alpha}}^{\intercal} C\hat{\mathbf{y}} = \xi^*_{\text{rel}} < \bar{\xi}_{\text{rel}} = \hat{\boldsymbol{\alpha}}^{\intercal} C\hat{\mathbf{y}}$, $\hat{\boldsymbol{\alpha}}$ is not an optimal solution to ($\text{GIO}^*_{\text{rel}}$), which contradicts the definition of $\hat{\boldsymbol{\alpha}}$. Therefore, such an $\{\boldsymbol{\alpha}^*, \xi^*_{\text{rel}}\}$ cannot exist, so $\{\bar{\boldsymbol{\alpha}}, \bar{\xi}_{\text{rel}}\}$ must be an optimal solution for (GIO_{rel}). □

Theorem A.1 shows that we can obtain the optimal solution to GIO_{rel} (which is bilinear) by solving $\text{GIO}^*_{\text{rel}}$ (which is a linear program).

As such, our modification to the algorithm proposed by Moghaddass and Terekhov (2020) minimizes the relative sub-optimality loss function, rather than absolute sub-optimality. The algorithm structure is shown in Algorithm 2. The main distinction is that our algorithm solves $\text{GIO}^*_{\text{rel}}$ and uses Theorem A.1 to obtain the solution to GIO_{rel}.

Theorem A.2. *Algorithm 2 terminates finitely with an optimal solution to GIO_{rel}.*

Proof. Proof of Theorem A.2: The proof follows directly from the proof of termination and correctness in Moghaddass and Terekhov (2020), together with the proof of Theorem A.1. □

Theorem A.2 demonstrates that Algorithm 2 produces the optimal solution to GIO_{rel} when the complete forward problem feasible region is known, in at most as many steps as there are extreme points to the forward multi-objective feasible region.

A.3 Adapting gap-gradient methods to the relative sub-optimality loss function

In the case of the relative optimality gap, we apply gap-gradient methods to an inverse formulation based on the linearization GIO^*_{REL}. Thus, for relative gap problems, we can formulate the corresponding gap function as $\xi_{\text{REL}}(\boldsymbol{\alpha}) = \boldsymbol{\alpha}^{\intercal}\mathbf{C}\hat{\mathbf{y}}$ defined over the gap function domain $\mathcal{A} = \{\boldsymbol{\alpha} \in \mathbb{R}^{\mathcal{K}} \mid \boldsymbol{\alpha}^{\intercal} C\mathbf{y} \geq 1, \boldsymbol{\alpha} \geq 0, \forall \mathbf{y} \in \mathcal{S}\}$

Similar to the absolute gap case, we note that the feasible region of the master problem $\text{GIO}^*_{\text{rel}}$ in the case where all extreme points are known is the epigraph of the relative gap function, and we obtain the following result.

Proposition A.3. *The relative gap function $\xi_{REL}(\boldsymbol{\alpha})$ is a convex function.*

Algorithm 2: Data-driven relative gap cutting-plane method.

input : C, $\boldsymbol{y}^0$, FP
output: $\boldsymbol{\alpha}^{\text{best}}$, $\xi_{\text{rel}}^{\text{best}}$
$\tilde{\mathcal{S}} \leftarrow \emptyset$, $\hat{\boldsymbol{\alpha}} \leftarrow \text{GenInvOp}^*_{\text{rel}}(\boldsymbol{y}^0, \tilde{\mathcal{S}})$;
$\boldsymbol{\alpha} \leftarrow \frac{\hat{\boldsymbol{\alpha}}}{||\hat{\boldsymbol{\alpha}}||_1}$, $\boldsymbol{y} \leftarrow \text{FP}(\boldsymbol{\alpha})$;
$\xi_{\text{rel}}^{\text{best}} \leftarrow \boldsymbol{\alpha}^\intercal C\boldsymbol{y}^0 - \boldsymbol{\alpha}^\intercal C\boldsymbol{y}$, $\boldsymbol{\alpha}^{\text{best}} \leftarrow \boldsymbol{\alpha}$;
while $\boldsymbol{\alpha}^\intercal C\boldsymbol{y}^0 > \boldsymbol{\alpha}^\intercal C\boldsymbol{y}$ **do**
  $\tilde{\mathcal{S}} \leftarrow \tilde{\mathcal{S}} \cup \boldsymbol{y}$;
  $\hat{\boldsymbol{\alpha}} \leftarrow \text{GenInvOp}^*_{\text{rel}}(\boldsymbol{y}^0, \tilde{\mathcal{S}})$;
  $\boldsymbol{\alpha} \leftarrow \frac{\hat{\boldsymbol{\alpha}}}{||\hat{\boldsymbol{\alpha}}||_1}$;
  $\boldsymbol{y} \leftarrow \text{FP}(\boldsymbol{\alpha})$;
  if $\boldsymbol{\alpha}^\intercal C\boldsymbol{y}^0 - \boldsymbol{\alpha}^\intercal C\boldsymbol{y} = \xi_{rel}^{best}$ **then**
    if $\boldsymbol{\alpha} = \boldsymbol{\alpha}^{best}$ **then**
      stop
    else
      $\boldsymbol{\alpha}^{\text{best}} \leftarrow \boldsymbol{\alpha}$;
    end
    if $\boldsymbol{\alpha}^\intercal C\boldsymbol{y}^0 - \boldsymbol{\alpha}^\intercal C\boldsymbol{y} < \xi_{abs}^{best}$ **then**
      $\xi_{\text{abs}}^{\text{best}} \leftarrow \boldsymbol{\alpha}^\intercal C\boldsymbol{y}^0 - \boldsymbol{\alpha}^\intercal C\boldsymbol{y}$;
      $\boldsymbol{\alpha}^{\text{best}} \leftarrow \boldsymbol{\alpha}$;
    end
  end
end

Proof. Proof of Theorem A.3: The gap function is a linear function, and as such is convex if its domain is convex. The domain of the gap function is the union of half-planes, and as such, is convex. □

Figure A.1(b) displays the domain of ξ_{REL} for an example inverse optimization problem with two objectives, and the vector $\mathbf{C}\hat{\mathbf{y}}$ which is equal to the gradient of the gap function $\nabla\xi_{\text{REL}}(\boldsymbol{\alpha})$. Minimizing the gap function yields the minimizer $\boldsymbol{\alpha}^*$.

Remark A.4. *Under the formulation of the relative gap space and relative gap function, each FOP extreme point $\mathbf{y}^{(k)}$ does not yield a tangent plane, but a facet of the boundary of the domain of the relative gap function. For the relative gap function the gradient is in fact uniform ($\nabla g_{REL}(\boldsymbol{\alpha}^{(k)}) = \mathbf{C}\hat{\mathbf{y}}, \forall \boldsymbol{\alpha}^{(k)}$ in the interior of the relative gap function's domain), but each facet gained from an extreme point can be used to find the projection of the gradient onto the boundary of the relative gap function domain at that point. Specifically, for a facet-defining boundary plane $\boldsymbol{\alpha}^\intercal C\mathbf{y}^{(k)} = 1$ discovered by solving the forward problem at objective weighting $\boldsymbol{\alpha}^{(k)}$, the gradient projected onto this boundary is*

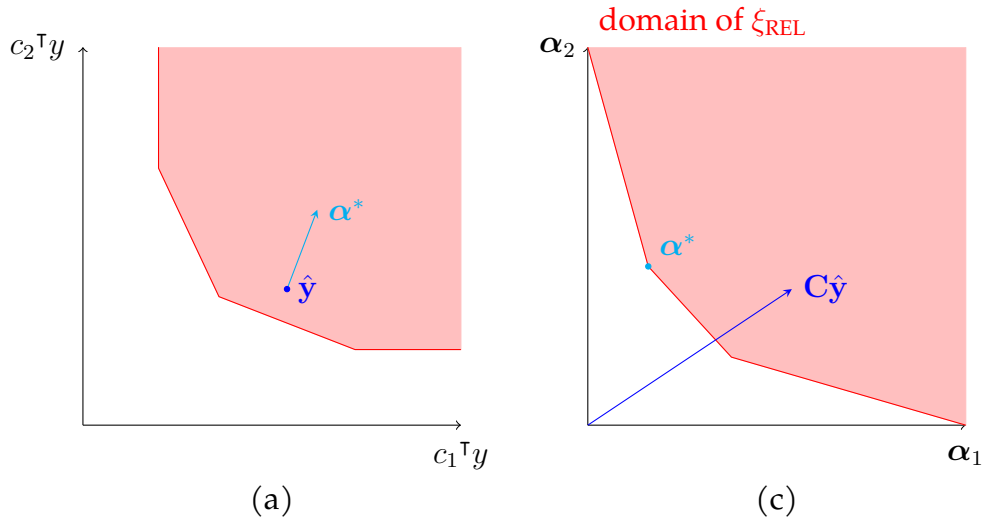


Figure A.1: An example inverse optimization problem with two sub-objectives. (a) The FOP objective feasible space and inverse input $\hat{\mathbf{y}}$, and (b) the corresponding relative gap function projected onto two dimensions, and the gradient of the gap function $\nabla\xi_{\text{REL}}(\boldsymbol{\alpha}) = \mathbf{C}\hat{\mathbf{y}}$.

$(\nabla g_{REL}(\boldsymbol{\alpha}^{(k)}))_{proj} = C\hat{\mathbf{y}} - (\frac{(C\hat{\mathbf{y}})^\intercal C\mathbf{y}^{(k)}}{||C\mathbf{y}^{(k)}||_2^2})C\mathbf{y}^{(k)}$. *Similar to the absolute sub-optimality loss function, this can be used to minimize the convex relative gap function in a bounded space.*

A gap-gradient solution method that uses this subgradient could proceed at each iteration by taking a step in the direction of the projected gradient. While it is possible that the $\boldsymbol{\alpha}^{(k)}$ generated by this step lies outside of the domain $\mathcal{A}$ due to incomplete knowledge of its boundary, we note that so long as $\boldsymbol{\alpha}^{(k)}$ lies within the positive orthant, solving FOP($\boldsymbol{\alpha}^{(k)}$) will yield the same solution $\mathbf{y}^{(k)}$ as an FOP with some scalar-multiplied objective weight $m\boldsymbol{\alpha}^{(k)}$, $m > 1$ such that $m\boldsymbol{\alpha}^{(k)} \in \mathcal{A}$. Once $\mathbf{y}^{(k)}$ is known then the fact-defining plane $\boldsymbol{\alpha}^\intercal\mathbf{C}\mathbf{y}^{(k)} = 1$ can be used to project $\boldsymbol{\alpha}^{(k)}$ onto the domain boundary by way of the projection $m = \frac{1}{\boldsymbol{\alpha}^{(k)\intercal}\mathbf{C}\mathbf{y}^{(k)}}$. One can then proceed in the gap-gradient algorithm by taking a step from $m\boldsymbol{\alpha}^{(k)}$ in the direction of the next projected (negative) gradient, $-\mathbf{C}\hat{\mathbf{y}} + (\frac{(C\hat{\mathbf{y}})^\intercal C\mathbf{y}^{(k)}}{||C\mathbf{y}^{(k)}||_2^2})C\mathbf{y}^{(k)}$.

A.4 Frank-Wolfe gap gradient methods with FOP early stopping

To operationalize this method of applying early stopping to solving the FOP when sufficient information is known, we can use the following approach. First, we create $|\mathcal{K}|$ different variations of the forward problem, each with an additional set of constraints $\mathbf{C}_k(\mathbf{y} - \hat{\mathbf{y}}) \leq \mathbf{C}_l(\mathbf{y} - \hat{\mathbf{y}})$, $\forall l \in 1 \ldots \mathcal{K} \setminus k$, which enforces that subobjective component k of any feasible solution minus that of the input solution must be less than or equal to that of every

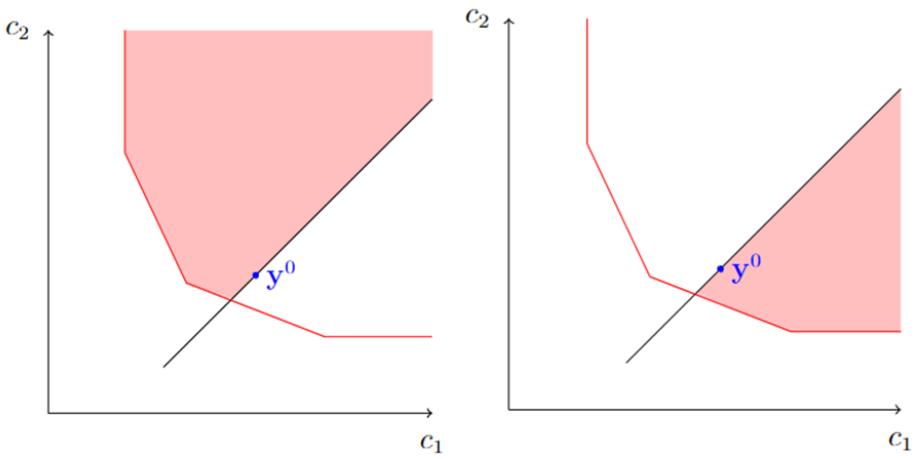


Figure A.2: The feasible regions of two concurrently partially solved variations of the FOP shown in Figure 2.1 (a), used in conjunction with the proposed Frank-Wolfe method for potentially faster descent steps.

other subobjective component. Figure A.2 illustrates the FOP variations created by these constraints for the example two-objective inverse optimization problem shown in Figure 2.1 (a). Next, we solve each of the FOP$_k$, $k \in \mathcal{K}$ in parallel. Once we reach a point where the lower objective bound of a given FOP$_j$, $j \in \mathcal{K}$, is greater than the lowest upper bound of any other running FOP$_k$, we can terminate the solving process of FOP$_j$. When only one such FOP$_k$ remains, we terminate because it is unnecessary to solve to optimality because we know that whatever the optimal solution is, the gradient will have subobjective k as its greatest component, which is sufficient information to execute the next step in algorithm (the next descent step is in the direction towards $\boldsymbol{e}_k$). This concurrent solving subalgorithm is detailed in the Appendix as Algorithm 6.

Using this method, we can also, at the point of terminating the final running subproblem, return the incumbent solution at the time of termination. This returns a feasible forward solution that produces a lower-bounding plane of the gap function. If it is an interior point, then it will not be tangent to the gap function, but strictly below. However, as our Frank-Wolfe method approaches $\boldsymbol{\alpha}^*$, the concurrent process will be terminating in smaller and smaller FOP optimality gaps, so the returned incumbents will be progressively closer to the boundary of conv$(\mathcal{B})$. Thus, once we are within a close neighborhood of the gap function minimizer, we may collect a set of proper tangent planes (at this point our method is likely to be returning vertices of conv$(\mathcal{B})$), and we can use these gathered tangent planes in a termination check that is similar to the termination methods used for our other proposed methods. The methods discussed in this section currently remain unimplemented because they will likely involve constructing an MILP solver from the ground up.

A.5 Formulation.

The overall mixed-integer linear formulation for the FOP is given by:

$$\underset{x,z^D,v^D,f,\rho,\sigma_A,\phi_{EG}}{\text{minimize}} \quad \alpha_1\rho + \alpha_2\sigma_A + \alpha_3\phi_{EG} \tag{A.1a}$$

subject to

$$\sum_{i\in V} x_{ii} = L, \tag{A.1b}$$

$$\sum_{i\in V} x_{ij} = 1, \qquad \forall\, j \in V, \tag{A.1c}$$

$$x_{ij} \le x_{ii}, \qquad \forall\, i,j \in V, \tag{A.1d}$$

$$x_{ij} + \sum_{v\in N(j)} (f_{ijv} - f_{ivj}) = 0, \qquad \forall i,j \in V, i \neq j, \tag{A.1e}$$

$$x_{ii} + \sum_{v\in N(i)} (f_{iiv} - f_{ivi}) - \sum_{v\in N(i)} x_{iv} = 0, \qquad \forall i \in V, \tag{A.1f}$$

$$|V|x_{ij} - \sum_{v\in N(j)} f_{ivj} \ge 0, \qquad \forall i,j \in V, \tag{A.1g}$$

$$(x_{ii} - \rho)\overline{P} \le \sum_{j\in V} p_j x_{ij}, \qquad \forall i \in V, \tag{A.1h}$$

$$(x_{ii} + \rho)\overline{P} \ge \sum_{j\in V} p_j x_{ij}, \qquad \forall i \in V, \tag{A.1i}$$

$$\sigma_A = \frac{\sum_{i,j\in V} d_{ij}a_j x_{ij}}{M}, \tag{A.1j}$$

$$\overline{P} \le \sum_{j\in V} (p_j^D - p_j^R)x_{ij}, -\overline{P}z_i^D \le 0, \qquad \forall i \in V, \tag{A.1k}$$

$$v_{ij}^D \le x_{ij}, \qquad \forall i,j \in V, \tag{A.1l}$$

$$v_{ij}^D \le z_i^D, \qquad \forall i,j \in V, \tag{A.1m}$$

$$v_{ij}^D \geq x_{ij} + z_i^D - 1, \qquad \forall i, j \in V, \tag{A.1n}$$

$$\sum_{j \in V} (\frac{3p_j^D - p_j^R}{2}) x_{ij} - \sum_{j \in V} (p_j^D + p_j^R) v_{ij}^D = w_i, \qquad \forall i \in V, \tag{A.1o}$$

$$\phi_{EG} \geq \frac{\sum_{i \in V} w_i}{\sum_{i \in V} (p_i^D + p_i^R)}, \tag{A.1p}$$

$$\phi_{EG} \geq -\frac{\sum_{i \in V} w_i}{\sum_{i \in V} (p_i^D + p_i^R)}, \tag{A.1q}$$

$$x_{ij}, z_i^D, v_{ij}^D \in \{0, 1\},\ f_{ivj} \geq 0, \qquad \forall i, j \in V, \forall v \in N(j). \tag{A.1r}$$

Constraints (2b) through (2d) enforce that every vertex is assigned to exactly one district out of L total districts. Constraints (2e) through (2g) construct a set of flow networks that maintain contiguity for the created districts. Constraints (2h) and (2i) define the ρ variable used in the objective. Constraint (2j) defines σ_A (used in the objective). Finally, Constraints (2k) - (2q) define ϕ_{EG}, which is used in the objective.

A.6 Experimental details

A.6.1 MIPLIB instances

The following are labels for the FOPs used in Section 2.5:

1. neos-1430701
2. gsvm2rl3
3. ran13x13
4. spd150x300d
5. supportcase17
6. ci-24
7. ic97_tension
8. fastxgemm-n2r60t2
9. timtab1CUTS

A.6.2 Generating sample states

To create a sample state G for a specified $|V|$, we randomly sample $|V|$ points uniformly over a unit square, and calculate the Delaunay triangulation to create an adjacency graph of our simulated census blocks. The distance matrix is calculated using the euclidean distances of the sampled points. Note that scaling of the distance matrix does not matter, as the compactness metric of a districting is normalized for the 1-median of the entire state. Land areas for the corresponding simulated census blocks are calculated as the areas of the Voronoi cells produced by the sampled points. At each census block, values for the number of voters for party A, party B, and non-voting people are randomly sampled from integers in the range $[10, 100]$. For each sampled state, an FOP is generated for the political districting optimization model with $L = 2$ districts.

A.7 Algorithm Structures

Algorithm 3: Moghaddass and Terekhov (2020)

input : $\mathbf{C}$, $\hat{\mathbf{y}}$, FOP
output: $\boldsymbol{\alpha}^{\text{best}}$, ξ^{best}
$\tilde{\mathcal{S}} \leftarrow \emptyset$, $\boldsymbol{\alpha} \leftarrow \text{GIO}_{\text{ABS}}(\hat{\mathbf{y}}, \tilde{\mathcal{S}})$, $\mathbf{y} \leftarrow \text{FOP}(\boldsymbol{\alpha})$;
$\xi^{\text{best}} \leftarrow \boldsymbol{\alpha}^\intercal \mathbf{C}\hat{\mathbf{y}} - \boldsymbol{\alpha}^\intercal \mathbf{C}\mathbf{y}$, $\boldsymbol{\alpha}^{\text{best}} \leftarrow \boldsymbol{\alpha}$;
while $\boldsymbol{\alpha}^\intercal \mathbf{C}\hat{\mathbf{y}} > \boldsymbol{\alpha}^\intercal C\mathbf{y}$ **do**
  $\tilde{\mathcal{S}} \leftarrow \tilde{\mathcal{S}} \cup \mathbf{y}$;
  $\boldsymbol{\alpha} \leftarrow \text{GIO}_{\text{ABS}}(\hat{\mathbf{y}}, \tilde{\mathcal{S}})$;
  $\mathbf{y} \leftarrow \text{FP}(\boldsymbol{\alpha})$;
  if $\boldsymbol{\alpha}^\intercal C\hat{\mathbf{y}} - \boldsymbol{\alpha}^\intercal C\mathbf{y} = \xi^{best}$ **then**
    if $\boldsymbol{\alpha} = \boldsymbol{\alpha}^{best}$ **then**
      stop
    else
      $\boldsymbol{\alpha}^{\text{best}} \leftarrow \boldsymbol{\alpha}$;
    end
    if $\boldsymbol{\alpha}^\intercal C\hat{\mathbf{y}} - \boldsymbol{\alpha}^\intercal C\mathbf{y} < \xi_{abs}^{best}$ **then**
      $\xi_{\text{abs}}^{\text{best}} \leftarrow \boldsymbol{\alpha}^\intercal C\hat{\mathbf{y}} - \boldsymbol{\alpha}^\intercal C\mathbf{y}$;
      $\boldsymbol{\alpha}^{\text{best}} \leftarrow \boldsymbol{\alpha}$;
    end
  end
end

Algorithm 4: Gap-gradient Projected Gradient Descent with Acceleration.

input : C, $\hat{\mathbf{y}}$, FOP, t, β
output: $\boldsymbol{\alpha}^{\text{best}}$, ξ^{final}
$k = 0$, $\mathcal{S}^k \leftarrow \emptyset$, $\boldsymbol{\alpha}^{(k)} \leftarrow \frac{1}{\mathcal{K}}\mathbf{1}$;
$\mathbf{y}^{(k)} \leftarrow \text{FOP}(\boldsymbol{\alpha}^{(k)})$;
$\xi_{\text{ABS}} \leftarrow \boldsymbol{\alpha}^{(k)\intercal} C\hat{\mathbf{y}} - \boldsymbol{\alpha}^{(k)\intercal} C\mathbf{y}^{(k)}$;
while *True* **do**
  $k \leftarrow k + 1$;
  $\mathcal{S}^k \leftarrow \mathcal{S}^{k-1} \cup \mathbf{y}^{(k-1)}$;
  $\boldsymbol{\alpha}^{(k)} \leftarrow \boldsymbol{\alpha}^{(k-1)} - t(C(\hat{\mathbf{y}} - \mathbf{y}^{(k-1)})) + \beta(\boldsymbol{\alpha}^{(k-1)} - \boldsymbol{\alpha}^{(k-2)})$;
  $\boldsymbol{\alpha}^{(k)} \leftarrow \text{proj}_{\Delta^{\mathcal{K}}}(\boldsymbol{\alpha}^{(k)})$;
  $\mathbf{y}^{(k)} \leftarrow \text{FOP}(\boldsymbol{\alpha}^{(k)})$;
  if $\boldsymbol{\alpha}^{(k)} = \boldsymbol{\alpha}^{(k-1)}$ ***or*** $\boldsymbol{\alpha}^{(k)\intercal} C\hat{\mathbf{y}} - \boldsymbol{\alpha}^{(k)\intercal} C\mathbf{y}^{(k)} > \xi_{ABS} + (\boldsymbol{\alpha}^{(k)} - \boldsymbol{\alpha}^{(k-1)})^\intercal (C(\hat{\mathbf{y}} - \mathbf{y}^{(k-1)}))$
   and $\mathbf{y}^{(k)} \in \mathcal{S}^k$ **then**
    $\boldsymbol{\alpha}^{\text{final}}, \xi^{\text{final}} \leftarrow \text{GIO}_{\text{ABS}}(\hat{\mathbf{y}}, \mathcal{S}^k)$;
    $k \leftarrow k + 1$;
    $\mathbf{y}^{(k)} \leftarrow \text{FOP}(\boldsymbol{\alpha}^{\text{final}})$;
    if $\boldsymbol{\alpha}^{final\intercal} C\hat{\mathbf{y}} - \boldsymbol{\alpha}^{final\intercal} C\mathbf{y}^{(k)} = \xi^{final}$ **then**
      stop
    else
      $\boldsymbol{\alpha}^{(k)} \leftarrow \boldsymbol{\alpha}^{\text{final}}$;
      $\mathcal{S}^k \leftarrow \mathcal{S}^{k-1} \cup \mathbf{y}^{(k-1)}$;
      $t \leftarrow \frac{t}{2}$;
    end
  else
    $\xi_{\text{ABS}} \leftarrow \boldsymbol{\alpha}^{(k)\intercal} C\hat{\mathbf{y}} - \boldsymbol{\alpha}^{(k)\intercal} C\mathbf{y}^{(k)}$;
  end
end

Algorithm 5: Frank-Wolfe Generalized Inverse Method, Absolute Gap

input : C, $\hat{\mathbf{y}}$, FOP
output: $\boldsymbol{\alpha}^{\text{final}}$, ξ^{final}
$k = 0$, $\mathcal{S}^k \leftarrow \emptyset$, $\boldsymbol{\alpha}^{(k)} \leftarrow \frac{1}{\mathcal{K}}\mathbf{1}$;
$\mathbf{y}^{(k)} \leftarrow \text{FOP}(\boldsymbol{\alpha}^{(k)})$;
$\xi_{\text{abs}} \leftarrow \boldsymbol{\alpha}^{(k)\intercal} C\hat{\mathbf{y}} - \boldsymbol{\alpha}^{(k)\intercal} C\mathbf{y}^{(k)}$;
while $\boldsymbol{\alpha}^{(k)\intercal} C\hat{\mathbf{y}} > \boldsymbol{\alpha}^{(k)\intercal} C\mathbf{y}^{(k)}$ **do**
  $k \leftarrow k + 1$;
  $\mathcal{S}^k \leftarrow \mathcal{S}^{k-1} \cup \mathbf{y}^{(k-1)}$;
  $i \leftarrow \arg\min_i C_i(\hat{\mathbf{y}} - \mathbf{y}^{(k-1)})$;
  $\boldsymbol{\alpha}^{(k)} \leftarrow \boldsymbol{\alpha}^{(k-1)} - \frac{2}{2+k}(\boldsymbol{e}_i - \boldsymbol{\alpha}^{(k-1)})$;
  $\mathbf{y}^{(k)} \leftarrow \text{FOP}(\boldsymbol{\alpha}^{(k)})$;
  if $\boldsymbol{\alpha}^{(k)} = \boldsymbol{\alpha}^{(k-1)}$ ***or*** $\boldsymbol{\alpha}^{(k)\intercal} C\hat{\mathbf{y}} - \boldsymbol{\alpha}^{(k)\intercal} C\mathbf{y}^{(k)} > \xi_{ABS} + (\boldsymbol{\alpha}^{(k)} - \boldsymbol{\alpha}^{(k-1)})^\intercal (C(\hat{\mathbf{y}} - \mathbf{y}^{(k-1)}))$
   and $\mathbf{y}^{(k)} \in \mathcal{S}^k$ **then**
    $\boldsymbol{\alpha}^{\text{final}}, \xi^{\text{final}} \leftarrow \text{GIO}_{\text{ABS}}(\hat{\mathbf{y}}, \mathcal{S}^k)$;
    $k \leftarrow k + 1$;
    $\mathbf{y}^{(k)} \leftarrow \text{FOP}(\boldsymbol{\alpha}^{\text{final}})$;
    if $\boldsymbol{\alpha}^{final\intercal} C\hat{\mathbf{y}} - \boldsymbol{\alpha}^{final\intercal} C\mathbf{y}^{(k)} = \xi^{final}$ **then**
      stop
    else
      $\boldsymbol{\alpha}^{(k)} \leftarrow \boldsymbol{\alpha}^{\text{final}}$;
      $\mathcal{S}^k \leftarrow \mathcal{S}^{k-1} \cup \mathbf{y}^{(k-1)}$;
    end
  else
    $\xi_{\text{ABS}} \leftarrow \boldsymbol{\alpha}^{(k)\intercal} C\hat{\mathbf{y}} - \boldsymbol{\alpha}^{(k)\intercal} C\mathbf{y}^{(k)}$;
  end
end

Algorithm 6: Frank-Wolfe Incomplete FP Solve Subalgorithm; Concurrent MIP Approach, Absolute Gap

input : FOP, C, $\boldsymbol{\alpha}$, $\boldsymbol{y}^0$
output: i, $\boldsymbol{y}$
for $i \in 1 \dots \mathcal{K}$ **do**
  FOP$_i \leftarrow$ FOP $\cap \{C_i(\boldsymbol{y} - \boldsymbol{y^0}) \leq C_j(\boldsymbol{y} - \boldsymbol{y^0}) \mid j \in 1 \dots \mathcal{K} \setminus i\}$;
end
$u \leftarrow \infty$;
$p \leftarrow \mathcal{K}$;
$\forall i$, concurrently initiate optimizing FOP$_i$ with objective coefficients $\boldsymbol{\alpha}$;
while *FOP$_i$ is solving* **do**
  if *FOP$_i$.upper_bound* $< u$ **then**
    $u \leftarrow$ FOP$_i$.upper_bound;
  end
  if *FOP$_i$.lower_bound* $> u$ **then**
    $p \leftarrow p - 1$;
    terminate FOP$_i$;
  end
  if $p = 1$ **then**
    $\boldsymbol{y} \leftarrow$ incumbent solution of FOP$_i$;
    terminate FOP$_i$;
    return i, $\boldsymbol{y}$
  end
end

Algorithm 7: Frank-Wolfe Partial FOP Inverse Method

input : $\mathbf{C}$, $\hat{\mathbf{y}}$, FOP
output: $\boldsymbol{\alpha}^{\text{final}}$, ξ^{final}
$k = 0$, $\mathcal{S}^k \leftarrow \emptyset$, $\boldsymbol{\alpha}^{(k)} \leftarrow \frac{1}{K}\mathbf{1}$;
i, $\mathbf{y}^{(k)} \leftarrow$ Alg7(FOP, $\mathbf{C}, \boldsymbol{\alpha}^{(k)}, \hat{\mathbf{y}}$);
$\xi_{\text{ABS}} \leftarrow \boldsymbol{\alpha}^{(k)\intercal} C\hat{\mathbf{y}} - \boldsymbol{\alpha}^{(k)\intercal} C\mathbf{y}^{(k)}$;
while *True* **do**
  $k \leftarrow k + 1$;
  $\mathcal{S}^k \leftarrow \mathcal{S}^{k-1} \cup \mathbf{y}^{(k-1)}$;
  $\boldsymbol{\alpha}^{(k)} \leftarrow \boldsymbol{\alpha}^{(k-1)} - \frac{2}{2+k}(\boldsymbol{e}_i - \boldsymbol{\alpha}^{(k-1)})$;
  i, $\mathbf{y}^{(k)} \leftarrow$ Alg7(FOP, $\mathbf{C}, \boldsymbol{\alpha}^{(k)}, \hat{\mathbf{y}}$);
  if $\boldsymbol{\alpha}^{(k)} = \boldsymbol{\alpha}^{(k-1)}$ ***or*** $\boldsymbol{\alpha}^{(k)\intercal} C\hat{\mathbf{y}} - \boldsymbol{\alpha}^{(k)\intercal} C\mathbf{y}^{(k)} > \xi_{ABS} + (\boldsymbol{\alpha}^{(k)} - \boldsymbol{\alpha}^{(k-1)})^\intercal (C(\hat{\mathbf{y}} - \mathbf{y}^{(k-1)}))$
   and $\mathbf{y}^{(k)} \in \mathcal{S}^k$ **then**
    $\boldsymbol{\alpha}^{\text{final}}, \xi^{\text{final}} \leftarrow \text{GIO}_{\text{ABS}}(\hat{\mathbf{y}}, \mathcal{S}^k)$;
    $k \leftarrow k + 1$;
    $\mathbf{y}^{(k)} \leftarrow \text{FOP}(\boldsymbol{\alpha}^{\text{final}})$;
    if $\boldsymbol{\alpha}^{final\intercal} C\hat{\mathbf{y}} - \boldsymbol{\alpha}^{final\intercal} C\mathbf{y}^{(k)} = \xi^{final}$ **then**
      stop
    else
      $\boldsymbol{\alpha}^{(k)} \leftarrow \boldsymbol{\alpha}^{\text{final}}$;
      $\mathcal{S}^k \leftarrow \mathcal{S}^{k-1} \cup \mathbf{y}^{(k-1)}$;
      $t \leftarrow \frac{t}{2}$;
    end
  else
    $\xi_{\text{ABS}} \leftarrow \boldsymbol{\alpha}^{(k)\intercal} C\hat{\mathbf{y}} - \boldsymbol{\alpha}^{(k)\intercal} C\mathbf{y}^{(k)}$;
  end
end

Algorithm 8: Boosted Coarsening Ensemble Generation Method

input : $G = \{V, E\},\ n,\ \eta$
output: $\mathbb{G}$
$\mathbb{G} \leftarrow \{\}$;
$\mathbf{w}_e \leftarrow 1, \quad \forall e \in E$;
for $i \in$ *range*(n) **do**
  $\mathbf{o}_e \leftarrow \mathbf{w}_e \text{Exp}(\lambda = 1), \quad \forall e \in E$;
  $O \leftarrow [e \in E | \mathbf{o}$ sorted in ascending order$]$;
  $\mathcal{C} \leftarrow \{\}$;
  for $e = (v_1, v_2) \in O$ **do**
    if $\exists c \in \mathcal{C} | v_1 \in c$ *or* $v_2 \in c$ **then**
      $\mathbf{w}_e \leftarrow \eta \mathbf{w}_e$
    else
      $\mathcal{C} \leftarrow \mathcal{C} \cup e$;
      $\mathbf{w}_e \leftarrow \frac{\mathbf{w}_e}{\eta}$
    end
  end
  $G' \leftarrow G$;
  contract all edges e of G' where $e \in \mathcal{C}$;
  $\mathbb{G} \leftarrow \mathbb{G} \cup G'$
end

Algorithm 9: Projected gradient descent method with stochastic subgradient estimation.

input : C, $\hat{\mathbf{y}}$, FOP, t, n_k, K
output: $\boldsymbol{\alpha}^{\text{best}}$
$k = 0$, $\mathcal{S}^k \leftarrow \emptyset$, $\boldsymbol{\alpha}^{(k)} \leftarrow \frac{1}{\mathcal{K}}\mathbf{1}$;
for $i \in 1 \dots n_k$ **do**
  $\text{FOP}_i \leftarrow \text{Coarsen}(\text{FOP})$;
  $\mathbf{y}_i^{(k)} \leftarrow \text{FOP}_i(\boldsymbol{\alpha}^{(k)})$;
end
$\mathbf{y}^{(k)} \leftarrow \underset{\mathbf{y}_i^{(k)}}{\arg\min} \boldsymbol{\alpha}^{(k)\intercal}\mathbf{C}\mathbf{y}_i^{(k)}$;
$\xi_{\text{ABS}} \leftarrow \boldsymbol{\alpha}^{(k)\intercal}C\hat{\mathbf{y}} - \boldsymbol{\alpha}^{(k)\intercal}C\mathbf{y}^{(k)}$;
while $k < K$ **do**
  $k \leftarrow k + 1$;
  $\mathcal{S}^k \leftarrow \mathcal{S}^{k-1} \cup \mathbf{y}^{(k-1)}$;
  $\boldsymbol{\alpha}^{(k)} \leftarrow \boldsymbol{\alpha}^{(k-1)} - t(C(\hat{\mathbf{y}} - \mathbf{y}^{(k-1)}))$;
  $\boldsymbol{\alpha}^{(k)} \leftarrow \text{proj}_{\Delta^{\mathcal{K}}}(\boldsymbol{\alpha}^{(k)})$;
  for $i \in 1 \dots n_k$ **do**
    $\text{FOP}_i \leftarrow \text{Coarsen}(\text{FOP})$;
    $\mathbf{y}_i^{(k)} \leftarrow \text{FOP}_i(\boldsymbol{\alpha}^{(k)})$;
  end
  $\mathbf{y}^{(k)} \leftarrow \underset{\mathbf{y}_i^{(k)}}{\arg\min} \boldsymbol{\alpha}^{(k)\intercal}\mathbf{C}\mathbf{y}_i^{(k)}$;
  $\xi_{\text{ABS}} \leftarrow \boldsymbol{\alpha}^{(k)\intercal}C\hat{\mathbf{y}} - \boldsymbol{\alpha}^{(k)\intercal}C\mathbf{y}^{(k)}$;
end
$\boldsymbol{\alpha}^{\text{best}} \leftarrow \frac{\sum_{i \in 0 \dots K} (i+1)^2 \boldsymbol{\alpha}^{(i)}}{\sum_{i \in 0 \dots K} (i+1)^2}$;
stop;

Appendix B

Appendices to Chapter 3

B.1 Crime Dataset Features

The UCI communities and crime dataset has a total of 127 features. All features with any missing values are excluded from our training dataset. Otherwise, all features are included except the following features, which are removed either due to being non-numerical or being highly correlated/colinear with other features:

- state
- county
- community
- communityname
- fold
- racePctWhite
- agePct12t29
- agePct16t24
- numbUrban
- PctUnemployed
- whitePerCap
- TotalPctDiv

- NumIlleg
- NumImmig
- HousVacant
- blackPerCap
- indianPerCap
- AsianPerCap
- OtherPerCap
- HispPerCap

After removing all these features and features with missing values, we are left with 85 features.

B.2 Forward Model Hyperparameters

For each model type in both the regression and classification setting, hyperparameters for the forward model to be audited were chosen by a grid search, evaluating models trained on $\mathcal{L}_0$ with the analysis set $\mathcal{R}$ as a validation set. Classification model hyperparameters were chosen to maximize area under the ROC curve (AUC), and regression model hyperparameters were chosen to maximize R^2. For Ridge regression, the set of searched values for λ was $\{1 * n, 0.5 * n, 0.1 * n, 0.05 * n, 0.01 * n, 0.005 * n, 0.001 * n\}$, where n is the number of observations in $\mathcal{L}$. For optimal trees, regression trees are trained to minimize MSE, classification trees are trained to minimize misclassification, the complexity parameter is set to 0, and maximum depth is searched over the set $\{2, 3, 4, 5\}$. For support vector machines, the kernel is selected between linear and rbf $(\text{with } \gamma = 1)$, the soft margin penalty λ is searched over $\{1 * \frac{1}{n}, 10 * \frac{1}{n}, 100 * \frac{1}{n}, 1000 * \frac{1}{n}, 10000 * \frac{1}{n}, 100000 * \frac{1}{n}\}$ where n is the number of observations in $\mathcal{L}$, with $\epsilon = 0.05$ for regression models. The multiplication by dataset sizes for the penalty term coefficients is done so that in inverse optimization analysis, the models ϕ and $\theta_{\mathbf{c}}$ can be trained with a penalty hyperparameter that is proportional to the differing size of the datasets used to train them. For each dataset, the selected hyperparameters for each of the four partitions are listed below.

- *Classification*

  Optimal Tree: depth = [3, 3, 2]

Support Vector Machine: kernel = [rbf, rbf, rbf], $\lambda = [1000 * \frac{1}{n}, 10000 * \frac{1}{n}, 10000 * \frac{1}{n}]$

- *Regression*

  Ridge Regression: $\lambda = [0.01 * n, 0.01 * n, 0.01 * n]$

  Optimal Tree: depth = [2, 4, 3]

  Support Vector Machine: kernel = [rbf, rbf, rbf], $\lambda = [1000 * \frac{1}{n}, 10000 * \frac{1}{n}, 10000 * \frac{1}{n}]$

B.3 Inverse Optimization Solution Algorithm

Algorithm 10 outlines pseudocode for finding the optimal solution for CIO($\hat{Y}_{\mathcal{R}}, \theta, C$). The loss function is minimized by the application of projected gradident descent with Nesterov Acceleration and Restarting momentum when the loss funtion increases at an iteration. Algorithm 10 calls on Algorithm 11 and Algorithm 12 as subalgorithms. The gradient $\nabla \ell$ is calculated according to the corresponding calculations in Section 3.6 for the model type at hand. Let η and K denote the chosen step-size and number of steps respectively.

Algorithm 10: Algorithm for solving CIO: Projected gradient descent with Nesterov Acceleration and Restarting.

input : $C,\ \mathcal{R},\ \theta,\ \hat{Y}_{\mathcal{R}}, \ell,\ \eta,\ K$
output: $\mathbf{c}^{(K)}$
$k \leftarrow 0,\ \mathbf{c}^{(k)} \leftarrow \nvDash,\ a_k \leftarrow 1, A_k \leftarrow 1$;
$P \leftarrow C^\intercal (CC^\intercal)^{-1} C$;
while $k < K$ **do**
  $\mathbf{c}^{(k)},\ k,\ a_k,\ A_k \leftarrow$ Algorithm 11$(P,\ \mathcal{R},\ \theta,\ \hat{Y}_{\mathcal{R}}, \ell,\ \eta,\ K,\ \mathbf{c}^{(k)},\ k,\ a_k,\ A_k)$;
  $\eta \leftarrow \frac{1}{2}\eta$;
end
stop

Algorithm 11: Subalgorithm for solving CIO: Projected gradient descent with Nesterov Acceleration.

input : $P,\ \mathcal{R},\ \theta,\ \hat{Y}_{\mathcal{R}}, \ell,\ \eta,\ K,\ \mathbf{c}^{(k)},\ k,\ a_k,\ A_k$
output: $\mathbf{c}^{\text{best}},\ k,\ a_k,\ A_k$
$\mathbf{c}^{\text{best}} \leftarrow \mathbf{c}^{(k)}$;
$\tilde{Y}_{\mathcal{R}} \leftarrow \{\theta_{\mathbf{c}}(\mathbf{x}, \mathcal{R}) | \mathbf{x} \in \mathcal{R}\}$;
$\xi_{\text{ABS}} \leftarrow \ell(\hat{Y}_{\mathcal{R}}, \tilde{Y}_{\mathcal{R}})$;
$\xi_{\text{best}} \leftarrow \xi_{\text{ABS}}$;
$\mathbf{g} \leftarrow \nabla \ell(\hat{Y}_{\mathcal{R}}, \tilde{Y}_{\mathcal{R}})$;
$\mathbf{u}_k \leftarrow \mathbf{c}^{(k)} - \eta \mathbf{g}$;
$_k \leftarrow \mathbf{u}_k$;
while $k < K$ **do**
  $k \leftarrow k + 1$;
  $a_k \leftarrow k + 1$;
  $A_k \leftarrow A_k + a_k$;
  $\mathbf{c}^{(k)} \leftarrow (1 - \frac{a_k}{A_k})\mathbf{u}_k + \frac{a_k}{A_k}{}_k$;
  $\mathbf{c}^{(k)} \leftarrow$ Algorithm 12$(\mathbf{c}^{(k)},\ P, n)$;
  $\tilde{Y}_{\mathcal{R}}(\mathbf{c}) \leftarrow \{\theta_{\mathbf{c}}(\mathbf{x}, \mathcal{R}) | \mathbf{x} \in \mathcal{R}\}$;
  $\xi_{\text{ABS}} \leftarrow \ell(\hat{Y}_{\mathcal{R}}, \tilde{Y}_{\mathcal{R}})$;
  if $\xi_{ABS} \geq \xi_{best}$ **then**
    stop
  else
    $\mathbf{c}^{\text{best}} \leftarrow \mathbf{c}^{(k)}$;
  end
  $\mathbf{g} \leftarrow \nabla \ell(\hat{Y}_{\mathcal{R}}, \tilde{Y}_{\mathcal{R}})$;
  $\mathbf{u}_k \leftarrow \mathbf{c}^{(k)} - \eta \mathbf{g}$;
  $_k \leftarrow_k - a_k \eta \mathbf{g}$;
end
stop

Algorithm 12: Projection onto the intersection of a scaled unit simplex and subspace with projection matrix P

input :, P, n, $\epsilon > 0$
output:
$\mathbf{p} \leftarrow \mathbf{0}$, $\mathbf{q} \leftarrow \mathbf{0}$;
while *True* **do**
  $\mathbf{u} \leftarrow P(+\mathbf{p})$;
  $\mathbf{p} \leftarrow +\mathbf{p} - \mathbf{u}$;
  $\leftarrow$ Blondel et al. (2014) Alg2$(\mathbf{u} + \mathbf{q}, n)$;
  $\mathbf{q} \leftarrow \mathbf{u} + \mathbf{q}-$;
  if $||\mathbf{p}|| < \epsilon$ *and* $||\mathbf{q}|| < \epsilon$ **then**
    stop
  end
end

Appendix C

Appendices to Chapter 5

C.1 10-fold cross validation

This section presents the results using 10-fold cross validation, including a comparison with leave-one-out cross validation.

Figure C.1 displays boxplots (acrosss the 10 folds) of the RMSE values for the best models of each type with all 10 features and with only 5 socio-demographic / contextual features. When using all features, the best KNN model has 25 neighbors, distance-based weighting, and minkowski distance; the best ElasticNet model had an alpha of 0.01 with l2 regularization; the best CART model had depth 4; and the best Random Forest model had 500 trees and depth 5. The overall best performing model was ElasticNet with an average RMSE of 1.096. This model improved upon the popular method by 0.218 (16.6%). The average RMSE for the popular, KNN, CART, and Random Forest methods was 1.314, 1.144, 1.163, and 1.098, respectively. The models that only had access to five features performed similarly to the popular approach. For example, the ElasticNet model with 10 features improved upon the five feature model by 0.198 (15.3%).

Figure C.2 displays boxplots of RMSE values for each model with leave-one-out and 10-fold cross validation. As expected, leave-one-out cross validation has a lower mean but a higher variance as compared to 10-fold cross validation. In both cases, the ElasticNet model performed best with an average RMSE of 1.096 and 1.061 for 10-fold and leave-one-out, respectively.

Figure C.3 displays boxplots (across the 10 folds) of RMSE values for each model with the 3-levels and 5-levels for the onboarding questions. The ElasticNet model with 5 levels for the onboarding features was able to improve upon the model with 3 levels by 0.018 (1.7%).

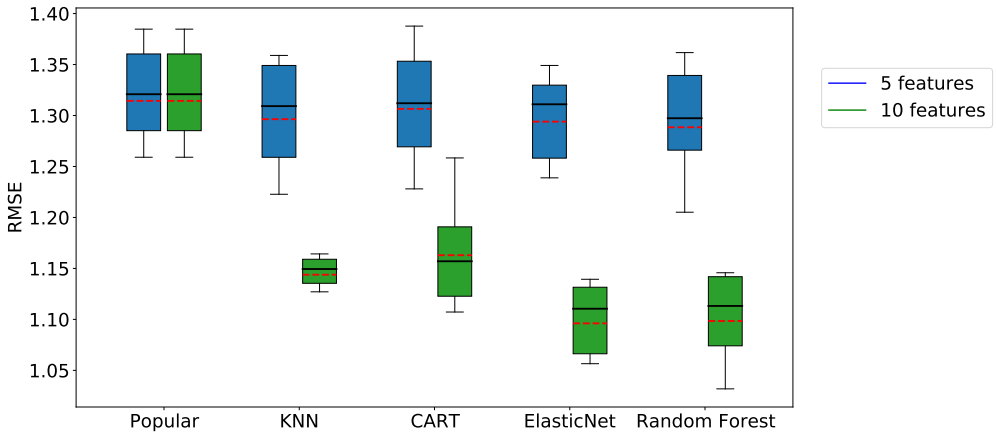


Figure C.1: Boxplots (across the 10 folds) of RMSE values for each model with the 3-levels and 5-levels for the onboarding questions. Note that the 5-level onboarding questions match the survey data. The solid black line denotes the median and the dashed red line denotes the mean.

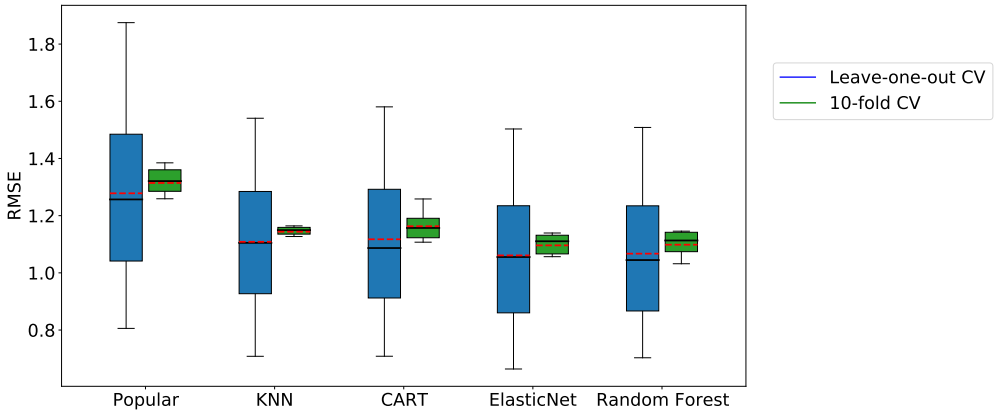


Figure C.2: Boxplots (across the 10 folds) of RMSE values for each model with leave-one-out and 10-fold cross validation. The solid black line denotes the median and the dashed red line denotes the mean.

Figure C.4 displays boxplots (across the 10 folds) of RMSE values for the ElasticNet model with an additional feature (noted on the x-axis) and the base model (shown on the far right). Note that the features are ordered according to mean RMSE, where the best additional feature is shown next to the base model. The best feature to add to the model was Q14.4, and this feature improved RMSE by 0.021 (2.0%) over the base model with 10 features.

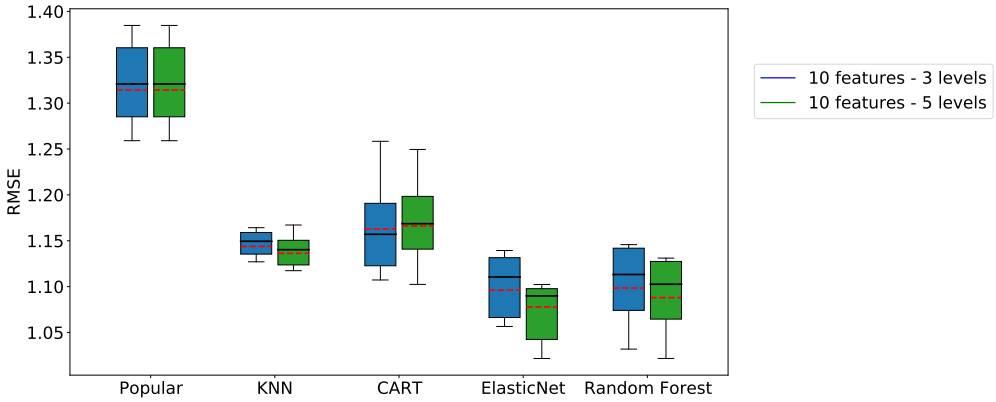


Figure C.3: Boxplots (across the 10 folds) of RMSE values for each model with the 3-levels and 5-levels for the onboarding questions. Note that the 5-level onboarding questions match the survey data. The solid black line denotes the median and the dashed red line denotes the mean.

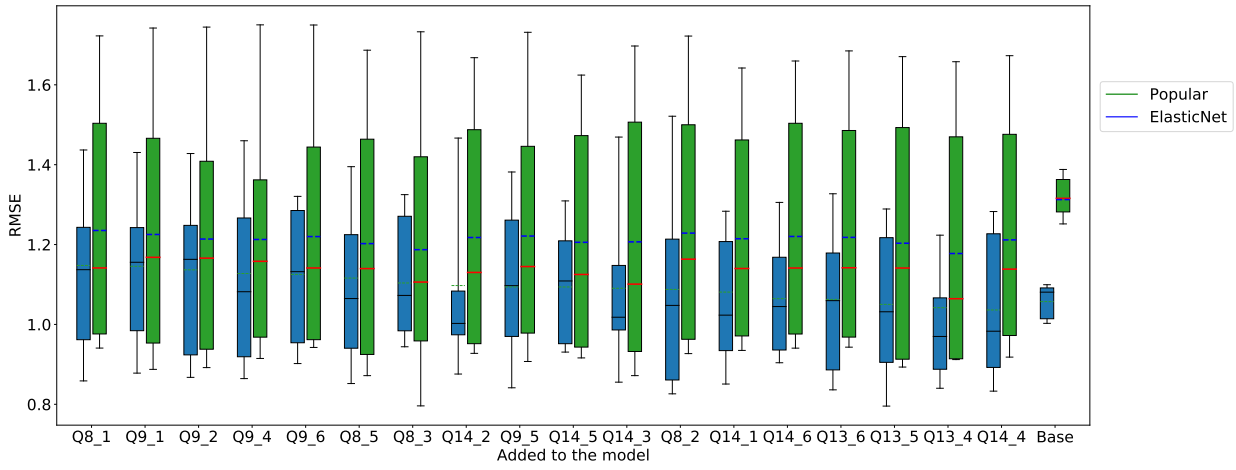


Figure C.4: Boxplots (across the 10 folds) of RMSE values for the ElasticNet model with an additional feature and the base model. The solid red line denotes the median and the dashed blue line denotes the mean.

C.2 Investigation of Q14.5

In this section, we run the recommender system experiments, but we remove Q14.5 from the set of potential recommendations and add it to the ElasticNet model. Since the set of recommendations has changed, the problem difficulty may also change. To account for this, we re-run the popular approach and ElasticNet models with only 5 features and with the original 10 features.

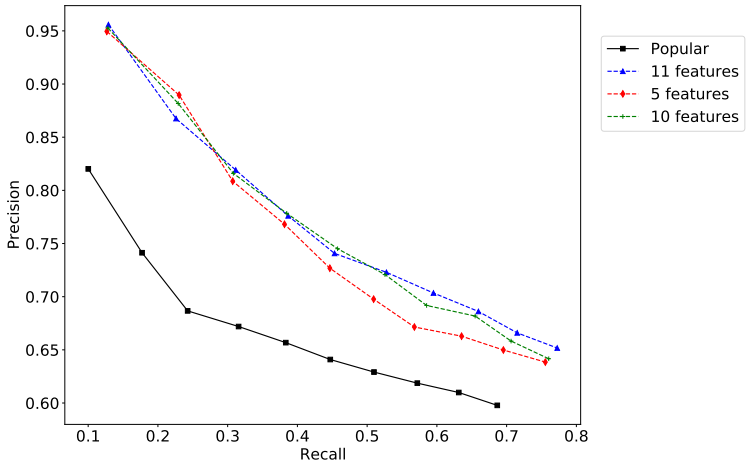


Figure C.5: Precision and recall values as a function of the number of recommendations for each model.

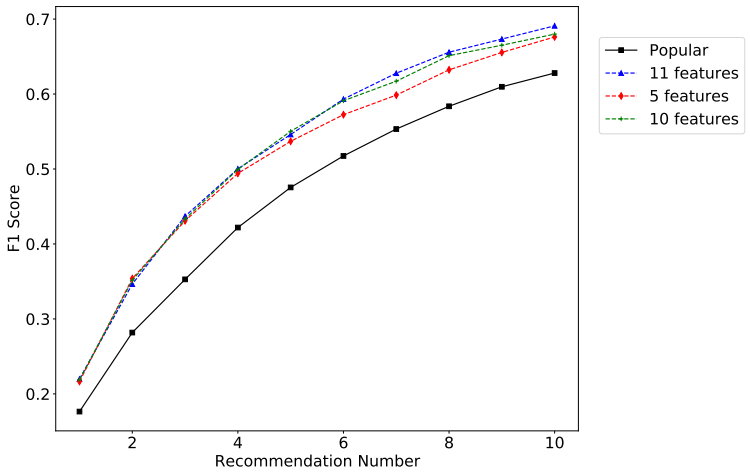


Figure C.6: F1-score as a function of the number of recommendations for each model.

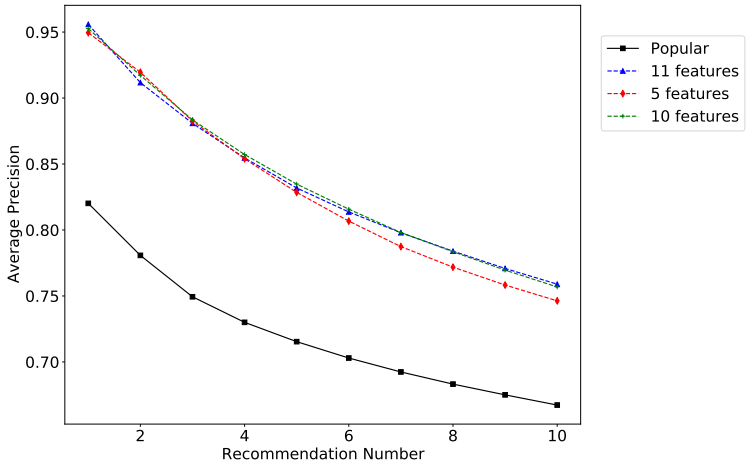


Figure C.7: Average precision as a function of the number of recommendations for each model.

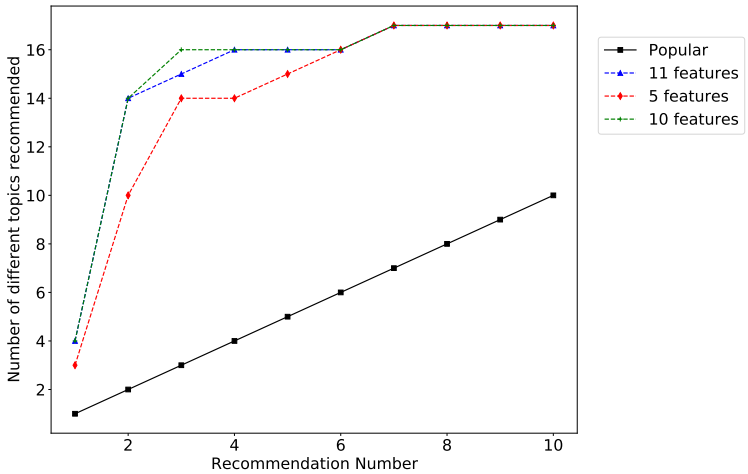


Figure C.8: Number of unique recommendations as a function of the number of recommendations for each model.

C.3 Investigation of Q14.4

In this section, we run the recommender system experiments, but we remove Q14.4 from the set of potential recommendations and add it to the ElasticNet model. Since the set of recommendations has changed, the problem difficulty may also change. To account for this, we re-run the popular approach and ElasticNet models with only 5 features and with the original 10 features.

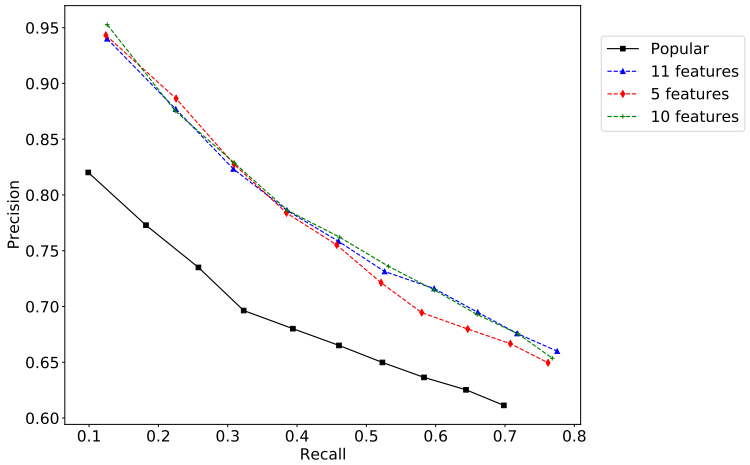


Figure C.9: Precision and recall values as a function of the number of recommendations for each model.

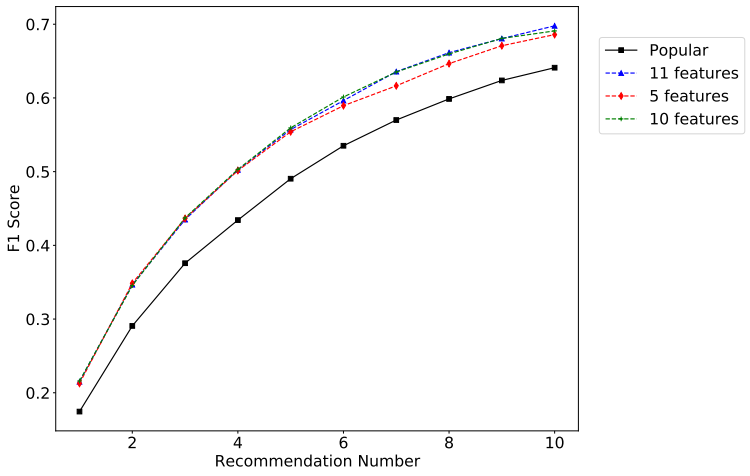


Figure C.10: F1-score as a function of the number of recommendations for each model.

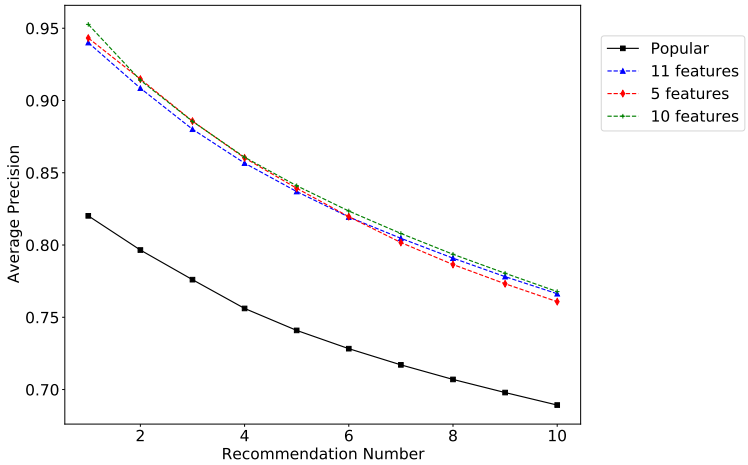


Figure C.11: Average precision as a function of the number of recommendations for each model.

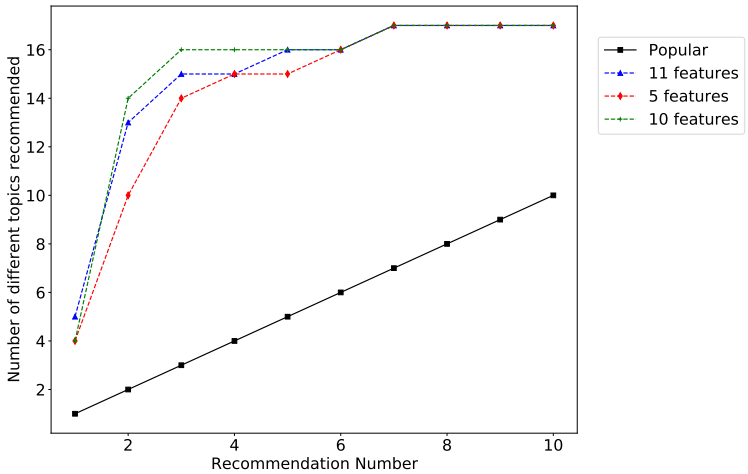


Figure C.12: Number of unique recommendations as a function of the number of recommendations for each model.

C.4 Evaluate using different relevant set

In this section, we run the recommender system experiments with a different set of relevant recommendations. We consider a recommendation as relevant if, for a given user, the target value of the topic is strictly less than the average target value over all 18 topics for that user (i.e., it is one of the topics they rated below average). Figure C.13 displays a histogram of the number of relevant topics for each individual. The mean number of relevant topics was 8.70.

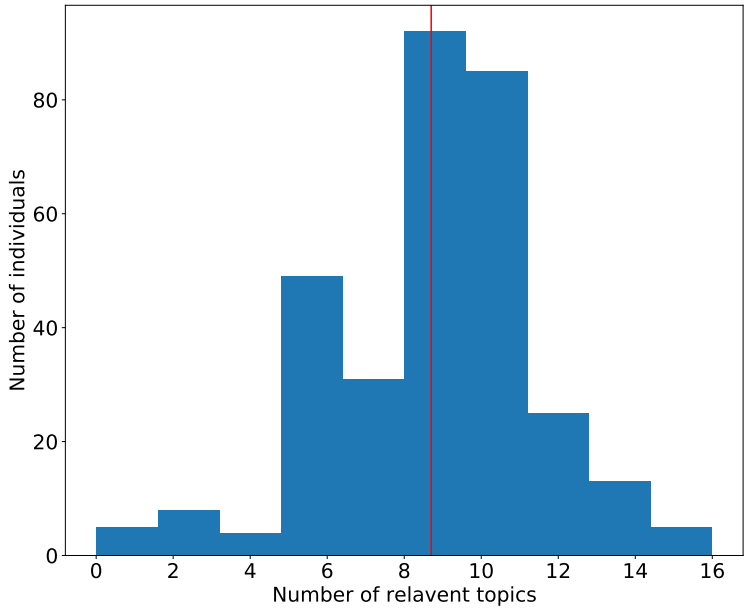


Figure C.13: A histogram of the number of relevant topics for each individual. The red line denotes the mean (8.70).

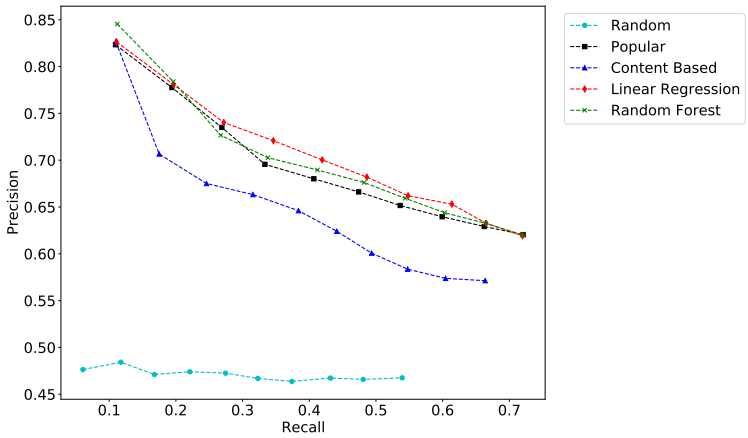


Figure C.14: Precision and recall values as a function of the number of recommendations for each model.

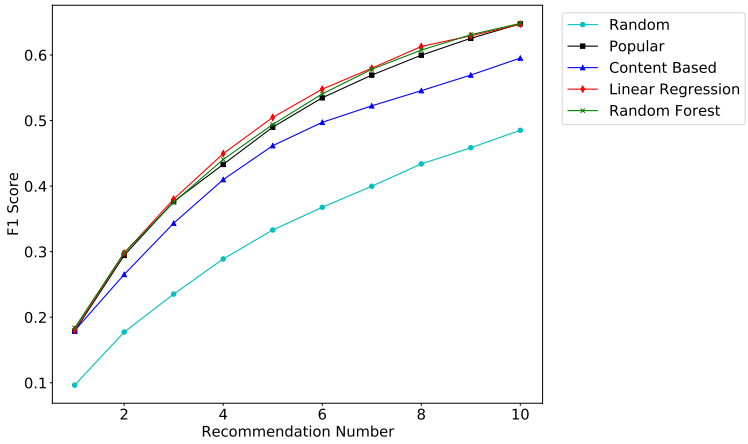


Figure C.15: F1-score as a function of the number of recommendations for each model.

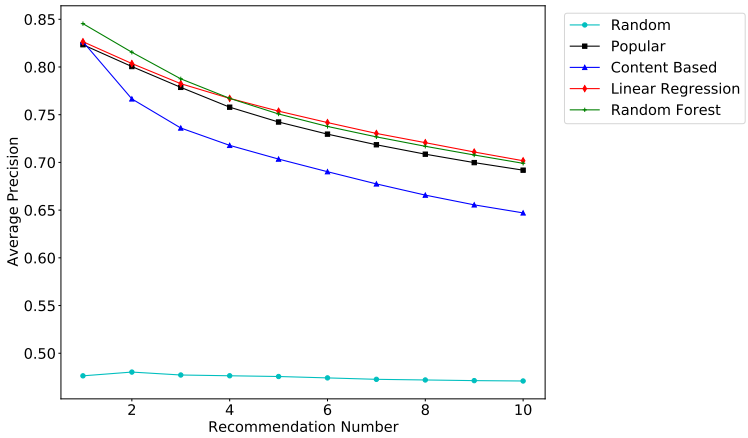


Figure C.16: Average precision as a function of the number of recommendations for each model.

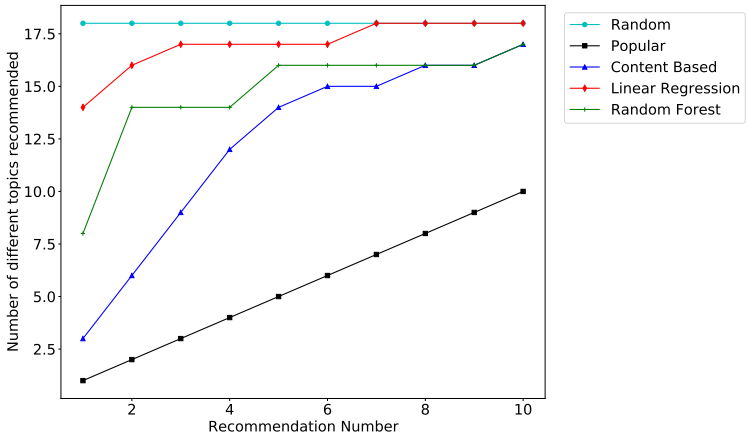


Figure C.17: Number of unique recommendations as a function of the number of recommendations for each model.

C.5 Evaluate using bottom set only

In this section, we run the recommender system experiments with a different set of relavent recommendations. We consider a recommendation as relevant if, for a given user, the target value of the topic is one of the lowest rated topics for that user. Figure C.18 displays a histogram of the number of relevant topics for each individual. The mean number of relevant topics was 4.78.

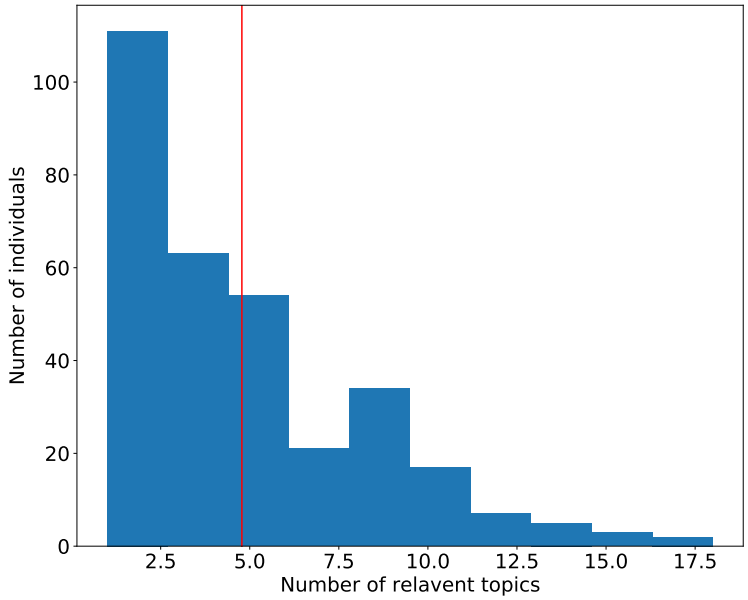


Figure C.18: A histogram of the number of relevant topics for each individual. The red line denotes the mean (4.78).

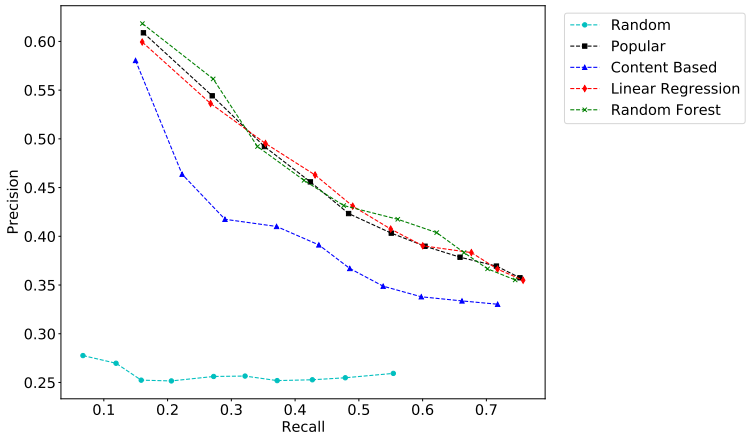


Figure C.19: Precision and recall values as a function of the number of recommendations for each model.

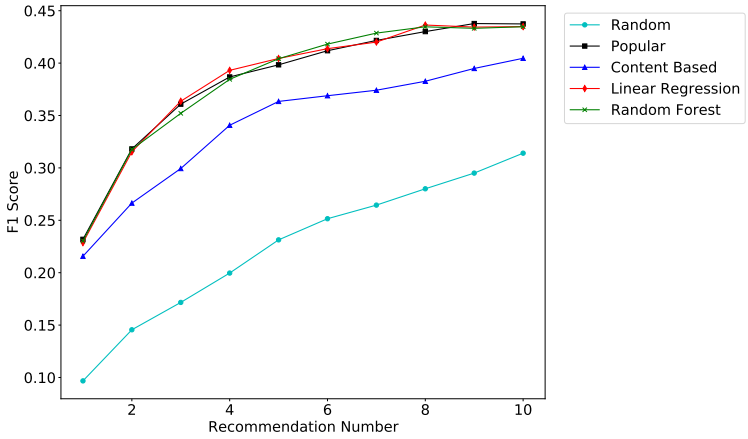


Figure C.20: F1-score as a function of the number of recommendations for each model.

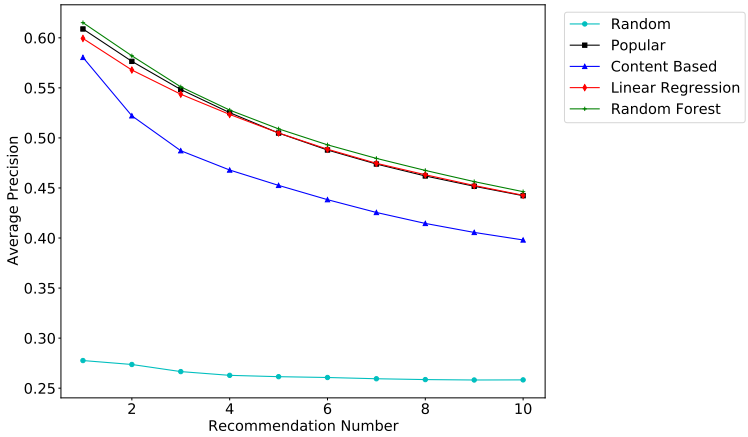


Figure C.21: Average precision as a function of the number of recommendations for each model.

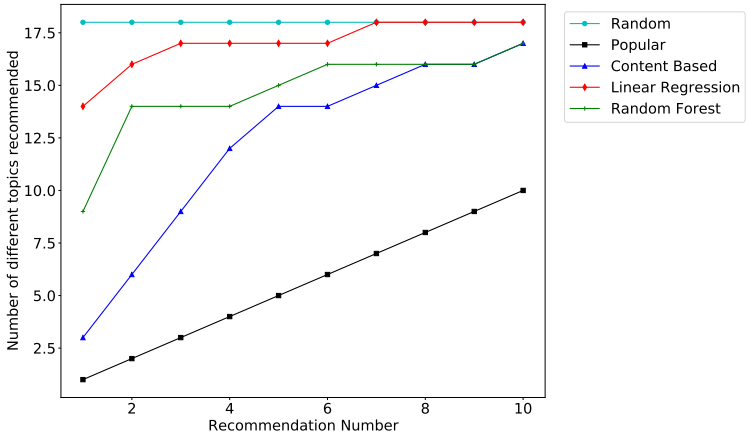


Figure C.22: Number of unique recommendations as a function of the number of recommendations for each model.

C.6 Evaluate using bottom two sets

In this section, we run the recommender system experiments with a different set of relavent recommendations. We consider a recommendation as relevant if, for a given user, the target value of the topic is in the set of lowest or second lowest rated topics for that user. Figure C.23 displays a histogram of the number of relevant topics for each individual. The mean number of relevant topics was 9.45.

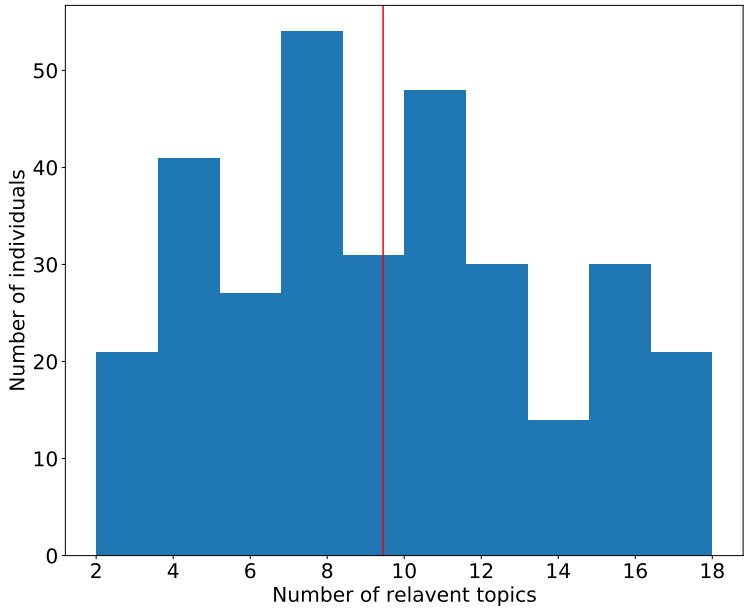


Figure C.23: A histogram of the number of relevant topics for each individual. The red line denotes the mean (9.45).

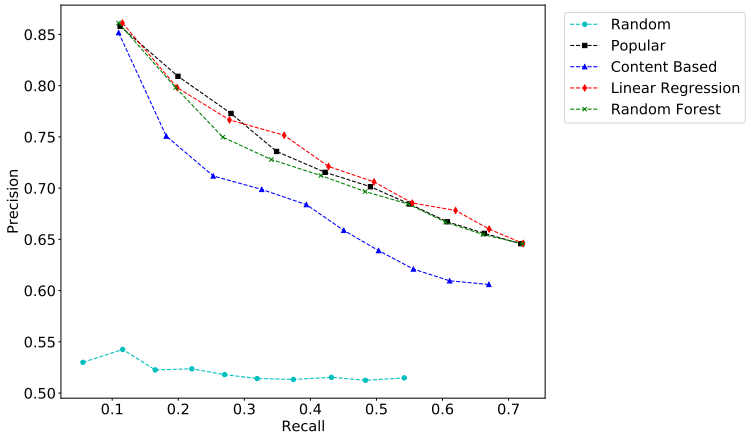


Figure C.24: Precision and recall values as a function of the number of recommendations for each model.

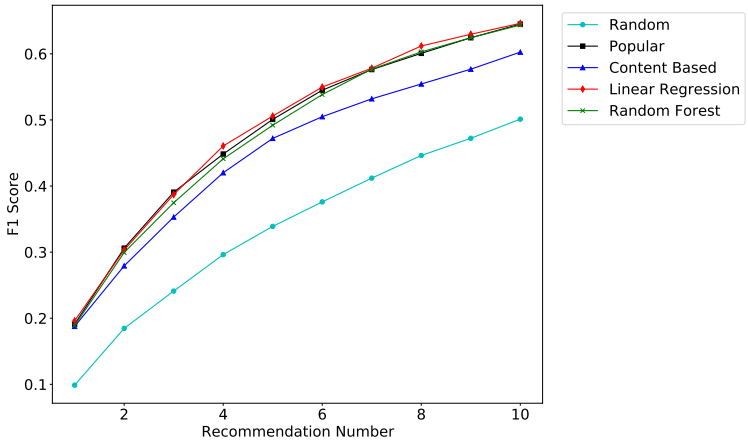


Figure C.25: F1-score as a function of the number of recommendations for each model.

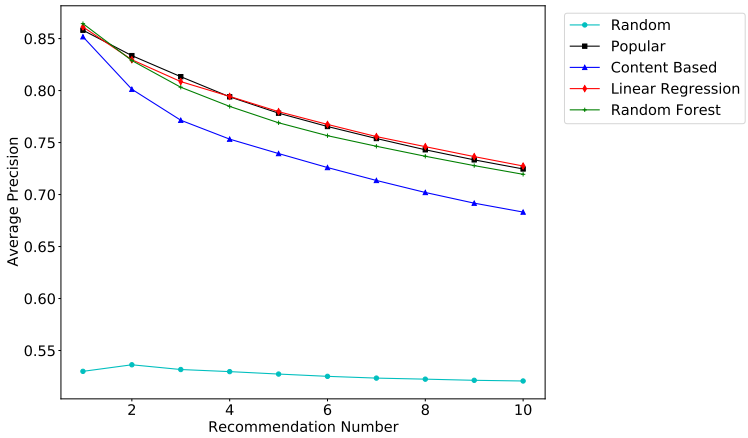


Figure C.26: Average precision as a function of the number of recommendations for each model.

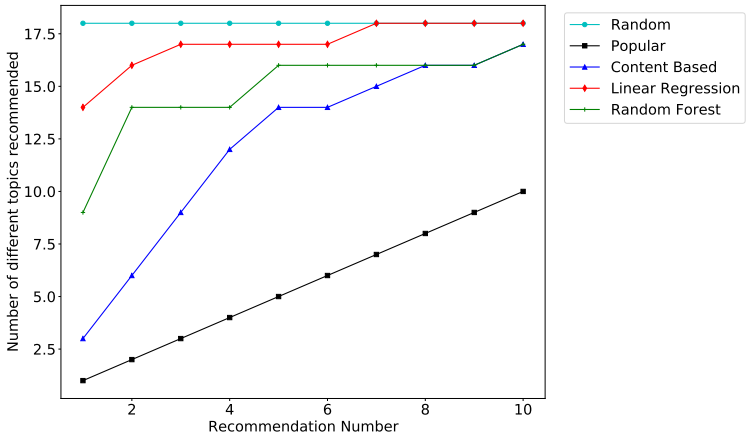


Figure C.27: Number of unique recommendations as a function of the number of recommendations for each model.

Appendix D

Multisite Evaluation of Prediction Models for Emergency Department Crowding Before and During the COVID-19 Pandemic

Ari Smith, Brian W. Patterson, MD, MPH, Michael S. Pulia, MD, MS,
John Mayer, PhD, Rebecca J. Schwei, MPH, Radha Nagarajan, PhD,
Frank Liao, PhD, Manish N. Shah, MD, MPH, Justin J. Boutilier, PhD

D.1 Abstract

Objective: To develop a machine learning framework to forecast emergency department (ED) crowding and to evaluate model performance under spatial and temporal data drift. **Materials and Methods:** We obtained four datasets, identified by the location: 1 – large academic hospital and 2 – rural hospital, and time period: pre-COVID (Jan 1, 2019 – Feb 1, 2020) and COVID-era (May 15, 2020 – Feb 1, 2021). Our primary target was a binary outcome that is equal to 1 if the number of patients with acute respiratory illness that were ED boarding for more than four hours was above a prescribed historical percentile. We trained a random forest and used the area under the curve (AUC) to evaluate out-of-sample performance for two experiments: 1) we evaluated the impact of sudden temporal drift by training models using pre-COVID data and testing them during the COVID-era, 2) we evaluated the impact of spatial drift by testing models trained at Location 1 on data from Location 2, and vice versa.

Results: The baseline AUC values for ED boarding ranged from 0.54 (pre-COVID at Location 2) to 0.81 (COVID-era at Location 1). Models trained with pre-COVID data performed similarly to COVID-era models (0.82 vs. 0.78 at Location 1). Models that were transferred from Location 2 to Location 1 performed worse than models trained at Location 1 (0.51 vs. 0.78).
Discussion and Conclusion: Our results demonstrate that ED boarding is a predictable metric for ED crowding, models were not significantly impacted by temporal data drift, and any attempts at implementation must consider spatial data drift.

D.2 Introduction

Emergency departments (EDs) experience high variability in patient visit volumes due to myriad factors, including temporal seasonality (Wargon et al. (2009)), weather (Røislien et al. (2018)), and events that change patient behavior (e.g., COVID-19) (Asheim et al. (2019). When this variation leads to an unexpectedly high ED or inpatient census, ED crowding can become a major challenge, affecting ED operations (e.g., equipment shortages), staff experience (e.g., burnout), and patient outcomes (e.g., delays in time sensitive treatments) (Hoot and Aronsky (2008), Sun et al. (2013)). Many hospitals have surge plans that trigger responsive action when ED crowding occurs, including increasing ED staff, opening temporary patient overflow areas, conducting early discharges, and in extreme cases, postponing elective surgeries (Kelen et al. (2021)). However, the effects of crowding on ED operations and the proposed solutions may differ depending on the cause of crowding (Kelen et al. (2021)).

Several modeling approaches have shown potential to anticipate certain types of crowding scenarios, including ARIMA and discrete event simulation models to predict short term crowding based on recent history and seasonality (Schweigler et al. (2009), Hoot et al. (2008), Hoot et al. (2009a)) presenting an opportunity to enact preemptive mitigation strategies. To date, research on forecasting methods for EDs has largely focused on predicting patient arrival and departure volumes with limited clinical data (Sun et al. (2009), Gopakumar et al. (2016), Jones et al. (2008)), or on predicting patient-level outcomes based on electronic medical record (EMR) data (Raita et al. (2019), Taylor et al. (2016), Chen et al. (2020), Klang et al. (2021)). More recently, and partially in response to the COVID-19 pandemic, several studies have developed methods to forecast population level spread (Alagoz et al. (2021), Hernández-Pereira et al. (2021), Bertsimas et al. (2021)), and to predict individual patient outcomes and pathways through the health system (e.g., length of stay, bed occupancy) (Hernández-Pereira et al. (2021), Haimovich et al. (2020),

Poirier et al. (2021)). Although these methods can be used to estimate the likelihood of ED crowding, there is limited research that focuses on directly predicting ED crowding in the context of COVID-19 or on predicting specific contributors to ED crowding such as acute respiratory illnesses, which can allow for even more targeted actionability. And while COVID-19 constitutes a universal disruptive event, there is limited understanding of its potentially disparate impact, spatially and temporally, on data drift across data settings.

In this paper, we develop a machine learning framework driven by EMR *and* community COVID-19 data to directly forecast the likelihood of ED crowding at three crucial junctures: ED arrival, hospital admission, and extended ED boarding. We defined crowding as a binary outcome that is equal to 1 if the number of patients at one of the aforementioned junctures was above a prescribed historical percentile (e.g., a 90th percentile event). We evaluate model performance for both general patient volumes and for patients with acute respiratory illness, like COVID-19 and influenza. Finally, we quantify the impact of data drift on model performance, both spatially (i.e., models trained in one location and applied to another location) and temporally (i.e., models trained pre-COVID and applied during the COVID-19 pandemic). The primary research gap that we address is forecasting crowding events (which we define using historical data) in in the context of COVID-19 and quantifying the impact of temporal and spatial data drift.

D.3 Materials and Methods

D.3.1 Study Setting

We obtained data from two hospitals in the United States of America. "Location 1" is a large academic hospital that has over 500 total beds and an ED with 65 beds. "Location 2" is a rural hospital that has 300 total beds and an ED with 25 beds.

D.3.2 Data Sources

Data was obtained from Epic Systems EMR at Location 1 and the internally developed EMR at Location 2. This study was reviewed by the IRB and granted exempt status as secondary use of medical records data for which consent was not required. The data comprised patient-level encounters with the ED, including demographic and medical information, plus higher-level operational data such as the National Emergency Department Overcrowding Scale (NEDOCS) score (Weiss et al. (2004)) and ED census information (e.g., bed availability). We also obtained community level COVID-19 data from the state's

Department of Health Services (DHS). These data included county-level counts of daily COVID-19 cases and hospitalizations, collected from May 15, 2020 onward.

D.3.3 Study Population

We extracted four unique datasets, identified by the location and time period from which the data was drawn. The pre-COVID datasets for Locations 1 and 2 included all ED encounters between January 1, 2019 and February 1, 2020, while the COVID-era datasets included all encounters between May 15, 2020 and February 1, 2021. We left a 104 day-gap between the two time periods for three reasons: 1) patient behavior and ED operations were highly variable during the first part of the pandemic; 2) data collection and testing for COVID-19 was not yet standardized leading to significant concerns about data validity; 3) this gap established a clear break between the pre-COVID and COVID-era datasets. We excluded all patients who were missing data on vital signs (e.g., temperature) or age. Missing socio-demographic data was encoded as a unique category.

D.3.4 Dependent Variables

Each of our four datasets was partitioned into 12-hour intervals (0000-1159 and 1200-2359). For each interval, we counted the total number of patients and the number of patients exhibiting Acute Respiratory Illness (ARI) at three junctures in the ED. We used a previously published composite definition for ARI that allowed us to capture a range of respiratory illnesses encompassing COVID-like illnesses, influenza-like illnesses, and possibly illnesses from unknown respiratory pathogens Pulia et al. (2020). The three junctures were:

1. *Arrival*: a patient arrives at the ED.

2. *Admission*: a disposition decision is made for the patient to be transferred to a hospital inpatient unit under the care of another service (as opposed to being discharged back to their community residence).

3. *Extended Boarding*: a patient is considered as an "extended boarder" if they have had an admission order placed, but have not been transferred and have been waiting for at least 4-hours. Although there are several metrics for quantifying boarding, this definition is used to correspond to a flag that is operationally used in the electronic medical record system at Location 1. We note that the existing literature discusses the use of 2, 4, and 6-hour boarding thresholds. The choice of 4 hours is supported by The

Joint Commission's Standard LD.04.03.11, EP 6, which sets 4 hours as a "reasonable goal" for boarding. We refer to patients who are in extended boarding as ED boarders, and a patient in extended boarding contributes to the total boarder count up until they depart the ED.

For each juncture, we defined a binary target that is equal to 1 if the total number of patients (or number of patients with ARI) exceeds a chosen historical percentile. We considered five different historical patient percentiles (75, 80, 85, 90, 95) as thresholds to define our binary target. For example, assume that the number of extended boarders at Location 1 pre-COVID over 12 AM-12PM during the first time period was 50 and that the 90th and 95th percentiles of historical observations at that time / location were 48 and 52, respectively. Then, the corresponding target value is equal to 1 for the 90th percentile model (and clearly also the 75, 80, 85th percentile models), but equal to 0 for the 95th percentile model.

D.3.5 Independent Variables

Each model used 203 independent variables (*features*) across the following 5 main categories: time descriptors, patient volume, ED patient descriptors, hospital-level descriptors, and community-level descriptors. The full list of features, grouped by category, is given below.

1. *Time descriptors*: Binary variables for each day of the week, a binary variable for the time of day (whether the time bucket is 12AM-12PM to 12PM-12AM), and a count of the number of days from the beginning of the dataset.

2. *Patient volume*: Total arrivals in the ED, total admission dispositions, total number of boarders, total number of arrivals with ARI, total admissions with ARI, and total boarders with ARI. We computed these variables, which are raw counts, for four 12-hour time intervals preceding the target time.

3. *ED patient descriptors*: Socio-demographic data (age, race, ethnicity, sex, and smoker status) and patients' most recent vitals (systolic blood pressure, diastolic blood pressure, temperature, pulse, and respirations). We encoded these variables as the following categories: for age (in years), the groups are 0-19, 20-39, 40-59, 60-79, 80+; for race, the groups are American Indian or Alaska Native, Asian, Black or African American, Native Hawaiian or Other Pacific Islander, and White; for ethnicity, the groups are Hispanic or Latino and Not Hispanic or Latino; for sex, the groups are Male and Female; for smoker status the groups are Current Smoker and Not Current Smoker; for systolic blood pressure (in mmHg), the groups are 0-119, 120-129, 130-139, and 140+; for diastolic blood pressure (in mmHg), the groups are 0-79, 80-89,

and 90+; for temperature (in °F), the groups are 0-96, 97-99, and 100+; for pulse (in bpm), the groups are 0-59, 60-99, and 100+; for respirations (per minute), the groups are 0-11, 12-19, and 20+. For each category, the associated feature describes the count of patients who arrived to the ED. We computed these variables for four 12-hour time intervals preceding the target time.

4. *Hospital-level descriptors*: ED Census, NEDOCS score, the most recent count of occupied ICU beds, and the most recent count of occupied ICU beds that are reserved for COVID-19 patients. We computed these variables for four 12-hour time intervals preceding the target time.

5. *Community-level descriptors*: The daily number of new confirmed COVID-19 cases, the number of COVID-related hospitalizations, and a 7-day rolling average of COVID hospitalizations as published by the Wisconsin DHS for the relevant Wisconsin County (Dane County for UW Health and Wood County for Location 2). We computed each daily count feature for the 10 days prior to the target time.

All feature data was scaled to a range of $[0, 1]$. All community-level descriptors and the feature representing the count of occupied ICU beds that are reserved for COVID-19 patients were excluded from the pre-COVID datasets.

D.3.6 Models

We investigated three predictive modelling methods: Logistic Regression with L1 regularization (Hosmer and Lemeshow (1989), Tibshirani (1996)), K-Nearest Neighbors (Altman (1992)), and Random Forest (Breiman (2001)). For Logistic Regression, we selected the best regularization coefficient on the logarithmic scale between 0.0001 and 10000. For K-Nearest Neighbors, we selected the best number of neighbors in the range from 1 to 20, weighted neighbors by distance, and tested full-feature models, as well models with feature selection via a greedy algorithm. For Random Forest, we selected the best max tree depth in the range from 1 to 10. These hyperparameters were conservatively chosen based on the best performance for 80th percentile predictions, and then applied to all other models. All other hyper-parameters were left as default selections used by the Sci-Kit Learn Python package.

For each model, we evaluated predictive performance by making predictions on all observations in the dataset beyond a given start date (Feb 01, 2019 for the pre-COVID dataset and July 1, 2020 for the COVID-Era dataset). For each prediction, a unique model is fit on all data prior to the observation being predicted (to simulate how the models would

be used in practice) and then used to predict the next 12-hour period. In this way, the models step forward and are retrained for each subsequent 12-hour period. We combined all out-of-sample predictions to evaluate model performance using receiver operating characteristic (ROC) curves and the associated area under the ROC curve (AUC).

D.3.7 Analysis

We conducted three primary analyses and a sensitivity analysis, as described in the following subsections.

Baseline model performance
First, to evaluate baseline model performance, we trained and tested each model on the same dataset (e.g., trained and tested at Location 1 pre-COVID) using a 90th percentile cutoff. For each location (1 and 2), this exercise involved twelve different datasets: four (pre-COVID all, pre-COVID ARI, COVID-era all, COVID-era ARI) for each event (ED arrival, ED admission, ED boarder).

Time period transfer performance
To evaluate the impact of temporal data drift, we conducted three experiments. First, in the COVID-era period, we evaluated two different models for each location trained using different feature sets; a full-feature model which used all 203 features, and a partial-feature model that only used features that were available pre-COVID (i.e., we excluded the county-level COVID data and any COVID related ICU bed occupancy data). This allowed us to evaluate the usefulness of the features generated from new data collection practices conducted during the pandemic. Second, we trained models using all data from the pre-COVID period and evaluated their performance during the COVID-era period. This allowed us to evaluate the usefulness of applying models developed using data pre-COVID during the COVID pandemic. Finally, we conducted an experiment to evaluate the impact of the changing dynamics throughout the COVID-era. To do this, we applied our models to different subsets of the data, representing different quarters of the overall testing window from the COVID-era data. For example, the AUC of the set of predictions for the entire testing window (July 1, 2020 – February 1, 2021) is compared to the set of predictions for the final quarter of the window (December 13, 2020 – February 1, 2021).

Location transfer performance
To assess the spatial transferability of the models, we evaluated the performance of models

that were trained on data collected at one location and used to create predictions at another location. We conducted this experiment in both directions: we trained models at Location 1 and tested them at Location 2, and we trained models at Location 2 and tested them at Location 1. We also conducted this experiment for both the pre-COVID and COVID-era time periods.

Sensitivity analysis

We conducted two sensitivity analyses of our models. First, we varied the choice of percentile that determined the target values using the following percentile cutoffs: 75, 80, 85, 90, and 95. Second, we varied the amount of additional lag that is applied to the features relative to the time of the target (only for a 90th percentile target). In particular, we investigated making predictions beyond the immediate next 12-hour window where there is more opportunity to take preemptive action, in five increments of 12 hours (i.e., the furthest prediction was made 60 hours in advance or with features that lagged 60 hours behind the target).

D.4 Results

The datasets at Location 1 encompassed 62,585 ED-patient encounters in the pre-COVID time period, and 34,244 ED-patient encounters in the COVID-era time period. At Location 2, the datasets encompassed 24,992 ED-patient encounters in the pre-COVID time period, and 17,359 ED-patient encounters in the COVID-era time period. Table 1 contains information on the number of ED encounters at both locations in both time periods, as well as information on the targets and missingness. See Appendix C for more details on excluded encounters. In both the pre-COVID and COVID-era periods, patients arriving at Location 2 were approximately twice as likely to have and to be admitted with ARI, as compared to patients arriving at Location 1.

Figures 1a and 1b display the raw counts of ED boarders at Location 1 that correspond to various percentiles with and without ARI, respectively. The results are stratified for the COVID-era and pre-COVID time periods, and for both 12-hour (AM and PM) windows. For example, for the 12-hour AM period, the 90th percentile of boarders with ARI during the COVID-era and pre-COVID time periods was 10 and 11, respectively. The number of boarders with ARI was consistently higher (across all percentiles) during the COVID-era as compared to pre-COVID, while the number of boarders was higher across all percentiles during the pre-COVID time period. For example, the 95th percentile of total ARI boarders was 40.8% higher during the pre-COVID era (49) as compared to the COVID-era (29).

Patient-encounter descriptor	Location 1 pre-COVID	Location 1 COVID-Era	Location 2 pre-COVID	Location 2 COVID-Era
Arrivals, n.	62,585	34,244	24,992	17,359
Admitted, n. (%)	15,523 (24.8%)	10,305 (30.1%)	8466 (39%)	6428 (37.0%)
Extended boarding, n. (%)	1997 (3.2%)	1760 (5.1%)	806 (3.2%)	857 (4.9%)
Arrivals with ARI, n. (%)	10,518 (16.8%)	8161 (23.8%)	6521 (26.1%)	9343 (53.8%)
Admitted with ARI, n. (%)	4713 (7.5%)	4097 (12.0%)	3067 (12.3%)	5752 (33.1%)
Extended boarding with ARI, n. (%)	645 (1.0%)	802 (2.3%)	251 (1.0%)	791 (4.6%)
Female sex, n. (%)	32,314 (51.6%)	17,369 (50.7%)	12,901 (51.6%)	8885 (51.2%)
Age in years, mean (stdev)	43.04 (24.30)	45.15 (23.70)	49.79 (26.85)	52.20 (25.43)
Daily county COVID cases, mean (stdev)	N/A	149.09 (135.95)	N/A	17.82 (21.31)
Daily county COVID hospitalizations, mean (stdev)	N/A	3.91 (4.00)	N/A	23.12 (20.04)
Excluded Encounters, n. (%)	5588 (8.1%)	2209 (6.1%)	9857 (28.3%)	4331 (20.0]%)

Table D.1: Study population details for both Location 1 and Location 2, during both pre-COVID and the COVID-era time periods.

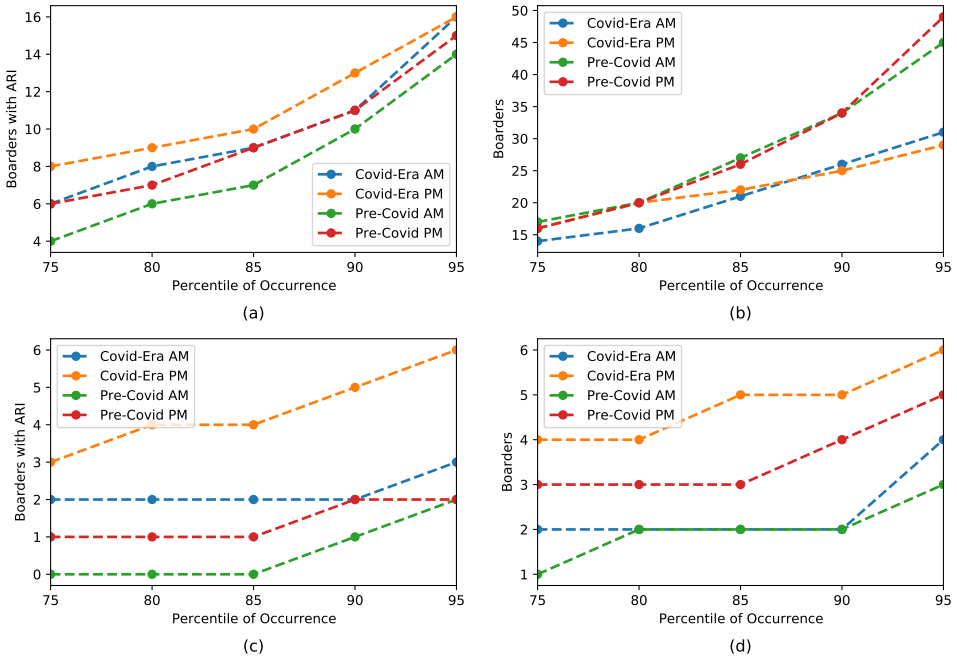


Figure D.1: Raw counts of ED boarders that correspond to various percentiles at Location 1 (a) with ARI and (b) total, and at Location 2 (c) with ARI and (d) total.

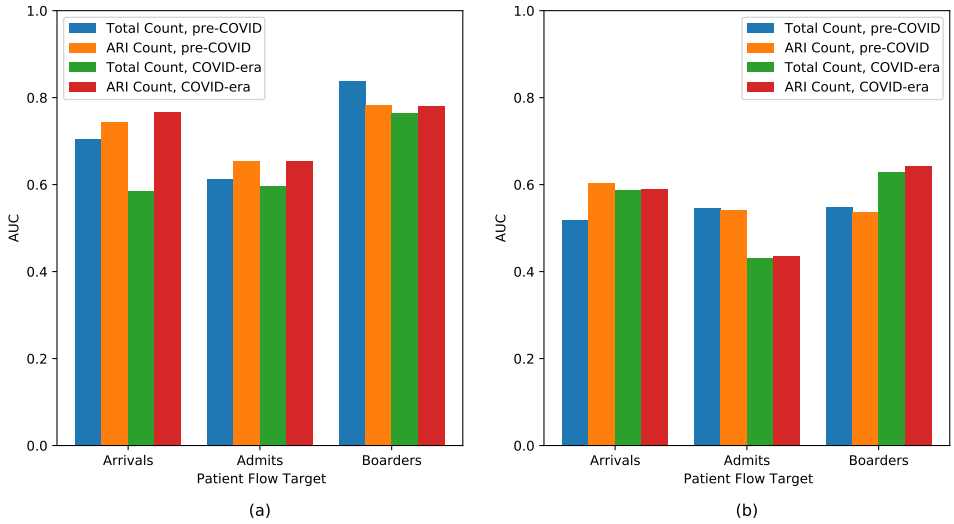


Figure D.2: AUCs for predicting 90th percentile events for the three different patient flow targets (arrivals, admits, boarders) for (a) Location 1 and (b) Location 2, stratified by time period and patient type.

Figures 1c and 1d display the raw counts of ED boarders at Location 2 that correspond to various percentiles with and without ARI, respectively. For example, for the 12-hour AM period, the 90th percentile of boarders with ARI during the COVID-era and pre-COVID time periods was 1 and 2, respectively. Similar to Location 1, the number of boarders with ARI was consistently higher (across all percentiles) during the COVID-era as compared to pre-COVID, while the total number of boarders was higher across all percentiles for PM time windows as compared to AM time windows. Notably, for AM time periods pre-COVID, the 85th percentile of counts of boarders with ARI was 0.

D.4.1 Baseline Model Performance

Figure 2a and 2b display the out-of-sample AUC values for the best performing modeling methods for all twelve datasets using a 90th percentile cutoff at Location 1 and Location 2, respectively. For Location 1, the models performed consistently well for predicting ARI boarders (with AUCs of 0.794 and 0.808 for pre-COVID and COVID-era, respectively) and ARI arrivals (with AUCs of 0.744 and 0.767 for pre-COVID and COVID-era, respectively). For Location 2, we found that ARI boarders during the COVID-era performed best, with an AUC of 0.642. The best performing model across all instances was random forest, and to keep the remaining exposition succinct, we focus on 90th percentile ARI boarder predictions from the random forest model. Feature importance values for the top five features from the random forest model at Location 1 are shown in Table D.2.

Pre-COVID		COVID-era	
Feature name	Importance	Feature name	Importance
Boarders, lagged 1	0.029	Boarders, lagged 1	0.066
Boarders, lagged 1, differenced 1	0.025	Boarders, lagged 1, differenced 1	0.035
ARI Boarders, lagged 1, differenced 1	0.012	ARI Boarders, lagged 1	0.032
ARI Boarders, lagged 1	0.012	Boarders, lagged 2, differenced 1	0.029
Boarders, lagged 2	0.011	ARI Boarders, lagged 1, differenced 1	0.025

Table D.2: Feature importance values for the top five features from the random forest model that predicts ARI boarders at Location 1.

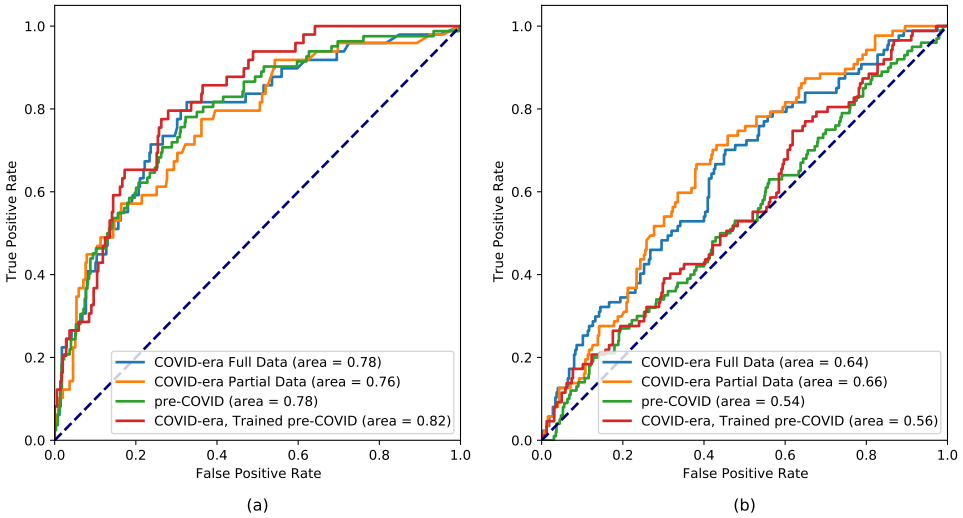


Figure D.3: ROC curves for the 90th percentile ARI boarders at (a) Location 1 and (b) Location 2.

D.4.2 Time Period Transfer Performance

Figures 3a and 3b display ROC curves for the 90th percentile ARI boarders at Location 1 and Location 2, respectively. For Location 1, we find that a model trained on data during the pre-COVID era, but used to make predictions during the COVID-era performs better (0.82) than both models (full – 0.78 and partial – 0.76) trained during the COVID-era. In contrast, at Location 2, the COVID-era models (full – 0.64 and partial – 0.66) outperform the model trained during the pre-COVID period (0.56).

Figure 4 shows the AUC for the 90th percentile of ARI boarders as a function of the testing time period window. Note that the first point contains predictions over the entire COVID-era testing period, while the last point contains predictions over the final quarter (i.e., December 13, 2020 – February 1, 2021) of the COVID-era testing period. In general, model performance remains stable over time.

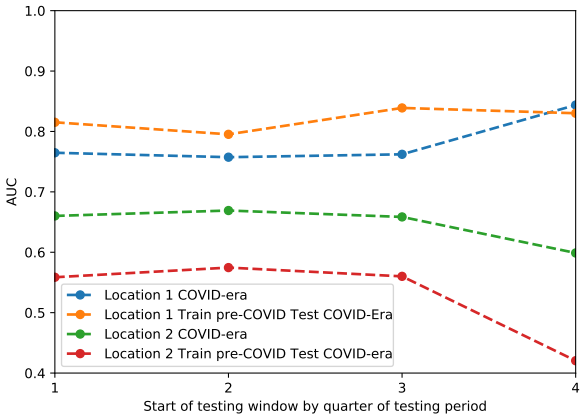


Figure D.4: AUCs for the 90th percentile of ARI boarders as a function of the testing time period window.

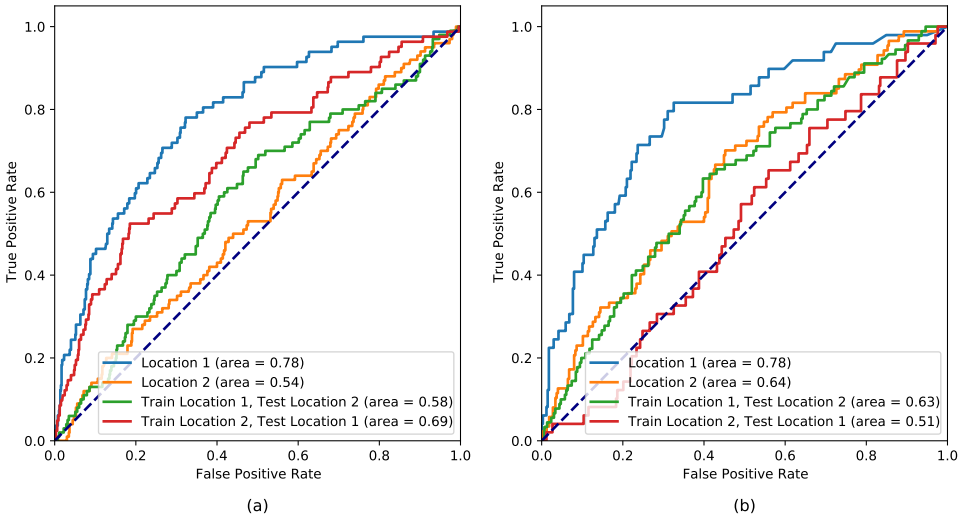


Figure D.5: ROC curves for the 90th percentile of ARI boarders for (a) pre-COVID and (b) COVID-era predictions.

D.4.3 Location Transfer Performance

Figures 5a and 5b display ROC curves for the 90th percentile of ARI boarders for location transfer predictions made during the pre-COVID and COVID-era time periods, respectively. We find that model transferability performed better during pre-COVID as compared to the COVID-era. For example, a model trained at Location 2 and tested at Location 1 had an AUC of 0.69 during pre-COVID (a drop of 0.09 from the Location 1 trained model) and an AUC 0.51 during the COVID-era (a drop of 0.27 from the Location 1 trained model).

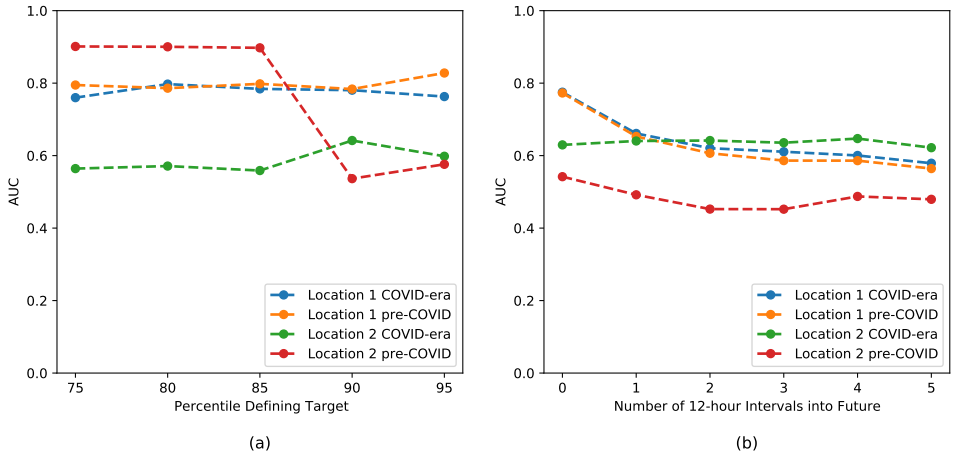


Figure D.6: AUC as a function of (a) the percentile for ARI boarders stratified by location and time period, and (b) the number of 12-hour intervals between the predicted time window and the most recent training data, stratified by location and time period.

D.4.4 Sensitivity Analysis

Figure 6a displays the AUCs for predicting ARI boarder events, when the percentile that determines targets is varied from: 75, 80, 85, 90, and 95. In both locations and both time periods, model performance remains stable across the percentiles. Note that at Location 2 in the pre-COVID period, the 75th, 80th, and 85th percentiles represented 0 ARI, which resulted in an upwards bias in the AUC values.

Figure 6b displays the AUCs for predicting 90th percentile ARI boarder events, where the time interval into the future is varied along the range [0-12 hours, 12-24 hours, 24-36 hours, 36-48 hours, 48-60 hours, 60-72 hours]. At Location 1, AUC decreases monotonically as distance into the future increases with a decrease of 27.3

D.5 Discussion

D.5.1 Summary of the Results

ED crowding is a widespread problem that causes patient harm and a critical indicator of hospital and health system performance. This paper developed machine learning models to directly predict ED crowding in the context of COVID-19 at three crucial junctures (ED arrival, hospital admission, extended ED boarding). To train and test our models, we used data from two time periods (pre-COVID, COVID-era) and from two different hospitals (Locations 1 and 2). Overall, we found that ED boarders were the easiest to predict, our predictions were robust to the choice of percentile used to define ED crowding,

and prediction accuracy remains stable up to 60 hours into the future. These are important implications for practitioners because they allow users to determine their own risk tolerance (through the choice of percentile) and because mitigation strategies are more effective when enacted as early as possible.

D.5.2 ED Boarders were Easiest to Predict

Extended ED boarding has received limited focus (as compared to arrivals and admissions) in the scientific literature on forecasting ED crowding, with no previous research focused directly on predicting extended boarding in the context of COVID-19. We found that predicting extended ED boarding performed best for both locations and both time periods, as compared to ED arrivals and hospital admissions (see Figure 2). One possible explanation is that extended ED boarding may be more stable over time compared to other metrics due to a higher degree of auto-regression. Appendix B shows the feature importance values for the top 5 features for the baseline models at Location 1 for ARI boarders. In both time periods, features measuring past boarder and ARI boarder counts have the highest importance, providing support for this hypothesis, but the magnitudes are still sufficiently low (0.066 at its highest) indicating that many other variables are contributing to the models.

Although limited previous literature (Morley et al. (2018), Hoot et al. (2009b)) has focused on predicting ED boarding, there are studies that examine the effects of extended ED boarding on patient outcomes (increased mortality), patient length of stay (longer), and quality of care (increased treatment delays). Directly predicting future "extreme" extended ED boarding events (i.e., above the 90th percentile of historical number of boarders) has the potential to be actionable for hospital administrators because they can enact mitigation strategies before boarding reaches harmful levels, resulting in the negative consequences described above. Additional research is needed to further evaluate the ability to predict extended ED boarding at other health systems with potentially different definitions and with different operational strategies and patient populations.

D.5.3 Time Transferability May be Acceptable

Temporal data drift can occur gradually as conditions change over time or suddenly as a result of a disruptive event like the COVID-19 pandemic. We evaluated the impact of sudden temporal data drift and found that a model trained pre-COVID performed well during the COVID-era at Location 1, but not at Location 2 (see Figure 3). Intuitively and as shown at Location 2, we might expect that a model trained pre-COVID would perform worse during the COVID-era because of differences in patient visit volumes, types of

patients, hospital operations, and overall community behavior. One explanation for the results observed at Location 1 is that the model trained during the COVID-era had access to a smaller training dataset as compared to the model trained pre-COVID. We also found that the additional data collected in response to COVID-19 (e.g., county cases, ICU beds occupied with COVID patients) did not significantly improve model performance at either location. These results are important for practice because they suggest that, depending on location, models may not need to be re-trained in response to disruptive events (like COVID-19) and that there may be limited value in developing and integrating COVID-specific features into models that predict ED crowding.

D.5.4 Location Matters

Several recent studies have highlighted the importance of quantifying the impact of the data collection setting on model performance, with most research focusing on quantifying the impact of spatial data drift, where models trained at one location are evaluated at another location (Wong et al. (2021), Singh et al. (2021)). We contribute to this literature by evaluating the impact of spatial data drift at two different locations and during two time periods. We found that models trained at Location 2 and tested at Location 1 performed significantly worse than models that were trained/tested at Location 1, while models trained at Location 1 and tested at Location 2 performed similar (or even better) to those trained/tested at Location 2. One possible explanation is that Location 1 serves as a tertiary care center and received transfers and referrals from Location 2, implying that models trained at Location 1 may have been exposed to patient data from (or similar to) Location 2. Overall, our findings reinforce prior research that suggests models trained (or conceived) at one location and tested at another location may not perform well, and extreme care is needed when attempting to apply models across locations.

D.5.5 Implementation Considerations

Our framework focuses on predicting the probability that the number of arrivals, admissions, and extended ED boarders will exceed a chosen historical percentile, which we use to represent ED crowding. Intuitively, this definition can be used to encode "extreme" ED crowding events, such as the number of extended ED boarders above the 95th percentile; an event that only occurs 5% of the time. We chose this approach because it allows practitioners to choose their own threshold (or risk) level and because we believe this metric is easily interpretable. We are currently in the process of implementing our models for everyday use at Location 1, with a future work focusing on understanding what actions

Status	Location 1 Pre-COVID	Location 1 COVID-Era	Location 2 Pre-COVID	Location 2 COVID-Era
Total Excluded	5588	2209	9857	4331
Admitted	20 (0.4%)	10 (0.5%)	979 (9.9%)	514 (11.9%)
Extended Boarding	0 (0%)	1 (0%)	108 (1.1 %)	78 (1.8%)

Table D.3: Counts of admission and boarder status of excluded patients

administrators may take when presented with our predictions. Our framework, including the training and testing code (but not including models trained on our data), is available for use through HIPxChange.

D.5.6 Limitations

This study has at least four limitations. First, the number of excluded encounters due to missing data on patient vitals was particularly high in the Location 2 datasets, which highlights one further difference in the study settings that could impact the efficacy of modelling efforts across different locations. However, patients who were admitted to the hospital or experienced extended boarding were less likely to have missing vitals, so the impact of missingness on predicting ED boarding may be less than expected (see Table D.3 for more details). Second, due to differences in the EMR systems at each location, the operationalization of our definitions for ARI and pulse were slightly different, which could have impacted our results. Third, an oxygenation feature was not included in our datasets. Fourth, we only evaluated the performance of our models using data from two hospitals, both in the same state. Given the heterogeneity in individual behavior and public health response to COVID-19 at the state and county-level (CDC COVID-19 Response Team et al. (2020)) it is unclear if our models will perform similarly at other locations with different levels of community transmission.

D.6 Conclusion

This paper developed a machine learning framework driven by EMR and community COVID-19 data to directly forecast the likelihood of ED crowding, defined as patient volumes above a prescribed historical percentile (e.g., a 90th percentile event), at three crucial junctures: ED arrival, hospital admission, and extended ED boarding.

References

Agarwal, Alekh, Miroslav Dudik, and Zhiwei Steven Wu. 2019. Fair regression: Quantitative definitions and reduction-based algorithms. In *Proceedings of the 36th international conference on machine learning*, ed. Kamalika Chaudhuri and Ruslan Salakhutdinov, vol. 97 of *Proceedings of Machine Learning Research*, 120–129. PMLR.

Ahuja, Ravindra K, and James B Orlin. 2001. Inverse optimization. *Operations research* 49(5):771–783.

Alagoz, Oguzhan, Ajay K. Sethi, Brian W. Patterson, Matthew Churpek, and Nasia Safdar. 2021. Effect of Timing of and Adherence to Social Distancing Measures on COVID-19 Burden in the United States: A Simulation Modeling Approach. *Annals of Internal Medicine* 174(1):50–57.

Altman, N. S. 1992. An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression. *The American Statistician* 46(3):175–185.

Anzaldúa, Gloria, et al. 1987. *Borderlands/la frontera*. na.

Asheim, Andreas, Lars P. Bache-Wiig Bjørnsen, Lars E. Næss-Pleym, Oddvar Uleberg, Jostein Dale, and Sara M. Nilsen. 2019. Real-time forecasting of emergency department arrivals using prehospital data. *BMC Emergency Medicine* 19(1):42.

Aswani, Anil, Zuo-Jun Shen, and Auyon Siddiq. 2018. Inverse optimization with noisy data. *Operations Research* 66(3):870–892.

Babier, Aaron, Timothy CY Chan, Taewoo Lee, Rafid Mahmood, and Daria Terekhov. 2021. An ensemble learning framework for model fitting and evaluation in inverse linear optimization. *Informs Journal on Optimization* 3(2):119–138.

Barad, Karen. 1998. Getting real: Technoscientific practices and the materialization of reality. *Differences: a journal of feminist cultural studies* 10(2):87–91.

———. 2007. Getting Real: Technoscientific Practices and the Materialization of Reality. In *Meeting the Universe Halfway*, 189–222. Duke University Press.

Barocas, Solon, Moritz Hardt, and Arvind Narayanan. 2023. *Fairness and machine learning: Limitations and opportunities*. MIT press.

Becker, Amariah, Moon Duchin, Dara Gold, and Sam Hirsch. 2021. Computational redistricting and the voting rights act. *Election Law Journal: Rules, Politics, and Policy* 20(4): 407–441. https://doi.org/10.1089/elj.2020.0704.

Bell, Robert M, Yehuda Koren, and Chris Volinsky. 2010. All together now: A perspective on the netflix prize. *Chance* 23(1):24–29.

Benjamin, Ruha. 2019. *Race after technology: Abolitionist tools for the new jim code*. John Wiley & Sons.

Berry, Kenneth J, Janis E Johnston, and Paul W Mielke. 2019. *A primer of permutation statistical methods*. Springer.

Bertsimas, Dimitris, Leonard Boussioux, Ryan Cory-Wright, Arthur Delarue, Vassilis Digalakis, Alexandre Jacquillat, Driss Lahlou Kitane, Galit Lukin, Michael Li, Luca Mingardi, Omid Nohadani, Agni Orfanoudaki, Theodore Papalexopoulos, Ivan Paskov, Jean Pauphilet, Omar Skali Lami, Bartolomeo Stellato, Hamza Tazi Bouardi, Kimberly Villalobos Carballo, Holly Wiberg, and Cynthia Zeng. 2021. From predictions to prescriptions: A data-driven response to COVID-19. *Health Care Management Science* 24(2):253–272.

Bertsimas, Dimitris, and Jack Dunn. 2017. Optimal classification trees. *Machine Learning* 106:1039–1082.

Better, MAPPING A. 2023. Alzheimer's disease facts and figures. *Alzheimers Dement* 19(4):1598–1695.

Blondel, Mathieu, Akinori Fujino, and Naonori Ueda. 2014. Large-scale multiclass support vector machine training via euclidean projection onto the simplex 1289–1294.

Bodur, Merve, Timothy CY Chan, and Ian Yihang Zhu. 2022. Inverse mixed integer optimization: Polyhedral insights and trust region methods. *INFORMS Journal on Computing*.

Boyd, Stephen, Lin Xiao, and Almir Mutapcic. 2003. Subgradient methods. *lecture notes of EE392o, Stanford University, Autumn Quarter* 2004:2004–2005.

Boyle, James P., and Richard L. Dykstra. 1986. A method for finding projections onto the intersection of convex sets in hilbert spaces. In *Advances in order restricted statistical inference*, ed. Richard Dykstra, Tim Robertson, and Farroll T. Wright, 28–47. New York, NY: Springer New York.

Breiman, Leo. 1996. Bagging predictors. *Machine learning* 24:123–140.

———. 2001. Random forests. *Machine learning* 45:5–32.

Breiman, Leo, Jerome Friedman, Richard A Olshen, and Charles J Stone. 2017. *Classification and regression trees*. Routledge.

Bulut, Aykut, and Ted K Ralphs. 2021. On the complexity of inverse mixed integer linear optimization. *SIAM Journal on Optimization* 31(4):3014–3043.

CDC COVID-19 Response Team, CDC COVID-19 Response Team, Stephanie Bialek, Virginia Bowen, Nancy Chow, Aaron Curns, Ryan Gierke, Aron Hall, Michelle Hughes, Tamara Pilishvili, Matthew Ritchey, Katherine Roguski, Benjamin Silk, Tami Skoff, Preethi Sundararaman, Emily Ussery, Michael Vasser, Hilary Whitham, and John Wen. 2020. Geographic Differences in COVID-19 Cases, Deaths, and Incidence — United States, February 12–April 7, 2020. *MMWR. Morbidity and Mortality Weekly Report* 69(15):465–471.

Chan, Timothy CY, Tim Craig, Taewoo Lee, and Michael B Sharpe. 2014. Generalized inverse multiobjective optimization with application to cancer therapy. *Operations Research* 62(3):680–695.

Chan, Timothy CY, Taewoo Lee, and Daria Terekhov. 2019. Inverse optimization: Closed-form solutions, geometry, and goodness of fit. *Management Science* 65(3):1115–1135.

Chan, Timothy CY, Rafid Mahmood, and Ian Yihang Zhu. 2023. Inverse optimization: Theory and applications. *Operations Research*.

Chen, Chien-Hua, Jer-Guang Hsieh, Shu-Ling Cheng, Yih-Lon Lin, Po-Hsiang Lin, and Jyh-Horng Jeng. 2020. Emergency department disposition prediction using a deep neural network with integrated clinical narratives and structured data. *International Journal of Medical Informatics* 139:104146.

Chen, Jie, and Ronny Luss. 2019. Stochastic gradient descent with biased but consistent gradient estimators. `1807.11880`.

Chouldechova, Alexandra, and Aaron Roth. 2018. The frontiers of fairness in machine learning. *arXiv preprint arXiv:1810.08810*.

Church, James D, and Edward L Wike. 1979. A monte carlo study of nonparametric multiple-comparison tests for a two-way layout. *Bulletin of the Psychonomic Society* 14(2): 95–98.

Cortes, Corinna, and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning* 20:273–297.

Cover, T. 1968. Estimation by the nearest neighbor rule. *IEEE Transactions on Information Theory* 14(1):50–55.

Crawford, Kate, and Trevor Paglen. 2021. Excavating ai: The politics of images in machine learning training sets. *Ai & Society* 36(4):1105–1116.

Daryl DeFord, Moon Duchin, and Justin Solomon. 2020. A computational approach to measuring vote elasticity and competitiveness. *Statistics and Public Policy* 7(1):69–86. https://doi.org/10.1080/2330443X.2020.1777915.

Deleuze, Gilles. 1992. Postscript on the Societies of Control. *October* 59:3–7.

Duchin, Moon. 2018. Gerrymandering metrics: How to measure? what's the baseline? *arXiv preprint arXiv:1801.02064*.

Duchin, Moon, Taissa Gladkova, Eugene Henninger-Voss, Ben Klingensmith, Heather Newman, and Hannah Wheelen. 2019. Locating the representational baseline: Republicans in massachusetts. *Election Law Journal: Rules, Politics, and Policy* 18(4):388–401. https://doi.org/10.1089/elj.2018.0537.

Dwork, Cynthia, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. 2012. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, 214–226.

Dziedzic, Adam, Haonan Duan, Muhammad Ahmad Kaleem, Nikita Dhawan, Jonas Guan, Yannis Cattan, Franziska Boenisch, and Nicolas Papernot. 2022. Dataset inference for self-supervised models. *Advances in Neural Information Processing Systems* 35:12058–12070.

Engel, Christoph, Lorenz Linhardt, and Marcel Schubert. 2024. Code is law: how compas affects the way the judiciary handles the risk of recidivism. *Artificial Intelligence and Law* 1–22.

Foucault, Michel. 1997. Technologies of the self/rabinow p.(ed.). ethics: Subjectivity and truth.

Frank, Marguerite, and Philip Wolfe. 1956. An algorithm for quadratic programming 3(1):95–110.

Friedman, Milton. 1937. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the american statistical association* 32(200):675–701.

Garfinkel, Robert S, and George L Nemhauser. 1970. Optimal political districting by implicit enumeration techniques. *Management Science* 16(8):B–495.

Gasser, Nolan. 2019. *Why you like it: The science and culture of musical taste*. Flatiron Books.

Ghobadi, Kimia, Taewoo Lee, Houra Mahmoudzadeh, and Daria Terekhov. 2018. Robust inverse optimization. *Operations Research Letters* 46(3):339–344.

Giovanola, Benedetta, and Simona Tiribelli. 2023. Beyond bias and discrimination: redefining the ai ethics principle of fairness in healthcare machine-learning algorithms. *AI & society* 38(2):549–563.

Gleixner, Ambros, Gregor Hendel, Gerald Gamrath, Tobias Achterberg, Michael Bastubbe, Timo Berthold, Philipp M. Christophel, Kati Jarck, Thorsten Koch, Jeff Linderoth, Marco Lübbecke, Hans D. Mittelmann, Derya Ozyurt, Ted K. Ralphs, Domenico Salvagnin, and Yuji Shinano. 2021. MIPLIB 2017: Data-Driven Compilation of the 6th Mixed-Integer Programming Library. *Mathematical Programming Computation*.

Goldberg, David, David Nichols, Brian M. Oki, and Douglas Terry. 1992. Using collaborative filtering to weave an information tapestry. *Communications of the ACM* 35(12): 61–70.

Gopakumar, Shivapratap, Truyen Tran, Wei Luo, Dinh Phung, and Svetha Venkatesh. 2016. Forecasting Daily Patient Outflow From a Ward Having No Real-Time Clinical Data. *JMIR Medical Informatics* 4(3):e5650.

Grofman, Bernard, and Gary King. 2007. The future of partisan symmetry as a judicial test for partisan gerrymandering after lulac v. perry. *Election Law Journal* 6(1):2–35.

Haimovich, Adrian D., Neal G. Ravindra, Stoytcho Stoytchev, H. Patrick Young, Francis P. Wilson, David van Dijk, Wade L. Schulz, and R. Andrew Taylor. 2020. Development and Validation of the Quick COVID-19 Severity Index: A Prognostic Tool for Early Clinical Decompensation. *Annals of Emergency Medicine* 76(4):442–453.

Haraway, Donna. 2016. *A Cyborg Manifesto: Science, Technology, and Socialist-Feminism in the Late Twentieth Century*. University of Minnesota Press.

Hardt, Moritz, Eric Price, and Nati Srebro. 2016. Equality of opportunity in supervised learning. *Advances in neural information processing systems* 29.

Hernández-Pereira, Elena, Oscar Fontenla-Romero, Verónica Bolón-Canedo, Brais Cancela-Barizo, Bertha Guijarro-Berdiñas, and Amparo Alonso-Betanzos. 2021. Machine learning techniques to predict different levels of hospital care of CoVid-19. *Applied Intelligence*.

Hess, Sidney Wayne, JB Weaver, HJ Siegfeldt, JN Whelan, and PA Zitlau. 1965. Nonpartisan political redistricting by computer. *Operations Research* 13(6):998–1006.

Hickey, James M., Pietro G. Di Stefano, and Vlasios Vasileiou. 2021. Fairness by explicability and adversarial shap learning. In *Machine learning and knowledge discovery in databases*, ed. Frank Hutter, Kristian Kersting, Jefrey Lijffijt, and Isabel Valera, 174–190. Cham: Springer International Publishing.

Hoerl, Arthur E, and Robert W Kennard. 1970. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics* 12(1):55–67.

Hoot, Nathan R., and Dominik Aronsky. 2008. Systematic Review of Emergency Department Crowding: Causes, Effects, and Solutions. *Annals of Emergency Medicine* 52(2): 126–136.e1.

Hoot, Nathan R., Stephen K. Epstein, Todd L. Allen, Spencer S. Jones, Kevin M. Baumlin, Neal Chawla, Anna T. Lee, Jesse M. Pines, Amandeep K. Klair, Bradley D. Gordon, Thomas J. Flottemesch, Larry J. LeBlanc, Ian Jones, Scott R. Levin, Chuan Zhou, Cynthia S. Gadd, and Dominik Aronsky. 2009a. Forecasting emergency department crowding: an external, multicenter evaluation. *Annals of Emergency Medicine* 54(4):514–522.e19.

Hoot, Nathan R., Larry J. LeBlanc, Ian Jones, Scott R. Levin, Chuan Zhou, Cynthia S. Gadd, and Dominik Aronsky. 2008. Forecasting emergency department crowding: a discrete event simulation. *Annals of Emergency Medicine* 52(2):116–125.

Hoot, Nathan R., Larry J. Leblanc, Ian Jones, Scott R. Levin, Chuan Zhou, Cynthia S. Gadd, and Dominik Aronsky. 2009b. Forecasting emergency department crowding: a prospective, real-time evaluation. *Journal of the American Medical Informatics Association: JAMIA* 16(3):338–345.

Hosmer, David W, and Stanley Lemeshow. 1989. *Applied Logistic Regression*. Hoboken: John Wiley & Sons. OCLC: 927738292.

Iyengar, Garud, and Wanmo Kang. 2005. Inverse conic programming with applications. *Operations Research Letters* 33(3):319–330.

Jin, Yingjie, and Chunyan Han. 2020. A music recommendation algorithm based on clustering and latent factor model. *MATEC Web of Conferences* 309:03009.

Jones, Spencer S., Alun Thomas, R. Scott Evans, Shari J. Welch, Peter J. Haug, and Gregory L. Snow. 2008. Forecasting daily patient volumes in the emergency department. *Academic Emergency Medicine: Official Journal of the Society for Academic Emergency Medicine* 15(2):159–170.

Karakayali, Nedim, Burc Kostem, and Idil Galip. 2017. Recommendation Systems as Technologies of the Self: Algorithmic Control and the Formation of Music Taste. *Theory, Culture & Society*.

Kelen, Gabor, Richard Wolfe, Gail D'Onofrio, Angela Mills, Deborah Diercks, Susan Stern, Michael Wadman, and Peter Sokolove. 2021. Emergency Department Crowding: The Canary in the Health Care System. *NEJM Catalyst Innovations in Care Delivery*.

Keshavarz, Arezou, Yang Wang, and Stephen Boyd. 2011. Imputing a convex objective function. In *2011 ieee international symposium on intelligent control*, 613–619. IEEE.

King, Douglas M, Sheldon H Jacobson, and Edward C Sewell. 2015. Efficient geo-graph contiguity and hole algorithms for geographic zoning and dynamic plane graph partitioning. *Mathematical Programming* 149(1):425–457.

Klang, Eyal, Benjamin R. Kummer, Neha S. Dangayach, Amy Zhong, M. Arash Kia, Prem Timsina, Ian Cossentino, Anthony B. Costa, Matthew A. Levin, and Eric K. Oermann. 2021. Predicting adult neuroscience intensive care unit admission from emergency department triage using a retrospective, tabular-free text machine learning approach. *Scientific Reports* 11(1):1381.

Koren, Yehuda, and Christian Bell. 2015. Advances in Collaborative Filtering. In *Recommender Systems Handbook*, 2nd ed., 77–118. New York: Springer Science+Business Media.

Lamperski, Jourdain B, and Andrew J Schaefer. 2015. A polyhedral characterization of the inverse-feasible region of a mixed-integer program. *Operations Research Letters* 43(6): 575–578.

Latour, Bruno. 2005. Reassembling the social. *Política y Sociedad* 43(3):127–130.

Laue, Sören, Matthias Mitterreiter, and Joachim Giesen. 2018. Computing higher order derivatives of matrix and tensor expressions. In *Advances in neural information processing systems* (*neurips*).

———. 2020. A simple and efficient tensor calculus. In *AAAI conference on artificial intelligence*, (*aaai*).

Lewis, George E. 2008. *A Power Stronger Than Itself: The AACM and American Experimental Music*. University of Chicago Press.

Li, Tian, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. 2020. Federated learning: Challenges, methods, and future directions. *IEEE signal processing magazine* 37(3):50–60.

Lieb, David. 2022. Explainer: How supreme court case could alter us house seats. *The Associated Press*.

Lundberg, Scott M, and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Advances in neural information processing systems*, ed. I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, vol. 30. Curran Associates, Inc.

Maini, Pratyush, Hengrui Jia, Nicolas Papernot, and Adam Dziedzic. 2024. Llm dataset inference: Did you train on my dataset? *Advances in Neural Information Processing Systems* 37:124069–124092.

Maini, Pratyush, Mohammad Yaghini, and Nicolas Papernot. 2021. Dataset inference: Ownership resolution in machine learning. *arXiv preprint arXiv:2104.10706*.

McCartan, Cory, Christopher T Kenny, Tyler Simko, George Garcia III, Kevin Wang, Melissa Wu, Shiro Kuriwaki, and Kosuke Imai. 2022. Simulated redistricting plans for the analysis and evaluation of redistricting in the united states. *Scientific Data* 9(1):689.

Mehrabi, Ninareh, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2021. A survey on bias and fairness in machine learning. *ACM computing surveys* (*CSUR*) 54(6):1–35.

Mehrotra, Anuj, Ellis L Johnson, and George L Nemhauser. 1998. An optimization based heuristic for political districting. *Management Science* 44(8):1100–1114.

Moghaddass, Mahsa, and Daria Terekhov. 2020. Inverse integer optimization with an imperfect observation. *Operations Research Letters* 48(6):763–769.

Morley, Claire, Maria Unwin, Gregory M. Peterson, Jim Stankovich, and Leigh Kinsman. 2018. Emergency department crowding: A systematic review of causes, consequences and solutions. *PLoS ONE* 13(8):e0203316.

Mulligan, Deirdre K, Joshua A Kroll, Nitin Kohli, and Richmond Y Wong. 2019. This thing called fairness: Disciplinary confusion realizing a value in technology. *Proceedings of the ACM on Human-Computer Interaction* 3(CSCW):1–36.

Nagle, John F. 2017. How competitive should a fair single member districting plan be? *Election Law Journal* 16(1):196–209.

Nesterov, Yurii. 1983. A method for solving the convex programming problem with convergence rate o (1/k2). In *Dokl akad nauk sssr*, vol. 269, 543.

Ning, Xia, Christian Desrosiers, and George Karypis. 2015. A Comprehensive Survey of Neighborhood-Based Recommendation Methods. In *Recommender Systems Handbook*, 2nd ed., 37–76. New York: Springer Science+Business Media.

Noble, Safiya Umoja. 2018. *Algorithms of oppression: How search engines reinforce racism*. New York university press.

Patriksson, Michael. 2015. *The traffic assignment problem: models and methods*. Courier Dover Publications.

Pessach, Dana, and Erez Shmueli. 2022. A review on fairness in machine learning. *ACM Computing Surveys (CSUR)* 55(3):1–44.

Poirier, Canelle, Yulin Hswen, Guillaume Bouzillé, Marc Cuggia, Audrey Lavenu, John S. Brownstein, Thomas Brewer, and Mauricio Santillana. 2021. Influenza forecasting for French regions combining EHR, web and climatic data sources with a machine learning ensemble approach. *PLOS ONE* 16(5):e0250890.

Polyak, Boris T. 1964. Some methods of speeding up the convergence of iteration methods. *Ussr computational mathematics and mathematical physics* 4(5):1–17.

Prey, Robert. 2018. Nothing personal: algorithmic individuation on music streaming platforms. *Media, Culture, and Society* 40(7):1086–1100.

Pulia, Michael, Daniel Hekman, Joshua Glazer, Ciara Barclay-Buchanan, Nicholas Kuehnel, Joshua Ross, Brian Sharp, Robert Batt, and Brian Patterson. 2020. Electronic Health Record-Based Surveillance for Community Transmitted COVID-19 in the Emergency Department. *Western Journal of Emergency Medicine* 21(4).

Raita, Yoshihiko, Tadahiro Goto, Mohammad Kamal Faridi, David F. M. Brown, Carlos A. Camargo, and Kohei Hasegawa. 2019. Emergency department triage prediction of clinical outcomes using machine learning models. *Critical Care (London, England)* 23(1):64.

Redmond, Michael. 2002. Communities and Crime. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C53W3X.

Ricca, Federica, Andrea Scozzari, and Bruno Simeone. 2013. Political districting: from classical models to recent approaches. *Annals of Operations Research* 204:271–299.

Rivas-Perea, Pablo, Juan Cota-Ruiz, David Garcia Chaparro, Jorge Arturo Perez Venzor, Abel Quezada Carreón, and Jose Gerardo Rosiles. 2012. Support vector machines for regression: a succinct review of large-scale and linear programming formulations.

Rudin, Cynthia, Caroline Wang, and Beau Coker. 2020. The age of secrecy and unfairness in recidivism prediction. *Harvard Data Science Review* 2(1):1.

Røislien, Jo, Signe Søvik, and Torsten Eken. 2018. Seasonality in trauma admissions – Are daylight and weather variables better predictors than general cyclic effects? *PLoS ONE* 13(2):e0192568.

Schaefer, Andrew J. 2009. Inverse integer programming. *Optimization Letters* 3:483–489.

Schapire, Robert E. 1999. A brief introduction to boosting. In *Ijcai*, vol. 99, 1401–1406. Citeseer.

Schmeier, Timothy, Joeseph Chisari, Sam Garrett, and Brett Vintch. 2019. Music recommendations in hyperbolic space: an application of empirical bayes and hierarchical poincaré embeddings. In *Proceedings of the 13th ACM Conference on Recommender Systems*, 437–441. RecSys '19, Copenhagen, Denmark: Association for Computing Machinery.

Schmidt, Mark. 2019. Cpsc 540: Machine learning lecture notes.

Schweigler, Lisa M., Jeffrey S. Desmond, Melissa L. McCarthy, Kyle J. Bukowski, Edward L. Ionides, and John G. Younger. 2009. Forecasting Models of Emergency Department Crowding. *Academic Emergency Medicine* 16(4):301–308.

Scroccaro, Pedro Zattoni, Bilge Atasoy, and Peyman Mohajerin Esfahani. 2023. Learning in inverse optimization: Incenter cost, augmented suboptimality loss, and algorithms. *arXiv preprint arXiv:2305.07730*.

Sherman, Mark. 2019. Supreme court allows partisan districts, blocks census query. *The Associated Press*.

Shmueli, Galit. 2010. To explain or to predict?

Singh, Karandeep, Thomas S. Valley, Shengpu Tang, Benjamin Y. Li, Fahad Kamran, Michael W. Sjoding, Jenna Wiens, Erkin Otles, John P. Donnelly, Melissa Y. Wei, Jonathon P. McBride, Jie Cao, Carleen Penoza, John Z. Ayanian, and Brahmajee K. Nallamothu. 2021. Evaluating a Widely Implemented Proprietary Deterioration Index Model among Hospitalized Patients with COVID-19. *Annals of the American Thoracic Society* 18(7):1129–1137.

Smith, Ari J, and Justin J Boutilier. 2024. Gap-gradient methods for solving generalized mixed integer inverse optimization: an application to political gerrymandering. *arXiv preprint arXiv:2406.09457*.

Smith, Ari J, Brian W Patterson, Michael S Pulia, John Mayer, Rebecca J Schwei, Radha Nagarajan, Frank Liao, Manish N Shah, and Justin J Boutilier. 2023. Multisite evaluation of prediction models for emergency department crowding before and during the covid-19 pandemic. *Journal of the American Medical Informatics Association* 30(2):292–300.

Star, Susan Leigh. 1999. The Ethnography of Infrastructure. *American Behavioral Scientist* 43(3):377–391.

———. 2015. Misplaced concretism and concrete situations: Feminism, method, and information technology. *Boundary Objects and Beyond: Working with Leigh Star* 143–167.

Star, Susan Leigh, and James R Griesemer. 1989. Institutional ecology,translations' and boundary objects: Amateurs and professionals in berkeley's museum of vertebrate zoology, 1907-39. *Social studies of science* 19(3):387–420.

Star, Susan Leigh, and Karen Ruhleder. 1996. Steps Toward an Ecology of Infrastructure: Design and Access for Large Information Spaces. *Information Systems Research* 7(1):111–134.

Stephanopoulos, Nicholas, and Eric McGhee. 2018. The measure of a metric: The debate over quantifying partisan gerrymandering.

Stephanopoulos, Nicholas O, and Eric M McGhee. 2015. Partisan gerrymandering and the efficiency gap. *The University of Chicago Law Review* 831–900.

Sun, Benjamin C., Renee Y. Hsia, Robert E. Weiss, David Zingmond, Li-Jung Liang, Weijuan Han, Heather McCreath, and Steven M. Asch. 2013. Effect of Emergency Department Crowding on Outcomes of Admitted Patients. *Annals of Emergency Medicine* 61(6):605–611.e6.

Sun, Yan, Bee Hoon Heng, Yian Tay Seow, and Eillyne Seow. 2009. Forecasting daily attendances at an emergency department to aid resource planning. *BMC emergency medicine* 9:1.

Swamy, Rahul, Douglas M King, and Sheldon H Jacobson. 2022. Multiobjective optimization for politically fair districting: A scalable multilevel approach. *Operations Research*.

Taylor, R. Andrew, Joseph R. Pare, Arjun K. Venkatesh, Hani Mowafi, Edward R. Melnick, William Fleischman, and M. Kennedy Hall. 2016. Prediction of In-hospital Mortality in Emergency Department Patients With Sepsis: A Local Big Data-Driven, Machine Learning Approach. *Academic Emergency Medicine* 23(3):269–278.

Tibshirani, Robert. 1996. Regression Shrinkage and Selection Via the Lasso. *Journal of the Royal Statistical Society: Series B (Methodological)* 58(1):267–288.

Validi, Hamidreza, Austin Buchanan, and Eugene Lykhovyd. 2022. Imposing contiguity constraints in political districting models. *Operations Research* 70(2):867–892.

Vayanos, Phebe, Duncan McElfresh, Yingxiao Ye, John Dickerson, and Eric Rice. 2020. Active Preference Elicitation via Adjustable Robust Optimization. *arXiv:2003.01899 [cs, math]*. ArXiv: 2003.01899.

Villenueve, Marina. 2022. Fight over gerrymandering argued at new york's highest court. *The Associated Press*.

Wang, Lizhi. 2009. Cutting plane algorithms for the inverse mixed integer linear programming problem. *Operations research letters* 37(2):114–116.

———. 2013. Branch-and-bound algorithms for the partial inverse mixed integer linear programming problem. *Journal of Global Optimization* 55(3):491–506.

Wargon, M., B. Guidet, T. D. Hoang, and G. Hejblum. 2009. A systematic review of models for forecasting the number of emergency department visits. *Emergency medicine journal: EMJ* 26(6):395–399.

Weerts, Hilde, Miroslav Dudík, Richard Edgar, Adrin Jalali, Roman Lutz, and Michael Madaio. 2023. Fairlearn: Assessing and improving fairness of ai systems.

Weiss, Steven J., Robert Derlet, Jeanine Arndahl, Amy A. Ernst, John Richards, Madonna Fernández-Frankelton, Robert Schwab, Thomas O. Stair, Peter Vicellio, David Levy, Mark Brautigan, Ashira Johnson, and Todd G. Nick. 2004. Estimating the Degree of Emergency Department Overcrowding in Academic Medical Centers: Results of the National ED Overcrowding Study (NEDOCS). *Academic Emergency Medicine* 11(1):38–50.

Wong, Andrew, Erkin Otles, John P. Donnelly, Andrew Krumm, Jeffrey McCullough, Olivia DeTroyer-Cooley, Justin Pestrue, Marie Phillips, Judy Konye, Carleen Penoza, Muhammad Ghous, and Karandeep Singh. 2021. External Validation of a Widely Implemented Proprietary Sepsis Prediction Model in Hospitalized Patients. *JAMA internal medicine* 181(8):1065–1070.

Young, H Peyton. 1988. Measuring the compactness of legislative districts. *Legislative Studies Quarterly* 105–115.

Yousefi, Nasrin, Timothy Chan, and Nathan Sandholtz. 2024. Uncertainty quantification in inverse optimization. Proceedings of INFORMS Annual Meeting 2024.

Zhang, Jianzhong, and Chengxian Xu. 2010. Inverse optimization for linearly constrained convex separable programming problems. *European Journal of Operational Research* 200(3): 671–679.

Zhao, Qi, Arion Stettner, Ed Reznik, Daniel Segrè, and Ioannis Ch Paschalidis. 2015. Learning cellular objectives from fluxes by inverse optimization. In *2015 54th ieee conference on decision and control (cdc)*, 1271–1276. IEEE.

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.