UNIVERSITY OF CALIFORNIA
SANTA CRUZ

LEARNING AND SOCIALLY RESPONSIBLE DECISION-MAKING WITH STRATEGIC FEEDBACK

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

DOCTOR OF PHILOSOPHY

in

COMPUTER SCIENCE AND ENGINEERING

by

Yatong Chen

June 2024

The Dissertation of Yatong Chen
is approved:

Professor Yang Liu, Chair

Professor Seshadhri Comandur

Professor Evangelos Chatziafratis

Professor Yiling Chen

Professor Aaron Roth

Dean Peter Biehl
Vice Provost and Dean of Graduate Studies

Table of Contents

List of Figures

List of Tables

Abstract

Learning and Socially Responsible Decision-Making with Strategic Feedback

by

Yatong Chen

In recent years, the concepts of "human-centered AI" and "responsible data science" have gained prominence across multiple sectors, including academia, industry, government, and civil society. This interdisciplinary field addresses the significant challenges posed by algorithmic decision-making, particularly issues of bias, fairness, robustness, and transparency in diverse applications such as education, facial recognition, and machine translation. These challenges often originate from biased training data, inadequate inclusion of minority classes, and inappropriate algorithmic choices during pre- and post-processing steps. Moreover, all of these potential problems might be reinforced by human responses, exacerbating more severe problems in the long term as people perform multiple rounds of interaction with deployed models.

To develop machine learning systems that are truly socially beneficial, it is essential to consider human behavior, especially how individuals may strategically alter their data to influence model predictions when they understand the model's mechanics. Recognizing this, the thesis focuses on learning and decision-making within the context of strategic human feedback. Despite significant progress in related research areas, the current state of research in this field still exhibits critical gaps, including a lack of practical models of human behavior, insufficient understanding of existing algorithms' limitations in scenarios involving human responses, and overlooked potential problems in algorithmic decision-making systems.

We focus on three key objectives: understanding the social impact of decision rules, designing interventions that are both socially beneficial and sustainable, and enhancing the practicality of algorithmic decision-making. By integrating theoretical and experimental methods from machine learning, algorithmic game theory, constrained optimization, and theoretical computer science, this thesis aims to demonstrate the necessity and benefits of incorporating human factors into algorithm design.

Ultimately, this thesis strives to advance the field by developing algorithms, modeling frameworks, theorems, and experimental evidence that bridge the gap between theoretical research and practical implementation in responsible machine learning.

To the memory of my grandma,

Lianfang Niu,

whose generosity and unconditional love will always be with me.

Acknowledgments

Ph.D. is a precious life experience. Throughout my Ph.D., I have had the opportunity to learn from many amazing colleagues, both at UC Santa Cruz and beyond. First and foremost, I owe an immeasurable debt of gratitude to my advisor, Yang Liu, for his generous guidance, support, and belief in me. At the beginning of my Ph.D., he introduced me to the work on strategic classification, which later proved to be perfectly aligned with my research tastes. Throughout the years, he trusted me to work on challenging projects, encouraged me to think beyond my scope, and granted me extraordinary flexibility in exploring different directions. Like many other Ph.D. students, I suffer from serious imposter syndrome. Yang, however, always has an undoubtful belief in my ability to strive in academia. He helped me appreciate my work and learned not to give up on a problem too easily. Overall, I consider myself to be extremely lucky to have Yang as my advisor. His kindness and wisdom have helped me walk through many difficult times. I also would like to express my deepest appreciation to my committee members, Professor Seshadhri Comandur, Professor Evangelos Chatziafratis, Professor Yiling Chen and Professor Aaron Roth for the time they devoted to the helpful discussions and insightful feedback.

I have had the privilege to work with many extraordinary collaborators. The successful completion of many projects would not have been possible without their invaluable collaboration and assistance. I especially thank Jimmy Wu, Andrew Estronell, and Wei Tang for their patience, sharp insights, solid theoretical backgrounds, and support. I also extend my gratitude to other collaborators: Jialu Wang, Reilly Raab, Zeyu Tang, Professor Chien-Ju Ho, Professor Yevgeniy Vorob-

eychik, and Professor Kun Zhang for their great advice and mentorship. Thank you for your insights, I have learned a lot from all of you. Also, for my collaborators when I interned at Google: Ehsan Amid, Abhishek Kumar, Manfred K Warmth, Rohan Anil, and Lawrence Peirson; thank you for giving me the industry experience during my Ph.D., which may be crucial in the future if I do not survive academia.

Furthermore, I appreciate valuable mentorship and academic inspiration from people at UC Santa Cruz and beyond. During my master's degree at Stanford, I transitioned from civil and environmental engineering to statistics. Professor Peter K. Kitanidis encouraged me to pursue my long-standing enthusiasm for mathematics and statistics. I am also extremely grateful to Professor Johan Ugandar for offering me my very first academic project in Computer Science, which further inspired me to do a Ph.D. in the field. Throughout my doctoral studies, I have closely followed Professor Moritz Hardt's research on strategic classification, performative prediction, algorithmic fairness, and more. It is an honor to continue my academic journey as a postdoc under his guidance. I also wish to acknowledge Professor Berk Ustun, Professor Lise Getoor, Professor David T Lee, Professor Eric Darve, Professor Yu Gu, and Professor Chunmiao Zheng, Linlin Chen, Samira Daruki, Lei Fang, Guanting Chen for their kindness and help throughout the years. My academic friends – Jennifer Chien, Yahav Bechavod, Siddartha Devic, Celestine Mendler-Dünner, Thomas Kleine Buening, and Tongxin Yin – have made attending large academic conferences enjoyable and reaffirm that I am not alone in my Ph.D. journey.

I am also extremely lucky to be surrounded by a group of kind and fun colleagues at UC Santa Cruz, especially everyone in the theory group (i.e., the "stronk" banana lab). Zack Jorquera, thank you for taking me on adventures and showing me how to explore the world in a way I could not have imagined. Sabyasachi Basu, we have known each other the longest at UCSC; thank you for always being there for me and pushing me outside my comfort zone. Suman Kalyan Bera, your advice in the early years of my Ph.D. was invaluable. In addition, I would also like to express my gratitude to Nicolas Menand, Reilly Raab, Shadi Haddad, Daniel Paul Pena, Omkar Bhalerao, Steven Kordonowy, Manolis Nikolakakis, Yiyuan Luo, Ehsan Mirafzali, Stuart Wayland, Olivia Anastassov, Amogh Lonkar, Omkar Lonkar, Andrea David, Lakshmi Krishnaswamy, Hari Kuttivelil, Connor Pryor, Charles Dickens, Sriram Srinivasan, Shweta Jain, William Bolden, Kostas Zampetakis, Noujan Pashanasangi and so on. Thank you for spending time together in the reading group, trivia, board games, and first Fridays. Also, to everyone in my immediate lab: Jialu Wang, Jiaheng Wei, Zhaowei Zhu, Jinlong Pang, Zonglin Di, Yaxuan Wang, Chris Liu, Jayanth Yetukuri, and Tianyi Luo. Thank you for all the great food and happy time we spent together. Special shout out to all three theory faculties at UCSC: Sesh, thank you for providing all the brilliant theory courses over the past few years and keeping the theory group at UCSC alive; Alex, you are truly my life inspiration and role model. You made me realize it is possible to do great both in and out of academia; and to Vaggos, knowing you have made it much easier for me to socialize in academic conferences.

I am also grateful to my dear friends outside UCSC. To Yiwen Guo, you are the type of friend anyone would dream of having. Your kindness and generosity have always warmed me up; thank you for taking care of me over the years. To Jiahui Wang, same as what you wrote in your thesis, thank you for being my gym buddy and the year-over-year company till today and in the future. To Dangna Li, you are a true inspiration for me in many different ways. I am grateful for all your advice and the great music events we attended together. To Jimmy Wu, I am in debt for the tremendous amount of encouragement and belief you have in me throughout the year, and all the generous help you offered me. To Jingrui Li, Ningyuan Zhang, Eric Gong, Shuoshi Qiu, Marios Galanis, Xinle Yao, Shenglin Huang, Max Xu, Xiwen Zhang, and many others, for your company and support throughout the adventure.

Special thanks to my family friends, Fang Huang, Dong Feng, Lucy Huang, and Eric Huang, for their hospitality and care for me. You are my family in California. My gratitude goes to Trader Joe's, Fillmore & 5th, Mountain View Public Library, Muse, The Strokes, and Franz Ferdinand. Thank you for your crucial contribution to my mental health throughout my PhD. Also, I thank my tortoise, Pengpeng (aka Alex), for the great company and distraction you provided me while I worked on my thesis.

Lastly, I thank my parents, Hong Zhao, and Xinzhen Chen, for their endless love and support. Having me far away is undoubtedly not easy for you. Thank you for growing up with me and providing me the courage to explore the world freely. I am profoundly indebted to them for their boundless love. This thesis is written in loving memory of my grandma, Lianfang Niu.

Chapter 1

Overture

1.1 Introduction

This thesis studies machine learning-based decision-making in the context of *strategic* human subjects. In the past few years, the concepts of "human-centered AI" and "responsible data science" have become a pivotal focus of academia, industry, government, and civil society. Researchers in this broad and interdisciplinary field are increasingly aware of the pitfalls of relying purely on algorithmic decision-making, in that there are fundamental issues around bias and unfairness [Hardt et al., 2016b, Barocas et al., 2019], robustness [Xu and Mannor, 2012], and transparency [Ustun et al., 2019]. For example, in school applications [Mehrabi et al., 2021], facial recognition [Xu et al., 2020], machine translation [Stanovsky et al., 2019], and more. Increased awareness of these issues has begun to shift the focus of machine learning research from pursuing solely predictive accuracy to including measures of fairness and social welfare.

In general, existing issues in the current machine learning literature, like bias and unfairness, arise from many sources: from the training data (which might reflect biases present in society) [Merler et al., 2019], from the training process (not sufficiently accounting for minority classes during the learning process), from seemingly innocuous pre-and post-processing steps, and even from the choice of the algorithm itself [Friedler et al., 2019]. Moreover, all of these potential problems might be reinforced by human responses, exacerbating more severe problems in the long term as people perform multiple rounds of interaction with deployed models [Liu et al., 2018].

To address these issues and create socially beneficial ML systems, researchers and practitioners need to take human behavior into account when designing algorithmic decision-making. For example, it is well-known that when a person subjected to the decision of a classification model gains information about the inner workings of that model, they have a strong incentive to modify their features so as to obtain a favorable prediction outcome — in other words, humans exhibit strategic behavior and can sometimes even reverse-engineer the decision rules.

1.1.1 Challenges and Critical Gaps in Algorithmic Decision-Making

Despite rapid advancement on related research topics, the state of the art research in algorithmic decision-making is still lacking on several fronts:

Lack of Practical Modeling of Human Behavior While much recent work in responsible ML has begun to explicitly incorporate social elements into algorithmic solutions, many of these efforts fall short of an adequate assessment of human behavior, resulting in algorithms that would be of limited use in deployment, or even introduce new issues. For example, in the vast majority of work on the influential literature on strategic classification [Hardt et al., 2016a, Chen et al., 2020b, Dong et al., 2018a], the goal is to discourage the effects of all strategic adaptation, and classify individuals as if they had not performed such adaptations; however, not all strategic behavior is socially undesirable: with the right incentives, it is often possible to leverage this behavior to the benefit of both decision-makers and subjects. An example of this comes from lending, in which a loan applicant presents their qualifications for a loan and receives a yes/no decision. If a decision subject improves their education level or salary, it should be both individual and societal beneficial; however, these improvement behaviors are treated as potentially malicious, and hence discouraged, by popular strategic classification algorithms.

Lack of Understanding the Limitations of Existing Algorithms and Frameworks in Settings with the Presence of Human Response Existing work in robust and fair ML jumps too quickly to prescribe specific algorithmic solutions or interventions before coming to an adequate understanding of the limitations of existing algorithms' performance. For example, one commonly used intervention to guarantee fair prediction among different social groups is to impose fairness constraints (e.g. enforce equal acceptance rate). However, recent work has shown that for the loan application setting, blindly enforcing fairness constraints on different societal groups might cause more harm to certain groups since in the long-term they might not actually be able to pay back their loan (see, e.g. [D'Amour et al., 2020, Liu et al., 2018, Zhang et al., 2020b]). In other words, a lack of understanding of the limitations on the standard intervention method in the presence of human response might cause more harm than no intervention at all.

Potential Issues in Algorithmic Decision-Making System Before blindly pursuing the development of socially beneficial algorithmic decision-making systems, it is essential to exercise caution and consider the potential issues that may arise. For example, while *algorithmic recourse* [Ustun et al., 2019], which aims to provide explanations and recommendations to individuals adversely affected by automated decision-making systems, is undoubtedly valuable, it is essential to recognize a potential downside – the availability of recourse may inadvertently create opportunities for strategic agents to exploit the transparency of the system, especially when agents strategically coordinate and share information. This inherent tension determines whether the system offers recourse, challenging the conventional assumption of a system's willingness to provide recourse without evaluating the rationality of such readiness. In general, failing to identify and address such potential issues proactively can lead to many problems. In an era where algorithmic systems are becoming integral to our daily lives, modeling potential issues is essential to ensure these systems serve the greater good while minimizing harm.

1.2 Overview of Results

Taking various human behaviors into account, I envision my work being able to strengthen the bond between theoretical and empirical analyses of the human-centered algorithm design. In particular, my thesis aims to advance the following directions:

1.2.1 Understanding the Social Impact of Decision Rules

Algorithms can potentially make consequential decisions that, in turn, induce complex social dynamics by influencing human outcomes. Given an algorithmic predictor that is accurate for a specific source population consisting of strategic human decision subjects, will it maintain accuracy if the population reacts to it? This question is particularly crucial when machine learning practitioners have access only to training data from the source distribution but expect changes due to human responses, and where retraining is either too costly or unavailable. To address this, in Chapter 3, we provide modeling frameworks and theoretical guarantees on how performance (i.e., accuracy and fairness) deteriorates due to distribution shifts primarily caused by human strategic behavior.

1.2.2 Socially Beneficial and Long-Term Intervention

Addressing the challenge of guiding human agents to act in socially beneficial ways when anticipating strategic behavior is central to my research. In Chapter 4, we propose a novel training approach that bridges the gap between traditional strategic classification and incentive-compatible machine learning. This approach offers an alternative to the predominantly pessimistic view found in conventional strategic classification literature. In Chapter 5, we analyze the incentives for decision-makers to offer recourse to a set of negatively affected applicants. In particular, we ask the question: does the decision-maker have the incentive to offer recourse to all rejected applicants? Contrary to the classic assumption that the algorithmic recourse system is always willing to provide recourse to individuals, we show that a utility-maximizing decision-maker does not have an incentive to offer recourse to all applicants, especially when the recourse process is possible to manipulate through imitation and collective behavior. We then propose efficient intervention tools to make recourse-providing incentive-compatible.

1.2.3 Making Algorithmic Decision-Making Practical

Ensuring the practicality of algorithmic decision-making is paramount in today's data-driven world. Practicality ensures that machine learning systems extend beyond academic exercises and effectively address real-world problems. Bridging the gap between theoretical advancements and real-world applications allows these systems to have a meaningful impact. Practicality includes considerations such as scalability, interpretability, and usability, enabling decision-makers to deploy these algorithms at scale and trust their results. Furthermore, practicality promotes responsible AI by ensuring that algorithms can adapt to the dynamic nature of human behavior and evolving data distributions, thereby making them more robust and fair in the face of real-world challenges. In Chapter 6, we study the derandomization of stochastic classifiers, which appear in various settings (e.g., as solutions to constrained optimization problems) but are impractical due to their inherent randomness. We provide a straightforward derandomization procedure with guarantees for both accuracy and individual fairness, making the deployment of stochastic classifiers more viable in real-life scenarios.

The ultimate aim of this thesis is to produce algorithms, modeling frameworks, theorems, and experimental findings that bring the state of the art in research closer to practical deployability.

1.2.4 Research Publications Underpinning This Thesis

This thesis draws much of its content from the following published manuscripts or preprints.

- *Model Transfereability with Responsive Decision Subjects* (Chapter 3). Joint work with Zeyu Tang, Kun Zhang and Yang Liu, appearing at ICML 2023. The preliminary version won the Best Paper Award at ICML 2022 Workshop on Adversarial Machine Learning Frontiers. [Chen et al., 2023].
- *Fair Transferability Subject to Distribution Shift* (Chapter 3). Joint work with Reilly Raab, Jialu Wang, and Yang Liu, appearing in NeurIPS 2022. [Chen et al., 2022].
- *Learning to Incentivize Improvements from Strategic Agents* (Chapter 4). Joint work with Jialu Wang, and Yang Liu, appearing in the Transactions on Machine Learning Research (TMLR). The preliminary version won the Best Paper Award at the ICML 2021 Workshop on Algorithmic Recourse. [Chen et al., 2020a].
- *To Give or Not to Give? In the Impacts of Strategically Withheld Recourse* (Chapter 5). Joint work with Andrew Estornel, Yevgeniy Vorobeychik, and Yang Liu. Preprint. [Chen et al., 2024].
- *Metric-Fair Classifier Derandomization* (Chapter 6). Joint work with Jimmy Wu and Yang Liu, appearing at ICML 2022. [Wu et al., 2022].

Chapter 2

Background, Preliminaries and Related Works

In conjunction with the literature review, this chapter will establish preliminary and basic notations by discussing key concepts and ideas drawn from existing research. We will cover foundational topics and their applications on strategic classification, performative prediction, algorithmic recourse, algorithmic fairness, and other related topics The goal is to ensure a comprehensive understanding of the field and lays the groundwork for the following discussions and analyses.

2.1 Preliminaries

Let $\mathscr{X} \subset \mathbb{R}^d$ and $\mathscr{Y} \equiv \{0, 1\}$ be a domain of features and labels respectively. We consider a classification task of training a classifier $f^1 \in \mathscr{F} : \mathscr{X} \to \mathscr{Y}$ from a dataset of n examples $\{(x_i, y_i)\}_{i=1}^n \sim \mathscr{D}$. Example i corresponds to an agent who wishes to receive a positive prediction and may alter their features to obtain such a prediction once the model is deployed. We assume that an agent's true

[1] We may use h to represent a classifier in later chapters.

qualification (or label), denoted as y, is always a function of its feature vector x, and define the true unknown qualification function $\mathbb{y} : \mathscr{X} \to \mathscr{Y}$ as the mapping between the feature vector $x \in \mathscr{X}$ and the true qualification/label $y \in \mathscr{Y}$.

Standard Empirical Risk Minimization In a standard prediction setting, a model designer trains a classifier that minimizes the *empirical risk*:

$$f^*_{\mathsf{ERM}} \in \underset{f \in \mathscr{F}}{\arg\min}\, \mathbb{E}_{(x,y)\sim\mathscr{D}} \mathbb{1}[(f(x) \neq y)]$$

Notice that this classifier may perform poorly in a setting with strategic adaptation since the model is deployed on a population with a different distribution over $\mathscr{X}$ as decision subjects alter their features. For instance, when a model is used to decide loan applications, candidates may adapt their features based on the model specification in order to maximize their chances of approval; thus the loan decision classifier observes a new shifted distribution caused by its own deployment (e.g., see Figure 2.1 for a demonstration). Similar observations can be articulated for application in the insurance sector, e.g., insurance companies may develop policy such that customers' behaviors might adapt to lower premium [Haghtalab et al., 2020], the education sector, e.g., teachers may want to design courses in a way that students are less incentivized to cheat [Kleinberg and Raghavan, 2020], and so on.

2.2 Strategic Classification

Strategic classification focuses on the problem of how to make predictions in the presence of agents who behave strategically to obtain desirable outcomes [Hardt et al., 2016a, Chen et al., 2020b, Dong et al., 2018a, Chen et al., 2020a, Miller et al., 2020]. In particular, [Hardt et al., 2016a] first formalizes strategic classification

Feature	Weight	Original Value		Adapted Value
Income	2	\$ 6,000	⟶	\$ 6,000
Education Level	3	College	⟶	College
Debt	**-10**	\$40,000	⟶	**\$20,000**
Savings	**5**	\$20,000	⟶	**\$0**

Table 2.1: An example of an agent who originally has both savings and debt, observes that the classifier penalizes debt (weight -10) more than it rewards savings (weight +5), and concludes that their most efficient adaptation is to use their savings to pay down debt.

tasks as a two-player sequential game (i.e., a Stackelberg game) between a model designer and strategic agents as follows:

Definition 2.2.1 (Full Information Strategic Classification Game, Hardt et al. [2016a])**.** *The two players are the model designer and strategic agents. Fix a population X, and a probability distribution $\mathscr{D}$ over $(\mathscr{X}, \mathscr{Y})$. Fix a cost function $c : \mathscr{X} \times \mathscr{X} \to \mathbb{R}^+$.*

1. *A model designer (who knows c and $\mathscr{D}$), publishes a classifier $f : \mathscr{X} \to \{-1, +1\}$ from a hypothesis class $\mathscr{F}$, which is also their action space.*
2. *Strategic agents, who adapt their features from x to x' so as to be assigned $f(x') = +1$ if possible. The action space for the decision subjects includes all feature vectors that are within a given manipulation budget B, namely $\forall x' \in \mathscr{X}$ such that $c(x, x') \leq B$.*

Denote the best response action of the agent with feature x as $\Delta(x)$. The payoff to the model designer is $\Pr_{((x,y)\sim\mathscr{D}}[f(\Delta(x)) = y]$, and the payoff to the strategic agent with feature $x \in X$ is $f(\Delta(x)) - c(x, \Delta(x))$.

Most existing work in strategic classification assumes that human agents are fully rational and will always perform *best response* to any given classifier. As a result, their behaviors can be fully characterized based on pre-specified human response models [Hardt et al., 2016a, Chen et al., 2020b]. Agent best response behavior is typically viewed as *malicious* in the traditional setting; as a result, the model designer seeks to disincentivize this behavior or limit its impact by publishing classifiers robust to any agent's adaptations.

Strategic Risk Minimization Existing approaches in strategic classification tackle these issues by training a robust classifier to *all* adaptation. This approach treats all adaptation as undesirable, and seeks to maximize accuracy by discouraging it entirely. Formally, they train a classifier that minimizes the *strategic risk*:

$$f^*_{\mathsf{SC}} \in \arg\min_{f \in \mathscr{F}} \mathbb{E}_{(x,y)\sim\mathscr{D}} \mathbb{1}[(\Delta(x) \neq y)]$$

However, this classifier still achieves only suboptimal accuracy, as it overlooks potential changes in the true label, y. Specifically, since y depends on the feature vector $\mathbf{x}$, an updated feature vector, $\Delta(x)$, results in a revised true label, $\mathbb{y}(\Delta(x))$. Additionally, this design choice does not take advantage of the opportunity to encourage an *improvement* in the profile x and its corresponding qualification, $\mathbb{y}(x)$.

Different Types of Strategic Behaviors In reality, there are two types of strategic adaptations:

1. *Gaming* corresponds to interventions that change the classification outcome $f(X)$, but do not change the true label Y. Classic strategic classification assumes that any adaptation is always gaming.

2. *Improvement* corresponds to interventions that change *both* the classification $f(X)$ and the true label Y. Incentivizing improvement requires inducing

agents to intervene on causal features that can change the label Y rather than non-causal features.

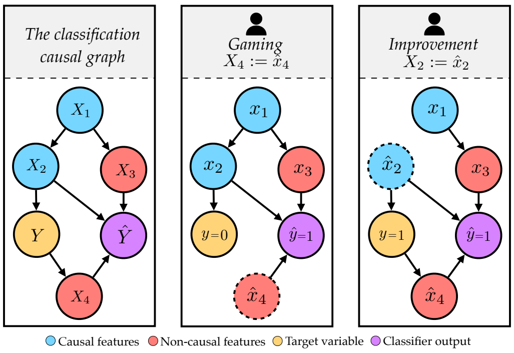


Figure 2.1: Illustration of the causal framework for strategic adaptation. Strategic adaptation is modeled as interventions in a counterfactual causal graph, conditioned on the individual's initial features X. *Gaming* corresponds to interventions that change the classification $f(X)$ (or $\hat{Y}$), but do not change the true label Y. Improvement corresponds to interventions that change both the classification $f(X)$ (or $\hat{Y}$) and the true label Y. Incentivizing improvement requires inducing agents to intervene on causal features that can change the label Y rather than non-causal features. Plot sourced from Miller et al. [2020].

Distinguishing between these two categories of features generally requires nontrivial causal analysis [Miller et al., 2020], see, e.g., Figure 2.1 for a demonstration. Ideally, the model designer should design a classifier that incentivizes improvement while discouraging gaming. In Chapter 4, we propose an intuitive way to incentivize improvement behavior from strategic agents.

2.3 Algorithic Recourse

Also related is the recent development of *algorithmic recourse* [Ustun et al., 2019, Venkatasubramanian and Alfano, 2020, Karimi et al., 2020b, Gupta et al., 2019, Karimi et al., 2020c, von Kügelgen et al., 2020]. Recourse is defined as the ability of a person to obtain a desired outcome from a *fixed* model.

In particular, given a person who is assigned an undesirable outcome $f(\boldsymbol{x}) = -1$, we aim to find an action $\boldsymbol{a}^*$ such that $f(\boldsymbol{x}+\boldsymbol{a}) = +1$ by solving the following optimization problem:

$$\begin{aligned} \boldsymbol{a}^* = \arg\min_{\boldsymbol{a}} \operatorname{cost}(\boldsymbol{a};\boldsymbol{x}) \\ \text{s.t. } f(\boldsymbol{x}+\boldsymbol{a}) = +1, \\ \boldsymbol{a} \in A(\boldsymbol{x}). \end{aligned}$$

Here, $A(\boldsymbol{x})$ is a set of feasible actions specified by the decision maker to $\boldsymbol{x}$, $\operatorname{cost}(\cdot;\boldsymbol{x}) : A(\boldsymbol{x}) \to \mathbb{R}$ is a cost function to choose between feasible actions.

The concept was first introduced to the machine learning community in [Ustun et al., 2019]. There, an integer programming solution was developed to offer actionable recourse from a linear classifier. Later, Venkatasubramanian and Alfano [2020] discusses a more adequate conceptualization and operationalization of recourse. Karimi et al. [2020b] provides a thorough survey of algorithmic recourse in terms of its definitions, formulations, solutions, and prospects. Rudin [2019] argues the sufficiency of recourse in explainable machine learning. Bellamy et al. [2018] builds toolkits for actionable recourse analysis. Furthermore, Gupta et al. [2019] studies how to mitigate disparities in recourse across populations.

Causal Recourse Recent developments in causal recourse propose to model algorithmic recourse through *counterfactual explanations* [Karimi et al., 2020c, 2021, von Kügelgen et al., 2020, 2022]. The formulation of causal recourse typically involves the *identification* and *modification* of causal relationships within a model. This requires a causal model or graph that specifies how features (variables) interact and influence each other, including the decision outcome. The goal is to find a set of interventions (or changes to input features) that will lead to a change in the output, ensuring that these interventions are feasible and have a *genuine* causal effect. We point the reader to [Karimi et al., 2020b] for a detailed read on this topic.

Comparison between Recourse and Pure Manipulation Recourse can be viewed as a *system-aided* tool to incentivize agent's good strategic behavior. There are two major differences between recourse and generic strategic behavior we mentioned in the previous section (Section 2.2):

1. Action Space: for recourse, the actions are specified by the decision maker (i.e., $a \in A(x)$, while for strategic manipulation, or strategic adaptation in general, the actions can be arbitrarily chosen by the agents as long as the manipulation cost is within budget.

2. Changes in True Qualification y: taking recourse implies that agents change their corresponding true value y, whereas when manipulation is malicious (i.e., gaming rather than gaming), it is simply a misreport rather than a change of one's features. Thus, the true qualification in that case remains the same.

2.4 Performative Prediction

Performative prediction is a new type of supervised learning problem in which the underlying data distribution shifts in response to the deployed model [Perdomo et al., 2020, Mendler-Dünner et al., 2020, Brown et al., 2020, Drusvyatskiy and Xiao, 2020, Izzo et al., 2021a, Li and Wai, 2022, Maheshwari et al., 2021].

In particular, Perdomo et al. [2020] first propose the notion of the *performative prediction risk* defined as

$$\textbf{Performative Risk:} \qquad \mathrm{PR}(\theta) := \mathbb{E}_{z\sim\mathscr{D}(\theta)}[\ell(\theta; z)] \tag{2.1}$$

where $z = (x, y)$ denote the feature and label pair, $\theta \in \Theta$ is the model parameter, ℓ is a loss function, and $\mathscr{D}(\theta)$ is the *induced* distribution as a result of the deployment of the model θ. We can think of $\mathscr{D}(\theta)$ as the distribution over features and outcomes that result from making decisions according to the model specified by θ.

One of the primary focuses of performative prediction is to find the optimal model θ_{OPT} which achieves the minimum performative prediction risk:

$$\theta_{\mathsf{OPT}} = \arg\min_{\theta} \mathop{\mathbb{E}}_{Z\sim\mathscr{D}} \ell(Z; \theta)$$

Another line of work focuses on finding the performative stable model θ_{ST}, which is optimal under its own induced distribution:

$$\theta_{\mathsf{ST}} = \arg\min_{\theta} \mathop{\mathbb{E}}_{Z\sim\mathscr{D}} \ell(Z; \theta)$$

In particular, one way to find a performative stable model θ_{ST} is to perform repeated retraining on the distribution resulting from the previous model, corresponding to the update rule:

$$\theta_{t+1} = \arg\min_{\theta} \mathop{\mathbb{E}}_{Z\sim\mathscr{D}(\theta_t)} \ell(Z; \theta).$$

In order to get meaningful theoretical guarantees on any proposed algorithms, works in this field generally require particular assumptions on the mapping between the model parameter and its induced distribution (e.g., the smoothness of the mapping), or requires multiple rounds of deployments and observing the corresponding induced distributions, which can be costly in practice [Jagadeesan et al., 2022a, Mendler-Dünner et al., 2020]. Other recent developments include the multiplayer version of the performative prediction problem [Piliouras and Yu, 2022, Narang et al., 2022], and the economic aspects of performative prediction [Hardt et al., 2022, Mendler-Dünner et al., 2022].

2.5 Algorithmic Fairness

This thesis also contributes to the broad study of algorithmic fairness in machine learning. Much of this study focuses on two notions of fairness: group fairness and individual fairness.

Group Fairness The general recipe for a notion of group fairness consists of three main components: (1) identify a *sensitive attribute*, which defines "protected groups" (e.g., gender, race, sexual orientation, etc). We will denote A as the sensitive attribute. (2) identify a statistic of interest, (e.g., prediction accuracy, true/false positive rate); (3) ensure that the statistics of interest are 'similar' across the groups defined by the sensitive attribute.

Most common notions of group fairness include disparate impact [Feldman et al., 2015], demographic parity [Agarwal et al., 2018], disparate mistreatment [Zafar et al., 2019], equality of opportunity [Hardt et al., 2016b] and calibration [Chouldechova, 2017]. Here, we introduce one notion that we will be using in the later chapters:

Definition 2.5.1 (Demographic Parity, [Dwork et al., 2012])**.** *A classifier f: $\mathscr{X} \to \mathscr{Y}$ satisfies demographic parity if for any pair of groups $A = a, a'$, we have:*

$$\Pr[f(x) = 1 | A = a] = \Pr[f(x) = 1 | A = a'] \tag{2.2}$$

This means that the rate at which the classifier accepts individuals from $A = a$ and from $A = a'$ (i.e. $f(x) = 1$) is equal. For instance, a company screening candidates for a job may enforce demographic parity to ensure they interview roughly the same number of men and women.

Individual (Metric-)Fairness: The idea behind individual fairness is similar individuals should be treated similarly. Mathematically speaking, it means that the classifier should be an approximately Lipschitz-continuous function relative to a given distance metric:

Definition 2.5.2 ((α, β, d)-metric fairness)**.** *Let* $\alpha \geq 1$ *and* $\beta \geq 0$*, let* $d : X^2 \to [0, 1]$ *be a metric, and let* $x, x' \in X$*. We say a classifier* $f : X \to [0, 1]$ *satisfies* (α, β, d)-metric fairness *on* (x, x')*, or is* (α, β, d)-fair *on* (x, x')*, if*

$$\left|f(x) - f(x')\right| \leq \alpha \cdot d(x, x') + \beta \tag{2.3}$$

In Chapter 6, we study classifier derandomization with such metric fairness guarantees.

Fairness in Recourse and Strategic Classification Fairness has also been explored in the algorithmic recourse and strategic classification literature. For example, existing works on fairness in recourse emphasize the importance of equitable recourse and explore various remedying unfair recourse decisions [Gupta et al., 2019, von Kügelgen et al., 2022, Ehyaei et al., 2023]. Among them, disparities in the recourse fraction can be viewed as equality of false positive rate (FPR) in the strategic classification setting. Fairness with the presence of strategic behavior has featured studies that highlight the inequity that results from strategic behavior by individuals [Hu et al., 2019], as well as inequity (e.g., social cost) resulting from making classifiers robust to strategic behavior [Milli et al., 2019, Estornell et al., 2023b]. Disparities in recourse costs and flipsets are empirically demonstrated in this thesis (see, e.g., Chapter 4).

2.6 Other Related Works

Incentive Design This thesis is also closely related to the literature on mechanism design in economy. Similar to the work in Chapter 4, Kleinberg and Raghavan [2020] discusses how to incentivize decision subjects to improve a certain subset of features. Next, Haghtalab et al. [2020] shows that an appropriate projection is an optimal linear mechanism for strategic classification, as well as an *approximate* linear threshold mechanism. Liu et al. [2020] considers the equilibria of a dynamic decision-making process in which individuals from different demographic groups invest rationally, and compares the impact of two interventions: decoupling the decision rule by group and subsidizing the cost of investment.

Domain Adaptation Work in Chapter 3 is closely related to the literature on domain adaptation. There has been tremendous work in domain adaptation studying different distribution shifts and learning from shifting distributions [Jiang, 2008, Ben-David et al., 2010a, Sugiyama et al., 2008, Zhang et al., 2019, Kang et al., 2019, Zhang et al., 2020a, Xie et al., 2022]. There are a few interesting subareas in domain adaptations: (1)Adversarial attack [Chakraborty et al., 2018, Papernot et al., 2016, Song et al., 2019]. Adversarial attack involves manipulating the input data to a machine learning model with the intent to cause the model to make errors. These attacks exploit vulnerabilities in the model's design or training data. (2) Domain generalization [Wang et al., 2021c, Li et al., 2017, Muandet et al., 2013]:

the goal of domain generalization is to learn a model that can be generalized to any unseen distribution. (3) Test-time adaptation [Varsavsky et al., 2020, Wang et al., 2021a, Nado et al., 2021]: the issue of test-time adaptation falls into the classical domain adaptation setting where the adaptation is independent of the model being deployed. Applying this technique to solve our problem requires accessing data (either unsupervised or supervised) drawn from both domains.

Chapter 3

Model Transferability with Strategic Decision Subjects

In this chapter, we provide a general framework for quantifying the *transferability of a decision rule* when facing responsive decision subjects. Specifically, we consider a setting where the deployed machine learning models interact with human agents, and will ultimately face data distributions that reflect how human agents respond to the models. In this case, how does the performance of the classifier primarily trained on the source distribution fares in the induced distribution?

3.1 Model Transferability Problem

Decision-makers are increasingly required to be transparent on their decision-making rules to offer the "right to explanation" [Goodman and Flaxman, 2017, Selbst and Powles, 2018, Ustun et al., 2019]. Being transparent also invites potential adaptations from the population, leading to potential shifts. We are motivated by settings where the deployed machine learning models interact with human agents, and will ultimately face data distributions that reflect human agents' responses to the models. For instance, when a model is used to decide loan applications, candidates may adapt their features based on the model specification to maximize their chances of approval; thus, the loan decision classifier observes a new shifted distribution caused by its own deployment (e.g., see Figure 2.1 for a demonstration).

In this chapter, we provide a general framework for quantifying the *transferability of a decision rule* when facing responsive decision subjects. What we would like to achieve is some characterizations of the *performance guarantee* of a classifier — that is, given a model primarily trained on the source distribution $\mathscr{D}_S$, how good or bad will it perform on the distribution it induces $\mathscr{D}(h)$, which depends on the model h itself. A key concept in our setting is the *induced risk*, defined as the error a model incurs on the distribution induced by itself:

$$\textbf{Induced Risk}: \quad \mathrm{Err}_{\mathscr{D}(h)}(h) := \mathbb{P}_{\mathscr{D}(h)}(h(X) \neq Y) \tag{3.1}$$

Most relevant to the above formulation are the works of literature on *strategic classification* Hardt et al. [2016a], and *performative prediction* [Perdomo et al., 2020]. In strategic classification, agents are modeled as rational utility maximizers, and under a specific agent's response model, game theoretical solutions were proposed to model the interactions between the agents and the decision-maker. In performative prediction, a similar notion of risk called the *performative prediction risk* is introduced to measure a given model's performance on the distribution itself induces. Unlike ours, one of their main focuses is finding the optimal classifier that achieves minimum induced risk after a sequence of model deployments and observing the corresponding response datasets, which might be computationally expensive.

3.1.1 Motivation

In particular, our results are motivated by the following challenges in more general scenarios:

- **Modeling assumptions being restrictive** In many practical situations, it is often hard to accurately characterize the agents' utilities. Furthermore, agents might not be fully rational when they respond. All the uncertainties can lead to a far more complicated distribution change in (X, Y), as compared to often-made assumptions that agents only change X but not Y [Hardt et al., 2016a, Chen et al., 2020b, Dong et al., 2018b].
- **Lack of access to response data** During training, machine learning practitioners may only have access to data from the source distribution, and even when they can anticipate changes in the population due to human agents' responses, they cannot observe the newly shifted distribution until the model is actually deployed.
- **Retraining being costly** Even when samples from the induced data distribution are available, retraining the model from scratch may be impractical due to computational constraints, and will result in another round of agents' response at its deployment.

The above observations motivate us to focus on understanding the transferability of a model before diving into finding the optimal solutions that achieve the minimum induced risk – the latter problem often requires more specific knowledge on the mapping between the model and its induced distribution, which might not be available during the training process. Another related research problem is to find models that will perform well on both the source and the induced distribution.

This question might be solved using techniques from *domain generalization* [Zhou et al., 2021, Sheth et al., 2022].

3.1.2 Our Contributions

Overall, we aim to provide answers to the following fundamental questions:

1. **Source risk $\Rightarrow$ Induced risk** For a given model h, how different is $\mathrm{Err}_{\mathscr{D}(h)}(h)$, the error on the distribution induced by h, from $\mathrm{Err}_{\mathscr{D}_S}(h) := \mathbb{P}_{\mathscr{D}_S}(h(X) \neq Y)$, the error on the source?

2. **Induced risk $\Rightarrow$ Minimum induced risk** How much higher is $\mathrm{Err}_{\mathscr{D}(h)}(h)$, the error on the induced distribution, than $\min_{h'} \mathrm{Err}_{\mathscr{D}(h')}(h')$, the minimum achievable induced error?

3. **Induced risk of *source optimal* $\Rightarrow$ Minimum induced risk** Of particular interest, and as a special case of the above, how does $\mathrm{Err}_{\mathscr{D}(h^*_S)}(h^*_S)$, the induced error of the optimal model trained on the source distribution $h^*_S := \arg\min_h \mathrm{Err}_{\mathscr{D}_S}(h)$, compare to $h^*_T := \arg\min_h \mathrm{Err}_{\mathscr{D}(h)}(h)$?

4. **Lower bound for learning tradeoffs** What is the minimum error a model must incur on either the source distribution $\mathrm{Err}_{\mathscr{D}_S}(h)$ or its induced distribution $\mathrm{Err}_{\mathscr{D}(h)}(h)$?

For the first three questions, we prove upper bounds on the additional error incurred when a model trained on a source distribution is transferred over to its induced domain. We also provide lower bounds for the trade-offs a classifier has to suffer on either the source training distribution or the induced target distribution. We then show how to specialize our results to two popular domain adaptation settings: *covariate shift* [Shimodaira, 2000, Zadrozny, 2004, Sugiyama et al., 2007, 2008, Zhang et al., 2013b] and *target shift* [Lipton et al., 2018, Guo et al., 2020, Zhang et al., 2013b].

3.1.3 How Does Our Work Relate to the Surrounding Literatures?

Our work most closely relates to strategic classification, domain adaptation, and performative prediction. This section discusses how our work relates to these three related literatures.

Strategic Classification In traditional strategic classification setting, agent's best response behavior is typically viewed as malicious (see, e.g., [Hardt et al., 2016a]). As a result, the model designer seeks to disincentivize this behavior or limit its impact by publishing classifiers that are robust to any agent's adaptations. In our work, the agents' strategic behaviors are not necessarily malicious; instead, we aim to provide a general framework that works for any distribution shift resulting from the human agency.

In addition, most existing work in strategic classification assumes that human agents are fully rational and will always perform *best response* to any given classifier. As a result, their behaviors can be fully characterized based on pre-specified human response models [Hardt et al., 2016a, Chen et al., 2020b]. While we are also interested in settings where agents respond to a decision rule, we focus on the distribution shift of human agents at a population level and characterize the induced distribution as a function of the deployed model. Instead of specifying a particular individual-level agent's response model, we only require the knowledge of the source data $\mathscr{D}_S$, as well as some characterizations of the relationship between the source and the induced distribution, e.g., they satisfy some particular distribution shift models, like covariate shift (see Section 3.4), or target shift (see Section 3.5), or we have access to some data points from the induced distribution so we can estimate their statistical differences like H-divergence (see Section 3.3).

Performative Prediction Similar to our definition of induced risk, performative risk (defined in Equation (2.1)), also measures a given model's performance on the distribution itself. The major difference between our work and performative prediction is that we focus on different aspects of the induced domain adaptation problem. Instead of focusing on finding the optimal model θ_{OPT} which achieves the minimum performative prediction risk, our work's primary focus is to study the *transferability* of a particular model trained primarily on the source distribution and provide theoretical bounds on its performance on its induced distribution, which is useful for estimating the effect of a given classifier when repeated retraining is unavailable. As a result, our work does not assume the knowledge of the supervision/label information on the transferred domain.

Domain Adaptation Our results differ from these previous works in domain adaptation: in our setting, changes in distribution are not passively provided by the environment, but rather an active consequence of model deployment. Part of our technical contributions is inspired by the transferability results in domain adaptations [Ben-David et al., 2010a, Zadrozny, 2004, Gretton et al., 2009, Sugiyama et al., 2008, Lipton et al., 2018, Azizzadenesheli et al., 2019].

Our work, at first sight, looks similar to several sub-areas within the literature of domain adaptation, e.g., domain generalization, adversarial attack, and test-time adaptation, to name a few. For instance, the notion of observing an "induced distribution" resembles similarity of the adversarial machine learning literature [Lowd and Meek, 2005, Huang et al., 2011, Vorobeychik and Kantarcioglu, 2018]. One of the major differences between ours and adversarial machine learning is that in adversarial machine learning, the true label Y stays the same for the attacked feature, while in our paper, both X and Y might change in the induced distribution $\mathscr{D}(h)$.

The details for reproducing our experimental results can be found at `https://github.com/UCSC-REAL/Model_Transferability`.

3.2 Notation and Formulation

Suppose we are given a parametric model $h \in \mathscr{H}$ primarily trained on the training data set $S := \{x_i, y_i\}_{i=1}^N$, which is drawn from a *source* distribution $\mathscr{D}_S$, where $x_i \in \mathbb{R}^d$ and $y_i \in \{-1, +1\}$. However, h will then be deployed in a setting where the samples come from a *test* or *target* distribution $\mathscr{D}_T$ that can differ substantially from $\mathscr{D}_S$. Therefore, instead of finding a classifier that minimizes the prediction error on the source distribution $\mathrm{Err}_{\mathscr{D}_S}(h) := \mathbb{P}_{\mathscr{D}_S}(h(X) \neq Y)$, ideally the decision maker would like to find h^* that minimizes $\mathrm{Err}_{\mathscr{D}_T}(h) := \mathbb{P}_{\mathscr{D}_T}(h(X) \neq Y)$. This is often referred to as the *domain adaptation problem*, where typically, the transition from $\mathscr{D}_S$ to $\mathscr{D}_T$ is assumed to be independent of the model h being deployed.

We consider a setting in which the distribution shift depends on h, or is thought of as being *induced* by h. We will use $\mathscr{D}(h)$ to denote the *induced domain* by h:

$$\mathscr{D}_S \quad \rightarrow \quad \textit{encounters model } h \quad \rightarrow \quad \mathscr{D}(h)$$

Strictly speaking, the induced distribution is a function of both $\mathscr{D}_S$ and h and should be better denoted by $\mathscr{D}_S(h)$. To ease the notation, we will stick with $\mathscr{D}(h)$, but we shall keep in mind its dependency of $\mathscr{D}_S$. For now, we do not specify the dependency of $\mathscr{D}(h)$ as a function of $\mathscr{D}$ and h, but later in Section 3.4 and 3.5 we will further instantiate $\mathscr{D}(h)$ under specific domain adaptation settings.

The challenge in the above setting is that when training h, the learner needs to carry the thoughts that $\mathscr{D}(h)$ should be the distribution it will be evaluated on and that the training cares about. Formally, we define the *induced risk* of a classifier h as the 0-1 error on the distribution h induces:[1]

$$\text{Induced risk}: \quad \mathrm{Err}_{\mathscr{D}(h)}(h) := \mathbb{P}_{\mathscr{D}(h)}(h(X) \neq Y). \tag{3.2}$$

Denote by $h^*_T := \arg\min_{h \in \mathscr{H}} \mathrm{Err}_{\mathscr{D}(h)}(h)$ the classifier with minimum induced risk. More generally, when the loss may not be the 0-1 loss, we define the *induced ℓ-risk* as

$$\text{Induced } \ell\text{-risk}: \quad \mathrm{Err}_{\ell,\mathscr{D}(h)}(h) := \mathbb{E}_{z \sim \mathscr{D}(h)}[\ell(h; z)]$$

The induced risks will be the primary quantities we are interested in quantifying. The following additional notation will also help present our theoretical results in the following few sections:

- Distributions of Y on a distribution $\mathscr{D}$: $\mathscr{D}_Y := \mathbb{P}_{\mathscr{D}}(Y = y)$, and in particular $\mathscr{D}_Y(h) := \mathbb{P}_{\mathscr{D}(h)}(Y = y)$, $\mathscr{D}_{Y|S} := \mathbb{P}_{\mathscr{D}_S}(Y = y)$.
- Distribution of h on a distribution $\mathscr{D}$: $\mathscr{D}_h := \mathbb{P}_{\mathscr{D}}(h(X) = y)$, and in particular $\mathscr{D}_h(h) := \mathbb{P}_{\mathscr{D}(h)}(h(X) = y)$, $\mathscr{D}_{h|S} := \mathbb{P}_{\mathscr{D}_S}(h(X) = y)$.
- Marginal distribution of X for a distribution $\mathscr{D}$: $\mathscr{D}_X := \mathbb{P}_{\mathscr{D}}(X = x)$, and in particular $\mathscr{D}_X(h) := \mathbb{P}_{\mathscr{D}(h)}(X = x)$, $\mathscr{D}_{X|S} := \mathbb{P}_{\mathscr{D}_S}(X = x)$.[2]
- Total variation distance [Ali and Silvey, 1966]: $d_{\mathrm{TV}}(\mathscr{D}, \mathscr{D}') := \sup_{\mathscr{O}} |\mathbb{P}_{\mathscr{D}}(\mathscr{O}) - \mathbb{P}_{\mathscr{D}'}(\mathscr{O})|$.

[1] The ":=" defines the RHS as the probability measure function for the LHS.

[2] For continuous X, the probability measure shall be read as the density function.

3.2.1 Example Induced Domain Adaptation Settings

We provide two example models to demonstrate the use cases of the distribution shift models described in our paper. We provide more detailed descriptions of both settings and instantiate our bounds in Section 3.4.3 and Section 3.5.3, respectively.

Strategic Response As mentioned before, one example of induced distribution shift is when human agents perform *strategic response* to a decision rule. In particular, it is natural to assume that the mapping between feature vector X and the qualification Y before and after the human agents' best response satisfies *covariate shift*: the feature distribution $\mathbb{P}(X)$ will change, but $\mathbb{P}(Y|X)$, the mapping between Y and X, remain unchanged. Notice that this is different from the assumption made in the classic strategic classification setting Hardt et al. [2016a], where *any* adaptations are considered malicious, which means any changes in the feature vector X *do not* change the underlying true qualification Y. In this example, we assume that changes in feature X could potentially lead to changes in the true qualification Y and that the mapping between Y and X remains the same before and after the adaptation. This is a common assumption made in a recent line of work on incentivizing improvement behaviors from human agents (see, e.g., Chen et al., 2020a, Shavit et al., 2020). We use Figure 3.1 (top) as a demonstration of how distribution might shift for the strategic response setting. In Section 3.4.3, we will use the strategic classification setup to verify our obtained results.

Replicator Dynamics Replicator dynamics is a commonly used model to study the evolution of an adopted "strategy" in evolutionary game theory [Tuyls et al., 2006, Friedman and Sinervo, 2016, Taylor and Jonker, 1978, Raab and Liu, 2021].

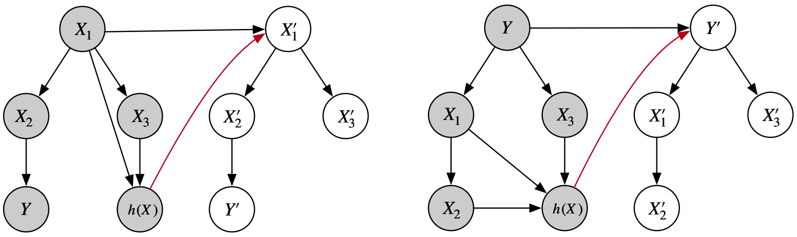


Figure 3.1: Example causal graph annotated to demonstrate covariate shift (the top panel) and target shift (the bottom panel) as a result of the deployment of h. Grey nodes indicate observable variables and transparent nodes are not observed at the training stage. Red arrow emphasizes h induces changes in certain variables.

The core notion of it is the growth or decline of the population of each strategy depends on its "fitness". Consider the label $Y = \{-1, +1\}$ as the strategy, and the following behavioral response model to capture the induced target shift:

$$\frac{\mathbb{P}_{\mathscr{D}(h)}(Y = +1)}{\mathbb{P}_{\mathscr{D}_S}(Y = +1)} = \frac{\textbf{Fitness}(Y = +1)}{\mathbb{E}_{\mathscr{D}_S}[\textbf{Fitness}(Y)]}$$

The intuition behind the above equation is that the change of the $Y = +1$ population depends on how predicting $Y = +1$ "fits" a certain utility function. For instance, the "fitness" can take the form of the prediction accuracy of h for class $+1$, namely $\textbf{Fitness}(Y = +1) := \mathbb{P}_{\mathscr{D}_S}(h(X) = +1 | Y = +1)$. Intuitively speaking, a higher "fitness" describes more success of agents who adopted a certain strategy ($Y = -1$ or $Y = +1$). Therefore, agents will imitate or replicate their successful peers by adopting the same strategy, resulting in an increase in the population ($\mathbb{P}_{\mathscr{D}(h)}(Y)$).

With the assumption that $\mathbb{P}(X|Y)$ stays unchanged, this instantiates one example of a specific induced *target shift*. We will provide detailed conditions for target shift in Section 3.5. We also use Figure 3.1 (bottom) as a demonstration of how distribution might shift for the replicator dynamic setting. In Section 3.5.3, we will use a detailed replicator dynamics model to further instantiate our results.

3.3 Transferability Bounds for General Distribution Shift

In this section, we first provide upper and lower bounds for *any* induced domain without specifying the particular type of distribution shift. In particular, we first provide upper bounds for the transfer error of any classifier h (that is, the difference between $\mathrm{Err}_{\mathscr{D}(h)}(h)$ and $\mathrm{Err}_{\mathscr{D}_S}(h)$), as well as between $\mathrm{Err}_{\mathscr{D}(h)}(h)$ and the minimum induced risk $\mathrm{Err}_{\mathscr{D}(h_T^*)}(h_T^*)$. We then provide lower bounds for $\max\{\mathrm{Err}_{\mathscr{D}_S}(h), \mathrm{Err}_{\mathscr{D}(h)}(h)\}$, that is, the minimum error a model h must incur on either the source distribution $\mathscr{D}_S$ or the induced distribution $\mathscr{D}(h)$.

3.3.1 Upper Bound For General Distribution Shift

We first investigate the upper bounds for the transfer errors. We begin by showing generic bounds and further instantiate the bound for specific domain adaptation settings in Section 3.4 and 3.5. We begin by answering the following question:

How does a model h trained on its training data set fare on the induced distribution $\mathscr{D}(h)$?

To that end, we define the minimum and h-dependent combined error of any two distributions $\mathscr{D}$ and $\mathscr{D}'$ as:

$$\lambda_{\mathscr{D}\to\mathscr{D}'} := \min_{h'\in\mathscr{H}} \mathrm{Err}_{\mathscr{D}'}(h') + \mathrm{Err}_{\mathscr{D}}(h'), \qquad \Lambda_{\mathscr{D}\to\mathscr{D}'}(h) := \max_{h'\in\mathscr{H}} \mathrm{Err}_{\mathscr{D}'}(h) + \mathrm{Err}_{\mathscr{D}}(h)$$

and their corresponding $\mathscr{H}$-divergence as

$$d_{\mathscr{H}\times\mathscr{H}}(\mathscr{D},\mathscr{D}') = 2 \sup_{h,h'\in\mathscr{H}} \left|\mathbb{P}_{\mathscr{D}}(h(X)\neq h'(X)) - \mathbb{P}_{\mathscr{D}'}(h(X)\neq h'(X))\right|.$$

The $\mathscr{H}$-divergence is a celebrated measure proposed in the domain adaptation literature [Ben-David et al., 2010a] which will be useful for bounding the difference in errors of any two classifiers. Following the classical arguments from Ben-David et al. [2010a], we can easily prove the following:

Theorem 3.3.1 (**Source risk ⇒ Induced risk**). *The difference between $Err_{\mathscr{D}(h)}(h)$ and $Err_{\mathscr{D}_S}(h)$ is upper bounded by: $Err_{\mathscr{D}(h)}(h) \leq Err_{\mathscr{D}_S}(h)+\lambda_{\mathscr{D}_S \to \mathscr{D}(h)}+\frac{1}{2}d_{\mathscr{H}\times\mathscr{H}}(\mathscr{D}_S, \mathscr{D}(h))$.*

Proof. We first establish two lemmas that will help bound the errors of a pair of classifiers. Both are standard results from the domain adaption literature Ben-David et al. [2010a].

Lemma 3.3.2. *For any hypotheses $h, h' \in \mathscr{H}$ and distributions $\mathscr{D}, \mathscr{D}'$,*

$$|Err_{\mathscr{D}}(h,h') - Err_{\mathscr{D}'}(h,h')| \leq \frac{d_{\mathscr{H}\times\mathscr{H}}(\mathscr{D},\mathscr{D}')}{2}.$$

Proof. Define the-cross prediction disagreement between two classifiers h, h' on a distribution $\mathscr{D}$ as $\mathrm{Err}_{\mathscr{D}}(h,h') := \mathbb{P}_{\mathscr{D}}(h(X) \neq h'(X))$. By the definition of the $\mathscr{H}-$divergence,

$$\begin{aligned} d_{\mathscr{H}\times\mathscr{H}}(\mathscr{D},\mathscr{D}') &= 2 \sup_{h,h'\in\mathscr{H}} \left|\mathbb{P}_{\mathscr{D}}(h(X) \neq h'(X)) - \mathbb{P}_{\mathscr{D}'}(h(X) \neq h'(X))\right| \\ &= 2 \sup_{h,h'\in\mathscr{H}} \left|\mathrm{Err}_{\mathscr{D}}(h,h') - \mathrm{Err}_{\mathscr{D}'}(h,h')\right| \\ &\geq 2 \left|\mathrm{Err}_{\mathscr{D}}(h,h') - \mathrm{Err}_{\mathscr{D}'}(h,h')\right|. \end{aligned}$$

□

Another helpful lemma for us is the well-known fact that the 0-1 error obeys the triangle inequality (see, e.g., Crammer et al. [2008]):

Lemma 3.3.3. *For any distribution $\mathscr{D}$ over instances and any labeling functions f_1, f_2, and f_3, we have $Err_{\mathscr{D}}(f_1, f_2) \leq Err_{\mathscr{D}}(f_1, f_3) + Err_{\mathscr{D}}(f_2, f_3)$.*

Denote by $\bar{h}^*$ the *ideal joint hypothesis*, which minimizes the combined error:

$$\bar{h}^* := \arg\min_{h' \in \mathscr{H}} \mathrm{Err}_{\mathscr{D}(h)}(h') + \mathrm{Err}_{\mathscr{D}_S}(h')$$

We have:

$$\begin{aligned}
\mathrm{Err}_{\mathscr{D}(h)}(h) &\leq \mathrm{Err}_{\mathscr{D}(h)}(\bar{h}^*) + \mathrm{Err}_{\mathscr{D}(h)}(h, \bar{h}^*) && \text{(Lemma 3.3.3)} \\
&\leq \mathrm{Err}_{\mathscr{D}(h)}(\bar{h}^*) + \mathrm{Err}_{\mathscr{D}_S}(h, \bar{h}^*) + \left|\mathrm{Err}_{\mathscr{D}(h)}(h, \bar{h}^*) - \mathrm{Err}_{\mathscr{D}_S}(h, \bar{h}^*)\right| \\
&\leq \mathrm{Err}_{\mathscr{D}(h)}(\bar{h}^*) + \mathrm{Err}_{\mathscr{D}_S}(h) + \mathrm{Err}_{\mathscr{D}_S}(\bar{h}^*) + \frac{1}{2} d_{\mathscr{H} \times \mathscr{H}}(\mathscr{D}_S, \mathscr{D}(h)) \\
& && \text{(Lemma 3.3.2)} \\
&= \mathrm{Err}_{\mathscr{D}_S}(h) + \lambda_{\mathscr{D}_S \to \mathscr{D}(h)} + \frac{1}{2} d_{\mathscr{H} \times \mathscr{H}}(\mathscr{D}_S, \mathscr{D}(h)). && \text{(Definition of } \bar{h}^*\text{)}
\end{aligned}$$

□

The transferability of a model h between $\mathrm{Err}_{\mathscr{D}(h)}(h)$ and $\mathrm{Err}_{\mathscr{D}_S}(h)$ looks precisely the same as in the classical domain adaptation setting [Ben-David et al., 2010a].

An arguably more interesting quantity in our setting to understand is the difference between the induced error of any given model h and the error induced by the optimal model h_T^*: $\mathrm{Err}_{\mathscr{D}(h)}(h) - \mathrm{Err}_{\mathscr{D}(h_T^*)}(h_T^*)$. We get the following bound, which differs from the one in Theorem 3.3.1:

Theorem 3.3.4 (Induced risk ⇒ Minimum induced risk). *The difference between*

$Err_{\mathscr{D}(h)}(h)$ and $Err_{\mathscr{D}(h_T^*)}(h_T^*)$ is upper bounded by:

$$Err_{\mathscr{D}(h)}(h) - Err_{\mathscr{D}(h_T^*)}(h_T^*) \leq \frac{\lambda_{\mathscr{D}(h) \to \mathscr{D}(h_T^*)} + \Lambda_{\mathscr{D}(h) \to \mathscr{D}(h_T^*)}(h)}{2} + \frac{1}{2} \cdot d_{\mathscr{H} \times \mathscr{H}}(\mathscr{D}(h_T^*), \mathscr{D}(h)).$$

The above theorem informs us that the induced transfer error is generally bounded by the "average" achievable error on both distributions $\mathscr{D}(h)$ and $\mathscr{D}(h_T^*)$, as well as the $\mathscr{H}$ divergence between the two distributions.

The major benefit of the results in Theorem 3.3.4 is that it provides the decision maker a way to estimate the minimum induced risk $\mathrm{Err}_{\mathscr{D}(h^*_T)}(h^*_T)$ even when she only has access to the induced risk of some available classifier h, as long as she can characterize the statistical difference between the two induced distribution. The latter, however, might not seem to be a trivial task itself. Later in Section 3.3.3, we briefly discuss how our bounds can still be useful even when we do not have the exact characterizations of this quantity.

3.3.2 Lower Bound For General Distribution Shift

Now we provide a lower bound on the induced transfer error. We particularly want to show that at least one of the two errors $\mathrm{Err}_{\mathscr{D}_S}(h)$, and $\mathrm{Err}_{\mathscr{D}(h)}(h)$, must be lower-bounded by a certain quantity.

Theorem 3.3.5 (**Lower bound for learning tradeoffs**). *Any model h must incur the following error on either the source or induced distribution:*

$$\max\{Err_{\mathscr{D}_S}(h), Err_{\mathscr{D}(h)}(h)\} \geq \frac{d_{TV}(\mathscr{D}_{Y|S}, \mathscr{D}_Y(h)) - d_{TV}(\mathscr{D}_{h|S}, \mathscr{D}_h(h))}{2}.$$

The proof leverages the triangle inequality of d_{TV}. This bound is dependent on h; however, by the data processing inequality of d_{TV} (and f-divergence functions in general) [Liese and Vajda, 2006], we have $d_{\mathrm{TV}}(\mathscr{D}_{h|S}, \mathscr{D}_h(h)) \leq d_{\mathrm{TV}}(\mathscr{D}_{X|S}, \mathscr{D}_X(h))$. Applying this to Theorem 3.3.5 yields:

Corollary 3.3.6. *For any model h,*

$$\max\{Err_{\mathscr{D}_S}(h), Err_{\mathscr{D}(h)}(h)\} \geq \frac{d_{TV}(\mathscr{D}_{Y|S}, \mathscr{D}_Y(h)) - d_{TV}(\mathscr{D}_{X|S}, \mathscr{D}_X(h))}{2}.$$

The benefit of Corollary 3.3.6 is that the bound does not contain any quantities that are functions of the induced distribution; as a result, for any classifier h, we can estimate the learning tradeoffs between its source risk and its induced risk using values that are computable without actually deploying the classifier at the first place.

Tightness of the General Bounds in Section 3.3 For the general bounds reported in Section 3.3, it is not trivial to fully quantify the tightness without further quantifying the specific quantities of the terms, e.g., the H-divergence of the source and the induced distribution and the average error a classifier have to incur for both distribution. This part of our results adapted from the classical literature in learning from multiple domains Ben-David et al. [2010a]. The tightness of using $\mathscr{H}$-divergence and other terms seem to be partially validated therein.

3.3.3 How to Use Our Bounds?

The upper and lower bounds we derived in the previous sections (Theorem 3.3.4 and Theorem 3.3.5) depend on the following two quantities either explicitly or implicitly: (1) the distribution $\mathscr{D}(h)$ induced by the deployment of the model h in question, and (2) the optimal target classifier h_T^* as well as the distribution $\mathscr{D}(h_T^*)$ it induces. The bounds may therefore seem to be of only theoretical interest since in reality we generally cannot compute $\mathscr{D}(h)$ without actual deployment, let alone compute h_T^*. Thus, in general, it is unclear how to compute the value of these bounds.

Nevertheless, our bounds can still be useful and informative in the following ways:

General modeling framework with flexible hypothetical shifting models The bounds can be evaluated if the decision maker has a particular shift model in mind, which specifies how the population would adapt to a model. A common special case is when the decision maker posits an individual-level agent response model (e.g., the strategic agent [Hardt et al., 2016a] - we demonstrate how to evaluate in Section 3.4.3). In these cases, the $\mathscr{H}$-divergence can be consistently estimated from finite samples of the population [Wang et al., 2005], allowing the decision maker to estimate the performance gap of a given h without deploying it. The general bounds provided can thus be viewed as a framework by which specialized, computationally tractable bounds can be derived.

Estimate the optimal target classifier h_T^* from a set of imperfect models Secondly, when the decision maker has access to a set of imperfect models $\tilde{h}_1, \tilde{h}_2 \cdots \tilde{h}_t \in H^T$ that will predict a range of possible shifted distribution $\mathscr{D}(\tilde{h}_1), \cdots \mathscr{D}(\tilde{h}_t) \in \mathscr{D}^T$ and a range of possibly optimal target distribution $h_T \in \mathscr{H}^T$, the bounds on h_T^* can be further instantiated by calculating the worst case in this predicted set :[3]

$$\mathrm{Err}_{\mathscr{D}(h)}(h) - \mathrm{Err}_{\mathscr{D}(h_T^*)}(h_T^*) \lesssim \max_{\mathscr{D}' \in \mathscr{D}^T, h' \in \mathscr{H}^T} \mathrm{UpperBound}(\mathscr{D}', h'),$$

$$\max\{\mathrm{Err}_{\mathscr{D}_S}(h), \mathrm{Err}_{\mathscr{D}(h_T^*)}(h_T^*)\} \gtrsim \min_{\mathscr{D}' \in \mathscr{D}^T, h' \in \mathscr{H}^T} \mathrm{LowerBound}(\mathscr{D}', h').$$

3.4 Transferability Bounds for Covariate Shift

In this section, we focus on a particular distribution shift model known as *covariate shift*, in which the distribution of features changes, but the distribution

[3] UpperBound and LowerBound are the RHS expressions in Theorem 3.3.4 and Theorem 3.3.5.

of labels conditioned on features remains the same:

$$\mathbb{P}_{\mathscr{D}(h)}(Y=y|X=x)=\mathbb{P}_{\mathscr{D}_S}(Y=y|X=x) \tag{3.3}$$

$$\mathbb{P}_{\mathscr{D}(h)}(X=x)\neq\mathbb{P}_{\mathscr{D}_S}(X=x) \tag{3.4}$$

Thus with covariate shift, we have

$$\begin{aligned}\mathbb{P}_{\mathscr{D}(h)}(X=x,Y=y)=&\mathbb{P}_{\mathscr{D}(h)}(Y=y|X=x)\cdot\mathbb{P}_{\mathscr{D}(h)}(X=x)\\=&\mathbb{P}_{\mathscr{D}_S}(Y=y|X=x)\cdot\mathbb{P}_{\mathscr{D}(h)}(X=x)\end{aligned}$$

Let $\omega_x(h) := \frac{\mathbb{P}_{\mathscr{D}(h)}(X=x)}{\mathbb{P}_{\mathscr{D}_S}(X=x)}$ be the *importance weight* at x, which characterizes the amount of adaptation induced by h at instance x. Then for any loss function ℓ we have:

Proposition 3.4.1 (Expected Loss on $\mathscr{D}(h)$ Under Covariate Shift)**.**

$$\mathbb{E}_{\mathscr{D}(h)}[\ell(h;X,Y)]=\mathbb{E}_{\mathscr{D}_S}[\omega_x(h)\cdot\ell(h;x,y)].$$

Proof.

$$\begin{aligned}&\mathbb{E}_{\mathscr{D}(h)}[\ell(h;X,Y)]\\=&\int\mathbb{P}_{\mathscr{D}(h)}(X=x,Y=y)\ell(h;x,y)\ dxdy\\=&\int\mathbb{P}_{\mathscr{D}_S}(Y=y|X=x)\cdot\mathbb{P}_{\mathscr{D}(h)}(X=x)\ell(h;x,y)\ dxdy\\=&\int\mathbb{P}_{\mathscr{D}_S}(Y=y|X=x)\cdot\mathbb{P}_{\mathscr{D}_S}(X=x)\cdot\frac{\mathbb{P}_{\mathscr{D}(h)}(X=x)}{\mathbb{P}_{\mathscr{D}_S}(X=x)}\cdot\ell(h;x,y)\ dxdy\\=&\int\mathbb{P}_{\mathscr{D}_S}(Y=y|X=x)\cdot\mathbb{P}_{\mathscr{D}_S}(X=x)\cdot\omega_x(h)\cdot\ell(h;x,y)\ dxdy\\=&\mathbb{E}_{\mathscr{D}_S}[\omega_x(h)\cdot\ell(h;x,y)]\end{aligned}$$

□

The above derivation is a classic trick and offers the basis for performing importance reweighting when learning under covariate shift [Sugiyama et al., 2008].

The particular form informs us that $\omega_x(h)$ controls the generation of $\mathscr{D}(h)$ and encodes its dependency on both $\mathscr{D}_S$ and h, and is critical for deriving our results below.

3.4.1 Upper Bound for Covariate Shift

We now derive an upper bound for transferability under covariate shift. We will particularly focus on the optimal model trained on the source data $\mathscr{D}_S$, which we denote as $h_S^* := \arg\min_{h\in\mathscr{H}} \mathrm{Err}_S(h)$. Recall that the classifier with minimum induced risk is denoted as $h_T^* := \arg\min_{h\in\mathscr{H}} \mathrm{Err}_{\mathscr{D}(h)}(h)$. We can upper bound the difference between h_S^* and h_T^* as follows:

Theorem 3.4.2 (Suboptimality of h_S^*)**.** *Let X be distributed according to $\mathscr{D}_S$. We have:*

$$Err_{\mathscr{D}(h_S^*)}(h_S^*) - Err_{\mathscr{D}(h_T^*)}(h_T^*) \leq \sqrt{Err_{\mathscr{D}_S}(h_T^*)} \cdot \left(\sqrt{Var(\omega_X(h_S^*))} + \sqrt{Var(\omega_X(h_T^*))}\right).$$

This result can be interpreted as follows: h_T^* incurs an irreducible amount of error on the source data set, represented by $\sqrt{\mathrm{Err}_{\mathscr{D}_S}(h_T^*)}$. Moreover, the difference in induced risks between h_S^* and h_T^* is at its maximum when the two classifiers induce adaptations in “opposite” directions; this is represented by the sum of the standard deviations of their importance weights, $\sqrt{\mathrm{Var}(\omega_X(h_S^*))} + \sqrt{\mathrm{Var}(\omega_X(h_T^*))}$.

3.4.2 Lower Bound For Covariate Shift

Recall that in Theorem 3.3.5, for the general setting, it is unclear whether the lower bound is strictly positive or not. In this section, we provide further understanding for when the lower bound $\frac{d_{\mathrm{TV}}(\mathscr{D}_{Y|S}, \mathscr{D}_Y(h)) - d_{\mathrm{TV}}(\mathscr{D}_{h|S}, \mathscr{D}_h(h))}{2}$ is indeed positive under covariate shift. Under several assumptions, our previously provided lower bound in Theorem 3.3.5 is strictly positive with covariate shift.

Assumption 3.4.3. $|\mathbb{E}_{X\in X_+(h),Y=+1}[1-\omega_X(h)]| \geq |\mathbb{E}_{X\in X_-(h),Y=+1}[1-\omega_X(h)]|$., *where* $X_+(h)=\{x:\omega_x(h)\geq 1\}$ *and* $X_-(h)=\{x:\omega_x(h)<1\}$.

This assumption states that increased $\omega_x(h)$ value points are more likely to have positive labels.

Assumption 3.4.4. $|\mathbb{E}_{X\in X_+(h),h(X)=+1}[1-\omega_X(h)]| \geq |\mathbb{E}_{X\in X_-(h),h(X)=+1}[1-\omega_X(h)]|$.

This assumption states that increased $\omega_x(h)$ value points are more likely to be classified as positive.

Assumption 3.4.5. $Cov\big(\mathbb{P}_{\mathscr{D}_S}(Y=+1|X=x)-\mathbb{P}_{\mathscr{D}_S}(h(x)=+1|X=x),\omega_x(h)\big)>0$.

This assumption is stating that for a classifier h, within all $h(X)=+1$ or $h(X)=-1$, a higher $\mathbb{P}_{\mathscr{D}}(Y=+1|X=x)$ associates with a higher $\omega_x(h)$.

Theorem 3.4.6. *Under Assumption 3.4.3 - Assumption 3.4.5, the following lower bound is strictly positive under covariate shift:*

$$\max\{Err_{\mathscr{D}_S}(h), Err_{\mathscr{D}(h)}(h)\} \geq \frac{d_{TV}(\mathscr{D}_{Y|S},\mathscr{D}_Y(h))-d_{TV}(\mathscr{D}_{h|S},\mathscr{D}_h(h))}{2}>0.$$

3.4.3 Covariate Shift via Strategic Response

As introduced in Section 3.2.1, we consider a setting caused by *strategic response* in which agents are classified by and adapt to a binary threshold classifier. In particular, each agent is associated with a d dimensional continuous feature $x\in\mathbb{R}^d$ and a binary true qualification $y(x)\in\{-1,+1\}$, where $y(x)$ is a function of the feature vector x. Consistent with the literature in strategic classification [Hardt et al., 2016a], a simple case where after seeing the threshold binary decision rule

$h(x) = 2 \cdot \mathbb{1}[x \geq \tau_h] - 1$, the agents will *best response* to it by maximizing the following utility function:

$$u(x, x') = h(x') - h(x) - c(x, x'),$$

where $c(x, x')$ is the *cost function* for decision subjects to modify their feature from x to x'. We assume all agents are rational utility maximizers: they will only *attempt* to change their features when the benefit of manipulation is greater than the cost (i.e. when $c(x, x') \leq 2$) and the agent will not change their feature if they are already accepted (i.e. $h(x) = +1$). For a given threshold τ_h and manipulation budget B, the theoretical best response of an agent with original feature x is:

$$\Delta(x) = \arg\max_{x'} u(x, x') \quad s.t. \ c(x, x') \leq B.$$

To make the problem tractable and meaningful, we further specify the following setups:

Setup 1. *(Initial Feature) Agents' initial features are uniformly distributed between* $[0, 1] \in \mathbb{R}^1$.

Setup 2. *(Agent's Cost Function) The cost of changing from x to x' is proportional to the distance between them:* $c(x, x') = \|x - x'\|$.

Setup 2 implies that only agents whose features are in between $[\tau_h - B, \tau_h)$ will *attempt* to change their feature. We also assume that feature updates are *probabilistic*, such that agents with features closer to the decision boundary τ_h have a greater *chance* of updating their feature and each updated feature x' is sampled from a uniform distribution depending on τ_h, B, and x (see Setup 3 & 4):

Setup 3. *(Agent's Success Manipulation Probability) For agents who* attempt *to* *update their features, the probability of a successful feature update is* $\mathbb{P}(X' \neq X) = 1 - \frac{|x-\tau_h|}{B}$.

Intuitively this setup means that the closer the agent's original feature x is to the decision boundary τ_h, the more likely they can successfully change their feature to cross the decision boundary.

Setup 4 (Adapted Feature's Distribution)**.** *An agent's updated feature* x'*, given original* x*, manipulation budget* B*, and classification boundary* τ_h*, is sampled as* $X' \sim$ *Unif*$(\tau_h, \tau_h + |B - x|)$.

Setup 4 aims to capture the fact that even though agent targets to change their feature to the decision boundary τ_h (i.e. the least cost action to get a favorable prediction outcome), they might end up reaching a feature that is beyond the decision boundary.

With the above setups, we can specify the bound in Theorem 3.4.2 for the strategic response setting as follows:

Proposition 3.4.7. *For the setting of strategic response described above, Theorem 3.4.2 implies* $Err_{\mathscr{D}(h_S^*)}(h_S^*) - Err_{\mathscr{D}(h_T^*)}(h_T^*) \leq \sqrt{\frac{2B}{3} Err_{\mathscr{D}_S}(h_T^*)}$.

We can see that the upper bound for strategic response depends on the manipulation budget B, and the error the ideal classifier made on the source distribution $\mathrm{Err}_{D_S}(h_T^*)$. This aligns with our intuition that the smaller the manipulation budget is, the fewer agents will change their features, thus leading to a tighter upper bound on the difference between $\mathrm{Err}_{h_S^*}(h_S^*)$ and $\mathrm{Err}_{h_T^*}(h_T^*)$. This expression also allows us to provide bounds even without the knowledge of the mapping between $\mathscr{D}(h)$ and h, since we can directly compute $\mathrm{Err}_{\mathscr{D}_S}(h_T^*)$ from the source distribution and an estimated optimal classifier h_T^*.

3.5 Transferability Bounds for Target Shift

We consider another popular domain adaptation setting known as *target shift*, in which the distribution of labels changes, but the distribution of features conditioned on the label remains the same:

$$\mathbb{P}_{\mathscr{D}(h)}(X = x|Y = y) = \mathbb{P}_{\mathscr{D}_S}(X = x|Y = y) \tag{3.5}$$

$$\mathbb{P}_{\mathscr{D}(h)}(Y = y) \neq \mathbb{P}_{\mathscr{D}_S}(Y = y) \tag{3.6}$$

For binary classification, let $p(h) := \mathbb{P}_{\mathscr{D}(h)}(Y = +1)$, and $\mathbb{P}_{\mathscr{D}(h)}(Y = -1) = 1 - p(h)$. Notice that $p(h)$ encodes the full adaptation information from $\mathscr{D}_S$ to $\mathscr{D}(h)$, since the mapping between Y and X, $\mathbb{P}(X = x|Y = y)$, is known and remains unchanged during target shift. We have for any proper loss function ℓ:

$$\begin{aligned}&\mathbb{E}_{\mathscr{D}(h)}[\ell(h; X, Y)]\\ =&p(h) \cdot \mathbb{E}_{\mathscr{D}(h)}[\ell(h; X, Y)|Y = +1] + (1 - p(h)) \cdot \mathbb{E}_{\mathscr{D}(h)}[\ell(h; X, Y)|Y = -1]\\ =&p(h) \cdot \mathbb{E}_{\mathscr{D}_S}[\ell(h; X, Y)|Y = +1] + (1 - p(h)) \cdot \mathbb{E}_{\mathscr{D}_S}[\ell(h; X, Y)|Y = -1]\end{aligned}$$

We will adopt the following shorthands: $\text{Err}_+(h) := \mathbb{E}_{\mathscr{D}_S}[\ell(h; X, Y)|Y = +1]$ $\text{Err}_-(h) := \mathbb{E}_{\mathscr{D}_S}[\ell(h; X, Y)|Y = -1]$. Note that $\text{Err}_+(h), \text{Err}_-(h)$ are both defined on the conditional source distribution, which is invariant under the target shift assumption.

3.5.1 Upper Bound for Target Shift

We first provide characterizations of the upper bound on the transferability of h^*_S under target shift. Denote by $\mathscr{D}_+$ the positive label distribution on $\mathscr{D}_S$ ($\mathbb{P}_{\mathscr{D}_S}(X = x|Y = +1)$) and $\mathscr{D}_-$ the negative label distribution on $\mathscr{D}_S$ ($\mathbb{P}_{\mathscr{D}_S}(X = x|Y = -1)$). Let $\underline{p := \mathbb{P}_{\mathscr{D}_S}(Y = +1)}$.

Theorem 3.5.1. *For target shift, the difference between* $Err_{\mathscr{D}(h_S^*)}(h_S^*)$ *and* $Err_{\mathscr{D}(h_T^*)}(h_T^*)$ *bounds as:*

$$\begin{aligned}&Err_{\mathscr{D}(h_S^*)}(h_S^*) - Err_{\mathscr{D}(h_T^*)}(h_T^*)\\ \leq&|\omega(h_S^*) - \omega(h_T^*)| + (1+p)\cdot(d_{TV}(\mathscr{D}_+(h_S^*), \mathscr{D}_+(h_T^*)) + d_{TV}(\mathscr{D}_-(h_S^*), \mathscr{D}_-(h_T^*))\,.\end{aligned}$$

The above bound consists of two components. The first quantity captures the difference between the two induced distributions $\mathscr{D}(h_S^*)$ and $\mathscr{D}(h_T^*)$. The second quantity characterizes the difference between the two classifiers h_S^*, h_T^* on the source distribution.

3.5.2 Lower Bound For Target Shift

Now we discuss lower bounds. Denote by $\mathrm{TPR}_S(h)$ and $\mathrm{FPR}_S(h)$ the true positive and false positive rates of h on the source distribution $\mathscr{D}_S$. We prove the following:

Theorem 3.5.2. *For target shift, any model* h *must incur the following error on either* $\mathscr{D}_S$ *or* $\mathscr{D}(h)$*:*

$$\max\{Err_{\mathscr{D}_S}(h), Err_{\mathscr{D}(h)}(h)\} \geq \frac{|p - p(h)|\cdot(1 - |TPR_S(h) - FPR_S(h)|)}{2}.$$

Proof of Theorem 3.5.2. We will make use of the following fact:

Lemma 3.5.3. *Under label shift,* $TPR_S(h) = TPR_h(h)$ *and* $FPR_S(h) = FPR_h(h)$.

In section 3.3.2 we showed a general lower bound on the maximum of $\mathrm{Err}_{\mathscr{D}_S}(h)$ and $\mathrm{Err}_{\mathscr{D}(h)}(h)$:

$$\max\{\mathrm{Err}_{\mathscr{D}_S}(h), \mathrm{Err}_{\mathscr{D}(h)}(h)\} \geq \frac{d_{\mathrm{TV}}(\mathscr{D}_{Y|S}, \mathscr{D}_Y(h)) - d_{\mathrm{TV}}(\mathscr{D}_{h|S}, \mathscr{D}_h(h))}{2}$$

In the case of label shift, and by the definitions of p and $p(h)$,

$$d_{\mathrm{TV}}(\mathscr{D}_{Y|S}, \mathscr{D}_Y(h)) = |\mathbb{P}_{\mathscr{D}_S}(Y = +1) - \mathbb{P}_{\mathscr{D}(h)}(Y = +1)| = |p - p(h)| \tag{3.7}$$

In addition, we have

$$\mathscr{D}_{h|S} = \mathbb{P}_S(h(X) = +1) = p \cdot \mathrm{TPR}_S(h) + (1 - p) \cdot \mathrm{FPR}_S(h) \tag{3.8}$$

Similarly

$$\begin{aligned}\mathscr{D}_h(h) &= \mathbb{P}_{\mathscr{D}(h)}(h(X) = +1)\\ &= p(h) \cdot \mathrm{TPR}_h(h) + (1 - p(h)) \cdot \mathrm{FPR}_h(h)\\ &= p(h) \cdot \mathrm{TPR}_S(h) + (1 - p(h)) \cdot \mathrm{FPR}_S(h) \qquad \text{(by Lemma 3.5.3)} \end{aligned} \tag{3.9}$$

Therefore

$$\begin{aligned}d_{\mathrm{TV}}(\mathscr{D}_{h|S}, \mathscr{D}_h(h)) =& |\mathbb{P}_{\mathscr{D}_S}(h(X) = +1) - \mathbb{P}_{\mathscr{D}(h)}(h(X) = +1)|\\ =& |(p - p(h)) \cdot \mathrm{TPR}_S(h) + (p(h) - p) \cdot \mathrm{FPR}_S(h)|\\ & \text{(By equation 3.9 and equation 3.8)}\\ =& |p - p(h)| \cdot |\mathrm{TPR}_S(h) - \mathrm{FPR}_S(h)| \end{aligned} \tag{3.10}$$

which yields:

$$d_{\mathrm{TV}}(\mathscr{D}_{Y|S}, \mathscr{D}_Y(h)) - d_{\mathrm{TV}}(\mathscr{D}_{h|S}, \mathscr{D}_h(h)) = |p - p(h)|(1 - |\mathrm{TPR}_S(h) - \mathrm{FPR}_S(h)|)$$

(By equation 3.7 and equation 3.10)

completing the proof. □

Taking a closer look, the lower bound is determined linearly by how much the label distribution shifts: $p - p(h)$. The difference is further determined by the performance of h on the source distribution through $1 - |\mathrm{TPR}_S(h) - \mathrm{FPR}_S(h)|$. For instance, when $\mathrm{TPR}_S(h) > \mathrm{FPR}_S(h)$, the quality becomes $\mathrm{FNR}_S(h) + \mathrm{FPR}_S(h)$, that is the more error h makes, the larger the lower bound will be.

3.5.3 Target Shift via Replicator Dynamics

We now further instantiate our theoretical bound for target shift (Theorem 3.5.1) using a particular replicator dynamics model previously used in Raab and Liu [2021]. In particular, the fitness function is specified as the prediction accuracy of h for class y:

$$\mathbf{Fitness}(Y = y) := \mathbb{P}_{\mathscr{D}_S}(h(X) = y|Y = y) \tag{3.11}$$

Then we have $\mathbb{E}\left[\mathbf{Fitness}(Y)\right] = 1 - \mathrm{Err}_{\mathscr{D}_S}(h)$, and $\frac{p(h)}{\mathbb{P}_{\mathscr{D}_S}(Y=+1)} = \frac{\mathrm{Pr}_{\mathscr{D}_S}(h(X)=+1|Y=+1)}{1-\mathrm{Err}_{\mathscr{D}_S}(h)}$. Plugging the result back into Theorem 3.5.1 we get the following bound for the above replicator dynamic setting:

Proposition 3.5.4. *Under the replicator dynamics model described in Equation* (3.11)*,* $|\omega(h^*_S) - \omega(h^*_T)|$ *bounds as:*

$$\begin{aligned}&|\omega(h^*_S) - \omega(h^*_T)|\\ \le&\mathbb{P}_{\mathscr{D}_S}(Y = +1) \cdot \frac{|Err_{\mathscr{D}_S}(h^*_S) - Err_{\mathscr{D}_S}(h^*_T)| \cdot |TPR_S(h^*_S) - TPR_S(h^*_T)|}{Err_{\mathscr{D}_S}(h^*_S) \cdot Err_{\mathscr{D}_S}(h^*_T)}.\end{aligned}$$

Proof.

$$\begin{aligned}&|p(h^*_S) - p(h^*_T)| \cdot \frac{1}{\mathbb{P}_{\mathscr{D}_S}(Y = +1)}\\ =&\frac{|(1 - \mathrm{Err}_{\mathscr{D}_S}(h^*_S))\mathrm{TPR}_S(h^*_S) - (1 - \mathrm{Err}_{\mathscr{D}_S}(h^*_T))\mathrm{TPR}_S(h^*_T)|}{(1 - \mathrm{Err}_{\mathscr{D}_S}(h^*_S)) \cdot (1 - \mathrm{Err}_{\mathscr{D}_S}(h^*_T))}\\ \le&\frac{|\mathrm{Err}_{\mathscr{D}_S}(h^*_S) - \mathrm{Err}_{\mathscr{D}_S}(h^*_T)| \cdot |\mathrm{TPR}_S(h^*_S) - \mathrm{TPR}_S(h^*_T)|}{(1 - \mathrm{Err}_{\mathscr{D}_S}(h^*_S)) \cdot (1 - \mathrm{Err}_{\mathscr{D}_S}(h^*_T))}\end{aligned} \tag{3.12}$$

The inequality above is due to Lemma 7 of Liu and Liu [2015]. □

The above result shows that the difference between the induced risks $\mathrm{Err}_{\mathscr{D}(h^*_S)}(h^*_S)$ and $\mathrm{Err}_{\mathscr{D}(h^*_T)}(h^*_T)$ only depends on the difference between the two classifiers' performances on the source data $\mathscr{D}_S$. This offers the decision maker a

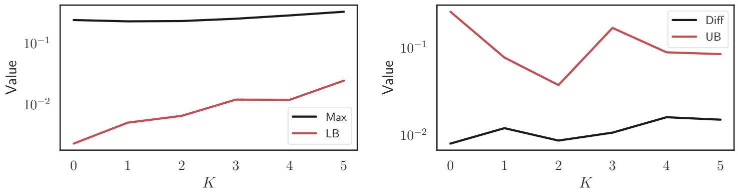


Figure 3.2: Results for synthetic experiments on real-world data. Diff := $\mathrm{Err}_{\mathscr{D}(h_S^*)}(h_S^*) - \mathrm{Err}_{\mathscr{D}(h_T^*)}(h_T^*)$, Max := $\max\{\mathrm{Err}_{\mathscr{D}_S}(h_T^*), \mathrm{Err}_{\mathscr{D}(h_T^*)}(h_T^*)\}$, UB := upper bound specified in Theorem 3.4.2, and LB := lower bound specified in Theorem 3.4.6. For each time step $K = k$, we compute and deploy the source optimal classifier h_S^* and update the credit score for each individual according to the received decision as the new reality for time step $K = k + 1$. Details of the data generation are deferred to Appendix A.7.

great opportunity to evaluate the performance gap by using their corresponding evaluations on the source data only without observing their corresponding induced distributions.

Tightness of the Bounds in Section 3.4 and Section 3.5 It is relatively easier to argue about the tightness of the boundss provided in Section 3.4 (for covariate shift) and Section 3.5 (target shift): the proofs there are more transparent and are easier to back out the conditions where the inequalities are relaxed. For example, in Theorem 5.1, the inequalities of our bound are introduced primarily in the following two places: 1) one is using the optimiality of h_S^* on the source distribution. 2) the other is bounding the statistical difference in h_S^* and h_T^*'s predictions on the positive and negative examples. Both are saying that if the differences in the two classifiers' predictions are bounded in a range, then the result in Theorem 5.1 is relatively tight.

3.6 Experiments

We present synthetic experimental results on both simulated and real-world data sets.

Synthetic experiments using simulated data We generate synthetic data sets from the structural equation models described on simple causal DAG in Figure 3.1 for covariate shift and target shift. To generate the induced distribution $\mathscr{D}(h)$, we posit a specific *adaptation function* $\Delta : \mathbb{R}^d \times \mathscr{H} \to \mathbb{R}^d$, so that when an input x encounters classifier $h \in \mathscr{H}$, its induced features are precisely $x' = \Delta(x, h)$. We provide details of the data generation processes and adaptation functions in Appendix A.7.

We take our training data set $\{x_1, \ldots, x_n\}$ and learn a "base" logistic regression model $h(x) = \sigma(w \cdot x)$.[4] We then consider the hypothesis class $\mathscr{H} := \{h_\tau \mid \tau \in [0, 1]\}$, where $h_\tau(x) := 2 \cdot \mathbb{1}[\sigma(w \cdot x) > \tau] - 1$. To compute h^*_S, the model that performs best on the source distribution, we simply vary τ and take the h_τ with the lowest prediction error. Then, we posit a specific adaptation function $\Delta(x, h_\tau)$. Finally, to compute h^*_T, we vary τ from 0 to 1 and find the classifier h_τ that minimizes the prediction error on its induced data set $\{\Delta(x_1, h_\tau), \ldots, \Delta(x_n, h_\tau)\}$. We report our results in Figure 3.3.

For all four datasets, we do observe positive gaps $\mathrm{Err}_{D(h^*_S)}(h^*_S) - \mathrm{Err}_{D(h^*_T)}(h^*_T)$, indicating the suboptimality of training on $\mathscr{D}_S$. The gaps are well bounded by the theoretical results. For the lower bound, the empirical observation and the theoretical bounds are roughly within the same magnitude except for one target shift dataset, indicating the effectiveness of our theoretical result. Regarding the

[4] $\sigma(\cdot)$ is the logistic function and $w \in \mathbb{R}^3$ denotes the weights.

upper bound, for target shift, the empirical observations are well within the same magnitude of the theoretical bounds while the results for the covariate shift are relatively loose.

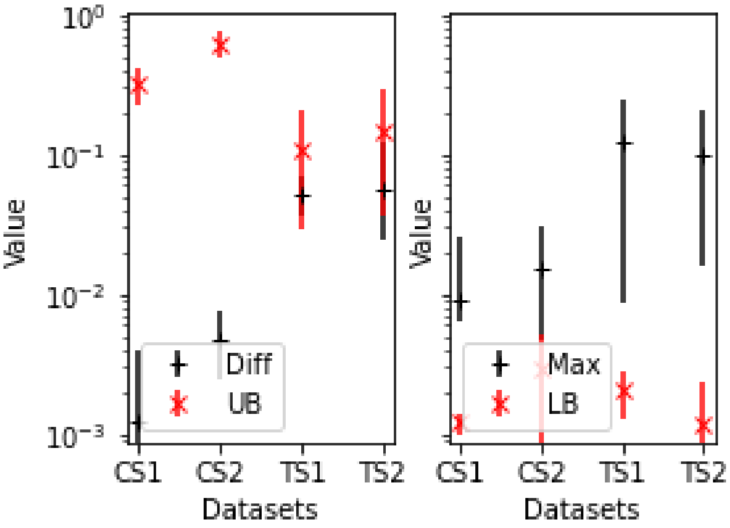


Figure 3.3: Results for synthetic experiments on simulated and real-world data. Diff := $\mathrm{Err}_{\mathscr{D}(h_S^*)}(h_S^*) - \mathrm{Err}_{\mathscr{D}(h_T^*)}(h_T^*)$, Max := $\max\{\mathrm{Err}_{\mathscr{D}_S}(h_T^*), \mathrm{Err}_{\mathscr{D}(h_T^*)}(h_T^*)\}$, UB := upper bound specified in Theorem 3.4.2, and LB := lower bound specified in Theorem 3.4.6.

Synthetic experiments using real-world data We also perform synthetic experiments using real-world data to demonstrate our bounds. In particular, we use the FICO credit score data set [Board of Governors of the Federal Reserve System (US), 2007] which contains more than 300k records of TransUnion credit scores of clients from different demographic groups. For our experiment on the preprocessed FICO data set [Hardt et al., 2016b], we convert the cumulative distribution function (CDF) of TransRisk score among different groups into group-wise credit score densities, from which we generate a balanced sample to represent a population where groups have equal representations. We demonstrate the application of our results in a series of resource allocations. Similar to the synthetic experiments on simulated data, we consider the hypothesis class of threshold classifiers and treat the classification outcome as the decision received by individuals.

For each time step $K = k$, we compute h_S^*, the statistical optimal classifier on the source distribution (i.e., the current reality for step $K = k$), and update the credit score for each individual according to the received decision as the new reality for time step $K = k + 1$. Details of the data generation are again deferred to Appendix A.7. We report our results in Figure 3.2. We do observe positive gaps $\text{Err}_{\mathscr{D}(h_S^*)}(h_S^*) - \text{Err}_{\mathscr{D}(h_T^*)}(h_T^*)$, indicating the suboptimality of training on $\mathscr{D}_S$. The gaps are well bounded by the theoretical upper bound (UB). Our lower bounds (LB) do return meaningful positive gaps, demonstrating the trade-offs that a classifier has to suffer on either the source distribution or the induced target distribution. We also provide additional experimental results using synthetic datasets generated according to causal graphs defined in Figure 3.1.

3.7 Fair Transferability

In this section, we briefly mention that we can also provide a general framework for quantifying the robustness of statistical group fairness guarantees [Chen et al., 2022].

Our primary result is a bound on a policy's potential "violation of statistical group fairness" defined in terms of the differences in policy outcomes between groups when applied to a target distribution shifted relative to the source distribution within known constraints. Such settings naturally arise whenever training data represents a random sample of a target population with different statistics or

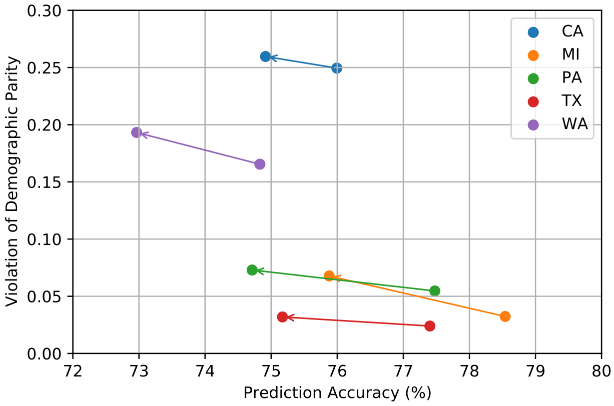


Figure 3.4: In Chen et al. [2022], we evaluate our bounds against historical, temporal distribution shifts in demographics and income recorded by the US Census Bureau [Ding et al., 2021]. The above figure depicts changes to income-prediction accuracy and demographic parity violation when a classifier initially trained on US state-specific demographic data for 2014 is reused on 2018 data, thus exemplifying the negative potential effects of distribution shift.

a sample from dynamic environments, when a policy is reused on a new distribution without retraining, or whenever policy deployment itself induces a distribution shift. As an example of this last case, strategic individuals seeking loans might change their features or abstain from future application (thus shifting the distribution of examples) in response to policies trained on historical data [Hardt et al., 2016a, Ustun et al., 2019, Zhang et al., 2020b]. Beyond policy selection, exogenous pressure such as economic trends and noise may also drive a distribution shift in this example. In Figure 3.4, we show how a real-world distribution shift in demographic and income data for US states between 2014 and 2018 may increase fairness violations while decreasing accuracy for a hypothetical classifier trained on the 2014 distribution. In such settings, it is useful to quantify how fairness guarantees transfer across distributions shifted within some bound, thus allowing the deployment of unfair machine learning policies to be avoided.

Our primary contribution is formulating a general, worst-case upper bound for a given policy's violation of statistical group fairness subject to group-dependent distribution shifts within presupposed bounds. The statistical group fairness notion we use is defined in Equation (2.2).

Overall, bounding violations of fairness subject to distribution shift allow us to recognize and avoid potentially inappropriate deployments of machine learning when the potential disparities of a prospective policy eclipse a given threshold within bounded distribution shifts of the training distribution.

3.8 Conclusions

Unawareness of the potential distribution shift might lead to unintended consequences when training a machine learning model. One goal of our paper is to raise awareness of this issue for the safe deployment of machine learning methods in high-stake scenarios. We also provide a general framework for characterizing the performance difference for a fixed-trained classifier when the decision subjects respond to it.

Chapter 4

Learning to Incentivize Improvements from Strategic Agents

Machine learning systems are often used in settings where individuals adapt their features to obtain a desired outcome. In such settings, strategic behavior leads to a sharp loss in model performance in deployment. In this chapter, we aim to address this problem by learning classifiers that encourage decision subjects to change their features in a way that leads to improvement in both predicted *and* true outcome. We frame the dynamics of prediction and adaptation as a two-stage game, and characterize optimal strategies for the model designer and its decision subjects. In benchmarks on simulated and real-world datasets, we find that classifiers trained using our method maintain the accuracy of existing approaches while inducing higher levels of improvement and less manipulation.

4.1 Incentivize Improvement From Strategic Agent

Individuals subject to a classifier's predictions may act strategically to influence their predictions. Such behavior, often referred to as *strategic manipulation*

[Hardt et al., 2016a], may lead to a sharp deterioration in classification performance. However, not all strategic behavior is detrimental: in many applications, model designers stand to benefit from strategic adaptation if they deploy a classifier that incentivizes decision subjects to perform adaptations that improve their true outcome [Haghtalab et al., 2020, Shavit et al., 2020]. For example:

- **Lending**: In lending, a classifier predicts a loan applicant's ability to repay their loan. If the classifier is designed so as to incentivize the applicants to improve their income, it will also improve the likelihood of repayment.
- **Content Moderation**: In online shopping, a recommender system suggests products to customers based on their relevance. Ideally, the algorithm should incentivize the product sellers to publish accurate product descriptions by aligning this with improved recommendation rankings.
- **Course design**: an instructor designs schoolwork to incentivize students to invest their efforts on studying rather than cheating on an exam [Kleinberg and Raghavan, 2020].
- **Car insurance determination**: an auto insurer tries to predict drivers' expected accident costs, and by designing a determination criterion, encourages safe driving behavior [Haghtalab et al., 2020, Shavit et al., 2020].

In this chapter, we study the following mechanism design problem: a *model designer* needs to train a classifier that will make predictions over *decision subjects* who will alter their features to obtain a specific prediction. Our goal is to learn a classifier that is accurate and that incentivizes decision subjects to adapt their features in a way that improves both their predicted *and* true outcomes.

4.1.1 Our Contributions

Our main contributions are as follows:

1. We introduce a new approach to handle strategic adaptation in machine learning, based on a new concept we call the *constructive adaptation risk*, which trains classifiers that incentivize decision subjects to adapt their features in ways that improve true outcomes. Under the assumption of a feature taxonomy that distinguishes improvable features (features that, if changed, lead to changes in the true qualification) from non-causal features (which do not lead to changes in the true qualification), we provide formal evidence that this risk captures both the strategic and constructive dimensions of decision subjects' behavior.
2. We characterize the dynamics of strategic decision subjects and the model designer in a classification setting using a two-player sequential game. We begin by generalizing cost functions used in previous works on strategic classification to the *Mahalanobis* distance, which provides a way to capture correlations between changes in different features. Under this generalization, we derive closed-form expressions for the decision subjects' optimal strategies (Theorem 4.4.2). These expressions (Section 4.4.3) reveal insights about decision subjects' behavior when the model designer uses non-causal features (features that do not affect the true outcome) as predictors.
3. We formulate the problem of training such a desired classifier as a risk minimization problem. We evaluate our method on simulated and real-world datasets to demonstrate how it can be used to incentivize improvement or discourage adversarial manipulation. Our empirical results show that our

method outperforms existing approaches, even when some feature types are misspecified. In addition, we provide a potential way to extend our main result into a non-linear setting using LIME [Ribeiro et al., 2016].

The details for reproducing our experimental results can be found at `https://github.com/UCSC-REAL/ConstructiveAdaptation`.

4.1.2 How Does Our Work Relate to the Surrounding Literatures?

Our paper builds on the strategic classification literature in machine learning. Departing from previous work, which aims to suppress *all* adaptations, we consider a setting in which strategic adaptation can consist of manipulation as well as improvement. Our broader goal of designing a classifier that encourages improvement is characteristic of recent work in this area [see e.g., Kleinberg and Raghavan, 2020, Haghtalab et al., 2020, Shavit et al., 2020, Rosenfeld et al., 2020]. Specifically, Haghtalab et al. [2020] study how to design an evaluation mechanism that incentivizes individuals to improve a desired quality. However, the success of their method requires explicit assumptions on the linear mapping of features to true qualifications, as well as a projection matrix P that maps the observed features back to the full features. Their setting also does not account for correlations between different features. Another recent work by Shavit et al. [2020] also focuses on finding a decision rule that maximizes decision subjects' true qualifications. Their setting is similar to ours, but they focus on how decision makers can perform causal interventions through the deployment of different decision rules, rather than designing a classifier relying only on observational data. Moreover, they assume that decision subjects take actions in some *action space* that maps linearly to features in *feature space*; this also does not capture correlations between features. Our approach may be useful for mitigating the disparate effects of strategic adaptation [Hu et al., 2019, Milli et al., 2019, Liu et al., 2020] that stem from differences in the cost of manipulation (see Proposition 4.4.6). Our results may also be helpful for developing robust classifiers in dynamic environments, where both decision subjects' features and the deployed models may vary across time periods [Kilbertus et al., 2020, Shavit et al., 2020, Liu and Chen, 2017a]. In contrast to the previous work on causal recourse [Karimi et al., 2020c], we explicitly separate improvable features from manipulated features when maximizing decision subjects' payoffs. Our work also broadly relates to the concept of *intervention* in the literature of causal inference [Eberhardt and Scheines, 2007]. In our work, the actionability of a feature is always factual, meaning it is always feasible to change those features. This is closely related to the concept of *last trial* in causal inference, which refers to the interventions that one could run in the real-world (which would rule out the interventions on age) [Hernán et al., 2022].

4.2 Problem Statement

In this section, we describe our approach to training a classifier that incentivizes improving actions. Similar to Definition 2.2.1, we formalize the dynamics between the model designer and strategic agent as a sequential game between the following two players: the model designer, and the decision subjects [1]. The objectives for the two players are as follows:

1. A model designer, who trains a classifier $h : \mathscr{X} \to \{-1, +1\}$ from a hypothesis class $\mathscr{H}$,which is also their action space.

[1] Throughout the paper, we will also use strategic agents, or agents, interchangeably.

2. Decision subjects, who adapt their features from x to x' so as to be assigned $h(x') = +1$ if possible. We assume that decision subjects incur a cost for altering their features, which we represent using a *cost function* $c : \mathscr{X} \times \mathscr{X} \to \mathbb{R}^+$. The action space for the decision subjects includes all feature vectors that are within a given manipulation budget B, namely $\forall x' \in \mathbb{R}^d$ such that $c(x, x') \leq B$.

We intentionally do not provide the formal definitions of the utilities for the two players here due to the need to provide a clear and accessible introduction to our framework. We will provide more detailed discussions on the agent and decision maker's utility function in Section 4.4.2 and Section 4.3, respectively.

We assume that decision subjects know the model designer's classifier, and the model designer knows the decision subjects' cost function. Decision subjects alter their features based on their current features x, the cost function c, and the classifier h, so that their altered features can be written $x_* = \Delta(x; h, c)$ where $\Delta(\cdot)$ is the *best response function.* The model designer only observes the altered feature x_* but not the original and private one x the decision subject holds. In other words, we consider the standard setting in strategic classification where the model designer has no strong verification power to verify truthfulness of x_*.

We allow adaptations that alter the true qualification y. In practice, the relationship between features and true qualification is unknown, and in fact, it is known that distinguishing causal features (features that affect the true outcome) from non-causal features reduces to solving a non-trivial causal inference problem [Miller et al., 2020]. Addressing this aspect is not the aim of the present work; instead, we will assume that changes in certain features are known to affect the qualification – for example, in loan application, such features can be the agent's education level and salary, and changing those features will affect qualification is the agent's ability to pay for the loan.

Remark 4.2.1. *When an agent adapts its feature vector from x to $\Delta(x)$, its qualification becomes $\mathbb{y}(\Delta(x))$, which may differ from $y(x)$. We consider a setting in which during the training process, the decision maker cannot observe how decision subjects' true qualifications change after they alter their features. We introduced the shorthand notation y to refer to $\mathbb{y}(x)$, the qualification for the* original *feature vector, for the sake of simplicity. For the rest of our paper, a label y always denotes the true qualification* before *adaptation.*

4.3 CA Risk: Minimizing Error While Encouraging Constructive Adaptation

In many applications, model designers are better off when decision subjects adapt their features in a way that yields a specific true outcome, such as $y = +1$. Consider a typical lending application where a model is used to predict whether a customer will repay a loan. In this case, a model designer benefits from $y = +1$, as this means that a borrower will repay their loan.

4.3.1 Ideal objective function for the decision maker

Ideally, the decision maker should aim to classify the agents correctly using their adapted features with respect to the corresponding new qualification. Mathematically, this corresponds to training a classifier h^* that minimizes the following quantity:

$$h^* \in \arg\min_{h \in \mathscr{H}} \mathbb{E}_{(x,y)\sim \mathscr{D}}[\mathbb{1}(h(\Delta(x)) \neq y(\Delta(x))] \tag{4.1}$$

where $\Delta(x)$ is the agent's adapted feature, and $y(\Delta(x))$ is the true qualification after the adaptation. However, since the true mapping function $\mathrm{y} : \mathscr{X} \rightarrow \mathscr{Y}$ is unknown, and the decision maker cannot observe how decision subjects' true qualifications change after they alter their features, we need to propose an alternative approach to achieve similar goals of this ideal objective function, which we call CA risk minimization.

4.3.2 Our Proposed CA risk

To help explain our proposed approach, we assume that we can write $x = [x_{\mathsf{I}} \mid x_{\mathsf{M}} \mid x_{\mathsf{IM}}]$ where x_{I}, x_{M} and x_{IM} denote the following categories of features:

- *Immutable* features (x_{IM}), which cannot be altered (e.g. race, age).
- *Improvable* features (x_{I}), which can be altered in a way that will either increase or decrease the true outcome $y(x)$ (e.g. increasing education level might help improve the probability of repayment).
- *Manipulable* features (x_{M}), which can be altered *without* changing the true outcome $y(x)$ (e.g. social media presence, which can be used as a proxy for influence). Notice that it is the *change* in these features that is undesirable; the features themselves may still be useful for prediction.

Incomplete taxonomy of features There may also be features that can be altered but whose effect is *unknown.* In this work, we treat them as manipulable features. We would like to point out that in practice, implementing our proposed solution does not require the decision-maker to know exactly how to characterize every single feature. In fact, our method can be applied to settings where the decision-makers only know some features are improvable and focus on incentivizing adaptations on them, while treating changes on the rest of the features as undesirable. In this case, using our training method is still strictly better than performing no intervention (i.e. simply letting decision subjects perform their unconstrained best response).

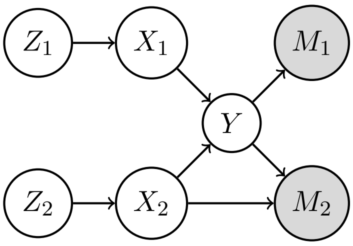


Figure 4.1: A causal DAG for the `toy` data. Z_1 and Z_2 are improvable features that determine the true qualification Y, $X_1 = Z_1$, and X_2 is a noisy proxy for Z_2. In our context, all we require is the knowledge that X_1, X_2 are the factors that causally affect Y, rather than complete knowledge of the DAG. We can directly observe X_1 and X_2 but not Z_1 or Z_2. In addition, M_1 and M_2 are manipulated features that correlate with Y.

Please see Figure 4.1 for a demonstration of the differences between improvable and manipulable features. We also use $x_{\mathsf{A}} = [x_{\mathsf{I}} \mid x_{\mathsf{M}}]$ to denote the *actionable* features, and d_{A} to denote its dimension. Note that the question of how to decide which features are of which type is beyond the scope of the present work; however, this is the topic of intense study in the causal inference literature [Miller et al., 2020]. Analogously, we define the following variants of the best response function Δ:

- $x_*^{\mathsf{I}} = \Delta_{\mathsf{I}}(x, h; c)$: the *improving best response*, which involves an adaptation that only alters improvable features.
- $x_*^{\mathsf{M}} = \Delta_{\mathsf{M}}(x, h; c)$: the *manipulating best response*, which involves an adapta-

tion that only alters manipulable features.

Note that in reality, a decision subject can still alter both types of features, which means that they will perform $\Delta(x, h; c)$, unless the model designer explicitly forbids changing certain features. However, it still worth distinguishing different types of best responses when the model designer designs the classifier: we can think of the improving best response Δ_{I} as the best possible adaptation which only consists of honest improvement, while the manipulating best response Δ_{M} is the worst possible adaptation that consists of pure manipulation. The model designer would like to design a classifier such that for the decision subjects, $\Delta(x, h; c)$ appears to be close to $\Delta_{\mathsf{I}}(x, h; c)$. We therefore propose to train a classifier that minimizes the *constructive adaptation* (CA) risk R_{CA}, which balances robustness to manipulation and incentivization of improvement:

$$h^*_{\mathsf{CA}} \in \underset{h \in \mathscr{H}}{\arg\min}\, R_{\mathsf{CA}}(h) := R_{\mathsf{M}}(h) + \lambda \cdot R_{\mathsf{I}}(h) \tag{4.2}$$

The first term, $R_{\mathsf{M}}(h) = \mathbb{E}_{(x,y)\sim\mathscr{D}}[\mathbb{1}(h(x^{\mathsf{M}}_*) \neq y)]$, is the *manipulation risk*, which penalizes pure manipulation. The second term, $R_{\mathsf{I}}(h) = \mathbb{E}_{(x,y)\sim\mathscr{D}}[\mathbb{1}(h(x^{\mathsf{I}}_*) \neq +1)]$, is the *improvement risk*, which rewards decision subjects for playing their improving best response. The parameter $\lambda > 0$ trades off between these competing objectives. Setting $\lambda \to 0$ results in an objective that simply discourages manipulation, whereas increasing $\lambda \to \infty$ yields a trivial classifier that always predicts $+1$.

A natural question to ask is: how good the proposed objective function Equation (4.2) is compared to the ideal objective function in Equation (4.1)? We show that the two terms in the objective function can be viewed as proxies for the ideal objective function. In particular, in Section 4.5, we show that under reasonable conditions, the following hold:

- The first term, $R_{\mathsf{M}}(h)$, is an upper bound on $R_{\mathsf{SC}}(h)$. Thus minimizing the manipulation risk also minimizes the traditional strategic risk (Proposition 4.5.1).
- A decrease in the second term, $R_{\mathsf{I}}(h)$ reflects an increase in $\Pr(y(x^{\mathsf{I}}_*) = +1)$. Thus improvement in the prediction outcome aligns with improvement in the true qualification (Proposition 4.5.2).

4.4 Decision Subjects' Best Response

We now characterize the decision subjects' best response.

4.4.1 Setup

We restrict our analysis to the setting in which a model designer trains a *linear classifier* $h(x) = \text{sign}(w^{\mathsf{T}}x)$, where $w = [w_0, w_1, \ldots, w_d] \in \mathbb{R}^{d+1}$ denotes a vector of $d+1$ weights. We capture the cost of altering x to x' through the *Mahalanobis* norm of the changes:[2]

$$c(x, x') = \sqrt{(x_{\mathsf{A}} - x_{\mathsf{A}}')^{\mathsf{T}} S^{-1} (x_{\mathsf{A}} - x_{\mathsf{A}}')}$$

Here, $S^{-1} \in \mathbb{R}^{d_{\mathsf{A}}} \times \mathbb{R}^{d_{\mathsf{A}}}$ is a symmetric *cost covariance matrix* in which $S^{-1}_{j,k}$ represents the cost of altering features j and k simultaneously. To ensure that $c(\cdot)$ is a valid norm, we require S^{-1} to be *positive definite*, meaning $x_{\mathsf{A}}{}^{\mathsf{T}} S^{-1} x_{\mathsf{A}} > 0$ for all $x_{\mathsf{A}} \neq \mathbf{0} \in \mathbb{R}^{d_{\mathsf{A}}}$. Additionally, we assume S^{-1} is a block matrix of the form

$$S^{-1} = \begin{bmatrix} (S^{-1})_{\mathsf{I}} & (S^{-1})_{\mathsf{IM}} \\ (S^{-1})_{\mathsf{MI}} & (S^{-1})_{\mathsf{M}} \end{bmatrix}, \text{or} \quad S = \begin{bmatrix} S_{\mathsf{I}} & S_{\mathsf{IM}} \\ S_{\mathsf{MI}} & S_{\mathsf{M}} \end{bmatrix} \tag{4.3}$$

[2] Since immutable features x_{IM} cannot be altered, the cost function involves only the actionable features x_{A}.

Notice that the I-th block of matrix S^{-1} (i.e. $(S^{-1})_{\mathsf{I}}$) does not necessarily equal to its inverse's I-th block component (i.e. ${S_{\mathsf{I}}}^{-1}$).

We allow the cost matrix to contain non-zero elements on non-diagonal entries. This means that our results hold even when there are interaction effects when altering multiple features. This generalizes prior work on strategic classification in which the cost is based on the ℓ_2 norm of the changes, which is tantamount to setting $S^{-1} = I$, and therefore assumes the change in each feature contributes independently to the overall cost [see e.g., Hardt et al., 2016a, Haghtalab et al., 2020].

4.4.2 Decision Subject's Best Response Model

Given the assumptions of Section 4.4.1, we can define and analyze the decision subjects' best response. We start by defining the decision subject's payoff function. Given a classifier h, a decision subject who alters their features from x to x' derives total utility

$$U(x, x') = h(x') - c(x, x')$$

Naturally, a decision subject tries to maximize their utility; that is, they play their *best response*:

Definition 4.4.1 (F-Best Response Function)**.** *Let* $\mathsf{F} \in \{\mathsf{I}, \mathsf{M}, \mathsf{A}\}$*, and let* $\mathscr{X}^*_{\mathsf{F}}(x)$ *denote the set of vectors that differ from* x *only in features of type* F*. Let* $\Delta_{\mathsf{F}} : \mathscr{X} \to \mathscr{X}$ *denote the* F-best response *of a decision subject with features* x *to* h*, defined as:*

$$\Delta_{\mathsf{F}}(x) = \operatorname*{arg\,max}_{x' \in \mathscr{X}^*_{\mathsf{F}}(x)} U(x, x')$$

Setting $\mathsf{F} = \mathsf{I}$ gives the *improving best response* $\Delta_{\mathsf{I}}(x)$, in which the adaptation changes only the improvable features; setting $\mathsf{F} = \mathsf{M}$ yields the *manipulating best response* $\Delta_{\mathsf{M}}(x)$, in which only manipulable features are changed. Setting $\mathsf{F} = \mathsf{A}$, we get the standard *unconstrained best response* $\Delta_{\mathsf{A}}(x)$ in which any actionable features can be changed. As we mentioned earlier, we will also use $x_*^{\mathsf{F}} := \Delta_{\mathsf{F}}(x)$ as shorthand for the F-best response, and we denote $\Delta(x) := \Delta_{\mathsf{A}}(x)$.

Intuitively, the cost of manipulation should be smaller than the cost of actual improvement. For example, improving one's coding skills should take more effort, and thus be more costly, than simply memorizing answers to coding problems. As a result, one would expect the gaming best response $\Delta_{\mathsf{M}}(x)$ and the unconstrained best response $\Delta(x)$ to flip a negative decision more easily than the improving best response $\Delta_{\mathsf{I}}(x)$. In Section 4.4.3, we formalize this notion (Proposition 4.4.4).

For ease of notation, let $\widehat{S}_{\mathsf{F}} := ((S^{-1})_{\mathsf{F}})^{-1}$.We prove the following theorem characterizing the decision subject's different best responses:

Theorem 4.4.2 (F-Best Response in Closed-Form)**.** *Given a linear threshold function $h(x) = \operatorname{sign}(w^{\mathsf{T}}x)$ and a decision subject with features x such that $h(x) = -1$, reorder the features so that $x = [x_{\mathsf{F}} \mid x_{\mathsf{A}\setminus\mathsf{F}} \mid x_{\mathsf{IM}}]$, and let $\Omega_{\mathsf{F}} = {w_{\mathsf{F}}}^{\mathsf{T}}\widehat{S}_{\mathsf{F}}w_{\mathsf{F}}$. Then x has F-best response*

$$\Delta_{\mathsf{F}}(x) = \begin{cases} \left[x_{\mathsf{F}} - \frac{w^{\mathsf{T}}x}{\Omega_{\mathsf{F}}}\widehat{S}_{\mathsf{F}}w_{\mathsf{F}}\right] \mid x_{\mathsf{A}\setminus\mathsf{F}} \mid x_{\mathsf{IM}}, & \textit{if } \frac{|w^{\mathsf{T}}x|}{\sqrt{\Omega_{\mathsf{F}}}} \leq 2 \\ x, & \textit{otherwise} \end{cases} \tag{4.4}$$

with corresponding cost

$$c(x, \Delta_{\mathsf{F}}(x)) = \begin{cases} \frac{|w^{\mathsf{T}}x|}{\sqrt{\Omega_{\mathsf{F}}}}, & \textit{if } \frac{|w^{\mathsf{T}}x|}{\sqrt{\Omega_{\mathsf{F}}}} \leq 2 \\ 0 & \textit{otherwise} \end{cases}.$$

All proofs in this section are included in Appendix B.1.

Example: When $\mathsf{F} = \mathsf{M}$, $x_{\mathsf{F}} = x_{\mathsf{M}}$ and $x_{\mathsf{A}\setminus\mathsf{F}} = [x_{\mathsf{I}}]$. After reordering features, we get the following closed-form expression for the manipulating best response:

$$\Delta_{\mathsf{M}}(x) = \begin{cases} \left[x_{\mathsf{I}} \mid x_{\mathsf{M}} - \frac{w^\intercal x}{\Omega_{\mathsf{M}}}\widehat{S}_{\mathsf{M}} w_{\mathsf{M}} \mid x_{\mathsf{IM}}\right] & \text{if } \frac{|w^\intercal x|}{\sqrt{\Omega_{\mathsf{M}}}} \leq 2 \\ x, & \text{otherwise} \end{cases}$$

with corresponding cost

$$c(x, \Delta_{\mathsf{M}}(x)) = \begin{cases} \frac{|w^\intercal x|}{\sqrt{\Omega_{\mathsf{M}}}}, & \text{if } \frac{|w^\intercal x|}{\sqrt{\Omega_{\mathsf{M}}}} \leq 2 \\ 0 & \text{otherwise} \end{cases}.$$

4.4.3 Discussion

We now discuss the implications of different decision subject's responses derived in Theorem 4.4.2. In this section, we consider a slightly more structured cost matrix that is diagonal blocked matrix (in which case, $S^{-1}_{\mathsf{IM}} = S^{-1}_{\mathsf{MI}} = \mathbf{0}$), which corresponds to a setting where there are no correlations between the *cost* of changing manipulated feature versus the cost of changing improvable features.

Notation For this section, we make use of the following additional notation:

- $v^{(i)}$ denotes the i-th element of a vector v
- For any $\mathsf{F} \in \{\mathsf{A}, \mathsf{I}, \mathsf{M}\}$, $\Delta^{\mathsf{F}} \in \mathbb{R}^{d_{\mathsf{F}}}$ denotes the vector containing only features of type F within the best response $\Delta(x)$.
- $\mathbf{0}$ denotes the vector whose elements are all 0
- $A \succ B$ indicates that matrix $A - B$ is positive definite
- e_i denotes the vector containing 1 in its i-th component and 0 elsewhere

Firstly, we demonstrate a basic limitation for the model designer: if the classifier uses any manipulable features as predictors, then decision subjects will find a way to exploit them. Hence the only way to avoid any possibility of manipulation is to train a classifier without such features.

Proposition 4.4.3 (Preventing Manipulation is Hard)**.** *Suppose there exists a manipulated feature $x^{(m)}$ whose weight in the classifier $w_{\mathsf{A}}^{(m)}$ is nonzero. Then for almost every $x \in \mathscr{X}$, $\Delta^{(m)}(x) \neq x^{(m)}$.* [3]

Proof. Let $w_{\mathsf{M}}^{(m)} \neq 0$, and consider an decision subject with original features x who was classified as -1. By Theorem 4.4.2, the actionable sub-vector of x's unconstrained best response is

$$\Delta^{\mathsf{A}}(x) = \frac{w^{\mathsf{T}}x}{w_{\mathsf{A}}{}^{\mathsf{T}}Sw_{\mathsf{A}}} S \cdot w_{\mathsf{A}} = \frac{w^{\mathsf{T}}x}{w_{\mathsf{A}}{}^{\mathsf{T}}Sw_{\mathsf{A}}} \begin{bmatrix} S_{\mathsf{I}} & 0 \\ \\ 0 & S_{\mathsf{M}} \end{bmatrix} \begin{bmatrix} w_{\mathsf{I}} \\ \\ w_{\mathsf{M}} \end{bmatrix} = \frac{w^{\mathsf{T}}x}{w_{\mathsf{A}}{}^{\mathsf{T}}Sw_{\mathsf{A}}} \begin{bmatrix} S_{\mathsf{I}} \cdot w_{\mathsf{I}} \\ \\ S_{\mathsf{M}} \cdot w_{\mathsf{M}} \end{bmatrix}$$

And in particular,

$$\Delta^{\mathsf{M}}(x) = \frac{w^{\mathsf{T}}x}{w_{\mathsf{A}}{}^{\mathsf{T}}Sw_{\mathsf{A}}} S_{\mathsf{M}} \cdot w_{\mathsf{M}}$$

Since x was initially classified as -1, we have $w^{\mathsf{T}}x < 0$, which means $\frac{w^{\mathsf{T}}x}{w_{\mathsf{A}}Sw_{\mathsf{A}}} \neq 0$. For convenience, let $c = \frac{w^{\mathsf{T}}x}{w_{\mathsf{A}}Sw_{\mathsf{A}}}$. We have

$$\Delta^{\mathsf{M}}(x) - x_{\mathsf{M}} = cS_{\mathsf{M}}w_{\mathsf{M}} - x_{\mathsf{M}} = S_{\mathsf{M}}(cw_{\mathsf{M}} - S_{\mathsf{M}}{}^{-1}x_{\mathsf{M}})$$

Now examine the following:

$$\begin{aligned} (cw_{\mathsf{M}} - S_{\mathsf{M}}{}^{-1}x_{\mathsf{M}})^{(m)} &= cw_{\mathsf{M}}^{(m)} - (S_{\mathsf{M}}^{-1}x_{\mathsf{M}})^{(m)} \\ &= cw_{\mathsf{M}}^{(m)} - \sum_{i=1}^{d_{\mathsf{M}}} (S_{\mathsf{M}}^{-1})^{(im)} x_{\mathsf{M}}{}^{(m)} \end{aligned}$$

Recall that $cw_{\mathsf{M}}^{(m)} \neq 0$. Hence if $\sum_{i=1}^{d_{\mathsf{M}}}(S_{\mathsf{M}}^{-1})^{(im)} = 0$, or if

$$x_{\mathsf{M}}^{(m)} \neq \frac{cw_{\mathsf{M}}^{(m)}}{\sum_{i=1}^{d_{\mathsf{M}}}(S_{\mathsf{M}}^{-1})^{(im)}},$$

[3] In our paper, the subscript (e.g. x_m) refers to the entire feature vector (e.g., $x_m \in R^{d_m}$, where d_m is the total number of the manipulative features), while the superscript (m) refers to the particular index of a particular manipulation feature.

then $(cw_{\mathsf{M}} - {S_{\mathsf{M}}}^{-1} x_{\mathsf{M}})^{(m)} \neq 0$, and therefore $cw_{\mathsf{M}} - S_{\mathsf{M}}^{-1} x_{\mathsf{M}} \neq \mathbf{0}$. Since S_{M} is positive definite, it has full rank, which implies

$$\Delta^{\mathsf{M}}(x) - x_{\mathsf{M}} = S_{\mathsf{M}}(cw_{\mathsf{M}} - S_{\mathsf{M}}^{-1} x_{\mathsf{M}}) \neq \mathbf{0}$$

as required. With this, we have shown that when there exists a manipulated feature $x^{(m)}$ whose corresponding coefficient ${w_{\mathsf{A}}}^{(m)} \neq 0$, the classifier is vulnerable to changes in the manipulated features by the vast majority of decision subjects. □

Next, we show that the unconstrained best response $\Delta(x)$ dominates the improving best response $\Delta_{\mathsf{I}}(x)$, thus highlighting the difficulty of inducing decision subjects to change only their improvable features when they are also allowed to change manipulable features.

Proposition 4.4.4 (Unconstrained Best Response Dominates Improving Best Response)**.** *Suppose there exists a manipulable feature $x^{(m)}$ whose weight in the classifier $w_A^{(m)}$ is nonzero. Then, if a decision subject can flip her decision by playing the improving best response, she can also do so by playing the unconstrained best response. The converse is not true: there exist decision subjects who can flip their predictions through their unconstrained best response but not their improving best response.*

Proof. Consider a decision subject with features x such that $h(x) = -1$. Suppose x can flip this classification result by performing the improving best response $\Delta_{\mathsf{I}}(x)$, which implies that the cost of that action is no greater than 2 for this decision subject. We therefore have:

$$2 \geq c(x, \Delta_{\mathsf{I}}(x)) = \frac{|w^{\mathsf{T}} x|}{\sqrt{{w_{\mathsf{I}}}^{\mathsf{T}} S_{\mathsf{I}} w_{\mathsf{I}}}} > \frac{|w^{\mathsf{T}} x|}{\sqrt{{w_{\mathsf{I}}}^{\mathsf{T}} S_{\mathsf{I}} w_{\mathsf{I}} + {w_{\mathsf{M}}}^{\mathsf{T}} S_{\mathsf{M}} w_{\mathsf{M}}}} = \frac{|w^{\mathsf{T}} x|}{\sqrt{{w_{\mathsf{A}}}^{\mathsf{T}} S w_{\mathsf{A}}}} = c(x, \Delta(x))$$

where the strict inequality is due to the fact that $S_{\mathsf{M}} \succ 0$ and $w_{\mathsf{M}} \neq \mathbf{0}$. As we have shown that $c(x, \Delta(x)) < 2$, we conclude whenever an decision subject can successfully flip her decision by the improving best response, she can also achieve it by performing the unconstrained best response.

On the other hand, consider the case when the unconstrained best response of a decision subject with features x^* has cost exactly 2:

$$\begin{aligned} 2 = c(x^*, \Delta(x^*)) = \frac{|w^{\mathsf{T}} x^*|}{\sqrt{{w_{\mathsf{A}}}^{\mathsf{T}} S w_{\mathsf{A}}}} =& \frac{|w^{\mathsf{T}} x^*|}{\sqrt{{w_{\mathsf{I}}}^{\mathsf{T}} S_{\mathsf{I}} w_{\mathsf{I}} + {w_{\mathsf{M}}}^{\mathsf{T}} S_{\mathsf{M}} w_{\mathsf{M}}}} \\ <& \frac{|w^{\mathsf{T}} x^*|}{\sqrt{{w_{\mathsf{I}}}^{\mathsf{T}} S_{\mathsf{I}} w_{\mathsf{I}}}} = c(x^*, \Delta_{\mathsf{I}}(x^*)) \end{aligned}$$

where the strict inequality is due to the fact that $S_{\mathsf{M}} \succ 0$ and $w_{\mathsf{M}} \neq \mathbf{0}$. As we have shown that $c(x^*, \Delta_{\mathsf{I}}(x^*)) > 2$, we conclude that while the unconstrained best response is viable for this decision subject, the improving best response is not. □

Next, we show how correlations between features affect the cost of adaptation. This can be demonstrated by looking at any cost matrix and adding a small nonzero quantity τ to some i, j-th and j, i-th entries. Such a perturbation can reduce every decision subject's best-response cost:

Proposition 4.4.5 (Correlations between Features May Reduce Cost)**.** *For any cost matrix S^{-1} and any nontrivial classifier h, there exist indices $k, \ell \in [d_{\mathsf{A}}]$ and $\tau \in \mathbb{R}$ such that every feature vector x has lower best-response cost under the cost matrix $\tilde{S}^{-1}$ given by*

$$\tilde{S}^{-1}_{ij} = \tilde{S}^{-1}_{ji} = \begin{cases} S^{-1}_{ij} + \tau, & \textit{if } i = k, j = \ell \\ S^{-1}_{ij}, & \textit{otherwise} \end{cases}$$

than under S^{-1}; that is, $c_{\tilde{S}^{-1}}(x, \Delta(x)) < c_{S^{-1}}(x, \Delta(x))$ for all x.

In many applications, decision subjects may incur different costs for modifying their features, resulting in disparities in prediction outcomes [see Hu et al., 2019, for a discussion]. To formalize this phenomenon, suppose Φ and Ψ are two groups whose costs of changing improvable features are identical, but members of Φ incur higher costs for changing manipulable features. Let $\phi \in \Phi$ and $\psi \in \Psi$ be two people from these groups who share the same profile, i.e. $x_\phi = x_\psi$. We show the following:

Proposition 4.4.6 (Cost Disparities between Subgroups)**.** *Suppose there exists a manipulated feature $x^{(m)}$ whose corresponding weight in the classifier $w_{\mathsf{A}}^{(m)}$ is nonzero. Then if decision subjects are allowed to modify any features, ϕ must pay a higher cost than ψ to flip their classification decision.*

Proof. Let the cost covariance matrices for groups Φ and Ψ be

$$S_\Psi^{-1} = \begin{bmatrix} S_{\mathsf{I}}^{-1} & 0 \\ 0 & S_{\mathsf{M},\Phi}^{-1} \end{bmatrix}, \qquad S_\Phi^{-1} = \begin{bmatrix} S_{\mathsf{I}}^{-1} & 0 \\ 0 & S_{\mathsf{M},\Psi}^{-1} \end{bmatrix}$$

Here, we see that both groups have the same cost of changing improvable features, as represented in the cost submatrix ${S_{\mathsf{I}}}^{-1}$. However, the cost of manipulation for group Φ is higher than that of group Ψ, namely $S_{\mathsf{M},\Phi}^{-1} \succ S_{\mathsf{M},\Psi}^{-1}$.

We are now equipped to compare the costs for the two decision subjects:

$$\begin{aligned} c(x_\phi, \Delta(x_\phi)) &= \frac{|w^\mathsf{T} x_\phi|}{\sqrt{{w_\mathsf{A}}^\mathsf{T} S_\Phi w_\mathsf{A}}} = \frac{|w^\mathsf{T} x|}{\sqrt{{w_\mathsf{I}}^\mathsf{T} S_\mathsf{I} w_\mathsf{I} + {w_\mathsf{M}}^\mathsf{T} \cdot S_{\mathsf{M},\Phi} \cdot w_\mathsf{M}}} \\ c(x_\psi, \Delta(x_\psi)) &= \frac{|w^\mathsf{T} x_\psi|}{\sqrt{{w_\mathsf{A}}^\mathsf{T} S_\Psi w_\mathsf{A}}} = \frac{|w^\mathsf{T} x|}{\sqrt{{w_\mathsf{I}}^\mathsf{T} S_\mathsf{I} w_\mathsf{I} + {w_\mathsf{M}}^\mathsf{T} \cdot S_{\mathsf{M},\Psi} \cdot w_\mathsf{M}}} \end{aligned}$$

Since $S_{\mathsf{M},\Phi}^{-1} \succ S_{\mathsf{M},\Psi}^{-1}$, we have $S_{\mathsf{M},\Phi} \prec S_{\mathsf{M},\Psi}$. And since $w_\mathsf{M} \neq \mathbf{0}$, this implies $0 < {w_\mathsf{M}}^\mathsf{T} S_{\mathsf{M},\Phi} w_\mathsf{M} < {w_\mathsf{M}}^\mathsf{T} \cdot S_{\mathsf{M},\Psi} \cdot w_\mathsf{M}$. As a result, $c(x_\phi, \Delta(x_\phi)) > c(x_\psi, \Delta(x_\psi))$ as required. □

Proposition 4.4.6 highlights the importance for a model designer to account for these differences when serving a population with heterogeneous subgroups. Indeed, when one group achieves more favorable prediction outcomes due to a lower cost of manipulation, our method mitigates the cost disparities between different subgroups by encouraging changes in improvable features and penalizing manipulation.

4.5 Constructive Adaptation Risk Minimization

In this section we analyze the training objective for the model designer, formulating it as an empirical risk minimization (ERM) problem. The omitted details can be found in Appendix B.3.

The model designer's goal is to publish a classifier h that maximizes the classification accuracy while incentivizing individuals to change their improvable features. By Theorem 4.4.2, we have

$$x_*^{\mathsf{M}} = \begin{cases} \left[x_{\mathsf{I}} \mid x_{\mathsf{M}} - \frac{w^\top x}{\Omega_{\mathsf{M}}} \widetilde{S}_{\mathsf{M}} w_{\mathsf{M}} \mid x_{\mathsf{IM}}\right] & \text{if} \frac{|w^\top x|}{\sqrt{\Omega_{\mathsf{M}}}} \leq 2 \\ x, & \text{otherwise} \end{cases} \tag{4.5}$$

$$x_*^{\mathsf{I}} = \begin{cases} \left[x_{\mathsf{I}} - \frac{w^\top x}{\Omega_{\mathsf{I}}} \widetilde{S}_{\mathsf{I}} w_{\mathsf{I}} \mid x_{\mathsf{M}} \mid x_{\mathsf{IM}}\right], & \text{if } \frac{|w^\top x|}{\sqrt{\Omega_{\mathsf{I}}}} \leq 2 \\ x, & \text{otherwise} \end{cases} \tag{4.6}$$

Recall from Section 4.3 that the model designer's optimization program is as follows:

$$\begin{aligned} \min_{h \in \mathscr{H}} \quad & \mathbb{E}_{x \sim \mathscr{D}} \left[\mathbb{1}(h(x_*^{\mathsf{M}}) \neq y)\right] + \lambda \mathbb{E}_{x \sim \mathscr{D}} \left[\mathbb{1}(h(x_*^{\mathsf{I}}) \neq +1)\right] \\ \text{s.t.} \quad & x_*^{\mathsf{M}} \text{ in Equation (4.5)}, \ x_*^{\mathsf{I}} \text{ in Equation (4.6)} \end{aligned} \tag{4.7}$$

Interpreting the Objective The two terms in the objective function can be viewed as proxies for two other familiar objectives. The first term, $\mathbb{E}_{x \sim \mathscr{D}} \left[\mathbb{1}(h(x_*^{\mathsf{M}}) \neq y)\right]$, directly penalizes pure manipulation. But as the following proposition suggests, minimizing this term also minimizes the traditional strategic risk when the true qualification does not change:

Proposition 4.5.1. *Assume that the manipulating best response is more likely to result in a positive prediction than the unconstrained best response, given that the true labels do not change. Then*

$$\mathbb{E}_{x\sim\mathscr{D}}\left[\mathbb{1}[h(x_*) \neq y] \mid \Delta(y) = y\right] \leq \mathbb{E}_{x\sim\mathscr{D}}\left[\mathbb{1}(h(x_*^{\mathsf{M}}) \neq y)\right].$$

Proof. We want to show that the standard strategic risk conditioned on an unchanged true label is upper-bounded by the first term in our model designer's objective, $R_{\mathsf{M}}(h)$:

$$\mathbb{E}_{x\sim\mathscr{D}}\left[\mathbb{1}[h(x_*) \neq y] \mid \Delta(y) = y\right] \leq \mathbb{E}_{x\sim\mathscr{D}}\left[\mathbb{1}(h(x_*^{\mathsf{M}}) \neq y)\right]$$

We assume that the manipulating best response is more likely to result in a positive prediction than the unconstrained best response, given that the true labels do not change:

$$\mathbb{E}_{x\sim\mathscr{D}}\left[\mathbb{1}[h(x_*) \neq y] \mid \Delta(y) = y\right] \leq \mathbb{E}_{\mathscr{D}}\left[\mathbb{1}[h(x_*^{\mathsf{M}}) \neq y] \mid \Delta_{\mathsf{M}}(y) = y\right] \tag{4.8}$$

We therefore have:

$$\begin{aligned}
&\mathbb{E}_{x\sim\mathscr{D}}\left[\mathbb{1}(h(x_*^{\mathsf{M}}) \neq y)\right] \\
=&\mathbb{E}_{x\sim\mathscr{D}}\left[\mathbb{1}(h(x_*^{\mathsf{M}}) \neq y) \mid \Delta_{\mathsf{M}}(y) \neq y\right] \cdot \Pr[\Delta_{\mathsf{M}}(y) \neq y] \\
&\qquad + \mathbb{E}_{x\sim\mathscr{D}}\left[\mathbb{1}(h(x_*^{\mathsf{M}}) \neq y) \mid \Delta_{\mathsf{M}}(y) = y\right] \cdot \Pr[\Delta_{\mathsf{M}}(y) = y] \\
=&\mathbb{E}_{x\sim\mathscr{D}}\left[\mathbb{1}(h(x_*^{\mathsf{M}}) \neq y) \mid \Delta_{\mathsf{M}}(y) = y\right] && (\Pr[\Delta_{\mathsf{M}}(y) = y] = 1) \\
\geq&\mathbb{E}_{x\sim\mathscr{D}}\left[\mathbb{1}(h(x_*) \neq y) \mid \Delta(y) = y\right] && (\text{by equation 4.8})
\end{aligned}$$

□

Intuitively, the assumption within Proposition 4.5.1 may be fulfilled in settings where a population of agents each have the same fixed budget on the cost or effort they are willing to expend, and manipulative or cheating-type actions (for instance, (controlling recent purchase behaviors and borrowing money from family members right before applying for a credit card) confer greater immediate advantages than honest improvement (e.g. spending frugally and accruing savings from personal income over several years).

The second term, $\mathbb{E}_{x \sim \mathscr{D}} \left[\mathbb{1}(h(x^{\mathsf{I}}_*) \neq +1) \right]$, explicitly rewards decision subjects for playing their improving best response (closely related to the notion of *recourse*). Of course, without positing a causal graph, we cannot know whether performing the improving best response leads to a positive change in the true qualification, namely whether $\Delta_{\mathsf{I}}(Y) = +1$; however, when the distribution of X may change but not the conditional label distribution $\Pr(Y|X)$, we can show that an increase in $\Pr(h(X) = +1)$ reflects an increase in $\Pr(Y = +1)$. This gives formal evidence that our prediction outcome aligns with improvement in the true qualification:

Proposition 4.5.2. *Let $\mathscr{D}^*$ be the new distribution after decision subject's best response. Denote $\omega_h(x) = \frac{\Pr_{\mathscr{D}^*}(X=x)}{\Pr_{\mathscr{D}}(X=x)}$ denote the amount of adaptation induced at feature vector x. Suppose $y(X)$ and $h(X)$ are both positively correlated with $\omega_h(X)$, and that the distribution of the true label Y given a particular feature vector X is unchanged is the same before and after adaptation. Then the following are equivalent:*

$$\Pr[h(x^{\mathsf{I}}_*) = +1] > \Pr[h(x) = +1] \Longleftrightarrow \Pr[y(x^{\mathsf{I}}_*) = +1] > \Pr[y(x) = +1].$$

Proof. Let $\mathscr{D}^*$ be the distribution induced by deploying classifier h. By the covariate shift assumption, $\Pr_{\mathscr{D}^*}(Y = y|X = x) = \Pr_{\mathscr{D}}(Y = y|X = x)$. Therefore

$$\begin{aligned}
\Pr_{x\sim\mathscr{D}^*}[y(x) = +1] =& \mathbb{E}_{\mathscr{D}^*}[\mathbb{1}[y(x) = +1]] \\
=& \int \mathbb{1}[y(x) = +1] \Pr_{\mathscr{D}^*}(X = x)dx \\
=& \int \mathbb{1}[y(x) = +1] \frac{\Pr_{\mathscr{D}^*}(X = x)}{\Pr_D(X = x)} \Pr_{\mathscr{D}}(X = x)dx \\
=& \int \mathbb{1}[y(x) = +1]\omega_h(x) \Pr_{\mathscr{D}}(X = x)dx \\
=& \mathbb{E}_{\mathscr{D}}\left[\omega_h(x)\mathbb{1}[y(x) = +1]\right]
\end{aligned}$$

This implies

$$\Pr_{x\sim\mathscr{D}^*}[y(x) = +1] \geq \Pr_{x\sim\mathscr{D}}[y(x) = +1] \iff \mathbb{E}_{\mathscr{D}}\left[(\omega_h(x) - 1)\mathbb{1}[y(x) = +1]\right] \geq 0 \quad (4.9)$$

By similar reasoning, we have

$$\Pr_{x\sim\mathscr{D}^*}[h(x) = +1] = \mathbb{E}_{\mathscr{D}^*}[\mathbb{1}[h(x) = +1]] = \mathbb{E}_{\mathscr{D}}\left[\omega_h(x)\mathbb{1}[h(x) = +1]\right]$$

which implies

$$\Pr_{x\sim\mathscr{D}^*}[h(x) = +1] \geq \Pr_{x\sim\mathscr{D}}[h(x) = +1] \iff \mathbb{E}_{\mathscr{D}}\left[(\omega_h(x) - 1)\mathbb{1}[h(x) = +1]\right] \geq 0 \quad (4.10)$$

It is easy to verify that $\mathbb{E}_{x\sim\mathscr{D}}[\omega_h(x)] = 1$, and this gives us

$$\mathbb{E}_{\mathscr{D}}\left[(\omega_h(x) - 1)\mathbb{1}[y(x) = +1]\right] = \mathrm{Cov}_{\mathscr{D}}(\omega_h(x), \mathbb{1}[y(x) = +1]) \quad (4.11)$$

$$\mathbb{E}_{\mathscr{D}}\left[(\omega_h(x) - 1)\mathbb{1}[h(x) = +1]\right] = \mathrm{Cov}_{\mathscr{D}}(\omega_h(x), \mathbb{1}[h(x) = +1]) \quad (4.12)$$

By equation 4.9, equation 4.10, and equation 4.11, the condition

$$\Pr_{x\sim\mathscr{D}^*}[h(x) = +1] \geq \Pr_{x\sim\mathscr{D}}[h(x) = +1] \iff \Pr_{x\sim\mathscr{D}^*}[y(x) = +1] \geq \Pr_{x\sim\mathscr{D}}[y(x) = +1]$$

is equivalent to the condition

$$\mathrm{Cov}_{\mathscr{D}}(\omega_h(x), \mathbb{1}[y(x) = +1]) \geq 0 \iff \mathrm{Cov}_{\mathscr{D}}(\omega_h(x), \mathbb{1}[h(x) = +1]) \geq 0$$

□

We also provide further derivation for model designer's objective function in Appendix B.3.

Here we provide some motivation for the premise of Proposition 4.5.2. An unchanged $\Pr(Y|X)$ means that the mapping from feature vector X to its corresponding true qualification $Y(X)$ remains the same despite a population-level distribution shift. This is a useful and natural simplification in numerous settings. An example is in credit card applications: suppose X is an applicant's credit score and Y is whether they are truly qualified. For people with the same credit score, we assume they have equal chances of being truly qualified.

Algorithm 1 Best Response for Non-Linear Model

```
Input: Non-Linear classifier $h$, an individual data point $x$
Result: $x_*^{\mathsf{M}}$ and $x_*^{\mathsf{I}}$
Step 1. Call LIME to get the approximated weights $\tilde{w}$ of a local linear classifier
    for non-linear model $h$ around the individual point $x$
Step 2. Substitute $\tilde{w}$ into Equation (4.5) and Equation (4.6) to get $x_*^{\mathsf{M}}$ and $x_*^{\mathsf{I}}$,
    respectively
```

Extension to Non-Linear Models The above approach in Equation (4.7) presumes a linear classifier such that we can derive a close-form solution of the agent's best response. However, the recourse scheme will be typically infeasible with non-linear classifiers. To extend our approach to nonlinear models, we propose to substitute x_*^{M} and x_*^{I} in Equation (4.7) with an approximated best response acquired from a local linear classifier. We note that a prior work LIME [Ribeiro et al., 2016] can provide an approximate linear decision boundary for arbitrary individual points to any non-linear models. The idea is to sample the spherical neighborhood of the data point and fit a local linear model with the target model's certified predictions. As shown in Algorithm 1, we integrate LIME into the oracle that can return us any decision subjects' best response in terms of the approximated local linear classifier. Once we get the best response x_*^{M} and x_*^{I}, we iteratively plug them back to Equation (4.7) as the learning objective of the non-linear classifier. We will demonstrate the effectiveness of this oracle procedure when optimizing a non-linear neural network with gradient descent in Appendix B.4.2. Nonetheless, even with the above extension, all of our theoretical guarantees is not straightforwardly clear to analysis with an oracle of non-linear models' best response, so we let the current paper focus on linear models.

4.6 Experiments

In this section, we present empirical results to benchmark our proposed method on synthetic and real-world datasets. We test the effectiveness of our approach in terms of its ability to incentivize improvement as well as to disincentivize manipulation (see **Evaluation Criteria** for details). We also compare its performance with other standard approaches (see **Methods**).

4.6.1 Setup

Datasets and Cost Matrix We consider five datasets:

1. `toy`, a synthetic dataset based on the causal DAG in Figure 4.1;
2. `credit`, a dataset for predicting whether an individual will default on an upcoming credit payment [Yeh and Lien, 2009];
3. `adult`, a census-based dataset for predicting adult annual incomes;

4. `german`, a dataset to assess credit risk in loans;
5. `spambase`, a dataset for email spam detection. The last three are from the UCI ML Repository [Dua and Graff, 2017b].

We conducted all experiments on a 3 GHz 6-Core Intel Core i5 CPU. All our methods have relatively modest computational costs and can be trained within a few minutes. We provide a detailed description of each dataset along with a partitioning of features in Table B.1 in the Appendix.

We assume the cost of manipulation is lower than that of improvement and refer to the specific cost matrix S below. In particular, we specify the cost matrix S as follows:

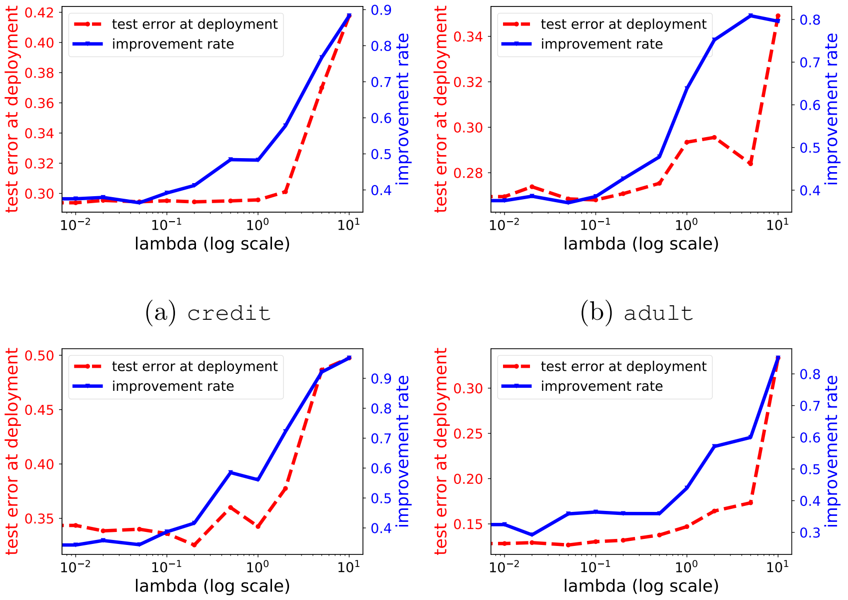
$$S_{ij}^{-1} = \begin{cases} 1, & \text{if } i = j \text{ and } i \in \mathsf{I} \\ 0.2, & \text{if } i = j \text{ and } j \in \mathsf{M} \\ 1, & \text{if the cost of changing features } i \text{ and } j \text{ are \textit{negatively} correlated} \\ -1, & \text{if the cost of changing features } i \text{ and } j \text{ are \textit{positively} correlated} \\ 0, & \text{otherwise} \end{cases}$$

We use the `credit` dataset as a demonstration of how we specify the non-diagonal element in the cost matrix. For two feature variables that have a positive correlation, e.g., *CheckingAccountBalance* and *SavingsAccountBalance*, we assign −1 to the corresponding elements in the cost matrix S. For two feature variables that have a negative correlation, e.g., *CheckingAccountBalance* and *MissedPayments*, we assign +1 to the corresponding elements in the cost matrix. In practice, the cost matrix S should be determined using domain expertise. The purpose of the cost matrix used in these experiments is not to accurately specify costs per se, but to demonstrate the relative difficulty of changing different features.

Methods We fit linear classifiers for each dataset using the following methods: ST, a static classifier trained using ℓ_2-logistic regression without accounting for strategic adaptation; DF, a classifier trained using ℓ_2-logistic regression without any manipulated features; MP, a classifier that considers the agent's unconstrained best response (i.e. with changes in any actionable features x_{A} allowed) during training, as typically done in the strategic classification literature [Hardt et al., 2016a]; CA, a linear logistic regression classifier that results from solving the optimization program in Equation (B.11), which is a smooth differentiable surrogate version of the objective function Equation (4.7). Please refer to Appendix B.3 for a detailed derivation. Using the BFGS algorithm [Byrd et al., 1995]. CA represents our approach.

Evaluation Criteria We run each method with 5-fold cross-validation and report the following:

- *Test Error*: the error of a classifier after training but *before* decision subjects' adaptations, i.e. $\mathbb{E}_{(x,y)\sim\mathscr{D}}\mathbb{1}[h(x) \neq y]$.
- *(Worst-Case) Deployment Error*: the test error of a classifier *after* decision subjects play their manipulating best response, i.e. $\mathbb{E}_{(x,y)\sim\mathscr{D}}\mathbb{1}[h(x_*^{\mathsf{M}}) \neq y]$.
- *(Best-Case) Improvement Rate*: the percent of improvement, defined as the proportion of the population who originally would be rejected but are accepted if they perform constructive adaptation (improving best response), i.e. $\mathbb{E}_{(x,y)\sim\mathscr{D}}\mathbb{1}[h(x_*^{\mathsf{I}}) = +1 \mid y(x) = -1]$.



(a) credit

(b) adult

(c) german

(d) Spambase

Figure 4.2: The trade-off between test error at deployment and improvement rate in the cost matrix. We observe that the test error increases consistently with the improvement rate.

4.6.2 Controlled Experiments on Synthetic Dataset

We perform controlled experiments using a synthetic toy dataset to test the effectiveness of our model at incentivizing improvement in various situations. As shown in Figure 4.1, we set Z_1 and Z_2 as improvable features, X_1 and X_2 as their corresponding noisy proxies, M_1 and M_2 as manipulable features, and Y as the true outcome. Since we have full knowledge of this DAG structure, we can observe the changes in the true outcome after the decision subject's best response. As shown in Table 4.1, Our method achieves the lowest deployment error (20.61%) and the best improvement rate (23.04%) when the model designer has full knowledge of the causal graph.

We also run experiments in which some features are *misspecified*, simulating realistic scenarios in which the model designer may not be able to observe all the

Table 4.1: Performance metrics for different specifications (**Spec.**) in which features may be misspecified. For each method, we report *test error*, *deployment error*, and *improvement rate*. In Full, the model designer has full knowledge of the causal DAG. In Mis. I, M_1 is mistaken for an improvable feature. In Mis. II, the improvable feature X_1 is miscategorized as manipulable.

		Methods			
Spec.	**Metrics**	ST	DF	MP	CA
Full	*test error*	10.29	28.0	11.91	11.62
	deployment error	35.79	35.15	24.1	20.61
	improvement rate	11.54	13.13	14.63	23.49
Mis. I	*test error*	11.39	10.52	11.26	11.04
	deployment error	37.37	10.53	19.79	25.30
	improvement rate	37.23	39.74	0.62	23.04
Mis. II	*test error*	10.58	35.77	29.52	10.80
	deployment error	12.37	41.51	27.68	23.58
	improvement rate	1.12	5.74	3.36	19.82

improvable features [Haghtalab et al., 2020, Shavit et al., 2020], or mistakes one type of feature for another. We model these situations by changing M_1 into an improvable feature and X_1 into a manipulable feature; the results, shown in Table 4.1, show that our classifier maintains a relatively high improvement rate in these cases, without sacrificing much deployment accuracy.

Table 4.2: Performance metrics for all methods over 4 real data sets with non-diagonal cost matrix. We report the mean and standard deviation for 5-fold cross validation. The constructive adaptation (CA) consistently achieves a high accuracy at deployment while providing the highest improvement rates across all four datasets.

		Methods			
Dataset	**Metrics**	ST	DF	MP	CA
CREDIT	*test error*	29.52 ± 0.37	29.66 ± 0.40	29.65 ± 0.41	29.60 ± 0.44
	deployment error	31.25 ± 0.56	29.66 ± 0.40	29.41 ± 0.32	29.49 ± 0.38
	improvement rate	46.35 ± 3.81	44.71 ± 4.75	36.76 ± 0.53	48.27 ± 5.50
ADULT	*test error*	23.05 ± 0.47	33.55 ± 0.73	24.94 ± 0.52	27.22 ± 0.65
	deployment error	38.64 ± 4.46	33.55 ± 0.73	26.85 ± 0.59	29.34 ± 0.45
	improvement rate	30.92 ± 3.31	60.63 ± 29.40	36.70 ± 1.62	63.79 ± 7.80
GERMAN	*test error*	30.85 ± 0.82	36.10 ± 1.97	33.25 ± 1.44	34.70 ± 2.15
	deployment error	33.40 ± 1.78	36.10 ± 1.97	34.60 ± 1.94	34.25 ± 1.78
	improvement rate	41.20 ± 5.77	42.10 ± 9.07	33.50 ± 2.53	56.10 ± 6.40
SPAMBASE	*test error*	7.11 ± 0.52	10.18 ± 0.45	11.52 ± 0.12	14.37 ± 0.24
	deployment error	22.40 ± 3.14	10.18 ± 0.45	12.92 ± 0.58	14.70 ± 0.36
	improvement rate	40.04 ± 13.06	32.46 ± 14.63	26.42 ± 4.80	43.98 ± 6.18

4.6.3 Results

We summarize the performance of each method in Table 4.2. To wrap up, our method produces classifiers that achieve almost the highest deployment accuracy while providing the highest percentage of improvement across all four datasets. The static classifier, which does not account for adaptations, is vulnerable to strategic manipulation and consequently has the highest deployment error on every dataset. Naively cutting off the manipulated features may harm the accuracy at test time – DF incurs high test errors on `Adult` (33.55%) and `German` (36.10%). In particular, the strategic classifier MP induces the lowest improvement rates on the `Credit` (36.76%) and `German` (34.50%) datasets.

Effect of Trade-off Parameter λ. Figure 4.2 shows the performance of linear classifiers for different values of λ on four real datasets. Note that, since the objective function is non-convex, the trends for test error at deployment are not necessarily monotonic. In general, we observe a trade-off between the improvement rate and deployment error: both increase as λ increases from 0.01 to 10 in all four datasets.

4.7 Conclusion and Limitations

In this work, we study how to train a linear classifier that encourages constructive adaption. We characterize the equilibrium behavior of both the decision subjects and the model designer, and prove other formal statements about the possibilities and limits of constructive adaptation. Finally, our empirical evaluations demonstrate that classifiers trained via our method achieve favorable trade-offs between predictive accuracy and inducing constructive behavior. Our work has several limitations:

1. As a first foray into strategic classification with constructive adaptation, our focus on linear threshold classifiers helps us capture the challenges unique to this setting; indeed, this is ultimately what allows for a closed-form best response (Theorem 4.4.2) even with a significantly more general cost function than in preceding literature. However, this is clearly not true of many models actually in deployment.

2. In order to focus on the *strategic* aspects of constructive adaptation, we assume that the feature taxonomy is simply given; however, distinguishing improvable features from non-improvable features is an interesting question in

its own right, and has been shown to be reducible to a nontrivial causal inference problem [Miller et al., 2020].

3. In real-world scenarios, causal features are often intertwined with non-causal features, and improving one may affect the other. While in our paper, we simplify the setting by assuming independence between the effects, we acknowledge that this is not always the case in practice. One potential way to address this issue is to incorporate additional modeling techniques that account for the causal interactions between features, such as causal inference methods or structural equation modeling.

Chapter 5

To Give or Not to Give? The Impacts of Strategically Withheld Recourse

Individuals often aim to reverse undesired outcomes in interactions with automated systems, like loan denials, through system-recommended actions (recourse) or manipulation actions (e.g., misreporting feature values). While providing recourse benefits users and enhances system utility, it also increases transparency, enabling more strategic exploitation by individuals, especially when groups share information. We show that this tension could potentially lead systems to strategically withhold recourse, challenging assumptions about universal recourse provision in current literature. We propose a framework to investigate the interplay of transparency, recourse, and manipulation and demonstrate that rational utility-maximizing systems frequently withhold recourse, leading to decreased population utility, particularly impacting sensitive groups. To mitigate these effects, we explore the role of recourse subsidies, finding them effective in increasing the provision of recourse actions by rational systems.

5.1 Is System Always Incentivized to Provide Recourse?

When individuals interacting with automated systems are denied a desired outcome (e.g., loan approval), they may seek a means of reversing this decision to obtain the desired outcome. This procedure is commonly referred to as *recourse* [Ustun et al., 2019]. In cases where the system's decision rule is opaque (e.g., lending), the system itself is responsible for supplying individuals with recourse, i.e., the system provides individuals with a minimum cost feature modification which is both feasible for the individual to make and will result in that individual being approved. When feature improvements change an agent's true qualification rate (e.g., paying off debt increases one's creditworthiness), providing recourse can benefit the system. This mutual benefit arises as the result of an increase in the true qualification rate among the population, e.g., the number of creditworthy individuals to whom the bank can lend increases.

However, by providing recourse actions, the system inadvertently reveals information about its decision rule, as each recourse action corresponds with a positively classified feature. This added transparency fosters opportunities for strategic individuals to exploit the system's decision rule by *manipulating* their features, especially when they share their knowledge about the decision rule with one another. For example, platforms like GradCafe for graduate school admissions and LendingClub for loan applications allow agents to see the features of other applicants and potentially misreport their features, leveraging publicly available information to their advantage. Manipulation typically refers to altering the reported features—in other words, deception—often resulting in incorrect predictions [Hardt et al., 2016a]. For instance, individuals might inflate their income or misreport their loan purpose.

This issue has been studied in the literature in strategic classification [Hardt et al., 2016a, Chen et al., 2020a, Bechavod et al., 2022]. Such feature alterations are undesirable from a model-designer perspective because the true qualification of the agent remains unchanged, even though their predicted outcome may be improved.

In the presence of greater manipulations, the system's utility decreases, as the system is more likely to make false positive errors. This natural tension results in settings where providing recourse to all, or even most, agents is no longer in the best interest of the system. This sharply contrasts with the common assumption in the algorithmic recourse literature, which typically considers agents taking recourse actions without the possibility of manipulation. In practice, systems are aware of the possibility of strategic agents' manipulations; for example, the existence of tax audits [1] serves as a simple example of such an awareness. As such, systems may *strategically withhold* recourse in the face of possible manipulations, jeopardizing the practicality of recourse in real-world settings. In particular, when the detriment of added transparency is too great, the system's choice not to provide recourse decreases the overall welfare of the population.

High-Level Overview of Our Model We first provide a high-level overview of our modeling framework using terms and notation that will be rigorously defined later. We model recourse providing in the presence of strategic and collective behavior as a game between a system and a set of n agents where the system can strategically not offer recourse to all agents. Each agent can be represented by a feature vector $\mathbf{x} \in \mathscr{X}$. The system trained a fixed, potentially opaque function $f : \mathscr{X} \to [0, 1]$ to decide who to provide a loan based on the feature vector $\mathbf{x}$.

[1] https://www.irs.gov/businesses/small-businesses-self-employed/irs-audits

The system is also responsible for providing recourse actions to negatively classified agents after they receive their prediction outcome. The central tension comes from the fact that agents can both (1) lie about their features and manipulate them to some publically known positively classified features and (2) take the recommended recourse actions that change their true features. The publically known features mainly come from either agents who are already classified positively, or agents who have successfully obtained a recourse action from the system. The latter is more within the control of the system. Thus, the system's main tool to combat this is strategically withholding recourse actions to the agents to maximize utility. Based on the relative cost of recourse and manipulation, agents choose to take the recommended action or manipulate known positively classified features.

Main Results With the modeling framework described above, our result shows that in many settings, the system is incentivized to strategically withhold recourse from most, if not all, agents. As far as we know, our work is the first to challenge this fundamental assumption and argues that without a third-party's intervention (e.g., the government regulation), a utility-maximizing algorithmic recourse system may be incentivized to withhold recourse from some agents to prevent manipulations strategically. As the system provides recourse to fewer agents, the *social cost*, i.e., the average cost required to achieve positive classification, increases. When the system chooses to withhold recourse actions, fewer agents can access positive classification through legitimate means. This lack of a legitimate path towards positive classification also results in larger portions of individuals having manipulation as their only means of achieving positive classification. The increased social cost associated with lower recourse rates can fall disproportionately on disadvantaged groups, meaning that strategically withheld recourse can further existing disparities in populations. To combat the negative effects of strategically withheld recourse and to increase the rate at which the system provides recourse, we investigate the use of *recourse-subsidies*, which are third-party payments that decrease the cost of recourse. As the cost of recourse decreases, more agents will choose recourse over manipulation. We find that subsidies are an effective tool to increase the number of agents who are provided recourse actions by a rational system.

5.1.1 How Does Our Work Relate to the Surrounding Literatures?

Our work is closely related to the literature on algorithmic recourse, strategic classification, and fairness in general.

Recourse Much of the works in recourse focus on the setting where the requested recourse is guaranteed to be provided out of ethical consideration [Venkatasubramanian and Alfano, 2020]. Our work is the first to challenge this fundamental assumption and argue that without a third-party's intervention, a utility-maximizing algorithmic recourse system may be incentivized to strategically withhold recourse from some agents to prevent manipulations.

Strategic Classification Our work considers a specific type of strategic behavior, namely the *imitation-based* manipulations: agents do not know the classifier f but are aware of a set of positively classified features and can misreport their feature by imitating another agent's feature that is positively classified. Such copycat behavior has been well-known in the literature of game theory, the behavioral economy, and strategic classification, e.g., [Bechavod et al., 2022, Barsotti et al., 2022]. While most of this line of work focuses on agents being strategic and could potentially modify their features to get a favorable prediction outcome, our work focuses on when the system is being strategic and potentially withholds recourse to the agents.

Fairness and Social Cost in Recourse and Strategic Classification Fairness has been explored in the literature algorithmic recourse and strategic classification. For example, existing works on fairness in recourse emphasize the importance of equitable recourse and explore various remedying unfair recourse decisions [Gupta et al., 2019, von Kügelgen et al., 2022, Ehyaei et al., 2023]. Fairness with the presence of strategic behavior has featured studies that highlight the inequity that results from strategic behavior by individuals [Hu et al., 2019], as well as inequity (e.g., social cost) resulting from making classifiers robust to strategic behavior [Milli et al., 2019, Estornell et al., 2023a]. Unlike previous work that primarily focuses on proposing fair classifiers with the presence of strategic agents, our work uniquely demonstrates how the system's strategic withholding impacts the fairness and social cost for different societal groups.

Transparency Works on transparency in machine learning are also related. In particular, Barsotti et al. [2022] find that the risks of transparent explanations are alleviated if effective methods to detect faking behaviors are in place. Unlike our modeling framework, they model transparency as how much noise is in the threshold of a threshold classifier. Akyol et al. [2016] examines the impact of users' strategic behavior on the design and performance of transparent machine learning algorithms, quantifying the "price of transparency" as the cost ratio for the algorithm designer when users exploit transparency compared to when the algorithm is opaque.

5.2 Preliminaries

Let $\mathscr{X} \subset \mathbb{R}^d$ and $\mathscr{Y} \equiv \{0,1\}$ be a domain of features and labels respectively. Let $f : \mathscr{X} \to \mathscr{Y}$ be a binary classifier. A population of n agents with features $\mathbf{X} = \{x : x \in \mathscr{X}\}$ and labels $Y \subset \mathscr{Y}$ are classified by f; all agents desired to be positively classified (e.g., all loan applicants desire approval). The classifier f is unknown to agents. Denote the domain of negatively classified features as $\mathscr{X}_-$ and the domain of positively classified features as $\mathscr{X}_+$, i.e. $f(\mathbf{x}) = 0$ for all $\mathbf{x} \in \mathscr{X}_-$ and $f(\mathbf{x}) = 1$ for all $\mathbf{x} \in \mathscr{X}_+$. All agent prefer positive classification over negative classification. For example, in the context of lending, where $f(\mathbf{x}) = 1$ corresponds to an agent being granted a loan, all agents prefer to have their loans approved, over having their loans denied. Agents who have features $\mathbf{x} \in \mathscr{X}_-$ have two means of obtaining positive classification, *recourse* and *manipulation* which are defined next.

Recourse Recourse is defined as the ability of an agent to obtain a desired outcome from a fixed model Ustun et al. [2019]. Let $c_R : \mathscr{X} \times \mathscr{X} \to \mathbb{R}_+$ be the cost of recourse, i.e. an agent with true features $\mathbf{x}$ pays cost $c_R(\mathbf{x}, \mathbf{x}')$ when modifying their features to be $\mathbf{x}'$. An agent with true feature $\mathbf{x} \in \mathscr{X}_-$ has *optimal* recourse action,

$$\mathbf{x}_R(\mathbf{x}) = \text{argmin}_{\mathbf{x}' \in \mathscr{X}_+} \; c_R(\mathbf{x}, \mathbf{x}') \tag{5.1}$$

$$\text{s.t. } f(\mathbf{x}') = 1, \quad \mathbf{x}' \in A(\mathbf{x}) \tag{5.2}$$

Where $A(\mathbf{x})$ is the set of all features that can be feasibly obtained by an agent with true features $\mathbf{x}$, i.e., $A(\mathbf{x})$ is the set of all *actionable* recourse actions, specified by the system. When agents perform recourse, both their true features and true qualification rate change, i.e., their true features become $\mathbf{x}_R(\mathbf{x})$, and their true qualification rate becomes $\mathbb{P}(y = 1|\mathbf{x}_R(\mathbf{x}))$.

Manipulation In addition to recourse, agents can also perform manipulations. Following Barsotti et al. [2022], we focus on *imitation-based* manipulations: agents do not know the classifier f, but are aware of a set of publically revealed positively classified features $\mathbf{Z} \subseteq \mathscr{X}_+$ (defined below) and can misreport their feature by imitating another agent's feature that is positively classified. For a manipulation cost function $c_M : \mathscr{X} \times \mathscr{X} \to \mathbb{R}_+$ the *optimal* manipulation for an agent with true feature $\mathbf{x}$ is

$$\mathbf{x}_M(\mathbf{x}) = \arg\min_{\mathbf{x}' \in \mathbf{Z}} c_M(\mathbf{x}, \mathbf{x}') \tag{5.3}$$

Remark 5.2.1 (Difference between Recourse and Manipulation)**.** *Recourse changes the agent's true features – when their features become* $\mathbf{x}_R(\mathbf{x})$*, and their true qualification rate becomes* $\mathbb{P}(y = 1|\mathbf{x}_R(\mathbf{x}))$*.. In contrast, manipulation is simply a misreport (rather than a change) of one's features and thus does not change* $\Pr[y|\mathbf{x}]$*. However, since the system only observes the reported features before classification, it does not know whether a report is truthful.*

Remark 5.2.2 (Terminology)**.** *Throughout this chapter, we will interchangeably use the terms 'recourse action' and 'recourse feature'. They both refer to the feature vector that will be classified positively after the agent's taking a particular recourse action. In other words, we assume that whenever an agent reveals their recourse* action*, it also reveals their original feature vector, which is equivalent to revealing the feature vector that corresponds to the vector*after *the agent performs recourse.*

Feature Disclosure and Publically Revealed Set $\mathbf{Z}$ We model the set of publically known features $\mathbf{Z} \subseteq \mathscr{X}_+$ resulting from agents sharing information with each other. In particular, the revealed set $\mathbf{Z}$ is composed of features from two sets: 1) the revealed recourse actions recommended by the system (i.e., $\mathbf{z} \in \mathbf{X}_R$ where $\mathbf{X}_R = \{\mathbf{x}_R(\mathbf{x}), \mathbf{x} \in \mathbf{X}_-\}$), and 2) the set of initial positively classified features (i.e., $\mathbf{z} \in \mathbf{X}_+$). Each element is made public with a *fixed* probability $p \in [0, 1]$, and all publically revealed elements make the reveal set $\mathbf{Z}$. We represent the set of recourse actions that are actually revealed as $\mathbf{Z}_R = \{\mathbf{z} \in \mathbf{X}_R : \text{Reveal}(\mathbf{z}) = 1\}$. Here, $\text{Reveal}(\mathbf{z})$ is a random indicator function that equals 1 with probability p (indicating that feature $\mathbf{z}$ is revealed) and 0 otherwise. Similarly, let $\mathbf{Z}_+$ represent the positively classified features that are actually revealed: $\mathbf{Z}_+ = \{\mathbf{z} \in \mathbf{X}_+ : \text{Reveal}(\mathbf{z}) = 1\}$. As a result, $\mathbf{Z} = \mathbf{Z}_R \cup \mathbf{Z}_+$.

This aims to capture the real-life scenarios where negatively classified agents act collectively to gather information about the classifier f, either by observing the features of their peers who are already classified positively by f, or by observing their peers who have successfully obtained a recourse action from the system.

5.3 Interaction between Agents and the System

Different from the traditional recourse setting where the system is always supposed to offer recourse to individuals requesting them, without any assumptions on regulation from external parties (e.g., the government may force the bank to provide recourses to any individual), a utility-maximizing system may have incentives to withhold recourse to prevent the manipulation behavior from agents strategically. In this section, we introduce our modeling framework to capture such dynamics.

A Motivating Example of Our Model With the above idea in mind, we begin with a motivating example. A bank publishes a classifier to decide who to issue a credit card. Each applicant (with application $\mathbf{x}$) is approved for the card if the bank's model predicts that the applicant could repay their loan. For agents denied credit cards, the bank may offer them access to recourse, i.e., a plan for making the applicant more creditworthy, such as paying off their outstanding debt or increasing income. Such recourse actions are provided through specific programs, e.g., offering financial classes for the agents to take. Agents also have access to an online forum where some applicants share their approved loan or recourse features with the public. Thus, with knowledge of both the recourse actions and the online forum, agents may misreport their features to potentially positively classified features to get approved instead of taking the recommended recourse actions. Thus, the bank has the incentive to only offer recourse to a fraction of individuals whose recourse features are not that easily misreported (e.g., it is easier for the bank to verify later).

5.3.1 Formulating the Dynamic between the System and Agents

We now formalize the dynamics as a sequential game between the system and the agents.

System The system trains a classifier $f : \mathscr{X} \to \mathscr{Y}$ to maximize the prediction accuracy: $f = \text{argmax}_{f \in \mathscr{F}} \sum_{x \in \mathbf{X}} \mathbb{1}[f(\mathbf{x}) = y]$. A collection of negatively classified agents with features $\mathbf{X}_- \subseteq \mathscr{X}_-$ will request recourse actions from the system after receiving their prediction outcome. The system first computes optimal recourse actions for all negatively classified agents but only chooses to *release* a subset of those recourse actions $\mathbf{Z}_R \subset \mathbf{X}_R$ to the public to maximize its utility, i.e., TP − FP:

$$(\textit{System's Objective}): \quad \max_{\mathbf{Z}_R \subset \mathbf{X}_R} \quad \underbrace{\mathsf{TP}(\mathbb{S}) - \mathsf{FP}(\mathbb{S})}_{\text{system's utility}} \tag{5.4}$$

$$\text{where} \quad \underbrace{\mathbb{S} = \{\mathbf{z}(\mathbf{x}, \mathbf{Z}) : \mathbf{x} \in \mathbf{X}\},}_{\substack{\text{the set of features after agent's} \\ \text{final actions described in Equation (5.5)}}} \quad \text{and} \quad \underbrace{\mathbf{Z} = \mathbf{Z}_R \cup \mathbf{Z}_+}_{\substack{\text{the set of all publically revealed features,} \\ \text{including revealed recourse features } \mathbf{Z}_R \\ \text{and positively classified features } \mathbf{Z}_+}}$$

$\mathsf{TP}(\mathbb{S})$ and $\mathsf{FP}(\mathbb{S})$ are the true positive and false positive rates on the set of features after the agent's final actions. We assume that the system either knows c_R and c_M, or can reasonably approximate these cost functions when optimizing their objective. Intuitively, this definition of system utility captures a bank gaining a utility of 1 for every loan that is repaid and −1 for each loan that is not repaid.

Agents: Agents who are already positively classified will keep their original feature $\mathbf{x}$. Agents who are negatively classified will request a recourse action from the system. Upon seeing the publically revealed features $\mathbf{Z}$ defined in Section 5.2, agents who are provided with a recourse action adapt their features from $\mathbf{x}$ to $\mathbf{z} = \mathbf{x}_M(\mathbf{x})$ or $\mathbf{z} = \mathbf{x}_R(\mathbf{x})$ such that $f(\mathbf{z}) = 1$, while minimizing the cost of the corresponding action. When both the recourse and manipulation actions are greater than 1[2], the agents will choose to stay with their original features $\mathbf{x}$, which corresponds to the *do-nothing* action. Agents who are not provided with a recourse action will choose to manipulate or *do nothing.* For already positively classified agents, their final action is always the *do-nothing* action.

Agent's Best Response Denote $\zeta_{\mathbf{x}} \in \{0, 1\}$ as an indicator for whether agent $\mathbf{x}$ is provided with a recourse or not (i.e., $\zeta(\mathbf{x}) = 1$ when provided with a recourse

[2] The strategic agent's utility for adapting their feature from x to x' is determined by the standard utility function in the literature of strategic classification (see, e.g., Hardt et al. [2016a]), which is $U(x, x') = f(x') - c(x, x')$. Thus, when the cost of adaptation $c(x, x') \geq 1$, the utility will be less than 0, in which case, the agent does nothing.

action). Then for all agents with $f(\mathbf{x}) = 0$, their final action is:

$$\mathbf{z}(\mathbf{x}, \mathbf{Z}) = \begin{cases} \mathbf{x}_R(\mathbf{x}) & \zeta_{\mathbf{x}} = 1 \text{ and } c_R(\mathbf{x}, \mathbf{x}_R(\mathbf{x})) < \min(1, c_M(\mathbf{x}, \mathbf{x}')), \\ & \forall \mathbf{x}' \in \mathbf{Z} \\ \mathbf{x}_M(\mathbf{x}) & \zeta_{\mathbf{x}} = 1 \text{ and } c_M(\mathbf{x}, \mathbf{x}_M(\mathbf{x})) < \min(1, c_R(\mathbf{x}, \mathbf{x}')), \\ & \forall \mathbf{x}' \in \mathbf{Z}, \text{ or } \zeta_{\mathbf{x}} = 0 \text{ and } \ c_M(\mathbf{x}, \mathbf{x}_M(\mathbf{x})) < 1 \\ \mathbf{x} & \zeta_{\mathbf{x}} = 1 \text{ and } c_R(\mathbf{x}, \mathbf{x}_R(\mathbf{x})), c_M(\mathbf{x}, \mathbf{x}_R(\mathbf{x})) \geq 1, \\ & \forall \mathbf{x}' \in \mathbf{Z}, \text{ or } \zeta_{\mathbf{x}} = 0 \text{ and } c_M(\mathbf{x}, \mathbf{x}_M(\mathbf{x})) \geq 1 \end{cases} \tag{5.5}$$

Summary of System-Agent Interaction

1. Agents arrive simultaneously, and the system trains a classifier $f : \mathscr{X} \to \mathscr{Y}$ for maximum prediction accuracy.

2. Negatively classified agents request recourse, and the system selects agents for recourse provision to maximize utility (Equation (5.4)).

3. Positively classified agents and those provided recourse have a probability $p \in [0, 1]$ to reveal features, contributing to the publicly revealed set $\mathbf{Z} \subseteq \mathscr{X}_+$.

4. Upon observing $\mathbf{Z}$, agents execute final actions based on Equation (5.5).

Our framework is intended to capture settings where black box models are used for decision-making. Any agent subjected to the decision rules will not have direct access to the model but will still act in their own best interest. In these opaque settings, recourse proposed by the system naturally offers a way for agents to learn more about the decision rule, thus increasing their ability to game the system. The tension between transparency and manipulability exists naturally; our framework, while stylized, is a means of capturing this tension when recourse increases model transparency.

5.3.2 Useful Definitions

We also provide two definitions to aid the discussions of strategic withheld algorithmic recourse systems.

Definition 5.3.1. ***(Recourse Rate)*** *Let $\boldsymbol{X}_-$ be the set of features of negatively classified agents. For a given set of disclosed features (i.e., recourse actions) $\boldsymbol{Z}$, the* recourse rate *$rec(\boldsymbol{Z}, \boldsymbol{X}_-)$ is defined as the fraction of agents who choose to perform recourse when shown $\boldsymbol{Z}$:*

$$rec(\boldsymbol{Z}, \boldsymbol{X}_-) = \frac{\sum\limits_{\mathbf{x} \in \boldsymbol{X}_-} \mathbb{1}\big[\min\limits_{\mathbf{z}' \in \boldsymbol{Z}} c_R(\mathbf{x}, \mathbf{z}') < \min\big(1, \min\limits_{\mathbf{z}'' \in \boldsymbol{Z}} c_M(\mathbf{x}, \mathbf{z}'')\big)\big]}{|\mathbf{X}_-|}$$

Definition 5.3.2. ***(Manipulation Rate)*** *Let $\boldsymbol{X}_-$ be the set of features of the negatively classified agents. For a given set of disclosed features (i.e., recourse actions) $\boldsymbol{Z}$, the* manipulation rate *$manip(\boldsymbol{Z}, \boldsymbol{X}_-)$ is defined as the fraction of the n agents which choose to manipulate when shown features $\boldsymbol{Z}$:*

$$manip(\boldsymbol{Z}, \boldsymbol{X}_-) = \frac{\sum\limits_{\mathbf{x} \in \boldsymbol{X}_-} \mathbb{1}\big[\min\limits_{\mathbf{z}' \in \boldsymbol{Z}} c_M(\mathbf{x}, \mathbf{z}') < \min\big(1, \min\limits_{\mathbf{z}'' \in \boldsymbol{Z}} c_R(\mathbf{x}, \mathbf{z}'')\big)\big]}{|\mathbf{X}_-|}$$

5.4 System Utility

Recall from the previous section, the system aims to select a set $\mathbf{Z}_R \subseteq \mathbf{X}_R$ to reveal as recourse recommendations simultaneously in order to minimize the number of agents who perform manipulation. We can first show that this problem is NP-hard:

Theorem 5.4.1 (Strategic Recourse Selection is Hard)**.** *The problem of selecting the optimal set of recourse actions to recommend, such that the system's utility is maximized (Equation 5.4), is NP-hard, even when the probability of disclosure $p = 1$.*

Proof Sketch. We reduce from the known NP-hard problem Minimum k-Union (MkU). Given an instance of MkU, we show that it can be mapped to an instance of our strategic recourse selection problem. We defer the rest of the proof to the appendix. □

Despite the hardness of this objective, the system's utility is *submodular* in the set of provided recourse actions. This characteristic enables the system to employ standard submodular optimization techniques to approximately get the optimal recourse actions to disclose to k agents.

Theorem 5.4.2 (System's Utility is Submodularity)**.** *The system's objective function is* submodular *with respect to the size of the set of revealed features.*

Proof Sketch. We defer the full proof to the Appendix. The intuition for this result follows from the fact that agents will select their action (recourse, manipulation, or do nothing) based on the set of publicly revealed features $\mathscr{X}$ and the recourse action recommended to them by the system $\mathbf{x}_R$. An agent who is given a recourse action will only manipulate if there exists some $\mathbf{x}' \in \mathbf{Z}$ such that $c_M(\mathbf{x}, \mathbf{x}') < c_R(\mathbf{x}, \mathbf{x}_R)$. Let $\mathscr{X}'_{\mathbf{x}} = \{\mathbf{x}' \in \mathscr{X}_+ : c_M(\mathbf{x}, \mathbf{x}') < c_R(\mathbf{x}, \mathbf{x}_R)\}$ be the set of all such features for a given agent with true features $\mathbf{x}$. Then the probability that this agent performs manipulation, i.e., the probability that some $\mathbf{x}' \in \mathscr{X}'_{\mathbf{x}}$ is revealed, is concave in the number of recourse actions recommended to other agents which are in $\mathscr{X}'_{\mathbf{x}}$ (this is due to the fact that each such feature is made public with probability α). □

Remark 5.4.3. *When the disclosure probability $p = 1$, the optimal set of recourse features to disclose can be found via an ILP (see Appendix C.2).*

5.4.1 Disconnect between System's Utility and Offering Recourse

We now demonstrate that the system's utility maximization and recourse maximizing are not always equivalent. We first show that, in expectation, the system benefits from agents *taking* recourse actions:

Theorem 5.4.4 (System's Expected Utility Changes)**.** *The system's expected utility increases for each recourse action taken by agents and decreases for every manipulation action taken by agents.*

Proof. Notice that only agents $\mathbf{x} \in X^{(0)}$ who are originally negatively classified would request a recourse from the system in the first place, and both the recourse action and the manipulation actions that they are potentially going to take will be positively classified by the system. From the system's perspective, when the classifier is non-trivial (better than random guessing), all positively classified $\mathbf{x}$ are more likely to have true label $y = 1$, and all negatively classified $\mathbf{x}$ are more likely to have true label 0. When an agent with feature $\mathbf{x}$ takes recourse, the expected system utility change is:

$$\begin{aligned}
&\Delta(\text{System's Utility})(\mathbf{x} \to \mathbf{z}_R)\\
&= \big(\mathbb{1}[y(\mathbf{z}_R) = 1, f(\mathbf{z}_R) = 1] - \mathbb{1}[y(\mathbf{z}_R) = -1, f(\mathbf{z}_R) = 1]\big) - 0\\
&= 2\Pr[y(\mathbf{z}_R) = 1 | X = \mathbf{z}_R] - 1 \geq 0 \qquad \text{(f is a non-trivial classifier, and } f(\mathbf{z}_R) = 1)
\end{aligned}$$

Similarly, when the agent takes manipulation, the expected system utility change is:

$$\begin{aligned}
&\Delta(\text{System's Utility})(\mathbf{x} \to \mathbf{z}_M)\\
&= \big(\mathbb{1}[y(\mathbf{x}) = 1, f(\mathbf{z}_M) = 1] - \mathbb{1}[y(\mathbf{x}) = -1, f(\mathbf{z}_M) = 1]\big) - 0\\
&= 2\Pr[y(\mathbf{x}) = 1 | X = \mathbf{x}] - 1 \leq 0 \;\; \text{(Since f is a non-trivial classifier, and } f(\mathbf{x}) = 0)
\end{aligned}$$

When the agent performs do-nothing, the system utility remains the same. □

Thus, we see that the system will benefit from agents *taking* a recourse action while suffering from agents taking a manipulation action. However, this does not imply that the system is always incentivized to provide as many recourse actions as possible since agents might not always take them if they collude, which creates a natural misalignment between the system's utility and recourse offering for the system.

5.5 Cost of Manipulation-Proof System

Having shown that the system could potentially be incentivized to withhold recourse from the agents, what are the consequences from the agent's perspective? In this section, we study the consequence of increased system manipulation-proofness, by proposing several metrics, including the social cost (Section 5.5.1), the difference in recourse ratio as well as the social cost for social groups (Section 5.5.2), and demonstrate our results in the experimental section.

5.5.1 Social Cost and Unfairness of Manipulation-proofness

How much has the average recourse cost increased due to the principal not providing optimal recourse actions for everyone due to the risk of agents manipulating? In particular, we propose the following definition to capture the *social cost* for a manipulation-proofness system:

Definition 5.5.1. ***(Social Cost of a Manipulation-proof System)*** *Given a particular set $S \subseteq \mathscr{X}$ that select k actions to reveal as recourse recommendations, the* social cost from the principal being manipulation-proof *refers to the additional* *cost agents must pay because the system recommends sub-optimal recourse actions to limit the total amount of manipulation. Denote $\mathbf{x}_R$ as the optimal recourse action provided by a non-strategic system, and $\mathbf{z}_R(\mathbf{x}, \boldsymbol{Z})$ as the recourse action that the agent takes given the revealed set $\boldsymbol{Z}$:*

$$\textit{cost}(\boldsymbol{Z}, \boldsymbol{X}_-) = \sum_{\mathbf{x} \in \boldsymbol{X}_-} \big(c_R(\mathbf{x}, \mathbf{z}_R(\mathbf{x}, \boldsymbol{Z})) - c_R(\mathbf{x}, \mathbf{x}_R)\big), \textit{ where } \mathbf{z}_R(\mathbf{x}, \boldsymbol{Z}) = \arg\min_{\mathbf{z} \in \boldsymbol{Z}} c_R(\mathbf{x}, \mathbf{z})$$

For the remainder of our results, we focus on univariate classifiers, i.e., the feature $\mathbf{x}$ is one-dimensional. There is a natural correspondence between univariate and multivariate classifiers in the sense that one can imagine the space of single-dimensional features as the scores produced a multi-dimensional classifier $f(\mathbf{x})$ (see, e.g., Lemma 3.1 in Milli et al. [2019]). Here, there is a natural equivalence between feature-based costs in a single dimension and score-based costs in multiple dimensions – that is, in the case when $f(\mathbf{x}) = [h(\mathbf{x}) \geq \theta]$ for some score function h and threshold theta, we can view f as a single dimensional classifier acting on the space of scores produced by h.

Theorem 5.5.2 (Monotonicity of Social Cost)**.** *When the recourse cost $c_R(x, x')$ is monotonic in $\|x - x'\|$, and consider a linear threshold classifier. The social cost monotonically decreases in the easiest obtained recourse action.*

Proof. Consider a 1-dimensional setting, where the system uses a linear threshold classifier $f(x) = \mathbb{1}[x \geq \tau]$. In this case, the optimal recourse action for any agent is always the minimum recourse action that has been revealed so far, namely $z_{\min} = \min_{z \in \mathbf{Z}} z$. Recall the definition of the social cost:

$$\text{cost}(\mathbf{Z}, \mathbf{X}_-) = \sum_{\mathbf{x} \in \mathbf{X}_-} \left(c_R(\mathbf{x}, \mathbf{z}_R(\mathbf{x}, \mathbf{Z}) - c_R(\mathbf{x}, \mathbf{x}_R)\right), \text{where } \mathbf{z}_R(\mathbf{x}, \mathbf{Z}) = \arg\min_{\mathbf{z} \in \mathbf{Z}} c_R(\mathbf{x}, \mathbf{z})$$

When the cost function is monotonic in the ℓ_2 norm, e.g., $c_R(x, x') = w_R \cdot \|x - x'\|$, we have

$$c_R(\mathbf{x}, \mathbf{z}_R(\mathbf{x}, \mathbf{Z})) = w_R \cdot \|x - \mathbf{z}_R(\mathbf{x}, \mathbf{Z})\| = w_R \cdot \min_{z \in \mathbf{Z}} \|x - z\| = w_R \cdot \left(\min_{z \in \mathbf{Z}} z - x\right)$$

$$c_R(\mathbf{x}, \mathbf{x}_R) = w_R \cdot \|\mathbf{x} - \mathbf{x}_R\| = w_R \cdot \|\mathbf{x} - \tau\| = w_R \cdot (\tau - x)$$

Thus,

$$\begin{aligned}
\mathrm{cost}(\mathbf{Z}, \mathbf{X}_-) &= \sum_{\mathbf{x} \in \mathbf{X}_-} \left(c_R(\mathbf{x}, \mathbf{z}_R(\mathbf{x}, \mathbf{Z})) - c_R(\mathbf{x}, \mathbf{z}_R)\right) \\
&= \sum_{\mathbf{x} \in \mathbf{X}_-} \left[w_R \cdot \left(\min_{z \in \mathbf{Z}} z - x\right) - w_R \cdot (\tau - x)\right] \\
&= |\mathbf{X}_-| \cdot w_R \cdot \left(\min_{z \in \mathbf{X}_-} z - \tau\right)
\end{aligned}$$

As the size of $\mathbf{Z}$ gets larger (more recourse actions get revealed), $\min_{z \in \mathbf{Z}} z$ will be non-increasing, which means that $\mathrm{cost}(\mathbf{Z}, \mathbf{X}_-)$ is monotonically decreasing.

□

Theorem 5.5.2 provides intuitions on the relationship between social cost and the size of the sets of recourse actions. As the size of $\mathbf{Z}$ gets larger, $\mathbf{z}_{\min} = \min_{z \in \mathbf{Z}} \mathbf{z}$ is only going to be non-increasing, indicating that social cost will be non-decreasing as the size of the revealed set becomes larger.

5.5.2 Unfairness in a Manipulation-Proof System

We also measure the disparities of different social groups in terms of their differences in 1) recourse ratios (defined in Definition 5.3.1) and 2) social cost (defined in Definition 5.3.1). Understanding the disparities in terms of recourse rate and social cost among different groups is crucial for addressing issues of unfairness in an algorithmic recourse system [Gupta et al., 2019, von Kügelgen et al., 2022]. These disparities often reflect systemic biases and inequalities, impacting marginalized communities disproportionately.

In particular, assume there are two groups of agents $\mathbf{X}^{(g_0)}$ and $\mathbf{X}^{(g_1)}$, where g_0, g_1 represents their group memberships, we are interested in the following quantities:

Definition 5.5.3. ***(Disparity in Social Cost)*** *The disparity in social cost for two group g_0, g_1 is defined as:*

$$\mathit{Diff}^{(cost)}(\boldsymbol{Z}, \boldsymbol{X}^{(g_0)}, \boldsymbol{X}^{(g_1)}) := \left| \mathit{cost}(\boldsymbol{Z}, \boldsymbol{X}_{-}^{(g_1)}) - \mathit{cost}(\boldsymbol{Z}, \boldsymbol{X}_{-}^{(g_0)}) \right|$$

Definition 5.5.4. ***(Disparity in Recourse Ratio)*** *The disparity in recourse ratio for two group g_0, g_1 is defined as:*

$$\mathit{Diff}^{(rec)}(\boldsymbol{Z}, \boldsymbol{X}^{(g_0)}, \mathscr{X}^{(g_1)}) := \left| \mathit{rec}(S, \boldsymbol{X}_{-}^{(g_1)}) - \mathit{rec}(\boldsymbol{Z}, \boldsymbol{X}_{-}^{(g_0)}) \right|$$

In the experiments section, we demonstrate the vast existence of these disparities across different datasets (see Figure 5.5 and Figure 5.6). By quantifying and illuminating these disparities, we gain crucial insights into the specific mechanisms of inequity and injustice within algorithmic recourse systems. This in-depth understanding is pivotal, particularly in comprehending how systems that aim to maximize utility might strategically withhold recourse, thereby exacerbating these disparities. Recognizing and addressing this interaction is vital for the recourse community, as it directly impacts the development of fairer and more effective policies and practices. It not only aids in the formulation of more equitable systems but also plays a significant role in raising public awareness.

5.6 Subsidies

As a means to remedy the adverse population- and group-level impacts previously observed, we investigate the use of subsidies (rigorously defined next). Subsidies correspond to a global decrease in the cost of recourse. For example, free educational material on financial literacy distributed to any agent petitioning the bank for recourse will increase the ease at which that agent can perform recourse actions. It is important to note that our investigation does not focus on the monetary value required to achieve a particular cost reduction but rather focuses on the question of how particular cost decreases change both the willingness of the system to provide recourse as well as the agents' choice of performing recourse over manipulation.

Definition 5.6.1. ***(Subsidies) [Hu et al., 2019]*** *A subsidy $0 \leq \alpha \leq 1$ is a scalar decrease to the cost of recourse. That is, for subsidy α, agents performing recourse pay only $(1-\alpha) \cdot c_R(\mathbf{x}, \mathbf{x}')$ instead of the full cost of $c_R(\mathbf{x}, \mathbf{x}')$.*

We denote $c_R(x, x'; \alpha) = (1-\alpha) \cdot c_R(\mathbf{x}, \mathbf{x}')$ as the new recourse cost at subsidy level α.

5.6.1 The Effect of Subsidy on Recourse Rate, Social Cost, Unfairness in Manipulation-Proof System

In the previous section, we observed that when a utility-maximizing system recognizes the potential for agents' manipulative behavior, it may strategically withhold recourse. This action can lead to increased social costs and unfairness for the agents. In this section, we demonstrate how subsidy help increase the recourse rate (Theorem 5.6.2) and system's utility (Theorem 5.6.4). Additionally, subsidies can mitigate disparities in recourse rate differences (Theorem 5.6.6) and social cost differences (Theorem 5.6.5) among various groups.

Subsidies and Recourse Rate We first show how subsidies influence the recourse rate. Recall that subsidy reduces the cost of recourse from $c_R(x, x')$ to $c_R(x, x'; \alpha)$. With that, the recourse rate becomes:

$$\operatorname{rec}(\mathbf{Z}, \mathbf{X}_-; \alpha) = \frac{\sum_{\mathbf{x} \in \mathbf{X}_-} \mathbb{1}\left[\min_{\mathbf{z}' \in \mathbf{Z}} c_R(\mathbf{x}, \mathbf{z}'; \alpha) < \min\big(1, \min_{\mathbf{z}'' \in \mathbf{Z}} c_M(\mathbf{x}, \mathbf{z}'')\big)\right]}{|\mathbf{X}_-|}$$

The key observation we make here is that with subsidy α, the cost of recourse becomes $(1 - \alpha) \cdot c_R(\mathbf{x}, \mathbf{z}')$ but the cost of manipulation remains the same. Both optimal recourse actions $\mathbf{z}_R$ and the optimal manipulation action $\mathbf{z}_M$ remain the same. With that, we can show that the recourse rate is a monotonic function in subsidy, namely as the subsidy level increases, the recourse rate will also increase:

Theorem 5.6.2 (Subsidy Influence on Recourse Rate)**.** *Given a reveal set $\boldsymbol{Z}$, the recourse rate rec$(\boldsymbol{Z}, \boldsymbol{X}_-, \alpha)$ is a monotonically increasing function of subsidies α.*

Proof. Recall that given a revealed set $\mathbf{Z}$, with subsidy α, the corresponding recourse rate becomes:

$$\operatorname{rec}(\mathbf{Z}, \mathbf{X}_-; \alpha) = \frac{\sum_{\mathbf{x} \in \mathbf{X}_-} \mathbb{1}\left[\min_{\mathbf{z}' \in \mathbf{Z}} c_R(\mathbf{x}, \mathbf{z}'; \alpha) < \min\big(1, \min_{\mathbf{z}'' \in \mathbf{Z}} c_M(\mathbf{x}, \mathbf{z}'')\big)\right]}{|\mathbf{X}_-|}$$

In particular, with subsidy α, the cost of recourse becomes $(1 - \alpha) \cdot c_R(\mathbf{x}, \mathbf{z}')$, the cost of manipulation remains the same. Both optimal actions $\mathbf{z}_R$ and $\mathbf{z}_M$ remain the same.

Thus, for the nominator, we have:

$$
\begin{aligned}
&\sum_{\mathbf{x}\in\mathbf{X}_-} \mathbb{1}\left[\min_{\mathbf{z}'\in\mathbf{Z}} c_R(\mathbf{x},\mathbf{z}';\alpha) \leq \min\left(1,\ \min_{\mathbf{z}''\in\mathbf{Z}} c_M(\mathbf{x},\mathbf{z}'')\right)\right] \\
&= \sum_{\mathbf{x}\in\mathbf{X}_-} \mathbb{1}\left[\min_{\mathbf{z}'\in\mathbf{Z}} (1-\alpha)\cdot c_R(\mathbf{x},\mathbf{z}') \leq \min\left(1,\ \min_{\mathbf{z}''\in\mathbf{Z}} c_M(\mathbf{x},\mathbf{z}'')\right)\right] \\
&= \sum_{\mathbf{x}\in\mathbf{X}_-} \mathbb{1}\left[(1-\alpha)\cdot \underbrace{\min_{\mathbf{z}'\in\mathbf{Z}} c_R(\mathbf{x},\mathbf{z}')}_{\text{fixed}} \leq \underbrace{\min\left(1,\ \min_{\mathbf{z}''\in\mathbf{Z}} c_M(\mathbf{x},\mathbf{z}'')\right)}_{\text{fixed}}\right] \\
&= \sum_{\mathbf{x}\in\mathbf{X}_-} \mathbb{1}\left[(1-\alpha)\cdot \underbrace{\min_{\mathbf{z}'\in\mathbf{Z}} c_R(\mathbf{x},\mathbf{z}')}_{\text{fixed for a particular x}} \leq \underbrace{\min\left(1,\ \min_{\mathbf{z}''\in\mathbf{Z}} c_M(\mathbf{x},\mathbf{z}'')\right)}_{\text{fixed for a particular x}}\right] \\
&= \sum_{\mathbf{x}\in\mathbf{X}_-} \mathbb{1}\left[\alpha \geq \underbrace{1 - \frac{\min\left(1,\ \min_{\mathbf{z}''\in\mathbf{Z}} c_M(\mathbf{x},\mathbf{z}'')\right)}{\min_{\mathbf{z}'\in\mathbf{Z}} c_R(\mathbf{x},\mathbf{z}')}}_{\text{fixed for a particular x}}\right]
\end{aligned}
$$

As α becomes larger, this quantity will be non-decreasing. This implies that the recourse rate is a monotonically non-decreasing function of subsidy for a given revealed set $\mathbf{Z}$. □

Subsidy and Social Cost With subsidy α, the social cost for a given revealed set $\mathbf{Z}$ becomes:

$$\mathrm{cost}(\mathbf{Z},\mathbf{X}_-;\alpha) = \sum_{\mathbf{x}\in\mathbf{X}_-} \left(c_R(\mathbf{x},\mathbf{z}_R(\mathbf{x},\mathbf{Z};\alpha);\alpha) - c_R(\mathbf{x},\mathbf{x}_R;\alpha)\right)$$

where $\mathbf{z}_R(\mathbf{x},\mathbf{Z};\alpha) = \arg\min_{\mathbf{z}\in\mathbf{Z}}(1-\alpha)c_R(\mathbf{x},\mathbf{z})$ is the optimal recourse action given revealed set $\mathbf{Z}$ and a particular subsidy level α, and $\mathbf{x}_R$ is the optimal default recourse action provided by the system without any strategic withholding. We can show that the social cost is also a monotonic non-increasing function in the subsidy level:

Theorem 5.6.3 (Subsidy Influence on Social Cost)**.** *Given a revealed set $\boldsymbol{Z}$, the social cost cost$(\boldsymbol{Z},\boldsymbol{X}_-;\alpha)$ is monotonically decreasing in subsidies.*

Subsidy with System's Utility Subsidies also help improve the system's utility. In particular, we show that under certain assumptions on the cost functions (i.e., monotonic in the distance and only cross once), the system's utility is monotonic in subsidies as well:

Theorem 5.6.4 (Subsidy's Influence on System's Utility)**.** *Given a revealed set $\boldsymbol{Z}$, when both $c_R(x, x')$ and $c_M(x, x')$ are monotonic in $\|x - x'\|$ and only cross once, the system utility is monotonically increasing in subsidies.*

Again, the key observation is that with subsidy α, the cost of recourse decreases and becomes $(1 - \alpha) \cdot c_R(\mathbf{x}, \mathbf{z}')$, but the cost of manipulation remains the same. Thus, more agents will choose to perform recourse over manipulation, leading to more increases in true positives for the system, which further leads to an increase in system's utility.

Subsidy and Social Cost Difference Next we examine the difference in social cost between groups as a function of subsidies. We find hat subsidies are an effective tool at mitigating disparities caused by the system strategically withholding recourse.

Theorem 5.6.5 (Subsidy Influence on Social Cost Disparity)**.** *With subsidy α, the disparity in social cost for two group g_0, g_1 becomes:*

$$Diff^{(cost)}(\boldsymbol{Z}, \boldsymbol{X}^{(g_0)}, \boldsymbol{X}^{(g_1)}; \alpha) := \left| cost(\boldsymbol{Z}, \boldsymbol{X}_-^{(g_1)}; \alpha) - cost(\boldsymbol{Z}, \boldsymbol{X}_-^{(g_0)}; \alpha) \right|$$

Given a revealed set $\boldsymbol{Z}$, the social cost difference monotonically decreases in subsidies.

Proof. Recall the definition of social cost difference:

$$\mathit{Diff}^{(\text{cost})}(S, \mathbf{X}_{-}^{(g_0)}, \mathbf{X}_{-}^{(g_1)}) := \left| \text{cost}(S, \mathbf{X}_{-}^{(g_1)}) - \text{cost}(S, \mathbf{X}_{-}^{(g_0)}) \right|$$

Again, consider a 1-dimensional setting, where the system uses a linear threshold classifier $f(x) = \mathbb{1}[x \geq \tau]$. In this case, the optimal recourse action for any agent is always the minimum recourse actions that has been revealed so far, namely $z_{\min} = \min_{z \in \mathbf{Z}} z$. Recall from the proof for social cost with subsidy, we have for a particular set $\mathscr{X}$:

$$\text{cost}(\mathbf{Z}, \mathbf{X}, \alpha) = (1 - \alpha) \cdot |\mathbf{X}| \cdot w_R \cdot \left(\min_{z \in \mathbf{Z}} z - \tau \right)$$

Plug it back to the definition of social cost difference at a certain subsidy level, we have:

$$\begin{aligned}
&\mathit{Diff}^{(\text{cost})}(\mathbf{Z}, \mathbf{X}^{(g_0)}, \mathbf{X}^{(g_1)}; \alpha) \\
&= \left| \text{cost}(\mathbf{Z}, \mathbf{X}^{(g_1)}) - \text{cost}(\mathbf{Z}, \mathbf{X}^{(g_0)}) \right| \\
&= \left| (1 - \alpha) \cdot |\mathbf{X}_{-}^{(g_0)}| \cdot (\min_{z \in \mathbf{Z}} - \tau) - (1 - \alpha) \cdot |\mathbf{X}_{-}^{(g_1)}| \cdot (\min_{z \in \mathbf{Z}} - \tau) \right| \\
&= \left| (1 - \alpha) \cdot (|\mathbf{X}_{-}^{(g_0)}| - |\mathbf{X}_{-}^{(g_1)}|) \cdot (\min_{z \in \mathbf{Z}} - \tau) \right|
\end{aligned}$$

which is monotonically decreasing as α increases.

□

Intuitively, as we increase the subsidy level, the cost of recourse decreases linearly as a function of the subsidy level, making it increasingly cheaper to perform the optimal recourse action. For both social groups, their social cost approaches 0 as we increase the subsidy level; as a result, the disparity in social cost between the two groups also decreases to 0.

Subsidy and Recourse Rate Difference With subsidy α, for a given a revealed set $\mathbf{Z}$, the disparity in recourse ratio for two group g_0, g_1 becomes:

$$\mathit{Diff}^{(\mathrm{rec})}(\mathbf{Z}, \mathbf{X}^{(g_0)}, \mathbf{X}^{(g_1)}; \alpha) := \left| \mathrm{rec}(\mathbf{Z}, \mathbf{X}_{-}^{(g_1)}; \alpha) - \mathrm{rec}(\mathbf{Z}, \mathbf{X}_{-}^{(g_0)}; \alpha) \right|$$

where $\mathrm{rec}(\mathbf{Z}, \mathbf{X}_{-}^{(g_i)})$ is the recourse rate for a particular subgroup g_i. We show that when subsidies are sufficiently large, the recourse rate difference is monotonically decreasing in subsidies:

Theorem 5.6.6 (Subsidy's Influence on Recourse Rate Disparity)**.** *Given two groups g_0 and g_1 of relatively equal negatively classified agents size $|\boldsymbol{X}_{-}^{(g_0)}| \approx |\boldsymbol{X}_{-}^{(g_1)}|$, there exists a subsidy level $0 \leq \alpha^* \leq 1$, such that $\forall \alpha \geq \alpha^*$, the recourse rate difference monotonically decreases.*

This result follows from the fact that when recourse is free, i.e., subsides are maximized, all agents can perform recourse and the recourse rate difference is 0. Thus, as subsidies increase there must exist a point (namely α^*) when both groups are able to take advantage of subsides at proportional rates, thus decreasing the gap between the number of agents performing recourse in both groups. We also verify empirically that for recourse rate difference, there indeed exists a peak subsidy value α^* where the recourse rate difference increases before and then decreases afterward (see Figure 5.7).

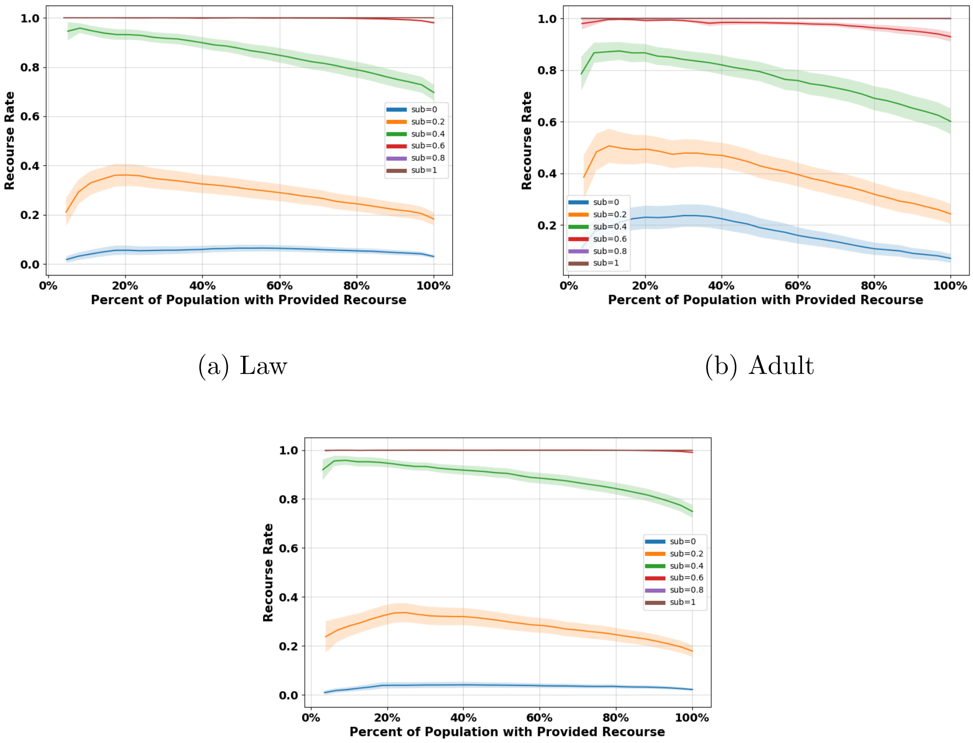
5.7 Empirical Studies

Setup We conduct experiments using three datasets: 1) **Law School** Wightman and Council [1998] dataset, in which the objective is to predict whether a student will pass the bar exam on the first attempt, **Adult Income** Dua and Graff [2017b] in which the objective is to predict whether an individual earns more than $50K$ annually, and **German Credit** Yeh and Lien [2009] in which the objective is to predict whether a given individual will *not* default on their credit.

In each dataset, agents have constant utility over approved features, i.e., the conventional recourse setting where $u_a(\mathbf{x}) = 1$ for all $\mathbf{x}$; the principal (system) has utility $u_p(\mathbf{x}) = 1$ when the agent is a true positive ($y = 1$, $f(x) = 1$) and $u_p(\mathbf{x}) = -1$ when the agent is a false positive ($y = -1$, $f(x) = 1$).

Qualification is predicted via a Logistic Regression model. Additionally, we present outcomes using a Gradient Boosting Decision Tree as the classifier across all datasets, where we observe comparable results. We defer the plots to Appendix C.8. Recourse and manipulation both carry an ℓ_2 cost, namely $c_R(\mathbf{x}, \mathbf{z}) = \|w_R \cdot (\mathbf{x} - \mathbf{z})\|_2$, and $c_M(\mathbf{x}, \mathbf{z}) = \|w_M \cdot (\mathbf{x} - \mathbf{z})\|_2$. In our experiments, we report outcomes over 100 runs using randomly initialized w_R and w_M and resampled subsets of positive and negative agents in the dataset in each run. We use the local search method provided in Orso et al. [2015] to compute the set of recourse actions that the system provides to agents. We set the probability that agent discloses their feature publicly at $p = 0.7$ for all experiments (when varying this value we observe qualitatively similar results).

Recourse Rate and Manipulation Rate We begin by examining the relationship between the fraction of the population choosing to perform recourse, and the fraction choosing to perform manipulation, as a function of the fraction of agents who are given a recourse action. Recall that 0 subsidy, i.e. sub = 0, is equivalent to the setting with no subsidies. In Figures 5.1 and 5.2, we see that in general, as the percentage of agents who are provided a recourse action increases, recourse



(a) Law

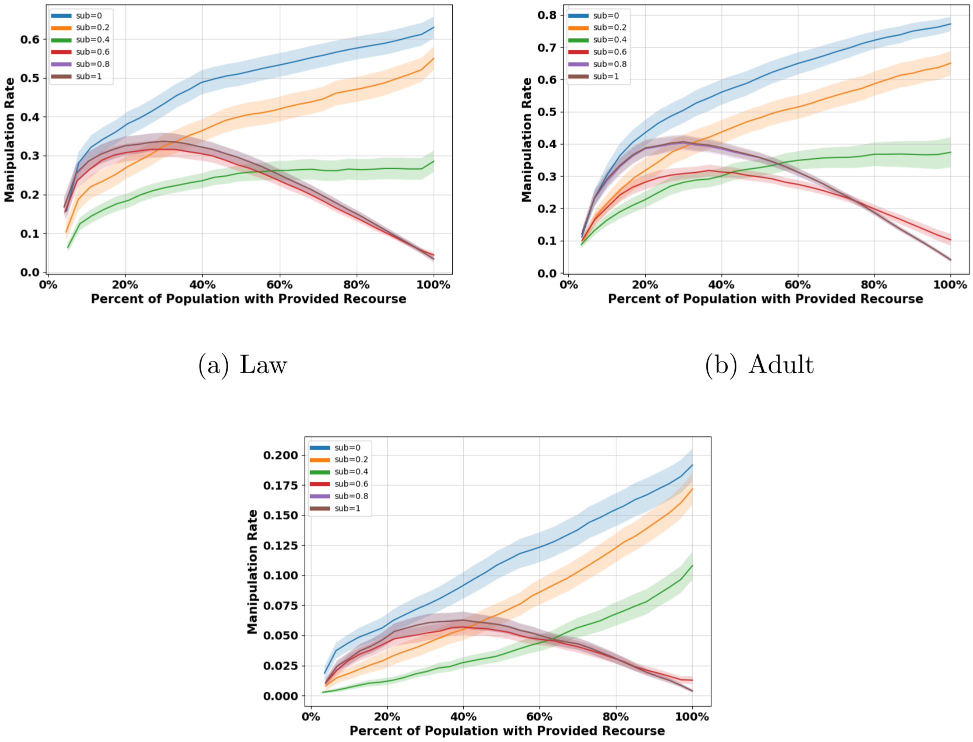
(b) Adult

(c) Credit

Figure 5.1: Fraction of the population performing recourse, with 95% confidence intervals. Each line corresponds to a different subsidy ratio “subs”, i.e., the cost reduction applied to recourse.

rate decreases while manipulation rate increases (this trend holds for each subsidy value). Thus, when agents themselves can strategically select between recourse and manipulation, the increased model transparency, created by providing more agents with recourse actions, results in more agents selecting to perform manipulation. Providing more recourse actions to agents, does not necessarily result in more agents performing recourse.

Despite this general trend, we also observe the effectiveness of subsidies. As subsidies converge to 1 (meaning recourse carries no cost), the fraction of agents choosing recourse converges to 1, while the fraction of agents choosing manipulation



(a) Law

(b) Adult

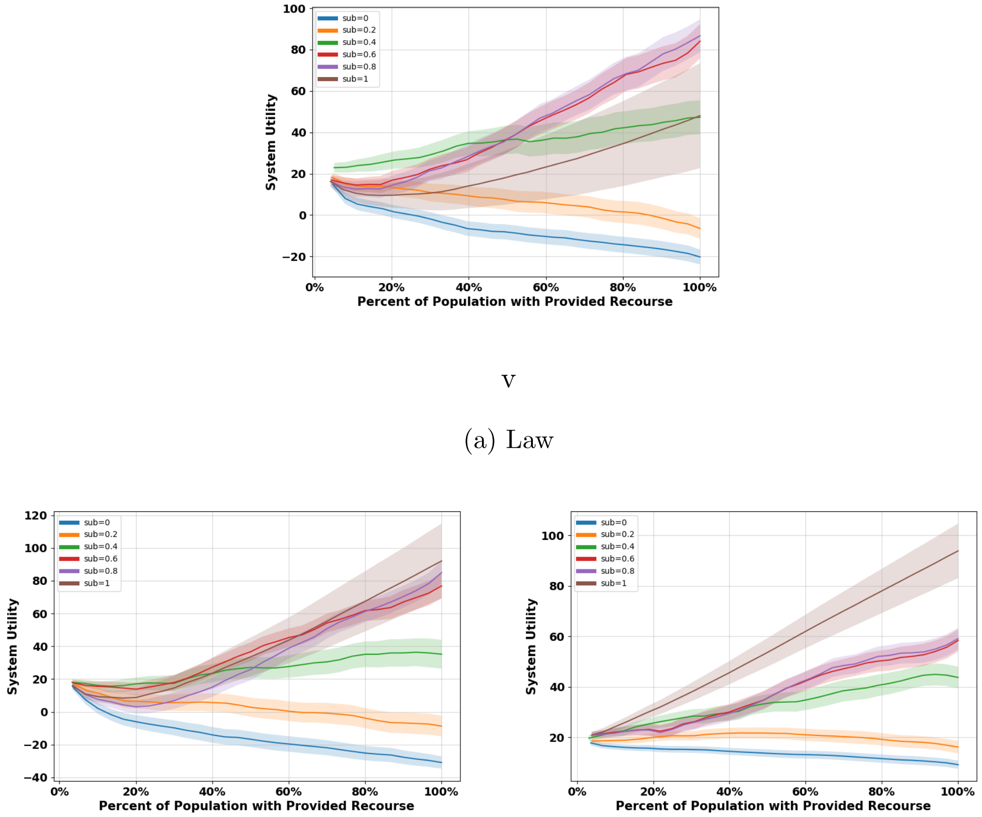
(c) Credit

Figure 5.2: Fraction of the population performing manipulation, with 95% confidence intervals. Each line corresponds to a different subsidy ratio “subs”, i.e., the cost reduction applied to recourse.

converges to 0. While it may be expensive in general to provide such subsides, and the question of how balance this expense against the system’s own utility remains open, these results indicate that subsides are an effective avenue for broadly promoting recourse and disincentivizng manipulation.

Social Cost and System Utility In Figure 5.4 and 5.1, we see both social cost and system utility as a function of the fraction of revealed features. As was the case with the recourse rate and manipulation rate, social cost has a roughly monotonic relationship with both the percentage of revealed features as well as the subsidy strength. This aligns with expectations as cost paid by agents should monotonically decrease with the number of possible recourse and manipulation actions. However, we also see that for most subsidy tradeoffs, system utility is monotonically decreasing, implying that the utility of the population (measured in terms of social cost) is fundamentally at odds with system utility in a wide range of cases. In particular, for subsidy scaling of 0.6 or greater, the system would prefer to *not provide a single agent with recourse* (since system utility is monotonically decreasing). Note that such an action from the system also corresponds to the highest possible social cost for each subsidy. Without strong enough subsidy tradeoffs, the system is incentivized to provide 0 recourse, resulting in the maximum possible harm to the population. However, for subsidy scaling of 0.4 and lower, system utility becomes loosely quadratic, with a maximum between 10% and 25%. In such cases, the system is incentivized to reveal a much larger portion of recourse actions to the population, resulting in overall lower social cost.

Disparity in Recourse and Social Cost Lastly we investigate the way in which strategic system behavior causes disparate impact among sensitive groups. In our experiments, groups are taken to be binary and are defined by race in the Law School dataset (White and Non-White), by gender in the Adult Income dataset (Male and Female), and by age in the German Credit dataset (Young and Old). In Figure 5.5, we see the difference in the amount of agents performing recourse, and in Figure 5.6 and the difference in social cost between groups. Higher values in these plots indicate a lower costs, or higher rates of recourse, for White individuals in the Law School dataset, Male individuals in the Adult income dataset, and Young individuals in the Credit dataset. First, we see that strong subsidies (particularly



v

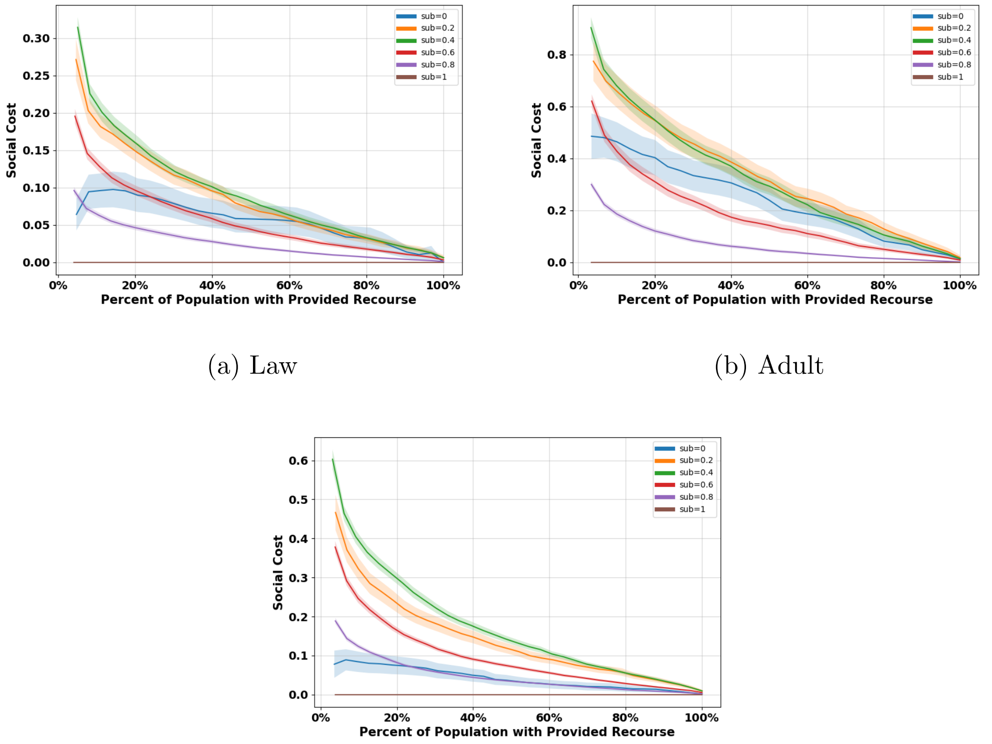
(a) Law

(b) Adult

(c) Credit

Figure 5.3: The system's utility as a function of the population percentage with provided recourse, with 95% confidence intervals. Each line corresponds to a different subsidy ratio "subs", i.e., the cost reduction applied to recourse.

subs ≤ 0.4) result in a large decrease in the disparities between groups for both recourse rate and social cost. For less strong subsidies (subs ≥ 0.6) we see that the gap in recourse rate between groups can increase. This is due to the fact that when subsidies are less strong, only agents with already low costs of recourse (which primarily come from the advantaged group) can benefit from those subsidies. When the percentage of revealed features is close to 0%, very few agents from either group have viable recourse actions; on the other hand, when large amounts of features are revealed (close to 100%), almost all agents from each group have viable recourse



(a) Law

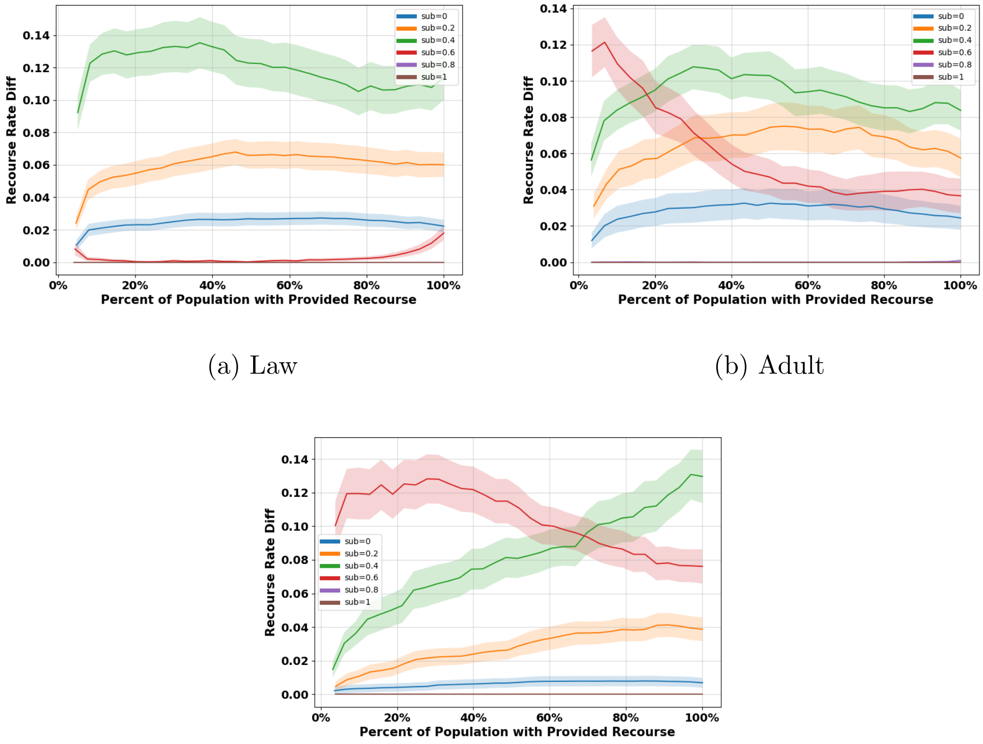
(b) Adult

(c) Credit

Figure 5.4: The social cost as a function of the population percentage with provided recourse, with 95% confidence intervals. Each line corresponds to a different subsidy ratio "subs", i.e., the cost reduction applied to recourse.

actions. While subsidies (particularly large subsidies) can help improve rates of disparity between groups, and always result in a lower total social cost, subsidies need not *always* decrease disparity between groups.

In Figure 5.7 we see the recourse rate difference between groups as a function of different values of subsidies. This figures serve to outline the parabolic nature relationship between subsidies and recourse rate difference. As mentioned previously, for smaller subsides, only those with already low recourse costs can benefit from subsidies. Thus we see that smaller subsidies can initially result in greater disparity between agents, however, as subsides increases, they eventually decrease



(a) Law

(b) Adult

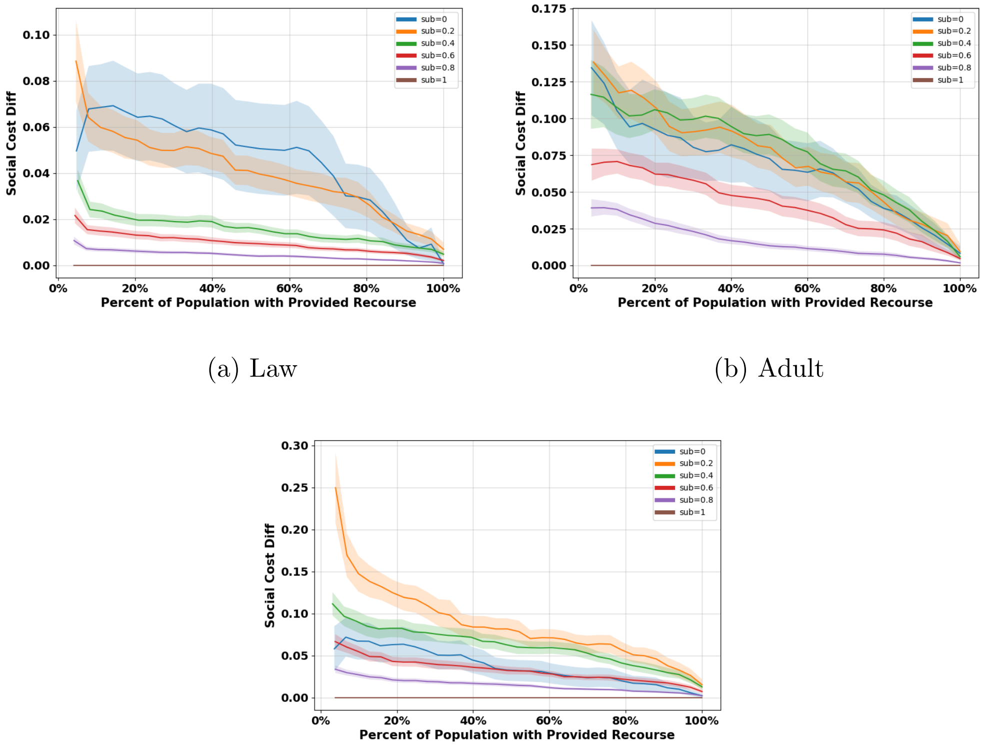
(c) Credit

Figure 5.5: Difference in recourse rate between different sensitive attribute groups with 95% confidence intervals. Each line corresponds to a different subsidy ratio "subs", i.e., the cost reduction applied to recourse.

disparity to rates which are lower than the disparity without subsidies ($sub = 0$). Thus when deciding the amount of subsidies to choose, it is important for systems to be aware of the potential negative impacts (larger disparities between groups) that can result from smaller subsidies.

5.8 Conclusion

When agents can manipulate a system, that system my have a decreased incentive to provide recourse as doing so increases model transparency, making manipulations easier. In such settings, a system seeking to preserver its own utility,



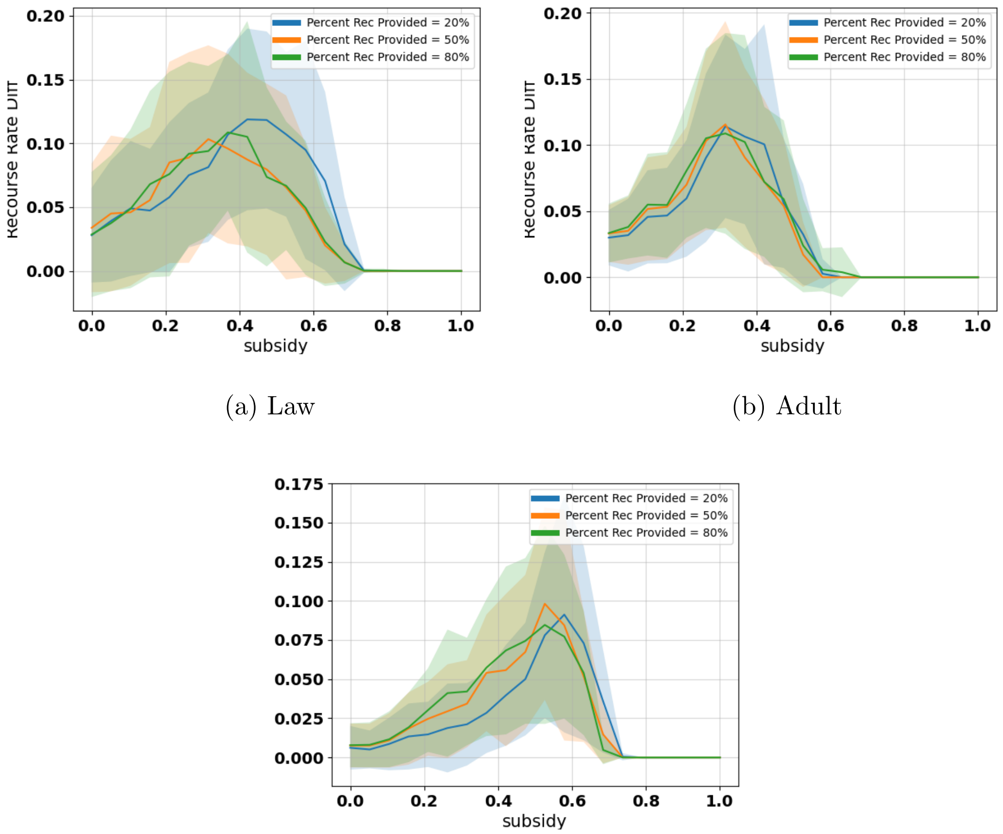
(a) Law

(b) Adult

(c) Credit

Figure 5.6: Difference in social cost between different sensitive attribute groups with 95% confidence intervals. Each line corresponds to a different subsidy ratio “subs”, i.e., the cost reduction applied to recourse.

will strategically withhold recourse from some (possibly all) individuals. This withholding of recourse in turn results in higher social cost to the population, and such costs can fall disproportionately on disadvantaged groups. We have demonstrated these relationships both theoretically and empirically. Despite the natural tension between the system’s utility and its desire to provide recourse, we showed that subsidies can be used as a useful tool to not only increase the rate at which the system provides recourse actions, but also to decrease the group-wise disparities caused by the system withholding recourse. Our work demonstrates that the presumption that a system will provide all individuals with recourse is unreasonable



(a) Law

(b) Adult

(c) Credit

Figure 5.7: Recourse rate difference as a function of subsidy with 95% confidence intervals. Each line corresponds to a different percentage of the population with provided recourse actions.

in cases when the system is self-interested (such as lending intuitions). However, the upshot is that mitigation techniques such as subsidies can be used to improve the rates at which self-interested systems provide recourse.

Chapter 6

Metric-Fair Classifier Derandomization

In this chapter, we study the problem of *classifier derandomization* in machine learning: given a stochastic binary classifier $f : X \to [0, 1]$, sample a deterministic classifier $\hat{f} : X \to \{0, 1\}$ that approximates the output of f in aggregate over any data distribution. Recent work revealed how to efficiently derandomize a stochastic classifier with strong output approximation guarantees, but at the cost of individual fairness — that is, if f treated similar inputs similarly, $\hat{f}$ did not. In this paper, we initiate a systematic study of classifier derandomization with metric fairness guarantees. We show that the prior derandomization approach is almost maximally metric-unfair, and that a simple "random threshold" derandomization achieves optimal fairness preservation but with weaker output approximation. We then devise a derandomization procedure that provides an appealing tradeoff between these two: if f is α-metric fair according to a metric d with a locality-sensitive hash (LSH) family, then our derandomized $\hat{f}$ is, with high probability, $O(\alpha)$-metric fair and a close approximation of f. We also prove generic results applicable to all (fair and unfair) classifier derandomization procedures, including a bias-variance decomposition and reductions between various notions of metric fairness.

6.1 Classifier Derandomization Problem

We study the general problem of *derandomizing* stochastic classification models. Consider a typical binary classification setting defined by a feature space $X \subseteq \mathbb{R}^n$ and labels $\{0, 1\}$; we wish to devise a procedure that, given a *stochastic* or *randomized* classifier $f : X \to [0, 1]$, efficiently samples a *deterministic* classifier $\hat{f} : X \to \{0, 1\}$ from some family of functions $\mathscr{F}$, such that $\hat{f}$ preserves various qualities of f.

Stochastic classifiers arise naturally in both theory and practice. For example, they are frequently the solutions to constrained optimization problems encoding complex evaluation metrics [Narasimhan, 2018], group fairness [Grgić-Hlača et al., 2017, Agarwal et al., 2018], individual fairness [Dwork et al., 2012, Rothblum and Yona, 2018, Kim et al., 2018, Sharifi-Malvajerdi et al., 2019], and robustness to adversarial attacks [Pinot et al., 2019, Cohen et al., 2019, Pinot et al., 2020, Braverman and Garg, 2020]. Stochastic classifiers are also the natural result of taking an ensemble of individual classifiers [Dietterich, 2000, Grgić-Hlača et al., 2017].

However, they may be undesirable for numerous reasons: a stochastic classifier is not robust to repeated attacks, since even one that is instance-wise 99% accurate will likely err after a few hundred attempts; by the same token, they violate intuitive notions of fairness since even the *same* individual may be treated differently over multiple classifications. For these reasons, Cotter, Gupta, and Narasimhan Cotter et al. [2019a] recently presented a procedure for derandomizing a stochastic classifier while approximately preserving the outputs of f with high probability. However, the authors observe that their construction results in similar individuals typically being given very different predictions — in other words, it does not satisfy *individual fairness* — and ask whether it is possible to obtain a family of deterministic classifiers that preserves both aggregate outputs and individual fairness.

Another motivation for studying individually fair decision making comes from the game-theoretic setting of *strategic classification*, wherein decision subjects may modify their features to obtain a desired outcome from the classifier [Hardt et al., 2016a, Cai et al., 2015, Chen et al., 2018, Dong et al., 2018b, Chen et al., 2020b]. A metric-fair stochastic classifier — and by extension, a metric-fair derandomization procedure — offers significant protection against such manipulations. See Appendix D.2 for more on this topic.

6.1.1 Our Contributions

In this paper, we initiate a systematic study of classifier derandomization with individual fairness preservation. In line with many recent works, we formalize individual fairness as *metric fairness*, which requires the classifier to output similar predictions on close point pairs in some metric space (X, d) Dwork et al. [2012], Kim et al. [2018], Friedler et al. [2016]. Roughly, f is metric-fair if there are constants $\alpha, \beta > 0$ such that for all $x, x' \in X$,

$$\left|f(x) - f(x')\right| \leq \alpha \cdot d(x, x') + \beta$$

A sampled deterministic classifier $\hat{f} \sim \mathscr{F}$ is metric-fair when this inequality holds in expectation.

Under this formalism, we obtain the following results:

1. We make precise the observation of Cotter et al. [2019a] that their derandomization procedure, based on pairwise-independent hash functions, does not preserve individual fairness. In fact, we prove that it is almost *maximally* metric-unfair regardless of how fair the original stochastic classifier was (Section 6.2.1).
2. We demonstrate that a very simple derandomization procedure, based on setting a single random threshold $r \sim [0, 1]$, attains near-perfect expected fairness preservation, and prove that no better fairness preservation is possible (Section 6.2.2). However, this procedure's output approximation has higher variance than the pairwise-independent hashing approach in general.
3. We devise a derandomization procedure that achieves nearly the best of both worlds, preserving aggregate outputs with high probability, with only modest loss of metric fairness (Section 6.3). In particular, when f has fairness parameters (α, β), sampling $\hat{f}$ from our family $\mathscr{F}_{\mathsf{LS}}$ yields expected fairness parameters at most $(\alpha + \frac{1}{2}, \beta + \epsilon)$. We also show a high-probability aggregate fairness guarantee: *most* deterministic classifiers in $\mathscr{F}$ assign *most* close pairs the same prediction. These guarantees hold for the class of metrics d that possess locality-sensitive hashing (LSH) schemes, which includes a wide variety of generic and data-dependent metrics.
4. We prove structural lemmas applicable to all classifier derandomization procedures: first, a bias-variance decomposition for the error of a derandomization $\hat{f}$ of f; second, a set of reductions showing that metric fairness-preserving derandomizations also preserve notions of *aggregate* and *threshold* fairness.

A practically appealing aspect of our LSH-based derandomization method is that it is completely oblivious to the original stochastic classifier, in that it requires no knowledge of how f was trained, and its fairness guarantee holds for whatever fairness parameters f happens to satisfy on each pair $(x, x') \in X^2$. The technique can therefore be applied as an independent post-processing step — for example, on the many fair stochastic classifiers detailed in recent works Rothblum and Yona [2018], Kim et al. [2018]. The burden on the model designer is thus reduced to selecting an LSHable metric feature space (X, d) that is appropriate for the classification task.

6.1.2 Preliminaries

Given a stochastic classifier $f : X \to [0, 1]$ and distance function $d : X \times X \to [0, 1]$, we wish to design an efficiently sampleable set $\mathscr{F}$ of deterministic binary classifiers $\hat{f} : X \to \{0, 1\}$; we call $\mathscr{F}$ a *family of deterministic classifiers*, or a *derandomization of* f. Moreover, we would like $\mathscr{F}$ to have the following properties:

Output approximation: $\hat{f}$ sampled uniformly[1] from $\mathscr{F}$ simulates or approximates f in aggregate over any distribution. More precisely, define the *pointwise* bias and variance of $\hat{f}$ with respect to f on a sample $x \in X$ as

$$\mathsf{Bias}(\hat{f}, f, x) := \mathbb{E}_{\hat{f}\sim\mathscr{F}}\left[\hat{f}(x)\right] - f(x) \qquad \text{and} \qquad \text{variance}(\hat{f}, x) := \mathrm{Var}_{\hat{f}\sim\mathscr{F}}\left(\hat{f}(x)\right)$$

Now let $\mathscr{D}$ be a distribution over X. The *aggregate* bias and variance of $\hat{f}$ with respect to f on $\mathscr{D}$ are

$$\mathsf{Bias}(\hat{f}, f, \mathscr{D}) := \mathbb{E}_{x\sim\mathscr{D}}\left[\mathsf{Bias}(\hat{f}, f, x)\right],$$
$$\text{variance}(\hat{f}, \mathscr{D}) := \mathrm{Var}_{\hat{f}\sim\mathscr{F}}\left(\mathbb{E}_{x\sim\mathscr{D}}\left[\hat{f}(x)\right]\right)$$

[1] In this paper, we will always sample uniformly from families of classifiers and hash functions; thus $\hat{f} \sim \mathscr{F}$ means $\hat{f} \sim \mathrm{Unif}(\mathscr{F})$, and $h \sim \mathscr{H}$ means $h \sim \mathrm{Unif}(\mathscr{H})$.

We seek a family $\mathscr{F}$ for which both of these quantities are small. This is a useful notion of a good approximation of f since in practice, classifiers are typically applied *in aggregate* on some dataset or in deployment. In Section 6.4.4 we also point out that low bias and variance in the above sense implies that $\hat{f}$ and f are nearly indistinguishable when compared according to any binary loss functions, such as accuracy, false positive rate, etc.

Individual fairness: Similar individuals are likely to be treated similarly. We formally define this notion as *metric fairness*, which says that that the classifier should be an approximately Lipschitz-continuous function relative to a given distance metric:

Definition 6.1.1 ((α, β, d)-metric fairness)**.** *Let* $\alpha \geq 1$[2] *and* $\beta \geq 0$*, let* $d : X^2 \to [0, 1]$ *be a metric, and let* $x, x' \in X$*. We say a stochastic classifier* $f : X \to [0, 1]$ *satisfies* (α, β, d)-metric fairness on (x, x')*, or is* (α, β, d)-fair on (x, x')*, if*

$$\left|f(x) - f(x')\right| \leq \alpha \cdot d(x, x') + \beta \tag{6.1}$$

Similarly, a deterministic classifier family $\mathscr{F}$ *is* (α, β, d)*-fair on* (x, x') *if*

$$\mathbb{E}_{\hat{f} \sim \mathscr{F}} \left[\left|\hat{f}(x) - \hat{f}(x')\right|\right] \leq \alpha \cdot d(x, x') + \beta \tag{6.2}$$

When this condition is satisfied for all $(x, x') \in X^2$*, we simply say the classifier (or family) is* (α, β, d)-fair*.*

To intuit this definition, notice that when a classifier satisfies metric fairness with $\beta = 0$, the difference between its predictions on some pair of points x and

[2] We enforce $\alpha \geq 1$, and not merely $\alpha \geq 0$, so that the codomain of f is $[0, 1]$ rather than potentially $[0, \alpha]$ (or some other interval of length $\alpha < 1$). Requiring $\alpha \geq 1$ thus makes f a proper stochastic classifier and enables direct comparisons between different fairness parameters. This is no loss of generality since (α, β, d)-fairness for $\alpha < 1$ can also be expressed as $(1, \frac{\beta}{\alpha}, \frac{d}{\alpha})$-fairness or, with some loss of generality, $(1, \beta + \alpha, d)$-fairness.

x' scales in proportion to their distance. To conform to this idea of fairness, it is important that the derandomization procedures we design do not substantially increase these fairness parameters, but especially β.

The above definition of metric fairness is most closely related to those of Rothblum and Yona Rothblum and Yona [2018], whose focus is learning a "probably approximately metric-fair" model that generalizes to unseen data; and Kim, Reingold, and Rothblum Kim et al. [2018], whose focus is in-sample learning when the metric d is not fully specified. Both works take inspiration from the metric-based notion of individual fairness introduced in Dwork et al. [2012]. Crucially however, the aforementioned works provide guarantees exclusively for stochastic classifiers, and to our knowledge, this is the case for all papers to date whose focus is learning metric-fair classifiers.

In addition to this pairwise notion of metric fairness, we will also develop *aggregate* fairness guarantees for various derandomization procedures. To that end, let

$$X^2_{\leq\tau} := \left\{(x, x') \in X^2 \;\middle|\; d(x, x') \leq \tau\right\}$$

denote the set of point pairs within some distance $\tau \in [0, 1]$. Our aggregate fairness bounds will state that, with high probability over the sampling of $\hat{f} \sim \mathscr{F}$, most pairs $(x, x') \in X^2_{\leq\tau}$ receive the same prediction from $\hat{f}$.

6.2 Output Approximation Versus Fairness

We begin our study of metric-fair classifier derandomization by contrasting two approaches: first, the "pairwise-independent" derandomization of Cotter et al. [2019a], which achieves a low-variance approximation of the original stochastic classifier, but does not preserve metric fairness; and second, a simple "random threshold" derandomization that perfectly preserves metric fairness, at the cost of higher output variance.

6.2.1 Pairwise-Independent Derandomization

The construction of Cotter, Narasimhan, and Gupta Cotter et al. [2019a] makes use of a pairwise-independent hash function family $\mathscr{H}_{\mathsf{PI}}$, i.e. a set of functions $h_{\mathsf{PI}} : B \to [k]$ such that

$$\Pr_{h\sim\mathscr{H}_{\mathsf{PI}}}[h(b) = i, h(b') = j] = \frac{1}{k^2} \quad \forall b \neq b' \in B, \quad i, j \in [k]$$

Observe that a family that satisfies this property is also uniform, i.e. $\Pr_{h\sim\mathscr{H}_{\mathsf{PI}}}[h(b) = i] = 1/k$ for all b, i.

The classifier family they propose is then[3]

$$\mathscr{F}_{\mathsf{PI}} := \left\{ \hat{f}_{h_{\mathsf{PI}}} \;\middle|\; h_{\mathsf{PI}} \in \mathscr{H}_{\mathsf{PI}} \right\}, \quad \text{where} \quad \hat{f}_{h_{\mathsf{PI}}}(x) := \mathbb{1}\left\{ f(x) \geq \frac{h_{\mathsf{PI}}(\pi(x))}{k} \right\} \tag{6.3}$$

where $\pi : X \to B$ is some fixed *bucketing* function that discretizes the input (since the pairwise-independent hash family has finite domain).

Let us develop some intuition for this construction. First, thinking of k as large, each $\hat{f}_{h_{\mathsf{PI}}} \in \mathscr{F}_{\mathsf{PI}}$ essentially assigns a pseudo-random threshold $\frac{h_{\mathsf{PI}}(\pi(x))}{k} \in [0, 1]$ to each input x, so that $\hat{f}(x) = 1$ if and only if $f(x)$ exceeds the threshold. Since h_{PI} is a uniform hash function family, $h_{\mathsf{PI}}(\pi(x))$ is uniform over $[k]$; this endows $\mathscr{F}_{\mathsf{PI}}$ with low bias with respect to f. Using this idea and the pairwise-independence of $\mathscr{H}_{\mathsf{PI}}$, the authors show that this classifier family exhibits low bias and variance of approximation:

[3] For the sake of clearer exposition, we simplify the deterministic classifier used in Cotter et al. [2019a], which is actually $\hat{f}_{h_{\mathsf{PI}}}(x) := \mathbb{1}\{f(x) \geq \frac{2h_{\mathsf{PI}}(x)-1}{2k}\}$; this does not change Theorem 6.2.1 or Proposition 6.2.2 beyond a $1/2k$ additive difference in the bias, variance, and β.

Theorem 6.2.1 (Bias and variance of pairwise-independent derandomization Cotter et al. [2019a] (simplified))**.** *Let f be a stochastic classifier, $\mathscr{D}$ a distribution over X, and $\pi : X \to B$ a bucketing function. Then $\hat{f} \sim \mathscr{F}_{\mathsf{PI}}$ satisfies*

$$\mathit{Bias}(\hat{f}_{\mathsf{PI}}, f, \mathscr{D}) \leq \frac{1}{k},$$

$$\text{variance}(\hat{f}_{\mathsf{PI}}, f, \mathscr{D}) \leq \max_{b \in B} \Pr_{x \sim \mathscr{D}}[\pi(x) = b] \cdot \mathbb{E}_{x \sim \mathscr{D}}[f(x)(1 - f(x))] + \frac{1}{k}$$

Moreover, $\hat{f}_{\mathsf{PI}}$ can be sampled using $O(\log |B| + \log k)$ uniform random bits.

To understand this variance bound, observe that for a given data distribution $\mathscr{D}$, the bound is stronger or weaker depending on how well π disperses samples into different buckets in B. When there exists some $b \in B$ such that $\Pr_{x \sim \mathscr{D}}[\pi(x) = b] \approx 1$, $\text{variance}(\hat{f}_{\mathsf{PI}}, f, \mathscr{D}) \approx \mathbb{E}_{x \sim \mathscr{D}}[f(x)(1 - f(x))]$ essentially tracks the stochasticity of f. At the other extreme when $\Pr_{x \sim \mathscr{D}}[\pi(x) = b] = 1/|B|$ for all $b \in B$, $\text{variance}(\hat{f}_{\mathsf{PI}}, f, \mathscr{D}) \approx 1/|B|$.

As the authors pointed out (but did not formalize), $\hat{f}_{\mathsf{PI}}$ does not preserve pairwise fairness in general. We make this observation precise by showing that it is always possible to design a dataset, of any desired size, such that the pairwise-independent derandomization treats *every* pair of points unfairly for nearly any $\beta < 1/2$.

Proposition 6.2.2 (Unfairness of pairwise-independent derandomization)**.** *For every $N \geq 2$, $\alpha \geq 1$, $\beta < \frac{1}{2} - \frac{1}{2k}$, and metric $d : \mathbb{R}^n \times \mathbb{R}^n \to [0, 1]$, there exist a set $X \subset \mathbb{R}^n$ of size N and stochastic classifier $f : X \to [0, 1]$ such that the following hold:*

1. f is nontrivial and $(1, 0, d)$-fair.

2. $\mathscr{F}_{\mathsf{PI}}$ violates (α, β, d)-metric fairness for every *pair $(x, x') \in X^2, x \neq x'$.*

If k is not too small, this says that derandomizing using pairwise-independent hashing is almost maximally unfair, as a uniform random binary function $\hat{g} : X \to \{0, 1\}$ satisfies $\mathbb{E}[|\hat{g}(x) - \hat{g}(x')|] = 1/2$, and therefore achieves $\beta = 1/2$.

Proof sketch of Proposition 6.2.2. Consider any $\alpha \geq 1$, $\beta \in \left(0, \frac{1}{2} - \frac{1}{2k}\right)$, and $N \geq 2$. We choose X to be some set of N points on a sufficiently small sphere about the origin, and let f be a classifier that maps half of the points in X to $\frac{1+\epsilon}{2}$ and the other half to $\frac{1-\epsilon}{2}$. When $\epsilon > 0$ is sufficiently small, it can be shown that f is $(1, 0, d)$-fair over X. However, $\mathscr{F}_{\mathsf{PI}}$ is not (α, β, d)-fair on any point pair $(x, x') \in X^2$. The reason is that since f is almost maximally stochastic (i.e. $f(x) \approx 1/2$ for all x), and $\mathscr{H}_{\mathsf{PI}}$ is pairwise-independent, the binary outputs $\hat{f}(x)$ and $\hat{f}(x')$ are about as likely to be the same as they are likely to be different. Hence $\mathbb{E}_{\hat{f} \sim \mathscr{F}_{\mathsf{PI}}}[|\hat{f}(x) - \hat{f}(x')|] \approx 1/2$, violating (α, β, d)-metric fairness. See Appendix D.1.1 for the full proof. □

6.2.2 Random Threshold Classifier

It turns out that there is a near-trivial derandomization that achieves optimal preservation of metric fairness, namely the following *random threshold* classifier family:

$$\mathscr{F}_{\mathsf{RT}} := \{\hat{f}_r \mid r \in [0, 1]\}, \text{ where } \hat{f}_r := \mathbb{1}\{f(x) \geq r\} \tag{6.4}$$

Formally we make the following observation, whose proof is in Appendix D.1.2.

Proposition 6.2.3 (Random threshold derandomization guarantees)**.** *Let f be an (α, β, d)-fair stochastic classifier and $\mathscr{D}$ a distribution over X. Then the deterministic classifier family $\mathscr{F}_{\mathsf{RT}}$ is also (α, β, d)-fair. Moreover,*

$$\mathit{Bias}(\hat{f}_{\mathsf{RT}}, f, \mathscr{D}) = 0 \qquad \text{and} \qquad \text{variance}(\hat{f}_{\mathsf{RT}}, f, \mathscr{D}) \leq \mathbb{E}_{x \sim \mathscr{D}}[f(x)(1 - f(x))]$$

Note that while this derandomization preserves the original fairness parameters perfectly, its variance can be substantially higher than that of $\mathscr{F}_{\mathsf{PI}}$ depending on the choice of bucketing function π in Equation (6.3).

One subtlety here is that $\mathscr{F}_{\mathsf{RT}}$ is an infinite set, and is therefore not sampleable in practice. For the more realistic scenario in which the threshold r is a number of some fixed precision $\epsilon > 0$, the statements in Proposition 6.2.3 hold up to additive error ϵ, and $\hat{f}_{\mathsf{RT}}$ can be sampled using $O(\log(1/\epsilon))$ uniform random bits. In this case $\mathscr{F}_{\mathsf{RT}}$ is $(\alpha, \beta + \epsilon, d)$-fair, and as we can show, this is in fact necessary:

Proposition 6.2.4 ($(\alpha, 0, d)$-metric fairness is impossible for finite deterministic families)**.** *Let $d : X \times X \to [0, 1]$ be a metric over a convex set $X \subseteq \mathbb{R}^n$, and let $\mathscr{F}$ be a finite family of deterministic classifiers, at least one of which is nontrivial. Then for every $\alpha \geq 1$ and $\beta < 1/|\mathscr{F}|$, $\mathscr{F}$ is not (α, β, d)-fair.*

Proof sketch. Since $\mathscr{F}$ contains a nontrivial classifier $\hat{f}$, we can pick sufficiently close points around a discontinuity of $\hat{f}$ and show that in expectation, $\mathscr{F}$ fails to achieve roughly $(\alpha, 1/|\mathscr{F}|, d)$-fairness on this point pair. See Appendix D.1.3 for details. □

The main consequence is that there is an irreducible amount of additive unfairness $\beta > 0$ that cannot be avoided when constructing a fair deterministic classifier family. Indeed, the derandomization $\mathscr{F}$ we present in Section 6.3 has $|\mathscr{F}| \geq 1/\beta$, thus avoiding the impossible regime indicated by Proposition 6.2.4.

6.3 Fair Derandomization via Locality-Sensitive Hashing

In this section, we construct a deterministic classifier family that combines much of the appeal of both the pairwise-independent derandomization (low output variance) and the random threshold derandomization (strong fairness preservation). This new approach utilizes two types of hashing: first, a pairwise-independent hash family $\mathscr{H}_{\mathsf{PI}}$ as before; and second, a locality-sensitive hash family:[4]

Definition 6.3.1 (Locality-sensitive hash (LSH) family)**.** *Let X be a set of hashable items, B a set of buckets, and $d : X^2 \to [0,1]$ a metric distance function. We say a set $\mathscr{H}_{\mathsf{LS}}$ of functions $h : X \to B$ is a* locality-sensitive family of hash functions *for d if for all $x, x' \in X$,*

$$\Pr_{h \sim \mathscr{H}_{\mathsf{LS}}} \left[h(x) \neq h(x') \right] = d(x, x')$$

Locality-sensitive hashing is a well-studied technique, and LSH families have been constructed for many standard distances and similarities, such as L_1 Indyk and Motwani [1998], L_2 Andoni and Indyk [2006], cosine Charikar [2002], Jaccard Broder [1997], various data-dependent metrics Jain et al. [2008], Andoni et al. [2014], Andoni and Razenshteyn [2015], and more.

Our derandomization works as follows: suppose $f : X \to [0,1]$ is a stochastic classifier, $\mathscr{H}_{\mathsf{LS}}$ is a family of locality-sensitive hash functions $h_{\mathsf{LS}} : X \to B$, and $\mathscr{H}_{\mathsf{PI}}$ is a family of pairwise-independent hash functions $h_{\mathsf{PI}} : B \to [k]$ for some positive

[4] We use the definition of LSH as coined by Charikar Charikar [2002]. See Indyk and Motwani [1998] for an alternative gap-based definition in the same spirit.

integer k. Our family of deterministic classifiers is then

$$\mathscr{F}_{\mathsf{LS}} := \left\{ \hat{f}_{h_{\mathsf{LS}}, h_{\mathsf{PI}}} \;\middle|\; h_{\mathsf{LS}} \in \mathscr{H}_{\mathsf{LS}}, h_{\mathsf{PI}} \in \mathscr{H}_{\mathsf{PI}} \right\}, \tag{6.5}$$

$$\text{where} \quad \hat{f}_{h_{\mathsf{LS}}, h_{\mathsf{PI}}}(x) := \mathbb{1}\left\{ f(x) \geq \frac{h_{\mathsf{PI}}(h_{\mathsf{LS}}(x))}{k} \right\}. \tag{6.6}$$

Let us develop some intuition for this construction. First, thinking of k as large, each $\hat{f} \in \mathscr{F}_{\mathsf{LS}}$ essentially assigns a pseudo-random threshold $\frac{h_{\mathsf{PI}}(h_{\mathsf{LS}}(x))}{k} \in [0, 1]$ to each input x, so that $\hat{f}(x) = 1$ if and only if $f(x)$ exceeds the threshold. Since the outer hash function h_{PI} is pairwise-independent, and therefore also uniform, $h_{\mathsf{PI}}(h_{\mathsf{LS}}(\cdot))$ is uniform over $[k]$. This endows $\mathscr{F}_{\mathsf{LS}}$ with low bias and variance with respect to f, as we explain in Section 6.3.1.

Second, the composition of two different hash functions gives us our fairness guarantee: h_{LS} maps close point pairs x, x' to the same bucket, then h_{PI} disperses pairs that were not hashed together — most of which are distant. This separation of point pairs by distance is precisely what enables good preservation of metric fairness, as we prove in Section 6.3.2.

6.3.1 Approximation of Outputs

We show the following bounds on the bias and variance of our derandomization. The proof is deferred to Appendix D.1.4.

Theorem 6.3.2 (Bias and variance of derandomized classifier)**.** *Let f be a stochastic classifier, $\hat{f} \sim \mathscr{F}_{\mathsf{LS}}$, and $\mathscr{D}$ a distribution over X. Then*

$$\mathit{Bias}(\hat{f}, f, \mathscr{D}) \leq \frac{1}{k}, \textit{ and}$$

$$\text{variance}(\hat{f}, f, \mathscr{D}) \leq \mathbb{E}_{h_{\mathsf{LS}} \sim \mathscr{H}_{\mathsf{LS}}} \left[\max_{b \in B} \Pr_{x \sim \mathscr{D}} [h_{\mathsf{LS}}(x) = b] \right] \cdot \mathbb{E}_{x \sim \mathscr{D}}[f(x)(1 - f(x))] + \frac{1}{k}.$$

The above variance bound is similar in form to that of the pairwise-independent derandomization (Theorem 6.2.1), but with added randomization over the sampling of locality-sensitive hash function: when most choices of h_{LS} distribute points $x \sim \mathscr{D}$ into buckets relatively evenly, the bound is as small as $O(1/|B|)$; when most hashes are collisions, the bound may be as large as $\mathbb{E}_{x\sim\mathscr{D}}[f(x)(1-f(x))]$, essentially tracking the stochasticity of f.

6.3.2 Preservation of Metric Fairness

We can now show that our derandomization procedure approximately preserves metric fairness, both in the sense of expected fairness for any pair of points (the usual convention in the metric fairness literature), as well as in aggregate over all point pairs.

Theorem 6.3.3 (Locality-sensitive derandomization preserves metric fairness)**.** *Let f be an (α, β, d)-fair stochastic classifier, where d is a metric with an LSH family $\mathscr{H}_{\mathsf{LS}}$ with $k \geq 2/\epsilon$ buckets. Then $\mathscr{F}_{\mathsf{LS}}$ is a deterministic classifier family satisfying the following:*

- *(Pairwise fairness) Consider any $x, x' \in X$, and assume without loss of generality that $f(x) \leq f(x')$. Then*

$$\mathbb{E}_{\hat{f}\sim\mathscr{F}_{\mathsf{LS}}}\left[\left|\hat{f}(x) - \hat{f}(x')\right|\right] \leq [\alpha + 2f(x)(1-f(x'))]\cdot d(x,x') + \beta + \epsilon$$

- *(Aggregate fairness) For any distance threshold $\tau \in [0,1]$, with probability at least $1-\delta$ over the sampling of $\hat{f}$,*

$$\Pr_{(x,x')\sim X^2_{\leq\tau}}\left[\hat{f}(x) \neq \hat{f}(x')\right] \leq \left(1 + \frac{1}{\sqrt{\delta}}\right)([\alpha + 2f(x)(1-f(x'))]\cdot\tau + \beta + \epsilon).$$

The above fairness guarantees can be simplified by noticing that since $f(x) \leq f(x')$ w.l.o.g., $f(x)(1-f(x')) \leq 1/4$; this yields the following worst-case bounds over f and (x, x'):

Corollary 6.3.4 (Worst-case fairness)**.** *When f is (α, β, d)-fair, $\mathscr{F}_{\mathsf{LS}}$ satisfies the following:*

- *(Pairwise fairness) $\left(\alpha + \frac{1}{2}, \beta + \epsilon, d\right)$-metric fairness on any $(x, x') \in X^2$, i.e.*

$$\mathbb{E}_{\hat{f} \sim \mathscr{F}_{\mathsf{LS}}}\left[\left|\hat{f}(x) - \hat{f}(x')\right|\right] \leq \left(\alpha + \frac{1}{2}\right) \cdot d(x, x') + \beta + \epsilon.$$

- *(Aggregate fairness) For any distance threshold $\tau \in [0, 1]$, with probability at least $1 - \delta$ over the sampling of $\hat{f}$,*

$$\Pr_{(x,x') \sim X^2_{\leq \tau}}\left[\hat{f}(x) \neq \hat{f}(x')\right] \leq \left(1 + \frac{1}{\sqrt{\delta}}\right)\left(\alpha\tau + \frac{\tau}{2} + \beta + \epsilon\right).$$

In expectation and with high probability, therefore, the generated deterministic classifier approximates the fairness guarantee of the original classifier to within a small constant factor when there exists an LSH family $\mathscr{H}$ for d. To get a better sense what kind of guarantees this gives us, consider the following example:

Example 1. *Let f be a $(1, 0, d)$-fair stochastic classifier, and suppose we derandomize it to some $\hat{f} \sim \mathscr{F}_{\mathsf{LS}}$, choosing $k = 500$. Then by Corollary 6.3.4,*

- *(Pairwise fairness) $\hat{f}$ is $(3/2, \epsilon, d)$-metric fair.*
- *(Aggregate fairness) With probability at least $1 - \delta = 3/4$ (over the sampling of $\hat{f}$), at least 76% of point pairs within distance $\tau = 1/20$ receive identical predictions.*

We present a sketch of the proof of Theorem 6.3.3; see Appendix D.1.5 for the complete proof.

Proof sketch of Theorem 6.3.3. Consider any $x, x' \in X$. Since $\hat{f}$ is binary and $\mathscr{H}_{\mathsf{LS}}$

is locality-sensitive,

$$\begin{aligned}\mathbb{E}_{\hat{f}\sim\mathscr{F}_{\mathsf{LS}}}\left[\left|\hat{f}(x)-\hat{f}(x')\right|\right] &= \Pr_{\substack{h_{\mathsf{LS}}\sim\mathscr{H}_{\mathsf{LS}}\\ h_{\mathsf{PI}}\sim\mathscr{H}_{\mathsf{PI}}}}\left[\hat{f}(x)\neq\hat{f}(x')\right]\\ &= \Pr_{h_{\mathsf{LS}},h_{\mathsf{PI}}}\left[\hat{f}(x)\neq\hat{f}(x')\;\middle|\;h_{\mathsf{LS}}(x)=h_{\mathsf{LS}}(x')\right]\cdot(1-d(x,x'))\\ &\quad+\Pr_{h_{\mathsf{LS}},h_{\mathsf{PI}}}\left[\hat{f}(x)\neq\hat{f}(x')\;\middle|\;h_{\mathsf{LS}}(x)\neq h_{\mathsf{LS}}(x')\right]\cdot d(x,x')\end{aligned}$$

From here, the proof is a systematic analysis of conditional probabilities. To give some intuition, notice that the event $[\hat{f}(x)\neq\hat{f}(x')\mid h_{\mathsf{LS}}(x)=h_{\mathsf{LS}}(x')]$ occurs precisely when $\frac{h_{\mathsf{PI}}(h_{\mathsf{LS}}(x))}{k}$ falls between $f(x)$ and $f(x')$; by the uniformity of $\mathscr{H}_{\mathsf{PI}}$, the probability of this is roughly $|f(x)-f(x')|\leq\alpha\cdot d(x,x')+\beta$. This is one of several cases that use the uniformity and symmetry properties of the composed hash function $h_{\mathsf{PI}}(h_{\mathsf{LS}}(\cdot))$ to express $|\hat{f}(x)-\hat{f}(x')|$ in terms of $|f(x)-f(x')|$. In some cases this is not possible, resulting in an additive $2f(x)(1-f(x'))$ loss in α. □

6.3.3 Sample Complexity

Since the LSH-based derandomization procedure involves sampling two hash functions $\mathscr{H}_{\mathsf{PI}}$ and $\mathscr{H}_{\mathsf{LS}}$, it samples $\hat{f}$ using $O(\log|B|+\log k+S_d(X,B))$ random bits, where $O(\log|B|+\log k)$ is the number of bits used to sample a pairwise-independent hash function Rubinfeld [2012], and $S_d(X,B)$ is the number of random bits required to sample a locality-sensitive hash function for metric d with domain X and range B. When the metric is the Euclidean distance, for example, $O(\dim X)$ random bits suffice Rashtchian [2019].

6.4 Structural Lemmas for Fair Classifier Derandomization

In this section, we present generic results applicable to all classifier derandomization procedures, as well as unify different definitions of fairness used in this paper and others.

6.4.1 Bias-Variance Decomposition

Up to this point, a "stochastic" classifier has signified any function f from X to $[0,1]$; in this sense, it does not necessarily contain any randomness of its own. However, when it comes time to perform a binary decision on some input x, $f(x)$ is typically interpreted as the probability of outputting 1, i.e. we use the (truly random) binary function $\mathbb{1}_f(x) \sim \mathrm{Bern}(f(x))$.

By how much does this prediction typically differ from that of some pre-sampled deterministic classifier $\hat{f}$? We show that this error can be decomposed into the bias of $\hat{f}$ and the variance of both $\hat{f}$ and f:

Lemma 6.4.1 (Bias-variance decomposition)**.** *Let $f : X \to [0,1]$ be a stochastic classifier and $\mathscr{F}$ a deterministic classifier family. Then for any $x \in X$,*

$$\mathbb{E}_{f,\hat{f}}\left[\left|\hat{f}(x) - \mathbb{1}_f(x)\right|\right] \leq \left|\mathit{Bias}(\hat{f}, \mathbb{1}_f, x)\right| + 2\left(\mathrm{Var}_f(\mathbb{1}_f(x)) + \mathrm{Var}_{\hat{f}\sim\mathscr{F}}\left(\hat{f}(x)\right)\right)^{2/3}$$

We defer the proof to Appendix D.1.6. For now, let us interpret this decomposition and see how it applies to the derandomization approaches laid out in previous sections. Recall that for all three derandomizations — $\mathscr{F}_{\mathsf{PI}}$, $\mathscr{F}_{\mathsf{RT}}$, and $\mathscr{F}_{\mathsf{LS}}$ — the bias was either zero or could be made arbitrarily small. As for the variance, we see two types: the first, $\mathrm{Var}_f(\mathbb{1}_f(x))$, is equal to $f(x)(1-f(x))$, i.e. the variance of a Bernoulli with parameter $f(x)$; it therefore quantifies the inherent stochasticity of the given classifier f, over which we have no control. In contrast, the second variance arises from sampling the deterministic classifier $\hat{f}$, which depends greatly on the procedure being used. Thus a comparison of the expected error of these approaches boils down to this latter variance, for which the pairwise-independent and locality-sensitive hashing approaches compare favorably against the simple random threshold.

6.4.2 Metric Fairness and Threshold Fairness

Friedler, Scheidegger, and Venkatasubramanian Friedler et al. [2016] propose an alternative threshold-based notion of individual fairness that implements the mantra that "similar individuals should receive similar treatment," but only extends this constraint to pairs of inputs within a certain distance of interest:

Definition 6.4.2 ((σ, τ, d)-threshold fairness)**.** *Fix some constants* $\sigma, \tau \in (0, 1)$*. We say a stochastic classifier* f *is* (σ, τ, d)-threshold fair *if for all* $x, x' \in X$ *such that* $d(x, x') \leq \sigma$*, we have* $|f(x) - f(x')| \leq \tau$*. We say a deterministic classifier family* $\mathscr{F}$ *is* (σ, τ, d)*-threshold fair if for all* $x, x' \in X$ *such that* $d(x, x') \leq \sigma$*, we have* $\mathbb{E}_{\hat{f} \sim \mathscr{F}}[|\hat{f}(x) - \hat{f}(x')|] \leq \tau$*.*

Neither metric fairness nor threshold fairness fully subsumes the other. However, we can still show the following *algorithmic* reduction: if we wish to derandomize a stochastic classifier while preserving threshold fairness, then it suffices to use any procedure that preserves metric fairness. For example, suppose we have a derandomization procedure that worsens the input classifier's fairness parameters α and β to at most $a \cdot \alpha$ and $b \cdot \beta$, respectively, for some small constants $a, b \geq 1$.

We should also expect this procedure to preserve threshold fairness, within certain parameters related to a, b. This is what we prove in the following lemma, but for more general fairness preservation functions:

Lemma 6.4.3 (Metric-fair derandomization preserves threshold fairness)**.** *Suppose we have a procedure that, given an (α, β, d)-metric fair stochastic classifier f, samples a deterministic classifier $\hat{f}$ from an $(A(\alpha), B(\beta), d)$-metric fair family $\mathscr{F}$, for some functions $A, B : \mathbb{R} \to \mathbb{R}$. Then this same procedure also derandomizes any (σ, τ, d)-threshold fair stochastic classifier to a deterministic classifier from a $(\sigma, A(0) \cdot \sigma + B(\tau), d)$-threshold fair family.*

Applying this to the random threshold and locality-sensitive derandomization procedures yields the following:

Corollary 6.4.4 (Threshold fairness-preserving derandomizations)**.** *Let f be a (σ, τ, d)-threshold fair stochastic classifier. Then*

- *The family $\mathscr{F}_{\mathsf{RT}}$ is (σ, τ, d)-threshold fair.*
- *If d is LSHable, the family $\mathscr{F}_{\mathsf{LS}}$, for a choice of $k \geq 4/\sigma$, is $(\sigma, \sigma + \tau, d)$-threshold fair.*

The proofs are deferred to Appendix D.1.7.

6.4.3 Pairwise Fairness and Aggregate Fairness

Throughout most of this paper (and in most of the individual fairness literature), we have been focused on pairwise notion of fairness, such as metric fairness (Definition 6.1.1) and threshold fairness (Definition 6.4.2). One shortcoming of these definitions is that even if a classifier satisfies them for any particular pair of points (x, x'), they do not hold simultaneously for all input pairs; thus once we sample a specific deterministic classifier $\hat{f}$, it may be unfair for many pairs. Fortunately, as we now show, these pairwise statements imply high-probability aggregate fairness guarantees: if $\mathscr{F}$ is a metric-fair family, then *most* deterministic classifiers in $\mathscr{F}$ assign *most* close pairs the same prediction.

To that end, for all distances $\tau \in [0,1]$, let $X^2_{\leq\tau} := \left\{(x,x') \in X^2 \;\middle|\; d(x,x') \leq \tau\right\}$ denote the set of point pairs within distance τ. Then we can bound the fraction of τ-close pairs that receive different predictions:

Lemma 6.4.5 (Pairwise fairness implies aggregate fairness)**.** *Let $\mathscr{F}$ be an (α,β,d)-fair deterministic classifier family. Then for any distance threshold $\tau \in [0,1]$, with probability at least $1-\delta$ over the sampling of $\hat{f} \sim \mathscr{F}$,*

$$\Pr_{(x,x')\sim X^2_{\leq\tau}}\left[\hat{f}(x) \neq \hat{f}(x')\right] \leq \left(1 + \frac{1}{\sqrt{\delta}}\right)(\alpha\tau + \beta).$$

The proof is deferred to Appendix D.1.8.

6.4.4 Output Approximation and Loss Approximation

In this paper, we have analyzed the output approximation qualities of various derandomization techniques using the definitions of bias and variance in Section 6.1.2, which say that the output of $\hat{f}$ should resemble that of f, either on a single point x or in aggregate over some distribution $\mathscr{D}$.

An alternative set of definitions of bias and variance, put forth in Cotter et al. [2019a], instead measures how well $\hat{f}$ preserves the *loss* of f according to one or more binary loss functions ℓ. This property, which we might call *loss approximation*, is useful since in practice, classifiers are typically compared based on criteria such as accuracy, false positive rate, etc. evaluated on a dataset — and these are essentially binary loss functions averaged over a data distribution.

Concretely, let $\ell : \{0,1\} \times \{0,1\} \to \{0,1\}$ be a loss function and let $(x,y) \in X \times \{0,1\}$ be an instance with its corresponding label. The loss on this instance incurred by a (stochastic or deterministic) classifier f is defined as

$$L(f,x,y) := f(x)\ell(1,y) + (1-f(x))\ell(0,y)$$

The (pointwise) bias and variance of $\hat{f}$ under this loss are then

$$\begin{aligned}\mathsf{Bias}(\hat{f},f,x,y,\ell) &:= \left|\mathbb{E}_{\hat{f}\sim\mathscr{F}}\left[L(\hat{f},x,y)\right] - L(f,x,y)\right|,\\ \text{variance}(\hat{f},x,y,\ell) &:= \mathrm{Var}_{\hat{f}\sim\mathscr{F}}\left(L(\hat{f},x,y)\right)\end{aligned}$$

We observe that these are closely related to the simpler definitions given in Section 6.1.2:

Lemma 6.4.6. *For any* $\ell : \{0,1\} \times \{0,1\} \to \{0,1\}$*,* $x \in X$*, and* $y \in \{0,1\}$*,*

$$\mathit{Bias}(\hat{f},f,x,y,\ell) \leq \left|\mathit{Bias}(\hat{f},f,x)\right| \qquad \text{and} \qquad \text{variance}(\hat{f},x,y,\ell) \leq \text{variance}(\hat{f},x)$$

Thus even when the goal is to compute a derandomization that simulates the performance of f on one or more binary loss functions, it essentially suffices to use a derandomization that merely simulates the raw output of f itself. See Appendix D.1.9 for the proof of this lemma.

6.5 Discussion

We offer some brief notes regarding practical considerations for our derandomization framework.

A framework for derandomization Our results give machine learning practitioners a time- and space-efficient way to remove randomness — with the inherent brittleness, security vulnerabilities, and other issues that stochasticity entails — from their deployed models while approximately preserving fairness constraints enforced during training. Notably, our derandomization procedure has the useful quality of being *oblivious* to f, its training process, and even its actual fairness parameters α and β. It can therefore be applied as an independent post-processing step — for example, on the stochastic classifiers generated by the algorithms of Rothblum and Yona [2018], Kim et al. [2018], and others. The burden on the model designer is thus reduced to selecting a metric feature space (X, d) that is both appropriate for the classification task and for which an LSH family exists.

This simplification comes with inherent constraints: it was shown in Charikar [2002] that only metrics (or similarities ϕ whose complement d is a metric) can have LSH schemes, though not all of them do. On the positive side, recent work has shown that various non-LSHable similarities can be approximated by LSHable similarities with some provable distortion bound Chierichetti et al. [2019].

Separation of feature sets Throughout this paper, we have assumed that the inner hash function h_{LS} and classifiers f and $\hat{f}$ all share the same domain X; however, this is in no way necessary. In fact, from a fairness perspective, it is often prudent to distinguish between the features used for ensuring fairness and those used purely for inference, i.e. we may have

$$f : X \to [0, 1], \ \hat{f} : X \to \{0, 1\}, \text{ and } h_{\mathsf{LS}} : Z \to B$$

The feature set Z should be chosen, in tandem with an appropriate LSHable metric $d : Z \to [0, 1]$, so as to measure similarity or difference between inputs on the basis of attributes that should be treated equitably; on the other hand, the feature set X can be designed primarily to maximize predictive accuracy, and need not have any overlap with Z. The fairness guarantees of Theorem 6.3.3 and Corollary 6.3.4 then hold with respect to the metric space (Z, d) rather than (X, d).

Chapter 7

Conclusion and Future Works

7.1 Conclusion

While algorithms can automate decisions and optimize outcomes, their design must thoughtfully account for the complexities of human behavior to avoid perpetuating biases and enable socially beneficial outcomes. Building upon existing literature on algorithmic game theory, machine learning, constrained optimization, and algorithmic fairness, our goal is to contribute to the ongoing dialogue in responsible machine learning, setting a precedent for future research to build upon. We mainly focus on three aspects: understanding the social impact of decision rules, designing interventions that are both socially beneficial and sustainable and enhancing the practicality of algorithmic decision-making. The ultimate aim of this thesis is to produce algorithms, modeling frameworks, theorems, and experimental findings that bring the state of the art in research closer to practical deployability.

7.2 Future Works

We separate the questions based on the relevant chapters in this thesis:

From Chapter 3 Our contributions in this chapter are mostly theoretical. A natural extension is to collect real human experiment data to verify the usefulness and tightness of our bounds. Another potential future direction is to develop algorithms to find an optimal model that achieves minimum induced risk, which has been an exciting ongoing research problem in the field of performative prediction [Perdomo et al., 2020]. Furthermore, using techniques from general domain adaptation to find robust classifiers that perform well in both the source and induced distribution is another promising direction.

From Chapter 5 There are several potential future works related to this chapter:

1. Given the fact that a utility-maximizing recourse system has an incentive to not provide optimal recourse to everyone as a result of strategic manipulation, how do we design a manipulate-proof recourse system?
2. What are the other efficient intervention tools besides subsidy to incentivize recourse offerings?
3. In the current work, we assume that agents perform naive collusion, meaning that agents share their true features as well as their original features. It will be interesting to study how various levels of cooperation and strength affect the result. In particular, we may want to consider the following types of collisions:
    - **No Collusion** Agents share no information with one another.
    - **Truthful Collusion** Agents only share their true feature, and do not share their recourse action.
    - **Naive Collusion** (the current setting) Agents share both their true

feature and their provided recourse feature, but the recourse feature is generated using their true feature.

- **Strategic Collusion** All agents share both their true feature and provided recourse feature, but agents now also manipulate the recourse generating process (via a misreport) with the goal of decreasing the cost of achieving positive classification for future agents, i.e., the population shares utility.
- **Adversarial Collusion** All agents share both their true features and provided recourse features. Agents manipulate the recourse generation process with the goal of decreasing the system utility gained by providing recourse. In addition to modeling agents who seek to harm the system, this case serves as a worst-case comparison in terms of when a system will elect not to provide recourse.

4. In this work, we explore a scenario in which agents simultaneously arrive and concurrently seek recourse. However, a potentially more realistic setting would involve considering a situation where agents arrive sequentially in an online manner. Within this framework, the central inquiry focuses on identifying the optimal strategy for a recourse system to employ in order to maximize its utility when providing recourse under such conditions.

From Chapter 6 This chapter has focused on classifier derandomization with individual fairness guarantees, but it is also worthwhile to investigate the effect of derandomization from a group fairness perspective — for example, if it is possible to design an LSHable metric such that the derandomization preserves notions of fairness with respect to a protected attribute?

Appendix A

Appendix for Chapter 3

A.1 Proof of Theorem 3.3.4

Proof. Invoking Theorem 3.3.1, and replacing h with h_T^* and S with $\mathscr{D}(h_T^*)$, we have

$$\mathrm{Err}_{\mathscr{D}(h)}(h_T^*) \leq \mathrm{Err}_{\mathscr{D}(h_T^*)}(h_T^*) + \lambda_{\mathscr{D}(h)\to\mathscr{D}(h_T^*)} + \frac{1}{2} d_{\mathscr{H}\times\mathscr{H}}(\mathscr{D}(h_T^*), \mathscr{D}(h)) \tag{A.1}$$

Now observe that

$$\begin{aligned}
\mathrm{Err}_{\mathscr{D}(h)}(h) &\leq \mathrm{Err}_{\mathscr{D}(h)}(h_T^*) + \mathrm{Err}_{\mathscr{D}(h)}(h, h_T^*) \\
&\leq \mathrm{Err}_{\mathscr{D}(h)}(h_T^*) + \mathrm{Err}_{\mathscr{D}(h_T^*)}(h, h_T^*) + \left|\mathrm{Err}_{\mathscr{D}(h)}(h, h_T^*) - \mathrm{Err}_{\mathscr{D}(h_T^*)}(h, h_T^*)\right| \\
&\leq \mathrm{Err}_{\mathscr{D}(h)}(h_T^*) + \mathrm{Err}_{\mathscr{D}(h_T^*)}(h, h_T^*) + \frac{1}{2} d_{\mathscr{H}\times\mathscr{H}}(\mathscr{D}(h_T^*), \mathscr{D}(h)) \\
&\qquad \text{(by Lemma 3.3.2)} \\
&\leq \mathrm{Err}_{\mathscr{D}(h)}(h_T^*) + \mathrm{Err}_{\mathscr{D}(h_T^*)}(h) + \mathrm{Err}_{\mathscr{D}(h_T^*)}(h_T^*) + \frac{1}{2} d_{\mathscr{H}\times\mathscr{H}}(\mathscr{D}(h_T^*), \mathscr{D}(h)) \\
&\qquad \text{(by Lemma 3.3.3)} \\
&\leq \mathrm{Err}_{\mathscr{D}(h_T^*)}(h_T^*) + \lambda_{\mathscr{D}(h)\to\mathscr{D}(h_T^*)} + \frac{1}{2} d_{\mathscr{H}\times\mathscr{H}}(\mathscr{D}(h_T^*), \mathscr{D}(h)) \\
&\qquad \text{(by equation A.1)} \\
&\quad + \mathrm{Err}_{\mathscr{D}(h_T^*)}(h) + \mathrm{Err}_{\mathscr{D}(h_T^*)}(h_T^*) + \frac{1}{2} d_{\mathscr{H}\times\mathscr{H}}(\mathscr{D}(h_T^*), \mathscr{D}(h))
\end{aligned}$$

Adding $\mathrm{Err}_{\mathscr{D}(h)}(h)$ to both sides and rearranging terms yields

$$\begin{aligned}
&2\mathrm{Err}_{\mathscr{D}(h)}(h) - 2\mathrm{Err}_{\mathscr{D}(h_T^*)}(h_T^*) \\
\leq &\mathrm{Err}_{\mathscr{D}(h)}(h) + \mathrm{Err}_{\mathscr{D}(h_T^*)}(h) + \lambda_{\mathscr{D}(h)\to\mathscr{D}(h_T^*)} + d_{\mathscr{H}\times\mathscr{H}}(\mathscr{D}(h_T^*), \mathscr{D}(h)) \\
= &\Lambda_{\mathscr{D}(h)\to\mathscr{D}(h_T^*)}(h) + \lambda_{\mathscr{D}(h)\to\mathscr{D}(h_T^*)} + d_{\mathscr{H}\times\mathscr{H}}(\mathscr{D}(h_T^*), \mathscr{D}(h))
\end{aligned}$$

Dividing both sides by 2 completes the proof. □

A.2 Proof of Theorem 3.3.5

Proof. Using the triangle inequality of d_{TV}, we have

$$d_{\mathrm{TV}}(\mathscr{D}_{Y|S}, \mathscr{D}_Y(h)) \leq d_{\mathrm{TV}}(\mathscr{D}_{Y|S}, \mathscr{D}_{h|S}) + d_{\mathrm{TV}}(\mathscr{D}_{h|S}, \mathscr{D}_h(h)) + d_{\mathrm{TV}}(\mathscr{D}_h(h), \mathscr{D}_Y(h)) \tag{A.2}$$

and by the definition of d_{TV}, the divergence term $d_{\mathrm{TV}}(\mathscr{D}_{Y|S}, \mathscr{D}_Y(h))$ becomes

$$\begin{aligned}
d_{\mathrm{TV}}(\mathscr{D}_{Y|S}, \mathscr{D}_{h|S}) &= |\mathbb{P}_{\mathscr{D}_S}(Y=+1) - \mathbb{P}_{\mathscr{D}_S}(h(x)=+1)| \\
&= \left|\frac{\mathbb{E}_{\mathscr{D}_S}[Y]+1}{2} - \frac{\mathbb{E}_{\mathscr{D}_S}[h(X)]+1}{2}\right| \\
&= \left|\frac{\mathbb{E}_{\mathscr{D}_S}[Y]}{2} - \frac{\mathbb{E}_{\mathscr{D}_S}[h(X)]}{2}\right| \\
&\leq \frac{1}{2}\cdot \mathbb{E}_{\mathscr{D}_S}\left[|Y - h(X)|\right] \\
&= \mathrm{Err}_{\mathscr{D}_S}(h)
\end{aligned}$$

Similarly, we have

$$d_{\mathrm{TV}}(\mathscr{D}_h(h), \mathscr{D}_Y(h)) \leq \mathrm{Err}_{\mathscr{D}(h)}(h)$$

As a result, we have

$$\begin{aligned}
\mathrm{Err}_{\mathscr{D}_S}(h) + \mathrm{Err}_{\mathscr{D}(h)}(h) &\geq d_{\mathrm{TV}}(\mathscr{D}_{Y|S}, \mathscr{D}_{h|S}) + d_{\mathrm{TV}}(\mathscr{D}_h(h), \mathscr{D}_Y(h)) \\
&\geq d_{\mathrm{TV}}(\mathscr{D}_{Y|S}, \mathscr{D}_Y(h)) - d_{\mathrm{TV}}(\mathscr{D}_{h|S}, \mathscr{D}_h(h)) \\
&\qquad \text{(by equation A.2)}
\end{aligned}$$

which implies

$$\max\{\mathrm{Err}_{\mathscr{D}_S}(h), \mathrm{Err}_{\mathscr{D}(h)}(h)\} \geq \frac{d_{\mathrm{TV}}(\mathscr{D}_{Y|S}, \mathscr{D}_Y(h)) - d_{\mathrm{TV}}(\mathscr{D}_{h|S}, \mathscr{D}_h(h))}{2} .$$

□

A.3 Proof of Theorem 3.4.2

Proof. We start from the error induced by h_S^*. Let the *average importance weight induced by* h_S^* be $\bar{\omega}(h_S^*) = \mathbb{E}_{\mathscr{D}_S}[\omega_x(h_S^*)]$; we add and subtract this from the error:

$$\begin{aligned}\mathrm{Err}_{\mathscr{D}(h_S^*)}(h_S^*) &= \mathbb{E}_{\mathscr{D}_S}\left[\omega_x(h_S^*) \cdot \mathbb{1}(h_S^*(x) \neq y)\right] \\ &= \mathbb{E}_{\mathscr{D}_S}\left[\bar{\omega}(h_S^*) \cdot \mathbb{1}(h_S^*(x) \neq y)\right] + \mathbb{E}_{\mathscr{D}_S}\left[(\omega_x(h_S^*) - \bar{\omega}(h_S^*)) \cdot \mathbb{1}(h_S^*(x) \neq y)\right]\end{aligned}$$

In fact, $\bar{\omega}(h_S^*) = 1$, since

$$\begin{aligned}\bar{\omega}(h_S^*) =& \mathbb{E}_{\mathscr{D}_S}[\omega_x(h_S^*)] = \int \omega_x(h_S^*)\mathbb{P}_{\mathscr{D}_S}(X = x)dx \\ &= \int \frac{\mathbb{P}_{\mathscr{D}(h)}(X = x)}{\mathbb{P}_{\mathscr{D}_S}(X = x)}\mathbb{P}_{\mathscr{D}_S}(X = x)dx = \int \mathbb{P}_{\mathscr{D}(h)}(X = x)dx = 1\end{aligned}$$

Now consider any other classifier h. We have

$$
\begin{aligned}
&\mathrm{Err}_{\mathscr{D}(h_S^*)}(h_S^*) \\
&= \mathbb{E}_{\mathscr{D}_S}\left[\mathbb{1}(h_S^*(x) \neq y)\right] + \mathbb{E}_{\mathscr{D}_S}\left[(\omega_x(h_S^*) - \bar{\omega}(h_S^*)) \cdot \mathbb{1}(h_S^*(x) \neq y)\right] \\
&\leq \mathbb{E}_{\mathscr{D}_S}\left[\mathbb{1}(h(x) \neq y)\right] + \mathbb{E}_{\mathscr{D}_S}\left[(\omega_x(h_S^*) - \bar{\omega}(h_S^*)) \cdot \mathbb{1}(h_S^*(x) \neq y)\right] \\
&\hfill \text{(by optimality of } h_S^* \text{ on } \mathscr{D}_S) \\
&= \mathbb{E}_{\mathscr{D}_S}\left[\bar{\omega}(h) \cdot \mathbb{1}(h(x) \neq y)\right] + \mathbb{E}_{\mathscr{D}_S}\left[(\omega_x(h_S^*) - \bar{\omega}(h_S^*)) \cdot \mathbb{1}(h_S^*(x) \neq y)\right] \\
&\hfill \text{(multiply by } \bar{\omega}(h_S^*) = 1) \\
&= \mathbb{E}_{\mathscr{D}_S}\left[\omega_x(h) \cdot \mathbb{1}(h(x) \neq y)\right] + \mathbb{E}_{\mathscr{D}_S}\left[(\bar{\omega}(h) - \omega_x(h)) \cdot \mathbb{1}(h(x) \neq y)\right] \\
&\hfill \text{(add and subtract } \bar{\omega}(h_S^*)) \\
&\quad + \mathbb{E}_{\mathscr{D}_S}\left[(\omega_x(h_S^*) - \bar{\omega}(h_S^*)) \cdot \mathbb{1}(h_S^*(x) \neq y)\right] \\
&= \mathrm{Err}_{\mathscr{D}(h)}(h) + \mathrm{Cov}(\omega_x(h_S^*), \mathbb{1}(h_S^*(x) \neq y)) - \mathrm{Cov}(\omega_x(h), \mathbb{1}(h(x) \neq y))
\end{aligned}
$$

Moving the error terms to one side, we have

$$
\begin{aligned}
&\mathrm{Err}_{\mathscr{D}(h_S^*)}(h_S^*) - \mathrm{Err}_{\mathscr{D}(h)}(h) \\
&\leq \mathrm{Cov}(\omega_x(h_S^*), \mathbb{1}(h_S^*(x) \neq y)) - \mathrm{Cov}(\omega_x(h), \mathbb{1}(h(x) \neq y)) \\
&\leq \sqrt{\mathrm{Var}(\omega_x(h_S^*)) \cdot \mathrm{Var}(\mathbb{1}(h_S^*(x) \neq y))} \qquad (|\mathrm{Cov}(X,Y)| \leq \sqrt{\mathrm{Var}(X) \cdot \mathrm{Var}(Y)}) \\
&\quad + \sqrt{\mathrm{Var}(\omega_x(h)) \cdot \mathrm{Var}(\mathbb{1}(h(x) \neq y))} \\
&= \sqrt{\mathrm{Var}(\omega_x(h_S^*)) \cdot \mathrm{Err}_S(h_S^*)(1 - \mathrm{Err}_S(h_S^*))} + \sqrt{\mathrm{Var}(\omega_x(h)) \cdot \mathrm{Err}_{\mathscr{D}_S}(h)(1 - \mathrm{Err}_{\mathscr{D}_S}(h))} \\
&\leq \sqrt{\mathrm{Var}(\omega_x(h_S^*)) \cdot \mathrm{Err}_S(h_S^*)} + \sqrt{\mathrm{Var}(\omega_x(h)) \cdot \mathrm{Err}_{\mathscr{D}_S}(h)} \qquad (1 - \mathrm{Err}_{\mathscr{D}_S}(h) \leq 1) \\
&\leq \sqrt{\mathrm{Err}_{\mathscr{D}_S}(h)} \cdot \left(\sqrt{\mathrm{Var}(\omega_x(h_S^*))} + \sqrt{\mathrm{Var}(\omega_x(h))}\right)
\end{aligned}
$$

Since this holds for any h, it certainly holds for $h = h_T^*$. □

A.4 Omitted Assumptions and Proof of Theorem 3.4.6

Denote $X_+(h) = \{x : \omega_x(h) \geq 1\}$ and $X_-(h) = \{x : \omega_x(h) < 1\}$. First, we observe that

$$\int_{X_+(h)} \mathbb{P}_{\mathscr{D}_S}(X = x)(1 - \omega_x(h))dx + \int_{X_-(h)} \mathbb{P}_{\mathscr{D}_S}(X = x)(1 - \omega_x(h))dx = 0$$

This is simply because of $\int_x \mathbb{P}_{\mathscr{D}_S}(X = x) \cdot \omega_x(h)dx = \int_x \mathbb{P}_{\mathscr{D}(h)}(X = x)dx = 1$.

Proof. Notice that in the setting of binary classification, we can write the total variation distance between $\mathscr{D}_{Y|S}$ and $\mathscr{D}_Y(h)$ as the difference between the probability of $Y = +1$ and the probability of $Y = -1$:

$$\begin{aligned}
&d_{\text{TV}}(\mathscr{D}_{Y|S}, \mathscr{D}_Y(h)) \\
=& \left|\mathbb{P}_{\mathscr{D}_S}(Y = +1) - \mathbb{P}_{\mathscr{D}(h)}(Y = +1)\right| \\
=& \left|\int \mathbb{P}_{\mathscr{D}_S}(Y = +1|X = x)\mathbb{P}_{\mathscr{D}_S}(X = x)dx - \int \mathbb{P}_{\mathscr{D}_S}(Y = +1|X = x)\mathbb{P}_{\mathscr{D}_S}(X = x)\omega_x(h)dx\right| \\
=& \left|\int \mathbb{P}_{\mathscr{D}_S}(Y = +1|X = x)\mathbb{P}_{\mathscr{D}_S}(X = x) \cdot (1 - \omega_x(h))dx\right| \qquad \text{(A.3)}
\end{aligned}$$

Similarly we have

$$d_{\text{TV}}(\mathscr{D}_{h|S}, \mathscr{D}_h(h)) = \left|\int \mathbb{P}_{\mathscr{D}_S}(h(x) = +1|X = x)\mathbb{P}_{\mathscr{D}_S}(X = x) \cdot (1 - \omega_x(h))dx\right| \tag{A.4}$$

We can further expand the total variation distance between $\mathscr{D}_{Y|S}$ and $\mathscr{D}_Y(h)$ as follows:

$$
\begin{aligned}
& d_{\mathrm{TV}}(\mathscr{D}_{Y|S}, \mathscr{D}_Y(h)) \\
=& \left| \int \mathbb{P}_{\mathscr{D}_S}(Y=+1|X=x)\mathbb{P}_{\mathscr{D}_S}(X=x)\cdot(1-\omega_x(h))dx \right| \\
=& \Big| \underbrace{\int_{X_+(h)} \mathbb{P}_{\mathscr{D}}(Y=+1|X=x)\mathbb{P}_{\mathscr{D}_S}(X=x)\cdot(1-\omega_x(h))dx}_{\leq 0} \\
& + \underbrace{\int_{X_-(h)} \mathbb{P}_{\mathscr{D}_S}(Y=+1|X=x)\mathbb{P}_{\mathscr{D}_S}(X=x)\cdot(1-\omega_x(h))dx \Big|}_{>0} \\
=& -\int_{X_+(h)} \mathbb{P}_{\mathscr{D}_S}(Y=+1|X=x)\mathbb{P}_{\mathscr{D}_S}(X=x)\cdot(1-\omega_x(h))dx \\
& -\int_{X_-(h)} \mathbb{P}_{\mathscr{D}_S}(Y=+1|X=x)\mathbb{P}_{\mathscr{D}_S}(X=x)\cdot(1-\omega_x(h))dx \\
& \hfill \text{(by Assumption 3.4.3)} \\
=& \int_{X_+(h)} \mathbb{P}_{\mathscr{D}_S}(Y=+1|X=x)\mathbb{P}_{\mathscr{D}_S}(X=x)\cdot(\omega_x(h)-1)dx \\
& + \int_{X_-(h)} \mathbb{P}_{\mathscr{D}_S}(Y=+1|X=x)\mathbb{P}_{\mathscr{D}_S}(X=x)\cdot(\omega_x(h)-1)dx \quad \text{(by equation A.3)} \\
=& \int \mathbb{P}_{\mathscr{D}_S}(Y=+1|X=x)\mathbb{P}_{\mathscr{D}_S}(X=x)\cdot(\omega_x(h)-1)dx
\end{aligned}
$$

Similarly, by assumption 3.4.4 and equation equation A.4, we have

$$
d_{\mathrm{TV}}(\mathscr{D}_{h|S}, \mathscr{D}_h(h)) = \int \mathbb{P}_{\mathscr{D}_S}(h(x)=+1|X=x)\mathbb{P}_{\mathscr{D}_S}(X=x)\cdot(\omega_x(h)-1)dx
$$

Thus we can bound the difference between $d_{\mathrm{TV}}(\mathscr{D}_{Y|S}, \mathscr{D}_Y(h))$ and $d_{\mathrm{TV}}(\mathscr{D}_{h|S}, \mathscr{D}_h(h))$

as follows:

$$
\begin{aligned}
& d_{\mathrm{TV}}(\mathscr{D}_{Y|S}, \mathscr{D}_Y(h)) - d_{\mathrm{TV}}(\mathscr{D}_{h|S}, \mathscr{D}_h(h)) \\
&= \int \mathbb{P}_{\mathscr{D}_S}(Y = +1|X = x)\mathbb{P}_{\mathscr{D}_S}(X = x) \cdot (\omega_x(h) - 1)dx \\
&\quad - \int \mathbb{P}_{\mathscr{D}}(h(x) = +1|X = x)\mathbb{P}_{\mathscr{D}_S}(X = x) \cdot (\omega_x(h) - 1)dx \\
&= \int [\mathbb{P}_{\mathscr{D}_S}(Y = +1|X = x) - \mathbb{P}_{\mathscr{D}_S}(h(x) = +1|X = x)]\mathbb{P}_{\mathscr{D}_S}(X = x) \cdot (\omega_x(h) - 1)dx \\
&= \mathbb{E}_{\mathscr{D}_S}[(\mathbb{P}_{\mathscr{D}_S}(Y = +1|X = x) - \mathbb{P}_{\mathscr{D}_S}(h(x) = +1|X = x))\,(\omega_x(h) - 1)] \\
&\hfill \text{(by Assumption 3.4.5)} \\
&> \mathbb{E}_{\mathscr{D}_S}[\mathbb{P}_{\mathscr{D}_S}(Y = +1|X = x) - \mathbb{P}_{\mathscr{D}_S}(h(x) = +1|X = x)]\mathbb{E}_{\mathscr{D}_S}[\omega_x(h) - 1] \\
&= 0
\end{aligned}
$$

Combining the above with Theorem 3.3.5, we have

$$
\max\{\mathrm{Err}_{\mathscr{D}_S}(h), \mathrm{Err}_{\mathscr{D}(h)}(h)\} \geq \frac{d_{\mathrm{TV}}(\mathscr{D}_{Y|S}, \mathscr{D}_Y(h)) - d_{\mathrm{TV}}(\mathscr{D}_{h|S}, \mathscr{D}_h(h))}{2} > 0
$$

□

A.5 Omitted Details for Section 3.4.3

With Setup 2 - Setup 4, we can further specify the important weight $w_x(h)$ for the strategic response setting:

Lemma A.5.1. *Recall the definition for the covariate shift important weight coef-*

ficient $\omega_x(h) := \frac{\mathbb{P}_{D(h)}(X=x)}{\mathbb{P}_{D_S}(X=x)}$, *for our strategic response setting, we have,*

$$w_x(h) = \begin{cases} 1, & x \in [0, \tau_h - B) \\ \frac{\tau_h - x}{B}, & x \in [\tau_h - B, \tau_h) \\ \frac{1}{B}(-x + \tau_h + 2B), & x \in [\tau_h, \tau_h + B) \\ 1, & x \in [\tau_h + B, 1] \end{cases} \tag{A.5}$$

Proof for Lemma A.5.1:

Proof. We discuss the induced distribution $\mathscr{D}(h)$ by cases:

- For the features distributed between $[0, \tau_h - B]$: since we assume the agents are rational, under assumption 2, agents with feature that is smaller than $[0, \tau_h - B]$ will not perform any kinds of adaptations, and no other agents will adapt their features to this range of features either, so the distribution between $[0, \tau_h - B]$ will remain the same as before.
- For the target distribution between $[\tau_h - B, \tau_h]$ can be directly calculated from assumption 3.
- For distribution between $[\tau_h, \tau_h + B]$, consider a particular feature $x^\star \in [\tau_h, \tau_h + B]$, under Setup 4, we know its new distribution becomes:

$$\begin{aligned} \mathbb{P}_{\mathscr{D}(h)}(x = x^\star) &= 1 + \int_{x^\star - B}^{\tau_h} \frac{1 - \frac{\tau_h - z}{B}}{B - \tau_h + z} dz \\ &= 1 + \int_{x^\star - B}^{\tau_h} \frac{1}{B} dz \\ &= \frac{1}{B}(-x^\star + \tau_h + 2B) \end{aligned}$$

- For the target distribution between $[\tau_h + B, 1]$: under assumption 2 and 4, we know that no agents will change their feature to this feature region. So the distribution between $[\tau_h + B, 1]$ remains the same as the source distribution.

Recall the definition for the covariate shift important weight coefficient $\omega_x(h) := \frac{\mathbb{P}_{D(h)}(X=x)}{\mathbb{P}_{D_S}(X=x)}$, the distribution of $\omega_x(h)$ after agents' strategic responding becomes:

$$\omega_x(h) = \begin{cases} 1, & x \in [0, \tau_h - B) \text{ and } x \in [\tau_h + B, 1] \\ \frac{\tau_h - x}{B}, & x \in [\tau_h - B, \tau_h) \\ \frac{1}{B}(-x + \tau_h + 2B), & x \in [\tau_h, \tau_h + B) \\ 0, & \text{otherwise} \end{cases} \tag{A.6}$$

□

Proof for Proposition 3.4.7:

Proof. According to Lemma A.5.1, we can compute the variance of $w_x(h)$ as $\mathrm{Var}(w_x(h)) = \mathbb{E}(w_x(h)^2) - \mathbb{E}(w_x(h)^2) = \frac{2}{3}B$. Then plugging it into the general bound for Theorem 3.4.2 gives us the desired result. □

A.6 Proof of Theorem 3.5.1

Proof. Defining $p := \mathbb{P}_{\mathcal{D}_S}(Y = +1)$, $p(h) = \mathbb{P}_{\mathcal{D}(h)}(Y = +1)$, we have

$$\mathrm{Err}_{\mathcal{D}(h_S^*)}(h_S^*) = p(h_S^*) \cdot \mathrm{Err}_+(h_S^*) + (1 - p(h_S^*)) \cdot \mathrm{Err}_-(h_S^*)$$

(by definitions of $p(h_S^*)$, $\mathrm{Err}_+(h_S^*)$, and $\mathrm{Err}_-(h_S^*)$)

$$= \underbrace{p \cdot \mathrm{Err}_+(h_S^*) + (1 - p) \cdot \mathrm{Err}_-(h_S^*)}_{\text{(I)}} + (p(h_S^*) - p)[\mathrm{Err}_+(h_S^*) - \mathrm{Err}_-(h_S^*)] \tag{A.7}$$

We can expand (I) as follows:

$$\begin{aligned}
& p \cdot \mathrm{Err}_+(h_S^*) + (1-p) \cdot \mathrm{Err}_-(h_S^*) \\
\leq\, & p \cdot \mathrm{Err}_+(h_T^*) + (1-p) \cdot \mathrm{Err}_-(h_T^*) \qquad \text{(by optimality of } h_S^* \text{ on } \mathscr{D}_S) \\
=\, & p(h_T^*) \cdot \mathrm{Err}_+(h_T^*) + (1-p(h_T^*)) \cdot \mathrm{Err}_-(h_T^*) + (p - p(h_T^*)) \cdot [\mathrm{Err}_+(h_T^*) - \mathrm{Err}_-(h_T^*)] \\
=\, & \mathrm{Err}_{\mathscr{D}(h_T^*)}(h_T^*) + (p - p(h_T^*)) \cdot [\mathrm{Err}_+(h_T^*) - \mathrm{Err}_-(h_T^*)]\ .
\end{aligned}$$

Plugging this back into equation A.7, we have

$$\begin{aligned}
& \mathrm{Err}_{\mathscr{D}(h_S^*)}(h_S^*) - \mathrm{Err}_{\mathscr{D}(h_T^*)}(h_T^*) \\
\leq & (p(h_S^*) - p)[\mathrm{Err}_+(h_S^*) - \mathrm{Err}_-(h_S^*)] + (p - p(h_T^*)) \cdot [\mathrm{Err}_+(h_T^*) - \mathrm{Err}_-(h_T^*)]
\end{aligned}$$

Notice that

$$\begin{aligned}
& 0.5(\mathrm{Err}_+(h) - \mathrm{Err}_-(h)) \\
=\, & 0.5 \cdot 1 - 0.5 \cdot \mathbb{P}(h(X) = +1 | Y = +1) - 0.5 \cdot \mathbb{P}(h(X) = +1 | Y = -1) \\
=\, & 0.5 - \mathbb{P}_{\mathscr{D}_u}(h(X) = +1)
\end{aligned}$$

where $\mathscr{D}_u$ is a distribution with a uniform prior. Then

$$\begin{aligned}
(p(h_S^*) - p)[\mathrm{Err}_+(h_S^*) - \mathrm{Err}_-(h_S^*)] &= 2(p(h_S^*) - p) \cdot (0.5 - \mathbb{P}_{\mathscr{D}_u}(h(X) = +1)) \\
(p - p(h_T^*))[\mathrm{Err}_+(h_T^*) - \mathrm{Err}_-(h_T^*)] &= 2(p - p(h_T^*)) \cdot (0.5 - \mathbb{P}_{\mathscr{D}_u}(h(X) = +1))
\end{aligned}$$

Adding together these two equations yields

$$
\begin{aligned}
&(p(h_S^*) - p)[\mathrm{Err}_+(h_S^*) - \mathrm{Err}_-(h_S^*)] + (p - p(h_T^*)) \cdot [\mathrm{Err}_+(h_T^*) - \mathrm{Err}_-(h_T^*)] \\
&= 2(p(h_S^*) - p) \cdot (0.5 - \mathbb{P}_{\mathscr{D}_u}(h_S^*(X) = +1)) + 2(p - p(h_T^*)) \cdot (0.5 - \mathbb{P}_{\mathscr{D}_u}(h_T^*(X) = +1)) \\
&= (p(h_S^*) - p(h_T^*)) - 2\left(p(h_S^*)\mathbb{P}_{\mathscr{D}_u}(h_S^*(X) = +1) - p(h_T^*)\mathbb{P}_{\mathscr{D}_u}(h_T^*(X) = +1)\right) \\
&\quad + 2p \cdot (\mathbb{P}_{\mathscr{D}_u}(h_S^*(X) = +1) - \mathbb{P}_{\mathscr{D}_u}(h_T^*(X) = +1)) \\
&\leq |p(h_S^*) - p(h_T^*)| \cdot (1 + 2|\mathbb{P}_{\mathscr{D}_u}(h_S^*(X) = +1) - \mathbb{P}_{\mathscr{D}_u}(h_T^*(X) = +1)|) \\
&\quad + 2p \cdot |\mathbb{P}_{\mathscr{D}_u}(h_S^*(X) = +1) - \mathbb{P}_{\mathscr{D}_u}(h_T^*(X) = +1)| \qquad \text{(A.8)}
\end{aligned}
$$

Meanwhile,

$$
\begin{aligned}
&|\mathbb{P}_{\mathscr{D}_u}(h_S^*(X) = +1) - \mathbb{P}_{\mathscr{D}_u}(h_T^*(X) = +1)| \\
&\leq 0.5 \cdot |\mathbb{P}_{\mathscr{D}|Y=+1}(h_S^*(X) = +1) - \mathbb{P}_{\mathscr{D}|Y=+1}(h_T^*(X) = +1)| \\
&\quad + 0.5 \cdot |\mathbb{P}_{\mathscr{D}|Y=-1}(h_S^*(X) = +1) - \mathbb{P}_{\mathscr{D}|Y=-1}(h_T^*(X) = +1)| \\
&= 0.5\left(d_{\mathrm{TV}}(\mathscr{D}_+(h_S^*), \mathscr{D}_+(h_T^*)) + d_{\mathrm{TV}}(\mathscr{D}_-(h_S^*), \mathscr{D}_-(h_T^*)\right) \qquad \text{(A.9)}
\end{aligned}
$$

Combining equation A.8 and equation A.9 gives

$$
\begin{aligned}
&|p(h_S^*) - p(h_T^*)| \cdot (1 + 2 \cdot |\mathbb{P}_{\mathscr{D}_u}(h_S^*(X) = +1) - \mathbb{P}_{\mathscr{D}_u}(h_T^*(X) = +1)|) \\
&\quad + 2p \cdot |\mathbb{P}_{\mathscr{D}_u}(h_S^*(X) = +1) - \mathbb{P}_{\mathscr{D}_u}(h_T^*(X) = +1)| \\
&\leq |p(h_S^*) - p(h_T^*)| \cdot (1 + d_{\mathrm{TV}}(\mathscr{D}_+(h_S^*), \mathscr{D}_+(h_T^*)) + d_{\mathrm{TV}}(\mathscr{D}_-(h_S^*), \mathscr{D}_-(h_T^*)) \\
&\quad + p \cdot (d_{\mathrm{TV}}(\mathscr{D}_+(h_S^*), \mathscr{D}_+(h_T^*)) + d_{\mathrm{TV}}(\mathscr{D}_-(h_S^*), \mathscr{D}_-(h_T^*)) \\
&\leq |p(h_S^*) - p(h_T^*)| + (1 + p) \cdot (d_{\mathrm{TV}}(\mathscr{D}_+(h_S^*), \mathscr{D}_+(h_T^*)) + d_{\mathrm{TV}}(\mathscr{D}_-(h_S^*), \mathscr{D}_-(h_T^*)) \ .
\end{aligned}
$$

□

A.7 Missing Experimental Details

A.7.1 Synthetic Experiments Using DAG

Here we provide details regarding the data-generating process for the simulated dataset.

Covariate Shift We specify the causal DAG for covariate shift setting in the following way:

$$X_1 \sim \text{Unif}(-1, 1)$$

$$X_2 \sim 1.2X_1 + \mathscr{N}(0, \sigma_2^2)$$

$$X_3 \sim -X_1^2 + \mathscr{N}(0, \sigma_3^2)$$

$$Y := 2\text{sign}(X_2 > 0) - 1$$

where σ_2^2 and σ_3^2 are parameters of our choices.

Adaptation function We assume the new distribution of feature X_1' will be generated in the following way:

$$X_1' = \Delta(X) = X_1 + c \cdot (h(X) - 1)$$

where $c \in \mathbb{R}^1 > 0$ is the parameter controlling how much the prediction $h(X)$ affect the generating of X_1', namely the magnitude of distribution shift. Intuitively, this adaptation function means that if a feature x is predicted to be positive ($h(x) = +1$), then decision subjects are more likely to adapt to that feature in the induced distribution; Otherwise, decision subjects are more likely to be moving away from x since they know it will lead to a negative prediction.

Target Shift We specify the causal DAG for target shift setting in the following way:

$$
\begin{aligned}
(Y+1)/2 &\sim \text{Bernoulli}(\alpha) \\
X_1|Y=y &\sim \mathscr{N}_{[0,1]}(\mu_y, \sigma^2) \\
X_2 &= -0.8X_1 + \mathscr{N}(0, \sigma_2^2) \\
X_3 &= 0.2Y + \mathscr{N}(0, \sigma_3^2)
\end{aligned}
$$

where $\mathscr{N}_{[0,1]}$ represents a truncated Gaussian distribution taken value between 0 and 1. α, μ_y, σ^2,σ_2^2 and σ_3^2 are parameters of our choices.

Adaptation function We assume the new distribution of the qualification Y' will be updated in the following way:

$$\mathbb{P}(Y' = +1|h(X) = h, Y = y) = c_{hy}, \text{ where } \{h, y\} \in \{-1, +1\}$$

where $0 \leq c_{hy} \in \mathbb{R}^1 \leq 1$ represents the likelihood for a person with original qualification $Y = y$ and get predicted as $h(X) = h$ to be qualified in the next step $(Y' = +1)$.

A.7.2 Synthetic Experiments Using Real-world Data

On the preprocessed FICO credit score data set [Board of Governors of the Federal Reserve System (US), 2007, Hardt et al., 2016b], we convert the cumulative distribution function (CDF) of TransRisk score among demographic groups (denoted as A, including Black, Asian, Hispanic, and White) into group-dependent densities of the credit score. We then generate a balanced sample where each group has equal representation, with credit scores (denoted as Q) initialized by sampling from the corresponding group-dependent density. The value of attributes for each data point is then updated under a specified dynamics (detailed in Appendix A.7.3) to model the real-world scenario of repeated resource allocation (with decision denoted as D).

A.7.3 Parameters for Dynamics

Since we are considering the dynamic setting, we further specify the data generating process in the following way (from time step $T = t$ to $T = t + 1$):

$$\begin{aligned}
X_{t,1} &\sim 1.5Q_t + U[-\epsilon_1, \epsilon_1] \\
X_{t,2} &\sim 0.8A_t + U[-\epsilon_2, \epsilon_2] \\
X_{t,3} &\sim A_t + \mathscr{N}(0, \sigma^2) \\
Y_t &\sim \text{Bernoulli}(q_t) \text{ for a given value of } Q_t = q_t \\
D_t &= f_t(A_t, X_{t,1}, X_{t,2}, X_{t,3}) \\
Q_{t+1} &= \{Q_t \cdot [1 + \alpha_D(D_t) + \alpha_Y(Y_t)]\}_{(0,1]} \\
A_{t+1} &= A_t \text{ (fixed population)}
\end{aligned}$$

where $\{\cdot\}_{(0,1]}$ represents truncated value between the interval $(0, 1]$, $f_t(\cdot)$ represents the decision policy from input features, and $\epsilon_1, \epsilon_2, \sigma$ are parameters of choices. In our experiments, we set $\epsilon_1 = \epsilon_2 = \sigma = 0.1$.

Within the same time step, i.e., for variables that share the subscript t, Q_t and A_t are root causes for all other variables $(X_{t,1}, X_{t,2}, X_{t,3}, D_t, Y_t)$. At each time step $T = t$, the institution first estimates the credit score Q_t (which is not directly visible to the institution, but is reflected in the visible outcome label Y_t) based on $(A_t, X_{t,1}, X_{t,2}, X_{t,3})$, then produces the binary decision D_t according to the optimal threshold (in terms of the accuracy).

For different time steps, e.g., from $T = t$ to $T = t + 1$, the new distribution at

$T = t + 1$ is induced by the deployment of the decision policy D_t. Such impact is modeled by a multiplicative update in Q_{t+1} from Q_t with parameters (or functions) $\alpha_D(\cdot)$ and $\alpha_Y(\cdot)$ that depend on D_t and Y_t, respectively. In our experiments, we set $\alpha_D = 0.01$ and $\alpha_Y = 0.005$ to capture the scenario where one-step influence of the decision on the credit score is stronger than that for ground truth label.

Appendix B

Appendix for Chapter 4

B.1 Proof of Theorem 4.4.2

In this section, we provide the proof of Theorem 4.4.2. To simplify our discussion, we focus on the unconstrained best response, i.e. the case in which $\mathsf{F} = \mathsf{A}$. The proofs for the other two types of best response ($\mathsf{F} = \mathsf{M}$, $\mathsf{F} = \mathsf{I}$) follow the same arguments except that the inverse of $(S^{-1})_{\mathsf{I}}$ does not equal to S, but equals to $((S^{-1})_{\mathsf{I}})^{-1}$.

We first prove two lemmas that allow us to reformulate the best response as an optimization problem. The first states that the decision subject's goal is to maximize their utility, but they are unwilling to pay a cost greater than 2:

Lemma B.1.1 (Decision Subject's Best-Response Function)**.** *Given a classifier $h : \mathscr{X} \to \{-1, +1\}$, a cost function $c : \mathscr{X} \times \mathscr{X} \to \mathbb{R}$, and a set of realizable feature vectors $\mathscr{X}^\dagger \subseteq \mathscr{X}$, the* best response *of a decision subject with features $x \in \mathscr{X}^\dagger$ is the solution to the following optimization program:*

$$\max_{x' \in \mathscr{X}^\dagger} \quad U(x, x') \quad \text{s.t.} \quad c(x, x') \leq 2$$

Proof. Since the classifier in our game outputs a binary decision (-1 or $+1$), decision subjects only have an incentive to change their features from x to x' when $c(x, x') \leq 2$. To see this, notice that an decision subject originally classified as -1 receives a default utility of $U(x, x) = f(x) - 0 = -1$ by presenting her original features x. Since costs are always non-negative, she can only hope to increase her utility by flipping the classifier's decision. If she changes her features to some x' such that $f(x') = +1$, then the new utility will be given by

$$U(x, x') = f(x') - c(x, x') = 1 - c(x, x')$$

Hence the decision subject will only change her features if $1 - c(x, x') \geq f(x) = -1$, or $c(x, x') \leq 2$. □

The next lemma turns the above maximization program into a minimization program, in which the decision subject seeks the minimum-cost change in x that crosses the decision boundary. If the cost exceeds 2, which is the maximum possible gain from adaptation, they would rather not modify any features.

Lemma B.1.2. *Let $x^\star$ be an optimal solution to the following optimization problem:*

$$\begin{aligned} x^\star = \ & \underset{x' \in \mathscr{X}^*_{\mathsf{A}}(x)}{\arg\min} \ \ c(x, x') \\ \text{s.t.} \quad & \operatorname{sign}(w^\mathsf{T} x') = 1 \end{aligned}$$

If no solution is returned, we say an $x^\star$ such that $c(x, x^\star) = \infty$ is returned. Define $\Delta(x)$ as follows:

$$\Delta(x) = \begin{cases} x^\star, & \textit{if} \ \ c(x, x^\star) \leq 2 \\ x, & \textit{otherwise} \end{cases}$$

Then $\Delta(x)$ is an optimal solution to the optimization problem in Lemma B.1.1.

Proof. Recall that the utility function of the decision subject is $U(x, x') = f(x') - c(x, x')$, and that, by Lemma B.1.1, they will only modify their features if the utility increases, i.e. if they achieve $f(x') = +1$ and while incurring cost $c(x, x') \leq 2$.

Consider two cases for $x' \neq x$:

1. When $c(x, x') > 2$, there are no feasible points for the optimization problem of Lemma B.1.1.
2. When $c(x, x') \leq 2$, we only need to consider those feature vectors x' that satisfy $f(x') = 1$, because if $f(x') = -1$, the decision subject with features x would prefer not to change anything. Since maximizing $U(x, x') = f(x') - c(x, x')$ is equivalent to minimizing $c(x, x')$ if $f(x') = 1$, we conclude that when $c(x, x') \leq 2$, the optimum of the program of Lemma B.1.1 is the same as the optimum of the program in Lemma B.1.2.

□

Lemma B.1.2 enables us to re-formulate the objective function as follows. Recall that $c(x, x') = \sqrt{(x_{\mathsf{A}} - x_{\mathsf{A}}')^\intercal S^{-1} (x_{\mathsf{A}} - x_{\mathsf{A}}')}$ where S^{-1} is symmetric positive definite. Thus S^{-1} has the following diagonalized form, in which Q is an orthogonal matrix and Λ^{-1} is a diagonal matrix:

$$S^{-1} = Q^\intercal \Lambda^{-1} Q = (\Lambda^{-\frac{1}{2}} Q)^\intercal (\Lambda^{-\frac{1}{2}} Q)$$

With this, we can re-write the cost function as

$$\begin{aligned}
c(x, x') &= \sqrt{(x_{\mathsf{A}} - x_{\mathsf{A}}')^\intercal S^{-1} (x_{\mathsf{A}} - x_{\mathsf{A}}')} \\
&= \sqrt{(x_{\mathsf{A}} - x_{\mathsf{A}}')^\intercal (\Lambda^{-\frac{1}{2}} Q)^\intercal (\Lambda^{-\frac{1}{2}} Q)(x_{\mathsf{A}} - x_{\mathsf{A}}')} \\
&= \sqrt{(\Lambda^{-\frac{1}{2}} Q (x_{\mathsf{A}} - x_{\mathsf{A}}'))^\intercal (\Lambda^{-\frac{1}{2}} Q (x_{\mathsf{A}} - x_{\mathsf{A}}'))} \\
&= \|\Lambda^{-\frac{1}{2}} Q (x_{\mathsf{A}} - x_{\mathsf{A}}')\|_2
\end{aligned}$$

Meanwhile, the constraint in Lemma B.1.2 can be written

$$\begin{aligned}\operatorname{sign}(w \cdot x') &= \operatorname{sign}(w_{\mathsf{A}} \cdot {x_{\mathsf{A}}}' + w_{\mathsf{IM}} \cdot x_{\mathsf{IM}}) \\ &= \operatorname{sign}(w_{\mathsf{A}} \cdot {x_{\mathsf{A}}}' - (-w_{\mathsf{IM}} \cdot x_{\mathsf{IM}})) = 1\end{aligned}$$

Hence the optimization problem can be reformulated as

$$\min_{{x_{\mathsf{A}}}' \in \mathscr{X}_A^*} \|(\Lambda^{-\frac{1}{2}} Q(x_{\mathsf{A}} - {x_{\mathsf{A}}}'))\|_2 \tag{B.1}$$

$$\text{s.t. } \operatorname{sign}(w_{\mathsf{A}} \cdot {x_{\mathsf{A}}}' - (-w_{\mathsf{IM}} \cdot x_{\mathsf{IM}})) = 1 \tag{B.2}$$

The above optimization problem can be further simplified by getting rid of the sign(·):

Lemma B.1.3. *If* ${x_{\mathsf{A}}}^{\mp}$ *is an optimal solution to Equation* (B.1) *under constraint Equation* (B.2)*, then it must satisfy* $w_{\mathsf{A}} \cdot {x_{\mathsf{A}}}^{\mp} - (-w_{\mathsf{IM}} \cdot x_{\mathsf{IM}}) = 0$.

Proof. We prove by contradiction. Let $x_{\mathsf{A}}^{\mp}$ is an optimal solution to Equation (B.1) and suppose towards contraction that $w_{\mathsf{A}} x_{\mathsf{A}}^{\mp} > -w_{\mathsf{IM}} \cdot x_{\mathsf{IM}}$. Since the original feature vector x was classified as -1, we have

$$w_{\mathsf{A}} \cdot x_{\mathsf{A}}^{\mp} > -w_{\mathsf{IM}} \cdot x_{\mathsf{IM}}, \quad w_{\mathsf{A}} \cdot x_{\mathsf{A}} < -w_{\mathsf{IM}} \cdot x_{\mathsf{IM}}$$

By the continuity properties of linear vector space, there exists $\mu \in (0, 1)$ such that:

$$w_{\mathsf{A}} \left(\mu \cdot {x_{\mathsf{A}}}^{\mp} + (1 - \mu) x_{\mathsf{A}}\right) = -w_{\mathsf{IM}} \cdot x_{\mathsf{IM}}$$

Let ${x_{\mathsf{A}}}'' = \mu \cdot {x_{\mathsf{A}}}^{\mp} + (1 - \mu) x_{\mathsf{A}}$. Then $\operatorname{sign}(w_{\mathsf{A}} {x_{\mathsf{A}}}'' - (-w_{\mathsf{IM}} \cdot x_{\mathsf{IM}})) = 1$, i.e., x''_{A} also satisfies the constraint. Since ${x_{\mathsf{A}}}^{\mp}$ is an optimum of Equation (B.1), we have

$$\|\Sigma^{-\frac{1}{2}} Q({x_{\mathsf{A}}}^{\mp} - x_{\mathsf{A}})\| \leq \|\Sigma^{-\frac{1}{2}} Q({x_{\mathsf{A}}}'' - x_{\mathsf{A}})\|$$

However, we also have:

$$\begin{aligned}
\|\Sigma^{-\frac{1}{2}}Q(x_{\mathsf{A}}'' - x_{\mathsf{A}})\| &= \|\Sigma^{-\frac{1}{2}}Q(\mu \cdot x_{\mathsf{A}}{}^{\mp} + (1-\mu)x_{\mathsf{A}} - x_{\mathsf{A}})\| \\
&= \|\Sigma^{-\frac{1}{2}}Q(\mu \cdot (x_{\mathsf{A}}{}^{\mp} - x_{\mathsf{A}}))\| \\
&= \mu\|\Sigma^{-\frac{1}{2}}Q(x_{\mathsf{A}}{}^{\mp} - x_{\mathsf{A}})\| \\
&< \|\Sigma^{-\frac{1}{2}}Q(x_{\mathsf{A}}{}^{\mp} - x_{\mathsf{A}})\|
\end{aligned}$$

contradicting our assumption that $x_{\mathsf{A}}^{\mp}$ is optimal. Therefore $x_{\mathsf{A}}^{\mp}$ must satisfy $w_{\mathsf{A}} x_{\mathsf{A}}^{\mp} = -w_{\mathsf{IM}} \cdot x_{\mathsf{IM}}$. □

As a result of Lemma B.1.3, we can replace the constraint in Equation (B.1) with its corresponding equality constraint without changing the optimal solution.[1] The decision subject's best-response program from Lemma B.1.1 is therefore equivalent to

$$\min_{x_{\mathsf{A}}' \in \mathscr{X}_A^*} \ \|(\Lambda^{-\frac{1}{2}}Q(x_{\mathsf{A}} - x_{\mathsf{A}}'))\|_2 \tag{B.3}$$

$$\text{s.t.} \ \ w_{\mathsf{A}} \cdot x_{\mathsf{A}}' - (-w_{\mathsf{IM}} \cdot x_{\mathsf{IM}}) = 0 \tag{B.4}$$

The following lemma gives us a closed-form solution for the above optimization problem:

Lemma B.1.4. *The optimal solution to the optimization problem defined in Equation* (B.3) *and Equation* (B.4)

has the following closed form:

$$x_{\mathsf{A}}{}^{\mp} = x_{\mathsf{A}} - \frac{w^{\mathsf{T}}x}{w_{\mathsf{A}}{}^{\mathsf{T}} S w_{\mathsf{A}}} S w_{\mathsf{A}}.$$

[1] A similar argument was made by Haghtalab et al. [2020] but here we provide a proof for a more general case, where the objective function is to minimize a weighted norm instead of simply $\|x_{\mathsf{A}} - x_{\mathsf{A}}'\|_2$.

Proof. Notice that the above program has the form

$$\min_{x_{\mathsf{A}}' \in x_{\mathsf{A}}^*} \quad \|A x_{\mathsf{A}}' - b\|_2$$
$$\text{s.t.} \quad C x_{\mathsf{A}}' = d$$

where $A = \Lambda^{-\frac{1}{2}} Q$, $b = \Lambda^{-\frac{1}{2}} Q x_{\mathsf{A}}$, $C = w_{\mathsf{A}}^{\mathsf{T}}$, and $d = -w_{\mathsf{IM}}^{\mathsf{T}} x_{\mathsf{IM}}$. Note the following useful equalities:

$$A^{\mathsf{T}} A = (\Lambda^{-\frac{1}{2}} Q)^{\mathsf{T}} \Lambda^{-\frac{1}{2}} Q = S^{-1}$$
$$(A^{\mathsf{T}} A)^{-1} = S$$
$$A^{\mathsf{T}} b = (\Lambda^{-\frac{1}{2}} Q)^{\mathsf{T}} \Lambda^{-\frac{1}{2}} Q x_{\mathsf{A}} = S^{-1} x_{\mathsf{A}}$$

The above is a norm minimization problem with equality constraints, whose optimum $x_{\mathsf{A}}^{\mp}$ has the following closed form [Boyd and Vandenberghe, 2004]:

$$\begin{aligned}
x_{\mathsf{A}}^{\mp} &= (A^{\mathsf{T}} A)^{-1} \left(A^{\mathsf{T}} b - C^{\mathsf{T}} (C (A^{\mathsf{T}} A)^{-1} C^{\mathsf{T}})^{-1} (C (A^{\mathsf{T}} A)^{-1} A^{\mathsf{T}} b - d) \right) \\
&= S \left(S^{-1} x_{\mathsf{A}} - w_{\mathsf{A}} (w_{\mathsf{A}}^{\mathsf{T}} S w_{\mathsf{A}})^{-1} (w_{\mathsf{A}}^{\mathsf{T}} S (S^{-1} x_{\mathsf{A}}) - (-w_{\mathsf{IM}}^{\mathsf{T}} x_{\mathsf{IM}})) \right) \\
&= x_{\mathsf{A}} - S \left(w_{\mathsf{A}} (w_{\mathsf{A}}^{\mathsf{T}} S w_{\mathsf{A}})^{-1} (w_{\mathsf{A}}^{\mathsf{T}} x_{\mathsf{A}} + w_{\mathsf{IM}}^{\mathsf{T}} x_{\mathsf{IM}}) \right) \\
&= x_{\mathsf{A}} - \frac{w^{\mathsf{T}} x}{w_{\mathsf{A}}^{\mathsf{T}} S w_{\mathsf{A}}} S w_{\mathsf{A}}
\end{aligned}$$

□

We can now compute the cost incurred by an individual with features x who plays their best response $x^{\mp}$:

$$\begin{aligned}
c(x, x\mp) &= \sqrt{(x_{\mathsf{A}} - x_{\mathsf{A}}^{\mp})^{\mathsf{T}} S^{-1} (x_{\mathsf{A}} - x_{\mathsf{A}}^{\mp})} \\
&= \sqrt{\left(\frac{w^{\mathsf{T}} x}{w_{\mathsf{A}}^{\mathsf{T}} S w_{\mathsf{A}}} S w_{\mathsf{A}} \right)^{\mathsf{T}} S^{-1} \left(\frac{w^{\mathsf{T}} x}{w_{\mathsf{A}}^{\mathsf{T}} S w_{\mathsf{A}}} S w_{\mathsf{A}} \right)} \\
&= \frac{|w^{\mathsf{T}} x|}{\sqrt{w_{\mathsf{A}}^{\mathsf{T}} S w_{\mathsf{A}}}}
\end{aligned}$$

Hence an decision subject who was classified as -1 with feature vector x has the unconstrained best response

$$\Delta(x) = \begin{cases} x, & \text{if } \frac{|w^\mathsf{T} x|}{\sqrt{w_\mathsf{A}{}^\mathsf{T} S w_\mathsf{A}}} \geq 2 \\ \left[x_\mathsf{A} - \frac{w^\mathsf{T} x}{w_\mathsf{A}{}^\mathsf{T} S w_\mathsf{A}} S w_\mathsf{A} \mid x_\mathsf{IM}\right], & \text{otherwise} \end{cases}$$

which completes the proof of Theorem 4.4.2.

B.2 Proof of Proposition 4.4.5

Proposition B.2.1 (Correlations between Features May Reduce Cost)**.** *For any cost matrix S^{-1} and any nontrivial classifier h, there exist indices $k, \ell \in [d_\mathsf{A}]$ and $\tau \in \mathbb{R}$ such that every feature vector x has lower best-response cost under the cost matrix $\tilde{S}^{-1}$ given by*

$$\tilde{S}^{-1}_{ij} = \tilde{S}^{-1}_{ji} = \begin{cases} S^{-1}_{ij} + \tau, & \textit{if } i = k, j = \ell \\ S^{-1}_{ij}, & \textit{otherwise} \end{cases}$$

than under S^{-1}; that is, $c_{\tilde{S}^{-1}}(x, \Delta(x)) < c_{S^{-1}}(x, \Delta(x))$ for all x.

Proof. Consider any cost matrix $S^{-1} \in \mathbb{R}^{d_\mathsf{A} \times d_\mathsf{A}}$ and any nontrivial classifier h (i.e. h does not assign every x the same prediction). Since S^{-1} is positive definite, so is its inverse S, and all of their diagonal entries are positive. And since h is nontrivial, it must contain a nonzero coefficient $w_i \neq 0$. Additionally, let w_j be any other coefficient.

Let $\tilde{S}^{-1} = S^{-1} + \tau(e_i e_j^\mathsf{T} + e_j e_i^\mathsf{T})$ for some constant $\tau \in \mathbb{R}$ to be set later. We claim that there exists τ such that the best-response adaptation always costs less under $\tilde{S}^{-1}$ than S^{-1}. To do so, we compute the inverse of $\tilde{S}^{-1}$ and invoke the closed-form cost expression given by Theorem 4.4.2.

To begin computing the inverse, note that by the Sherman-Morrison-Woodbury formula [Golub and Van Loan, 2013],

$$\tilde{S} = \left(\tilde{S}^{-1}\right)^{-1} = S - \tau S \begin{bmatrix} e_i & e_j \end{bmatrix} \left(I + \tau \begin{bmatrix} e_j^\mathsf{T} \\ e_i^\mathsf{T} \end{bmatrix} S \begin{bmatrix} e_i & e_j \end{bmatrix} \right)^{-1} \begin{bmatrix} e_j^\mathsf{T} \\ e_i^\mathsf{T} \end{bmatrix} S \tag{B.5}$$

$$= S - \tau S \begin{bmatrix} e_i & e_j \end{bmatrix} \left(I + \tau \begin{bmatrix} S_{ij} & S_{jj} \\ S_{ii} & S_{ij} \end{bmatrix} \right)^{-1} \begin{bmatrix} e_j^\mathsf{T} \\ e_i^\mathsf{T} \end{bmatrix} S \tag{B.6}$$

$$= S - \tau S \begin{bmatrix} e_i & e_j \end{bmatrix} \left[\tau \left(\frac{1}{\tau} I + \begin{bmatrix} S_{ij} & S_{jj} \\ S_{ii} & S_{ij} \end{bmatrix} \right) \right]^{-1} \begin{bmatrix} e_j^\mathsf{T} \\ e_i^\mathsf{T} \end{bmatrix} S \tag{B.7}$$

$$= S - \tau S \begin{bmatrix} e_i & e_j \end{bmatrix} \tau^{-1} \begin{bmatrix} \frac{1}{\tau} + S_{ij} & S_{jj} \\ S_{ii} & \frac{1}{\tau} + S_{ij} \end{bmatrix}^{-1} \begin{bmatrix} e_j^\mathsf{T} \\ e_i^\mathsf{T} \end{bmatrix} S \tag{B.8}$$

$$= S - S \begin{bmatrix} e_i & e_j \end{bmatrix} \underbrace{\begin{bmatrix} \frac{1}{\tau} + S_{ij} & S_{jj} \\ S_{ii} & \frac{1}{\tau} + S_{ij} \end{bmatrix}}_{T}^{-1} \begin{bmatrix} e_j^\mathsf{T} \\ e_i^\mathsf{T} \end{bmatrix} S \tag{B.9}$$

Clearly, we can ensure that T is invertible by setting τ so that $\det(T) \neq 0$. But as the following lemmas show, we can actually say much more: $\det(T)$ can be made either positive or negative, and moreover, both can be accomplished with a choice of $\tau > 0$ or $\tau < 0$. This flexibility in choosing τ will become crucial later.

First, we need the following useful fact about positive definite matrices:

Lemma B.2.2 (Off-diagonal entries of a positive definite matrix). *If $A \in \mathbb{R}^{n\times n}$ is symmetric positive definite, then for all $i, j \in [n]$, $\sqrt{A_{ii}A_{jj}} > |A_{ij}|$.*

Proof. By positive definiteness, we have, for any nonzero $\alpha, \beta \in \mathbb{R}$,

$$(\alpha e_i + \beta e_j)^\mathsf{T} A (\alpha e_i + \beta e_j) = \alpha^2 A_{ii} + \beta^2 A_{jj} + 2\alpha\beta A_{ij} > 0$$

For a choice of $\alpha = -A_{ij}$ and $\beta = A_{ii}$, we have

$$A_{ij}^2 A_{ii} + A_{ii}^2 A_{jj} - 2A_{ij}^2 A_{ii} = A_{ii}(A_{ii}A_{jj} - A_{ij}^2) > 0$$

Since $A_{ii} > 0$, we must have $A_{ii}A_{jj} - A_{ij}^2 > 0$, from which the claim follows. □

Now we can characterize the possible settings of τ and $\det(T)$:

Lemma B.2.3 (Possible settings of τ). *There exist $\tau_{\max}, \tau_{\min} > 0$ such that the following hold:*

1. $\det(T) > 0$ *for any* $\tau \in \mathbb{R}$ *such that* $\tau_{\max} \geq |\tau| > 0$.

2. $\det(T) < 0$ *for any* $\tau \in \mathbb{R}$ *such that* $\tau_{\min} \leq |\tau|$.

Proof. To prove the first claim, note that having

$$\det(T) = \left(\frac{1}{\tau} + S_{ij}\right)^2 - S_{ii}S_{jj} > 0$$

is equivalent to

$$\left|\frac{1}{\tau} + S_{ij}\right| > \sqrt{S_{ii}S_{jj}}$$

It suffices to choose τ such that

$$\left|\frac{1}{\tau}\right| - |S_{ij}| > \sqrt{S_{ii}S_{jj}}$$
$$\frac{1}{|\tau|} > \sqrt{S_{ii}S_{jj}} + |S_{ij}|$$

So any τ such that $0 < |\tau| < \left(\sqrt{S_{ii}S_{jj}} + |S_{ij}|\right)^{-1}$ results in $\det(T) > 0$. Analogously, for the second claim, a sufficient condition for $\det(T) < 0$ is that

$$\frac{1}{|\tau|} < \sqrt{S_{ii}S_{jj}} - |S_{ij}|$$

By Lemma B.2.2, the right-hand side is positive. Hence it suffices to pick any τ such that

$$|\tau| > \left(\sqrt{S_{ii}S_{jj}} - |S_{ij}|\right)^{-1}.$$

□

With this lemma in place, we can describe the difference between the inverses of S^{-1} and $\tilde{S}^{-1}$. Denote this matrix by $E = S - \tilde{S}$. We show the following:

Lemma B.2.4 (Difference between inverse cost matrices)**.** *The k, ℓ-th entry of E has the following form:*

$$E_{k\ell} = \frac{1}{\det(T)} \left(E'_{k\ell} + \frac{1}{\tau} E''_{k\ell} \right)$$

where $E'_{k\ell}$ and $E''_{k\ell}$ do not depend on τ.

Proof. Assume that τ has been chosen so that $\det(T) \neq 0$, as Lemma B.2.3 showed to be possible. We then have

$$T^{-1} = \frac{1}{\det(T)} \begin{bmatrix} \frac{1}{\tau} + S_{ij} & -S_{jj} \\ -S_{ii} & \frac{1}{\tau} + S_{ij} \end{bmatrix}$$

Thus continuing from equation B.9, we have

$$\tilde{S} = S - \frac{1}{\det(T)} S \underbrace{\begin{bmatrix} e_i & e_j \end{bmatrix} \begin{bmatrix} \frac{1}{\tau} + S_{ij} & -S_{jj} \\ -S_{ii} & \frac{1}{\tau} + S_{ij} \end{bmatrix} \begin{bmatrix} e_j^{\mathsf{T}} \\ e_i^{\mathsf{T}} \end{bmatrix}}_{V} S$$

It can be verified that V is a $d_{\mathsf{A}} \times d_{\mathsf{A}}$ matrix whose only nonzero entries are

$$V_{ii} = -S_{jj}, \qquad V_{jj} = -S_{ii}, \qquad V_{ij} = V_{ji} = \frac{1}{\tau} + S_{ij}$$

Next we evaluate the $d_{\mathsf{A}} \times d_{\mathsf{A}}$ matrix SVS. For any $k, \ell \in [d_{\mathsf{A}}]$, we have

$$\begin{aligned}
(SVS)_{k\ell} &= \sum_{i'=1}^{d_{\mathsf{A}}} \sum_{j'=1}^{d_{\mathsf{A}}} S_{ki'} V_{i'j'} S_{j'\ell} \\
&= S_{ki} V_{ii} S_{i\ell} + S_{ki} V_{ij} S_{j\ell} + S_{kj} V_{ji} S_{i\ell} + S_{kj} V_{jj} S_{j\ell} \\
& \qquad\qquad\qquad\qquad\qquad\qquad ((V \text{ has four nonzero entries})) \\
&= V_{ii} S_{ki} S_{i\ell} + V_{jj} S_{kj} S_{j\ell} + V_{ij}(S_{ki} S_{j\ell} + S_{kj} S_{i\ell}) \qquad ((V_{ij} = V_{ji})) \\
&= -S_{jj} S_{ki} S_{i\ell} - S_{ii} S_{kj} S_{j\ell} + \left(\frac{1}{\tau} + S_{ij}\right)(S_{ki} S_{j\ell} + S_{kj} S_{i\ell}) \\
&= \underbrace{-S_{jj} S_{ki} S_{i\ell} - S_{ii} S_{kj} S_{j\ell} + S_{ij}(S_{ki} S_{j\ell} + S_{kj} S_{i\ell})}_{E'_{k\ell}} + \frac{1}{\tau} \underbrace{(S_{ki} S_{j\ell} + S_{kj} S_{i\ell})}_{E''_{k\ell}}
\end{aligned}$$

which proves the claim. □

We now compute the marginal best-response cost incurred due to the difference between the inverse cost matrices, $E = S - \tilde{S}$. We have

$$\begin{aligned}
{w_{\mathsf{A}}}^{\mathsf{T}} E w_{\mathsf{A}} &= \sum_{k=1}^{d_{\mathsf{A}}} \sum_{\ell=1}^{d_{\mathsf{A}}} w_k w_\ell E_{k\ell} \\
&= \frac{1}{\det(T)} \sum_{k=1}^{d_{\mathsf{A}}} \sum_{\ell=1}^{d_{\mathsf{A}}} w_k w_\ell \left(E'_{k\ell} + \frac{1}{\tau} E''_{k\ell}\right) \qquad \text{(by Lemma B.2.4)} \\
&= \frac{1}{\det(T)} \left[\underbrace{\sum_{k=1}^{d_{\mathsf{A}}} \sum_{\ell=1}^{d_{\mathsf{A}}} w_k w_\ell E'_{k\ell}}_{E'} + \frac{1}{\tau} \underbrace{\sum_{k=1}^{d_{\mathsf{A}}} \sum_{\ell=1}^{d_{\mathsf{A}}} w_k w_\ell E''_{k\ell}}_{E''} \right]
\end{aligned}$$

By Lemma B.2.3, there exists $\tau \neq 0$ such that

$$\operatorname{sign}(\det(T)) = -\operatorname{sign}(E') \quad \text{and} \quad \operatorname{sign}(\tau) = -\operatorname{sign}(\det(T)) \cdot \operatorname{sign}(E'')$$

Such a choice of τ results in ${w_{\mathsf{A}}}^{\mathsf{T}} E w_{\mathsf{A}} < 0$. Finally by Theorem 4.4.2, we have for all x that

$$\begin{aligned}
c_{\tilde{S}^{-1}}(x, \Delta_{\tilde{S}^{-1}}(x)) = \frac{|w^{\mathsf{T}} x|}{\sqrt{{w_{\mathsf{A}}}^{\mathsf{T}} \tilde{S} w_{\mathsf{A}}}} &= \frac{|w^{\mathsf{T}} x|}{\sqrt{{w_{\mathsf{A}}}^{\mathsf{T}} S w_{\mathsf{A}} - {w_{\mathsf{A}}}^{\mathsf{T}} E w_{\mathsf{A}}}} \\
&< \frac{|w^{\mathsf{T}} x|}{\sqrt{{w_{\mathsf{A}}}^{\mathsf{T}} S w_{\mathsf{A}}}} = c_{S^{-1}}(x, \Delta_{S^{-1}}(x))
\end{aligned}$$

which completes the proof. □

B.3 Derivations For The Model Designer's Objective Function

Now that we have obtained a closed-form expression for both the unconstrained and improving best response from the decision subjects, we can analyze the objective function for the model designer and the model that would be deployed at equilibrium. Recall that the objective function for the model designer is

$$\min_{w \in \mathbb{R}^{d+1}} \quad \mathbb{E}_{x \sim \mathscr{D}} \left[\mathbb{1}(h(\Delta_{\mathsf{M}}(x)) \neq y) \right] + \lambda \mathbb{E}_{x \sim \mathscr{D}} \left[\mathbb{1}(h(\Delta_{\mathsf{I}}(x)) \neq +1) \right]$$

By Theorem 4.4.2, $h(\Delta_{\mathsf{M}}(x))$ has the closed form

$$\begin{aligned} h(\Delta_{\mathsf{M}}(x)) &= \begin{cases} +1 & \text{if } \; w \cdot x \geq -2\sqrt{{w_{\mathsf{M}}}^{\mathsf{T}} S_{\mathsf{M}} w_{\mathsf{M}}} \\ \\ -1 & \text{otherwise} \end{cases} \\ &= 2 \cdot \mathbb{1} \left[w \cdot x \geq -2\sqrt{{w_{\mathsf{M}}}^{\mathsf{T}} S_{\mathsf{M}} w_{\mathsf{M}}} \right] - 1 \end{aligned}$$

and similarly,

$$h(\Delta_{\mathsf{I}}(x)) = 2 \cdot \mathbb{1} \left[w \cdot x \geq -2\sqrt{{w_{\mathsf{I}}}^{\mathsf{T}} S_{\mathsf{I}} w_{\mathsf{I}}} \right] - 1$$

The model designer's objective can then be re-written as follows:

$$\begin{aligned} & \mathbb{E}_{x \sim D} \left[\mathbb{1}[h(\Delta_{\mathsf{M}}(x)) \neq y] + \lambda \mathbb{1}[h(\Delta_{\mathsf{I}}(x)) \neq +1] \right] \\ =& \mathbb{E}_{x \sim \mathscr{D}} \left[1 - \frac{1}{2}(1 + h(\Delta_{\mathsf{M}}(x)) \cdot y) + \lambda(1 - \frac{1}{2}(1 + h(\Delta_{\mathsf{I}}(x)) \cdot 1)) \right] \\ =& \mathbb{E}_{x \sim \mathscr{D}} \left[\frac{1}{2}(1 + \lambda) - \frac{1}{2} h(\Delta_{\mathsf{M}}(x)) \cdot y - \frac{\lambda}{2} h(\Delta_{\mathsf{I}}(x)) \right] \end{aligned}$$

Removing the constants, the objective function becomes:

$$
\begin{aligned}
&\min_{w} \mathbb{E}_{x\sim\mathscr{D}} \left[\lambda - h(\Delta_{\mathsf{M}}(x)) \cdot y - \lambda h(\Delta_{\mathsf{I}}(x))\right] \\
=&\min_{w} \mathbb{E}_{x\sim\mathscr{D}} \Bigg[- \Big(2 \cdot \mathbb{1}\left[w \cdot x \geq -2\sqrt{w_{\mathsf{M}}{}^{\mathsf{T}} S_{\mathsf{M}} w_{\mathsf{M}}}\right] - 1\Big) \cdot y(x) \\
&\qquad - 2\lambda \cdot \mathbb{1}\left[w \cdot x \geq -2\sqrt{w_{\mathsf{I}}{}^{\mathsf{T}} S_{I} w_{\mathsf{I}}}\right] \Bigg]
\end{aligned}
$$

Re-organizing the above equations, we can turn the model designer's *constrained* optimization problem in equation 4.7 into the following *unconstrained* problem:

$$
\min_{w\in\mathbb{R}^d} \mathbb{E}_{x\sim\mathscr{D}} \Big[- \Big(2 \cdot \mathbb{1}\left[w^{\mathsf{T}} x \geq -2\sqrt{\Omega_{\mathsf{M}}}\right] - 1\Big) \cdot y - 2\lambda \cdot \mathbb{1}\left[w^{\mathsf{T}} x \geq -2\sqrt{\Omega_{\mathsf{I}}}\right] \Big] \tag{B.10}
$$

The optimization problem in equation B.10 is intractable since both the objective and the constraints are non-convex. To overcome this difficulty, we train our classifier by replacing the 0-1 loss function with a convex surrogate loss $\sigma(x) = \log\left(\frac{1}{1+e^{-x}}\right)$. This results in the following ERM problem:

$$
\min_{w\in\mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \Big[-\sigma\Big(y_i \cdot (w^{\mathsf{T}} x_i + 2\sqrt{\Omega_{\mathsf{M}}})\Big) - \lambda \cdot \sigma(w^{\mathsf{T}} x_i + 2\sqrt{\Omega_{\mathsf{I}}})\Big] \tag{B.11}
$$

Conditionally Actionable Features In practice, individuals can often only change some features in either a positive or negative direction, but not both. However, modeling this restriction on the decision subject's side precludes a closed-form solution. Instead, we strongly disincentivize such moves in the model designer's objective function. The idea is that if the model designer is punished for encouraging an illegal action, the announced classifier will not incentivize such moves from decision subjects. The result is that decision subjects encounter an *implicit* direction constraint on the relevant variables. To that end, we construct a vector $\mathsf{dir} \in \{-1, 0, +1\}^d$ where dir_i represents the prohibited direction of change for the corresponding feature x_i; that is, $\mathsf{dir}_i = +1$ if x_i should not be allowed to increase, -1 if it should not decrease, and 0 if there are no direction constraints. We then append the following penalty term to the model designer's objective in Equation (4.7):

$$-\eta \cdot \sum_{i=1}^{d} \max(\mathsf{dir}_i \cdot (\Delta(x) - x)_i, 0) \tag{B.12}$$

where $\eta > 0$ is a hyperparameter representing the weight given to this penalty term. Equation (B.12) penalizes the weights of partially actionable features so that decision subjects would prefer to move towards a certain direction.

B.4 Additional Experimental Details and Results

In this section, we provide additional experimental information and results.

B.4.1 Basic Information Of Each Dataset.

Table B.1: Basic Information Of Each Dataset.

Dataset	Size	Dimension	Prediction Task
`credit`	$20,000$	16	To predict if a person can repay their credit card loan.
`adult`	$48,842$	14	To predict whether income exceeds $50K/yr$ based on census data.
`german`	$1,000$	26	To predict whether a person is good or bad credit risk.
`spam`	4601	57	To predict if an email is a spam or not.

B.4.2 Additional Experimental Results for Non-Linear Models

We also work with a three-layer neural network to validate the effectiveness of the oracle best response in Algorithm 1. We note that the LIME program needs to learn a local linear model for each instance, which is very time-consuming. Therefore, we downsample only 10% of data examples from the `credit` dataset. We follow the same setting as the linear classifier experiments. We compare our method with the static classifier in Table B.2. We find out for this non-linear model setting, our approach has a higher improvement rate while preventing manipulations with the deploy error 27.72% vs. 35.64%.

Table B.2: Performance Metrics for Non-linear Models.

	Methods	
Metrics	ST	CA
test error	30.72%	30.01%
deployment error	35.64%	27.72%
improvement rate	0.99%	2.97%

Appendix C

Appendix for Chapter 5

C.1 Notation Table for Chapter 5

Symbol	Usage
$\mathscr{X} \subset \mathbb{R}^d$	The domain of the feature $\mathbf{x}$
$\mathscr{Y} \equiv \{0,1\}$	The domain of labels
$\mathbf{X} \in \mathbb{R}^{n \times d}$	A set of features of n agents
$Y \in \{0,1\}^{\lvert\mathbf{X}\rvert}$	The labels for the set of features $\mathbf{X}$
$\mathbf{x} \in \mathscr{X}$	A random variable representing an example's features.
$y \in \mathscr{Y}$	A random variable representing an example's *ground truth label*
$f : \mathscr{X} \rightarrow \mathscr{Y}$	a binary classifier, unknown to the agents
$\mathscr{X}_-, \mathscr{X}^{(0)} \subseteq \mathscr{X}$	The domain of negatively classified features, i.e. $\forall \mathbf{x} \in \mathscr{X}_-$, $f(\mathbf{x}) = 0$
$\mathscr{X}_+ \subseteq \mathscr{X}$	The domain of positively classified features, i.e., $\forall \mathbf{x} \in \mathscr{X}_+, f(\mathbf{x}) = 1$
$\mathbf{X}_- \subseteq \mathscr{X}$	The set of negatively classified features, i.e. $\forall \mathbf{x} \in \mathbf{X}_-$, $f(\mathbf{x}) = 0$
$\mathbf{X}_+ \subseteq \mathscr{X}$	The set of positively classified features, i.e., $\forall \mathbf{x} \in \mathbf{X}_+, f(\mathbf{x}) = 1$
$\mathbf{X}^{(g_i)} \subseteq \mathbf{X}$	The subset of features belongs to group $G = g_i$
$c_R : \mathscr{X} \times \mathscr{X} \rightarrow \mathbb{R}_+$	The cost function of recourse
$c_M : \mathscr{X} \times \mathscr{X} \rightarrow \mathbb{R}_+$	The cost function of recourse
$\mathbf{X}_R$	The set of all possible recourse actions
$\mathbf{Z}_R$	The set of revealed recourse actions
$\mathbf{Z}_+$	The set of revealed positively classified features
$\mathbf{Z} = \mathbf{Z}_R \cup \mathbf{Z}_+$	A publicly revealed feature set
$\mathbf{x}_R(\mathbf{x})$	The optimal recourse action for agent with feature $\mathbf{x}$
$\mathbf{x}_M(\mathbf{x})$	The optimal manipulation action for agent with feature $\mathbf{x}$
$\mathbf{z}(\mathbf{x}, \mathbf{Z})$	The agent's final action
$\text{rec}(\mathbf{Z}, \mathbf{X})$	The recourse ratio for feature sets $\mathbf{X}$ given revealed set is $\mathbf{Z}$
$\alpha \in [0,1]$	A subsidy level
$u_0 \in \mathbb{R}$	The initial utility of a system without providing recourse.

Table C.1: Primary Notation

C.2 ILP for system when $p = 1$

We provide the ILP formula for the system to find optimal recourse actions when the revealing probability $p = 1$:

$$\max_{\mathbf{a} \in \{0,1\}^{\lvert\mathbf{Z}_{\max}\rvert}, \mathbf{b} \in \{0,1\}^{\lvert\mathbf{X}_-\rvert}} \sum_{j=1}^{\lvert\mathbf{X}_-\rvert} b_j$$

(maximize the number of agents performing recourse)

$$\text{s.t.} \quad b_j c_R(\mathbf{x}_j, \mathbf{z}_R) \leq a_i c_M(\mathbf{x}_j, \mathbf{z}_i) + (1 - a_i)$$

(only do recourse if all manipulation costs are greater)

$$b_j \leq a_{j_R}$$

C.3 Proofs for Theorem 5.4.1

Proof. To demonstrate the intractability of this objective, we reduce from the known NP-hard problem Minimum k-Union (MkU), an instance of which is defined via a universe of n elements $U = \{s_1, \ldots, s_n\}$,a collection of n sets $\mathbf{S} = \{S_1, \ldots S_m\}$ with elements in U, and a budget k. The objective in MkU is to select an index set I of size exactly k such that $\cup_{j \in I} S_j$ is minimized. Given an instance of MkU can be mapped to an instance of simultaneous recourse as follows. Let $\mathbf{X}^{(0)} \times \mathbf{Z} = \big\{(\mathbf{x}, \mathbf{z}_j) : s_i \in U \text{ and } S_j \in \mathbf{S}\big\}$, and define c_R and c_M as follows,

$$c_R(\mathbf{x}, \mathbf{z}_j) = \begin{cases} 1 & \text{if } \ i \neq j \\ \\ 0 & \text{if } \ i = j \end{cases} \qquad c_M(\mathbf{x}, \mathbf{z}_j) = \begin{cases} 1 & \text{if } \ s_i \notin S_j \\ \\ 1/2 & \text{if } \ s_i \in S_j \end{cases}$$

Under this construction of the cost functions, each agent $\mathbf{x}$ will perform recourse if and only if $\mathbf{z}_i$ is revealed, and the disclosure probability $p = 1$. In the case that $\mathbf{z}_i$ is not revealed, the agent will elect to perform manipulation when any $\mathbf{z}_j$ is revealed where $j \neq i$ and $s_i \in S_j$. If neither criterion is met, the agent will elect to do nothing (remaining negatively classified). Combining these cases, we see that revealing each $\mathbf{z}_j$ causes exactly one agent to perform recourse, namely $\mathbf{x}_j$, and causes all $\mathbf{x}$ (with $s_i \in S_j$) to manipulate. Let $I = \{j_1, \ldots, j_k\}$ be the index set of the revealed features, then the number of agents manipulating is equal to $\big| \cup_{j \in I} S_j \big| - k$. Therefore providing k recourse actions to agents while minimizing the number of agents manipulating is equivalent to minimizing $\big| \cup_{j \in I} S_j \big|$. □

C.4 Proof for Theorem 5.4.2

Proof. Given a revealed set $\mathbf{Z} \subseteq \mathbf{X}_R$, for agent with feature $\mathbf{x} \in \mathbf{X}_-$, let $S_m(\mathbf{x}, \mathbf{Z}) := \{z \in \mathbf{Z} : c_M(\mathbf{x}, z) \leq c_R(\mathbf{x}, \mathbf{z}_R(\mathbf{x}, \mathbf{Z}))\}$ be the set of manipulation features that are cheaper than the minimum recourse action $\mathbf{z}_R(x, \mathbf{Z})$ given the revealed set $\mathbf{Z}$. Then the agent will perform recourse if and only if $S_m(\mathbf{x}, \mathbf{Z}) = \emptyset$. Given the cost function c_M and c_R, the principal can pre-compute each agent's manipulation set $S_m(\mathbf{x}, \mathbf{Z})$.

The probability for the manipulation set $S_m(\mathbf{x}, \mathbf{Z})$ to overlap with a given revealed set $\mathbf{Z}$ is $P(\mathbf{x}; \mathbf{Z}) = \Pi_{z \in \mathbf{Z}_m(\mathbf{x},\mathbf{Z})}(1-p)$, where p is the disclosure probability for any criteria $\mathbf{z}$.

The goal for the system is to select a disclosure set $\mathbf{Z} \subseteq \mathbf{X}_R$ to *minimize* the overlap between $\mathbf{Z}$ and $S_m(\mathbf{x}, \mathbf{Z})$ for all agents, namely:

$$\min_{\mathbf{Z} \subseteq \mathbf{X}_R} u(\mathbf{Z}, \mathbf{X}_-) := \sum_{\mathbf{x} \in \mathbf{X}_-} (1 - P(\mathbf{x}; \mathbf{Z})) = \sum_{\mathbf{x} \in \mathbf{X}_-} \left(1 - \Pi_{z \in \mathbf{Z}_m(\mathbf{x},\mathbf{Z})}(1-p)\right) \tag{C.1}$$

To ease the notation, we use $u(\mathbf{Z})$ to shorthand $u(\mathbf{Z}, \mathbf{X}_-)$ since $\mathbf{X}_-$ is fixed in our setting. To show that Equation (C.1) is submodular, it is equivalent to prove that the objective function $u(\mathbf{Z}, \mathscr{X}_-)$ satisfies the *diminishing returns property*, which means $\forall A, B \subseteq \mathbf{Z}$ with $A \subseteq B \subseteq \mathbf{Z}$, and any criteria $z \in \mathbf{Z} \backslash B$, we want to show

$$u(A \cup \{z\}) - u(A) \geq u(B \cup \{z\}) - u(B)$$

Only four types of agents could potentially contribute to the marginal gain for U when the revealed sets are $A \cup \{z\}$ v.s. $B \cup \{z\}$:

1. when $S_m(\mathbf{x}, B \cup \{z\}) = B \cup \{z\}$

2. when $S_m(\mathbf{x}, B \cup \{z\}) = A \cup \{z\}$

3. when $S_m(\mathbf{x}, B \cup \{z\}) = \{z\}$

4. when $S_m(\mathbf{x}, B \cup \{z\}) = B \backslash A \cup \{z\}$

For the first three cases, we can verify that the two marginal gains are the same. For the last case, the two marginal gains are:

$$\begin{aligned}
u(A + \{z\}) - u(A) &= [1 - (1 - p)] - 0 = p \\
u(B + \{z\}) - u(B) &= [1 - \Pi_{t \in \{B \backslash A \cup \{z\}\}}(1 - p)] - [1 - \Pi_{t \in \{B \backslash A\}}(1 - p)] \\
&= p \times \Pi_{t \in \{B \backslash A\}}(1 - p) \\
&\leq p
\end{aligned}$$

Since this holds for all agents, we show that adding a criterion z to a larger set B provides an equal or smaller marginal gain in the objective function compared to adding it to a smaller set A, satisfying the diminishing returns property. Therefore, the objective function defined in Equation (C.1) is submodular.

□

C.5 Proof for Theorem 5.6.3

Proof. Again, consider a 1-dimensional setting, where the system uses a linear threshold classifier $f(x) = \mathbb{1}[x \geq \tau]$. In this case, the optimal recourse action for any agent is always the minimum recourse actions that has been revealed so far, namely $z_{\min} = \min_{z \in \mathbf{Z}} z$. Recall the definition of the social cost with subsidy level α:

$$\text{cost}(\mathbf{Z}, \mathbf{X}_{-}; \alpha) = \sum_{\mathbf{x} \in \mathbf{X}_{-}} \big(c_R(\mathbf{x}, \mathbf{z}_R(\mathbf{x}, \mathbf{Z}; \alpha); \alpha) - c_R(\mathbf{x}, \mathbf{z}_R)\big),$$

$$\text{where } \mathbf{z}_R(\mathbf{x}, \mathbf{Z}; \alpha) = \arg\min_{\mathbf{z} \in \mathbf{Z}} (1 - \alpha) c_R(\mathbf{x}, \mathbf{z})$$

In the 1-dimension case, we have

$$\begin{aligned}
c_R(\mathbf{x}, \mathbf{z}_R(\mathbf{x}, \mathbf{Z}; \alpha); \alpha) &= (1-\alpha) \cdot w_R \cdot \|\mathbf{x} - \mathbf{z}_R(\mathbf{x}, \mathbf{Z}; \alpha)\| \\
&= (1-\alpha) \cdot w_R \cdot \min_{z \in \mathbf{Z}} \|x - z\| = (1-\alpha) \cdot w_R \cdot \left(\min_{z \in \mathbf{Z}} z - x \right) \\
c_R(\mathbf{x}, \mathbf{z}_R; \alpha) &= (1-\alpha) \cdot w_R \cdot \|\mathbf{x} - \mathbf{z}_R\| \\
&= (1-\alpha) \cdot w_R \cdot \|\mathbf{x} - \tau\| \\
&= (1-\alpha) \cdot w_R \cdot (\tau - x)
\end{aligned}$$

Thus,

$$\begin{aligned}
\mathrm{cost}(\mathbf{Z}, \mathbf{X}_-; \alpha) &= \sum_{\mathbf{x} \in \mathbf{X}_-} \left(c_R(\mathbf{x}, \mathbf{z}_R(\mathbf{x}; \mathbf{Z}; \alpha)) - c_R(\mathbf{x}, \mathbf{x}_R; \alpha)\right) \\
&= \sum_{\mathbf{x} \in \mathbf{X}_-} \left[(1-\alpha) \cdot w_R \cdot \left(\min_{z \in \mathbf{Z}} z - x \right) - (1-\alpha) \cdot w_R \cdot (\tau - x) \right] \\
&= (1-\alpha) \cdot |\mathbf{X}_-| \cdot w_R \cdot \left(\min_{z \in \mathbf{Z}} z - \tau \right)
\end{aligned}$$

As the level of subsidy gets larger (α gets bigger, cheaper to perform recourse), $\mathrm{cost}(\mathbf{Z}, \mathbf{X}_-; \alpha)$ will get smaller, which corresponds to a smaller social cost.

□

C.6 Proof for Theorem 5.6.4

Proof. The system utility is defined as the difference between true positive and false positive *after* agent's actions. Let $\Pr[Y = 1|X = x]$ be the true qualification rate given a feature $X = x$, and assume it's also monotonic in X. u_0 is the system's initial utility (before providing recourse).

Let the recourse region R_R and manipulation region R_M are defined as:

$$R_M = \{x \in \mathbf{X}_- : c_M(x, z_{\min}) < \min(1, c_R(x, z_{\min}))\}$$

$$R_R = \{x \in \mathbf{X}_- : c_R(x, z_{\min}) < \min(1, c_M(x, z_{\min}))\}$$

where $\mathbf{X}_-$ is the set of negatively classified agents. Then we have

System's utility($z_{\min}$)

$$= \mathsf{TP} - \mathsf{FP}$$

$$= u_0 + \underbrace{\int_{x \in R_M} \Pr(y = 1|X = x)dx}_{\mathsf{TP}\text{ from agents taking manipulation}} + \underbrace{\int_{x \in R_R} \Pr(y = 1|X = z_{min})dx}_{\mathsf{TP}\text{ from agents taking recourse}}$$

$$- \underbrace{\int_{x \in R_M} (1 - \Pr(y = 1|X = x))\, dx}_{\mathsf{FP}\text{ from agents taking manipulation}} - \underbrace{\int_{x \in R_R} (1 - \Pr(y = 1|X = z_{min}))dx}_{\mathsf{FP}\text{ from agents taking recourse}}$$

$$= u_0 + \int_{x \in R_M} (2 \cdot \Pr(y = 1|X = x) - 1)\, dx + \int_{x \in R_R} (2 \Pr(y = 1|X = z_{\min}) - 1)\, dx$$

$$= u_0 + \int_{x \in R_M} (2 \cdot \Pr(y = 1|X = x) - 1)\, dx + (2 \Pr(y = 1|X = z_{\min}) - 1) \int_{x \in R_R} dx$$

where $z_{\min} = \arg\min_{z \in \mathbf{Z}} z$ is the cheapest recourse actions.

Useful facts:

1. Suppose the classifier is a threshold classifier: $f = \mathbb{I}[x \geq \theta]$, we can further characterize the $\mathscr{X}^{(0)} = \{x \in \mathscr{X} : x \leq \theta\}$.

2. the minimum value of $z_{\min}$ is θ (the decision boundary).

3. Since $\Pr[y = 1|X = x]$ is monotonic in x, $\forall x \in R_M, \Pr[y = 1|X = x] \leq \Pr[y = 1|X = \mathbf{z}_{min}]$

When we change the subsidy level α, the two regions change as:

$$\mathscr{X}_M^{(\alpha)} = \{x \in \mathbf{X}^{(0)} : c_M(x, z_{\min}) < \min(1, c_R(x, z_{\min}; \alpha))\}$$

$$R_R^{(\alpha)} = \{x \in \mathbf{X}^{(0)} : c_R(x, z_{\min}; \alpha) < \min(1, c_M(x, z_{\min}; \alpha))\}$$

where $c_R(x, x'; \alpha) = (1 - \alpha) \cdot c_R(x, x')$. As α becomes larger, we should expect $|\mathscr{X}_R^{(\alpha)}|$ to be larger and $|\mathscr{X}_M^{(\alpha)}|$ to be smaller.

When $c_R(x, x')$ and $c_M(x, x')$ are both monotonic in $\|x - x'\|$ and only cross once. wlog, assume

$$c_M(x, x') = \|x - x'\|, c_R(x, x'; \alpha) = \alpha \cdot w_R \cdot \|x - x'\| + b \quad (0 < w_R \leq 1, b < 1)$$

(to guarantee they only cross once)

we can further characterize the two regions:

$$\mathscr{X}_M^{(a)} = \{x : x \in [z_{\min} - \sqrt{\frac{b}{1 - \alpha \cdot w_R}}, \theta]\},$$
$$\mathscr{X}_R^{(a)} = \{x : x \in [z_{\min} - \sqrt{\frac{1 - b}{\alpha \cdot w_R}}, z_{\min} - \sqrt{\frac{b}{1 - \alpha \cdot w_R}}]\}$$

which gives us the size for the two regions as:

$$\left|\mathscr{X}_M^{(a)}\right| = \theta - z_{\min} + \sqrt{\frac{b}{1 - \alpha \cdot w_R}}, \quad \left|\mathscr{X}_R^{(a)}\right| = \sqrt{\frac{1 - b}{\alpha \cdot w_R}} - \sqrt{\frac{b}{1 - \alpha \cdot w_R}}$$

For $\alpha \in [0, 1]$, the rate in which the size of $\mathscr{X}_M^{(a)}$ and $\mathscr{X}_R^{(a)}$ changes as a function of the subsidy level α can be expressed as:

$$\frac{\partial |\mathscr{X}_M^{(a)}|}{\partial \alpha} = \frac{1}{2} \cdot b^{1/2} \cdot w \cdot (1 - aw)^{-3/2},$$
$$\frac{\partial |\mathscr{X}_R^{(a)}|}{\partial \alpha} = -\frac{1}{2}\sqrt{\frac{1 - b}{w}} \cdot a^{-3/2} - \frac{1}{2} \cdot b^{1/2} \cdot w \cdot (1 - aw)^{-3/2}$$

we can see the increase rate in the size of $R_R^{(\alpha)}$ is higher than the decrease rate in the size of $R_M^{(\alpha)}$. This, together with the fact that useful fact (3), tell us that the system's utility will be a monotonically increasing function in subsidy level α. □

C.7 Proof for Theorem 5.6.6

Proof. Recall from the proof for the recourse rate with subsidy, for a particular reveal set $\mathbf{Z}$ and a given set of negatively classified feature set $\mathbf{X}_-$, we have:

$$\operatorname{rec}(\mathbf{Z}, \mathbf{X}_-; \alpha) = \frac{\sum_{\mathbf{x}\in\mathbf{X}_-} \mathbb{1}\left[\alpha \geq 1 - \frac{\min\left(1, \min_{\mathbf{z}''\in\mathbf{Z}} c_M(\mathbf{x},\mathbf{z}'')\right)}{\min_{\mathbf{z}'\in\mathbf{Z}} c_R(\mathbf{x},\mathbf{z}')}\right]}{|\mathbf{X}_-|}$$

To ease the notation, let's define $\gamma(x) = \frac{\sum_{\mathbf{x}\in\mathbf{X}_-} \mathbb{1}\left[\alpha \geq 1 - \frac{\min\left(1, \min_{\mathbf{z}''\in\mathbf{Z}} c_M(\mathbf{x},\mathbf{z}'')\right)}{\min_{\mathbf{z}'\in\mathbf{Z}} c_R(\mathbf{x},\mathbf{z}')}\right]}{|\mathbf{X}_-|}$. Plug the expression into the definition for the disparity in recourse ratio for two groups g_0, g_1, we have:

$$\begin{aligned}
\mathit{Diff}^{(\mathrm{rec})}(\mathbf{Z}, \mathbf{X}_-^{(g_0)}, \mathbf{X}_-^{(g_1)}) &= \left|\operatorname{rec}(\mathbf{Z}, \mathbf{X}_-^{(g_1)}, \alpha) - \operatorname{rec}(\mathbf{Z}, \mathbf{X}_-^{(g_0)}, \alpha)\right| \\
&= \left|\frac{\sum_{\mathbf{x}\in\mathbf{X}_-^{(g_1)}} \mathbb{1}\left[\alpha \geq 1 - \gamma(x)\right]}{|\mathbf{X}_-^{(g_1)}|} - \frac{\sum_{\mathbf{x}\in\mathbf{X}_-^{(g_0)}} \mathbb{1}\left[\alpha \geq 1 - \gamma(x)\right]}{|\mathbf{X}_-^{(g_0)}|}\right|
\end{aligned}$$

when the size of the two groups are similar, namely when $|\mathbf{X}_-^{(g_0)}| \approx |\mathbf{X}_-^{(g_1)}|$, we can roughly approximate the recourse difference by:

$$\mathit{Diff}^{(\mathrm{rec})}(\mathbf{Z}, \mathbf{X}_-^{(g_0)}, \mathbf{X}_-^{(g_1)}, \alpha) \approxeq \left|\sum_{\mathbf{x}\in\mathbf{X}_-^{(g_1)}} \mathbb{1}\left[\alpha \geq 1 - \gamma(x)\right] - \sum_{\mathbf{x}\in\mathbf{X}_-^{(g_0)}} \mathbb{1}\left[\alpha \geq 1 - \gamma(x)\right]\right|$$

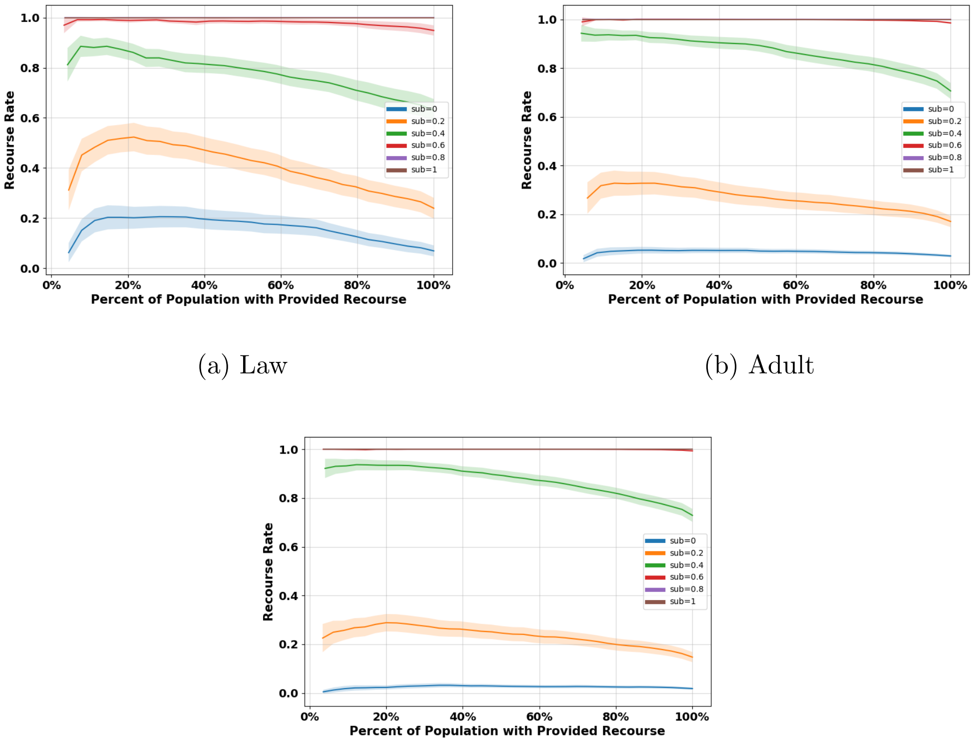
We make the following observation:

- When $\alpha = 0$: it corresponds to the situation where no subsidy is provided. This is the original disparity $\mathit{Diff}^{(\mathrm{rec})}(\mathbf{Z}, \mathbf{X}_-^{(g_0)}, \mathbf{X}_-^{(g_1)})$.
- When $\alpha = \alpha_{\max} = 1$, it corresponds to when the cost of recourse is 0, in this case, everyone takes recourse, which means the recourse difference is zero. Since $1 - \gamma(x) \leq 1 = \alpha_{\max}$ is also an upper bound on the value $1 = \gamma(x)$ for all $x \in \mathbf{X}_-$.

For each group g_0 and g_1, if we rank x by their $1 - \gamma(x)$ value, then as we move α from 0 to 1, all the points that are to the left of the α will be counted towards $\mathbb{1}[\alpha \geq 1 - \gamma(x)]$. Thus the disparity will depend on the distribution of $1 - \gamma(x)$, which will mainly depend on the distribution of x, as well as the cost functions c_R and c_M. However, we are guaranteed to at least find an $1 < \alpha^* < 1$, such that after $\alpha > \alpha^*$, there is only one $x \in \mathbf{X}_-^{(g_0)}$ such that $\alpha \geq 1 - \gamma(x)$ is true. In this case, increasing α will only leads to decreasing in the disparity. □

C.8 Additional Experimental Results

In this section, we present further empirical findings obtained by employing a Gradient Boosting Decision Tree as the training method. Overall, we observe similar behavior compared with training with logistic regression.

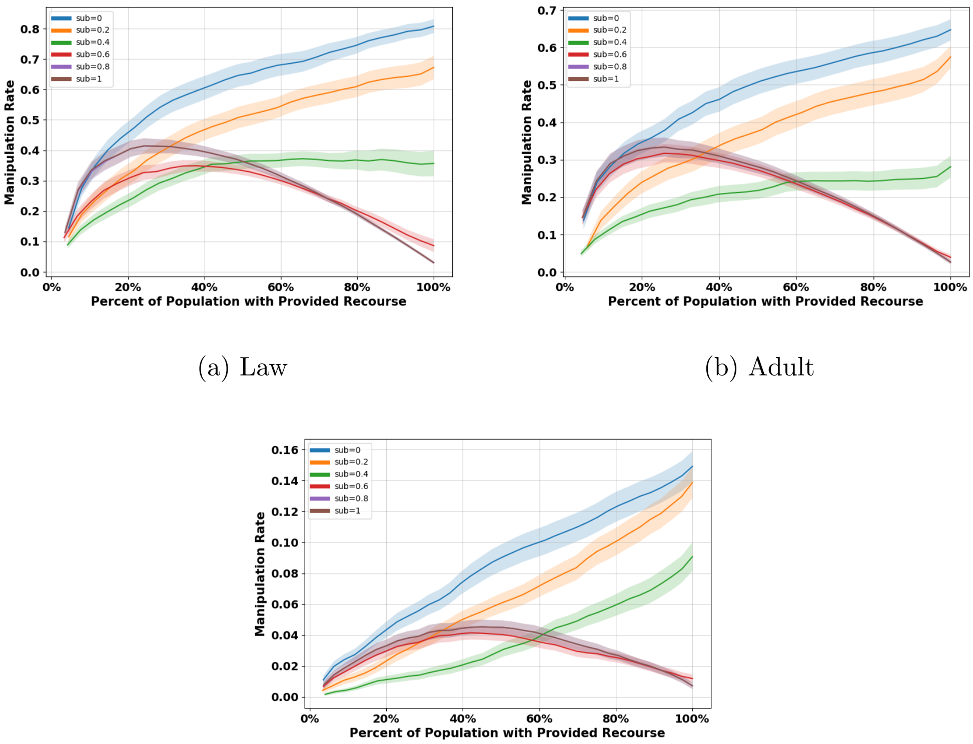


(a) Law

(b) Adult

(c) Credit

Figure C.1: Fraction of the population performing recourse, with 95% confidence intervals. Each line corresponds to a different subsidy ratio “subs”, i.e., the cost reduction applied to recourse.

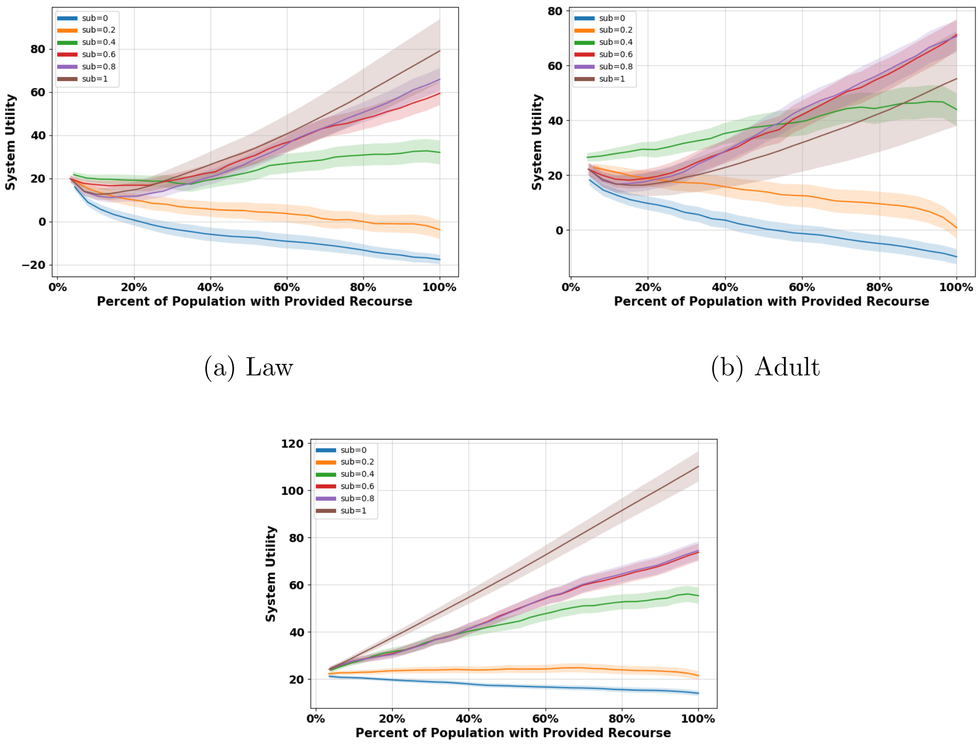


(a) Law

(b) Adult

(c) Credit

Figure C.2: Fraction of the population performing manipulation, with 95% confidence intervals. Each line corresponds to a different subsidy ratio “subs”, i.e., the cost reduction applied to recourse.

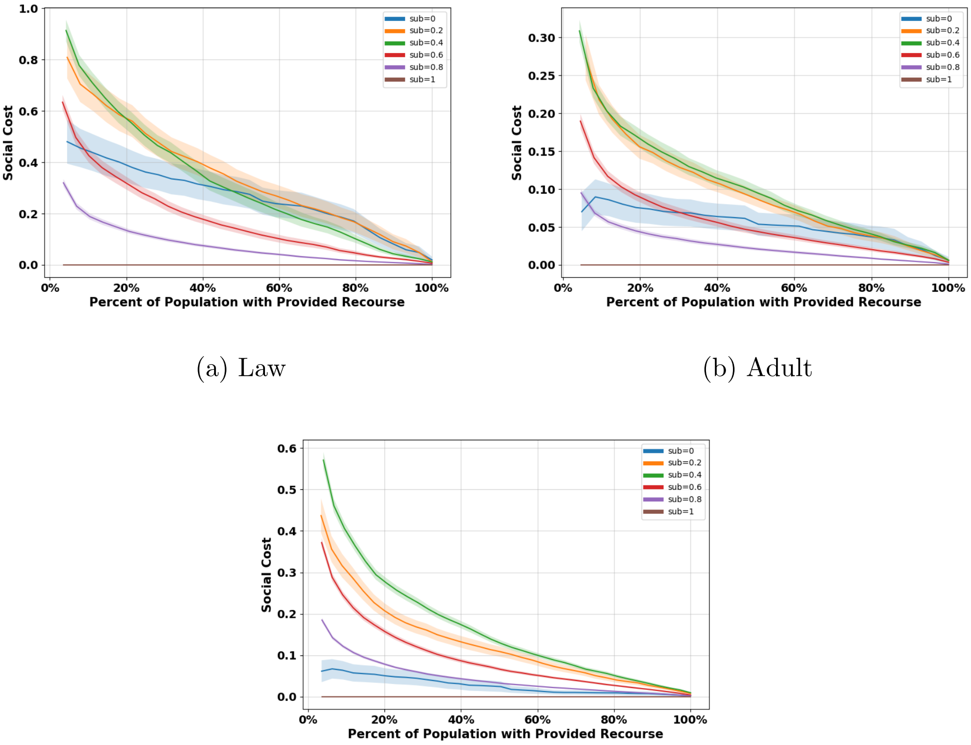


(a) Law

(b) Adult

(c) Credit

Figure C.3: The system's utility as a function of the population percentage with provided recourse, with 95% confidence intervals. Each line corresponds to a different subsidy ratio "subs", i.e., the cost reduction applied to recourse.

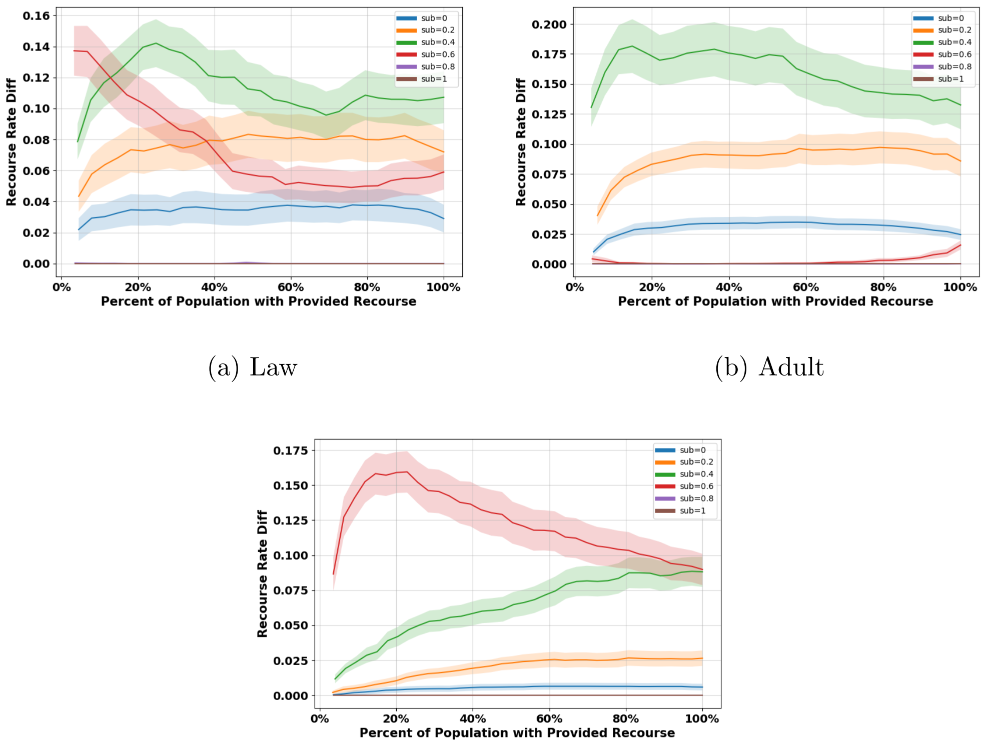


(a) Law

(b) Adult

(c) Credit

Figure C.4: The social cost as a function of the population percentage with provided recourse, with 95% confidence intervals. Each line corresponds to a different subsidy ratio "subs", i.e., the cost reduction applied to recourse.

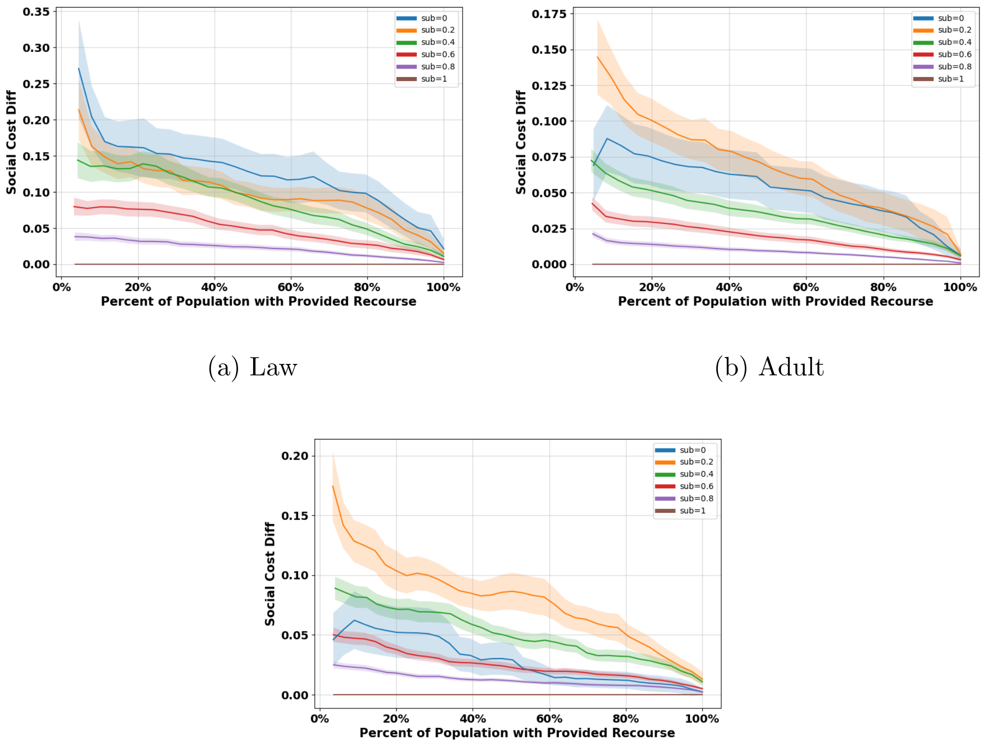


(a) Law

(b) Adult

(c) Credit

Figure C.5: Difference in recourse rate between different sensitive attribute groups with 95% confidence intervals. Each line corresponds to a different subsidy ratio “subs”, i.e., the cost reduction applied to recourse.



(a) Law

(b) Adult

(c) Credit

Figure C.6: Difference in social cost between different sensitive attribute groups with 95% confidence intervals. Each line corresponds to a different subsidy ratio “subs”, i.e., the cost reduction applied to recourse.

Appendix D

Appendix for Chapter 6

D.1 Omitted Proofs

D.1.1 Unfairness of Pairwise-Independent Derandomization

Proof of Proposition 6.2.2. For any $\delta > 0$, let $\mathbb{S}_\delta := \{x \in \mathbb{R}^n \mid d(x, \mathbf{0}) = \delta\}$ be the sphere of radius δ around the origin. Consider any $\alpha \geq 1$ and $\beta \in \left(0, \frac{1}{2} - \frac{1}{2k}\right)$, and choose X to be some subset of $\mathbb{S}_\delta$ of size $|X| = N$ in which the closest two points are positioned at distance ϵ from one another, where

$$0 < \epsilon := \min_{x,x' \in X} d(x, x') < \frac{1}{2} - \frac{1}{2k} - \beta.$$

Now let f be a classifier that maps half of the points in X to $\frac{1+\epsilon}{2}$, and the other half to $\frac{1-\epsilon}{2}$. f is $(1, 0, d)$-fair over X, since for any $x, x' \in X$,

$$|f(x) - f(x')| \leq \left|\frac{1+\epsilon}{2} - \frac{1-\epsilon}{2}\right| = \epsilon \leq d(x, x')$$

However, $\mathscr{F}_{\mathsf{PI}}$ is not (α, β, d)-fair on any point pair. To see this, consider any $x \neq x' \in X$; we show that for $\hat{f} \sim \mathscr{F}_{\mathsf{PI}}$, $|\hat{f}(x) - \hat{f}(x')|$ is typically large relative to

$d(x, x')$:

$$
\begin{aligned}
&\mathbb{E}_{\hat{f}\sim\mathscr{F}_{\mathsf{PI}}}\left[\left|\hat{f}(x) - \hat{f}(x')\right|\right] \\
&= \Pr_{\hat{f}\sim\mathscr{F}_{\mathsf{PI}}}\left[\hat{f}(x) \neq \hat{f}(x')\right] && (\hat{f} \in \{0,1\}) \\
&= \Pr_{\hat{f}\sim\mathscr{F}_{\mathsf{PI}}}\left[\hat{f}(x) = 1, \hat{f}(x') = 0\right] + \Pr_{\hat{f}\sim\mathscr{F}_{\mathsf{PI}}}\left[\hat{f}(x) = 0, \hat{f}(x') = 1\right] \\
&= \Pr_{h\sim\mathscr{H}_{\mathsf{PI}}}\left[f(x) \geq \frac{h(x)}{k}, f(x') < \frac{h(x')}{k}\right] + \Pr_{h\sim\mathscr{H}_{\mathsf{PI}}}\left[f(x) < \frac{h(x)}{k}, f(x') \geq \frac{h(x')}{k}\right] \\
&\geq \Pr_{h\sim\mathscr{H}_{\mathsf{PI}}}\left[\frac{1-\epsilon}{2} \geq \frac{h(x)}{k}, \frac{1+\epsilon}{2} < \frac{h(x')}{k}\right] + \Pr_{h\sim\mathscr{H}_{\mathsf{PI}}}\left[\frac{1+\epsilon}{2} < \frac{h(x)}{k}, \frac{1-\epsilon}{2} \geq \frac{h(x')}{k}\right] \\
&= \Pr_{h\sim\mathscr{H}_{\mathsf{PI}}}\left[\frac{h(x)}{k} \leq \frac{1-\epsilon}{2}\right] \cdot \Pr_{h\sim\mathscr{H}_{\mathsf{PI}}}\left[\frac{h(x')}{k} > \frac{1+\epsilon}{2}\right] \\
&\qquad + \Pr_{h\sim\mathscr{H}_{\mathsf{PI}}}\left[\frac{h(x)}{k} > \frac{1+\epsilon}{2}\right] \cdot \Pr_{h\sim\mathscr{H}_{\mathsf{PI}}}\left[\frac{h(x')}{k} \leq \frac{1-\epsilon}{2}\right] && \text{(by pairwise independence)} \\
&\geq \left(\frac{1-\epsilon}{2} - \frac{1}{k}\right)\left(1 - \frac{1+\epsilon}{2} - \frac{1}{k}\right) + \left(1 - \frac{1+\epsilon}{2} - \frac{1}{k}\right)\left(\frac{1-\epsilon}{2} - \frac{1}{k}\right) \\
& && \text{(by equation D.2)} \\
&= \frac{1}{2}\left(1 - 2\epsilon + \epsilon^2\right) - \frac{1-\epsilon}{2k} + \frac{1}{k^2} \\
&\geq \frac{1}{2} - \epsilon - \frac{1}{2k}
\end{aligned}
$$

The distance between any two points in $\mathbb{S}_\delta$, and therefore X, is at most 2δ; hence for a choice of $\delta \in \left(0, \frac{1/2-\beta-\epsilon-1/2k}{2\alpha}\right)$ (which is possible since $\beta < \frac{1}{2} - \frac{1}{2k}$ and $\epsilon < \frac{1}{2} - \frac{1}{2k} - \beta$), we have

$$
\begin{aligned}
&\mathbb{E}_{h\sim\mathscr{H}}\left[\left|\hat{f}(x) - \hat{f}(x')\right|\right] \\
\geq& \frac{1}{2} - \epsilon - \frac{1}{2k} \\
=& 2\alpha \cdot \frac{1/2 - \beta - \epsilon - 1/2k}{2\alpha} + \beta \\
>& \alpha \cdot 2\delta + \beta \\
\geq& \alpha \cdot d(x, x') + \beta
\end{aligned}
$$

which is a violation of (α, β, d)-metric fairness (Equation (6.2)) and applies to all pairs $x, x' \in X$. □

D.1.2 Random Threshold Derandomization Guarantees

Proof of Proposition 6.2.3. Let f be an (α, β, d)-fair classifier, and consider any $x, x' \in X$. We have

$$\begin{aligned}
&\mathbb{E}_{\hat{f}_r \sim \mathscr{F}_{\mathsf{RT}}}\left[\left|\hat{f}_r(x) - \hat{f}_r(x')\right|\right] \\
&= \Pr_{\hat{f}_r \sim \mathscr{F}_{\mathsf{RT}}}\left[\hat{f}_r(x) \neq \hat{f}_r(x')\right] && (\hat{f} \in \{0,1\}) \\
&= \Pr_{\hat{f}_r \sim \mathscr{F}_{\mathsf{RT}}}\left[\hat{f}_r(x) = 0, \hat{f}_r(x') = 1\right] + \Pr_{\hat{f}_r \sim \mathscr{F}_{\mathsf{RT}}}\left[\hat{f}_r(x) = 1, \hat{f}_r(x') = 0\right] \\
&= \Pr_{r \sim [0,1]}[f(x) < r \leq f(x')] + \Pr_{r \sim [0,1]}[f(x') < r \leq f(x)] \\
&= |f(x) - f(x')| \\
&\leq \alpha \cdot d(x, x') + \beta && (f \text{ is } (\alpha, \beta, d)\text{-fair})
\end{aligned}$$

which shows that $\mathscr{F}_{\mathsf{RT}}$ is also (α, β, d)-fair. To compute the bias, note that for any $x \in X$,

$$\mathbb{E}_{\hat{f}_r \sim \mathscr{F}_{\mathsf{RT}}}\left[\hat{f}_r(x)\right] = \Pr_{r \sim [0,1]}[f(x) \geq r] = f(x) \tag{D.1}$$

which implies $\mathsf{Bias}(\hat{f}_r, f, x) = 0$ for all x and hence $\mathsf{Bias}(\hat{f}, f, \mathscr{D})$ for all $\mathscr{D}$. Finally for the variance, we have

$$
\begin{aligned}
&\mathrm{variance}(\hat{f}_r, \mathscr{D}) \\
&:= \mathrm{Var}_{\hat{f}_r \sim \mathscr{F}_{\mathsf{RT}}} \left(\mathbb{E}_{x\sim\mathscr{D}}[\hat{f}_r(x)]\right) \\
&= \mathbb{E}_{r\sim[0,1]}\left[\left(\mathbb{E}_{x\sim\mathscr{D}}\left[\hat{f}_r(x)\right]\right)^2\right] - \left(\mathbb{E}_{r\sim[0,1]}\left[\mathbb{E}_{x\sim\mathscr{D}}\left[\hat{f}_r(x)\right]\right]\right)^2 \\
&= \mathbb{E}_{r\sim[0,1]}\left[\left(\mathbb{E}_{x\sim\mathscr{D}}\left[\hat{f}_r(x)\right]\right)^2\right] - \left(\mathbb{E}_{x\sim\mathscr{D}}\left[\mathbb{E}_{r\sim[0,1]}\left[\hat{f}_r(x)\right]\right]\right)^2 \\
&= \mathbb{E}_{r\sim[0,1]}\left[\mathbb{E}_{x,x'\sim\mathscr{D}}\left[\hat{f}_r(x)\hat{f}_r(x')\right]\right] - \mathbb{E}_{x,x'\sim\mathscr{D}}\left[\mathbb{E}_{r\sim[0,1]}\left[\hat{f}_r(x)\right]\mathbb{E}_{r\sim[0,1]}\left[\hat{f}_r(x')\right]\right] \\
&= \mathbb{E}_{x,x'\sim\mathscr{D}}\left[\mathbb{E}_{r\sim[0,1]}\left[\hat{f}_r(x)\hat{f}_r(x')\right] - \mathbb{E}_{r\sim[0,1]}\left[\hat{f}_r(x)\right]\mathbb{E}_{r\sim[0,1]}\left[\hat{f}_r(x')\right]\right] \\
&= \mathbb{E}_{x,x'\sim\mathscr{D}}\left[\mathrm{Cov}_{r\sim[0,1]}\left(\hat{f}_r(x), \hat{f}_r(x')\right)\right] \\
&\leq \mathbb{E}_{x,x'\sim\mathscr{D}}\left[\sqrt{\mathrm{Var}_{r\sim[0,1]}\left(\hat{f}_r(x)\right)\mathrm{Var}_{r\sim[0,1]}\left(\hat{f}_r(x')\right)}\right] \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{(Cauchy-Schwarz inequality)} \\
&= \left(\mathbb{E}_{x\sim\mathscr{D}}\left[\sqrt{\mathrm{Var}_{r\sim[0,1]}\left(\hat{f}_r(x)\right)}\right]\right)^2 \\
&\leq \mathbb{E}_{x\sim\mathscr{D}}\left[\mathrm{Var}_{r\sim[0,1]}\left(\hat{f}_r(x)\right)\right] && \text{(Jensen's inequality)} \\
&= \mathbb{E}_{x\sim\mathscr{D}}\left[\mathbb{E}_{r\sim[0,1]}\left[\hat{f}_r(x)\right]\left(1 - \mathbb{E}_{r\sim[0,1]}\left[\hat{f}_r(x)\right]\right)\right] \\
&= \mathbb{E}_{x\sim\mathscr{D}}[f(x)(1-f(x))] && \text{(Equation (D.1))}
\end{aligned}
$$

as required. □

D.1.3 Perfect Deterministic Fairness is Impossible for Finite Families

Proof of Proposition 6.2.4. Consider any $\alpha \geq 1$ and $\beta \in (0, 1/|\mathscr{F}|)$; it suffices to exhibit a pair of points $x, x' \in X$ such that

$$\mathbb{E}_{\hat{f}\sim\mathscr{F}}\left[\left|\hat{f}(x) - \hat{f}(x')\right|\right] > \alpha \cdot d(x, x') + \beta.$$

For any $\delta > 0$, define the *ball of radius δ around x* to be $\mathbb{B}_\delta(x) := \{x' \in X \mid d(x, x') \leq \delta\}$. By assumption, $\mathscr{F}$ contains at least one nontrivial classifier (i.e. one function that is not identically 1 or 0); let $\hat{f}$ be one such classifier. Since $X \subseteq \mathbb{R}^n$ is convex and d is a metric, $\hat{f}$ must be discontinuous at some point $x \in X$, meaning that for all $\delta > 0$, there exists $x' \in \mathbb{B}_\delta(x)$ such that $\hat{f}(x) = 1 - \hat{f}(x')$. Choose any $\delta^* \in \left(0, \frac{1/|\mathscr{F}| - \beta}{\alpha}\right)$, and consider some $x^* \in \mathbb{B}_{\delta^*}(x)$. We have

$$\begin{aligned}
\mathbb{E}_{\hat{f}\sim\mathscr{F}}\left[\left|\hat{f}(x) - \hat{f}(x^*)\right|\right] &\geq \frac{1}{|\mathscr{F}|} && \text{(at least one function in } \mathscr{F} \text{ is discontinuous at } x) \\
&= \alpha\left(\frac{1/|\mathscr{F}| - \beta}{\alpha}\right) + \beta \\
&> \alpha \cdot \delta^* + \beta && (\delta^* < \tfrac{1/|\mathscr{F}| - \beta}{\alpha}) \\
&\geq \alpha \cdot d(x, x^*) + \beta && (x^* \in \mathbb{B}_{\delta^*}(x))
\end{aligned}$$

which shows that $\mathscr{F}$ is not (α, β, d)-fair. □

D.1.4 Output Approximation of Locality-Sensitive Derandomization

Proof of Theorem 6.3.2. We will repeatedly use the following fact: by the uniformity of $\mathscr{H}_{\mathsf{PI}}$, for all $0 \leq a < b \leq 1$ and $x \in X$ we have

$$\Pr_{\substack{h_{\mathsf{LS}}\sim\mathscr{H}_{\mathsf{LS}} \\ h_{\mathsf{PI}}\sim\mathscr{H}_{\mathsf{PI}}}}\left[a \leq \frac{h_{\mathsf{PI}}(h_{\mathsf{LS}}(x))}{k} \leq b\right] \in \left(b - a - \frac{1}{k}, b - a + \frac{1}{k}\right) \tag{D.2}$$

Thus for all $x \in X$,

$$\begin{aligned}
\mathbb{E}_{\hat{f}\sim\mathscr{F}_{\mathsf{LS}}}\left[\hat{f}(x)\right] &= \Pr_{\hat{f}\sim\mathscr{F}_{\mathsf{LS}}}\left[\hat{f}(x) = 1\right] \\
= \Pr_{\substack{h_{\mathsf{LS}}\sim\mathscr{H}_{\mathsf{LS}} \\ h_{\mathsf{PI}}\sim\mathscr{H}_{\mathsf{PI}}}}\left[f(x) \geq \frac{h_{\mathsf{PI}}(h_{\mathsf{LS}}(x))}{k}\right] &\in \left(f(x) - \frac{1}{k}, f(x) + \frac{1}{k}\right)
\end{aligned}$$

which implies $\mathsf{Bias}(\hat{f}, f, x) \leq \frac{1}{k}$ for all $x \in X$ and hence $\mathsf{Bias}(\hat{f}, f, \mathscr{D}) \leq \frac{1}{k}$ for all $\mathscr{D}$.

Now we bound the variance. Define the *bucketed* stochastic classifier

$$g(x) = \frac{1}{k} \sum_{i=1}^{k} \mathbb{1} \left\{ f(x) \geq \frac{i}{k} \right\}$$

In other words, $g(x)$ is the smallest multiple of $1/k$ greater than $f(x)$. Note that $|g(x) - f(x)| \leq \frac{1}{k}$ for all x. Additionally, define the *deterministic* classifier family $\mathscr{G}_{\mathsf{LS}}$ from g just as $\mathscr{F}_{\mathsf{LS}}$ was defined from f in Equation (6.5), i.e.

$$\mathscr{G}_{\mathsf{LS}} := \{\hat{g}_{h_{\mathsf{LS}}, h_{\mathsf{PI}}} \mid h_{\mathsf{LS}} \in \mathscr{H}_{\mathsf{LS}}, h_{\mathsf{PI}} \in \mathscr{H}_{\mathsf{PI}}\}, \tag{D.3}$$

$$\text{where} \quad \hat{g}_{h_{\mathsf{LS}}, h_{\mathsf{PI}}}(x) := \mathbb{1} \left\{ g(x) \geq \frac{h_{\mathsf{PI}}(h_{\mathsf{LS}}(x))}{k} \right\}. \tag{D.4}$$

It essentially suffices to analyze $\hat{g}$ instead of $\hat{f}$, since in the end, we simply incur an additional bias or variance of $\frac{1}{k}$. To begin, observe that for any distribution $\mathscr{D}$ over X,

$$\begin{aligned}
\text{variance}(\hat{f}, f, \mathscr{D}) &= \text{variance}(\hat{g}, g, \mathscr{D}) \\
&:= \mathrm{Var}_{\hat{g} \sim \mathscr{G}_{\mathsf{LS}}} \left(\mathbb{E}_{x \sim \mathscr{D}}[\hat{g}(x)] \right) \\
&= \mathbb{E}_{\substack{h_{\mathsf{LS}} \sim \mathscr{H}_{\mathsf{LS}} \\ h_{\mathsf{PI}} \sim \mathscr{H}_{\mathsf{PI}}}} \left[\left(\mathbb{E}_{x \sim \mathscr{D}}[\hat{g}(x)] \right)^2 \right] - \left(\mathbb{E}_{x \sim \mathscr{D}}[\hat{g}(x)] \right)^2 \\
&= \mathbb{E}_{\substack{h_{\mathsf{LS}} \sim \mathscr{H}_{\mathsf{LS}} \\ h_{\mathsf{PI}} \sim \mathscr{H}_{\mathsf{PI}}}} \left[\left(\mathbb{E}_{x \sim \mathscr{D}}[\hat{g}(x)] \right)^2 \right] - \left(\mathbb{E}_{x \sim \mathscr{D}}[g(x)] \right)^2
\end{aligned}$$

To evaluate the first term, note that for any $x, x' \in X$,

$$\begin{aligned}
&\mathbb{E}_{\substack{h_{\mathsf{LS}} \sim \mathscr{H}_{\mathsf{LS}} \\ h_{\mathsf{PI}} \sim \mathscr{H}_{\mathsf{PI}}}} \left[\hat{g}(x) \hat{g}(x') \right] \\
&= \mathbb{E}_{h_{\mathsf{LS}} \sim \mathscr{H}_{\mathsf{LS}}} [\mathbb{E}_{h_{\mathsf{PI}} \sim \mathscr{H}_{\mathsf{PI}}} \left[\mathbb{1}\{h_{\mathsf{LS}}(x) = h_{\mathsf{LS}}(x')\} \hat{g}(x) \hat{g}(x') \right] \\
&\qquad + \mathbb{E}_{h_{\mathsf{PI}} \sim \mathscr{H}_{\mathsf{PI}}} \left[\mathbb{1}\{h_{\mathsf{LS}}(x) \neq h_{\mathsf{LS}}(x')\} \hat{g}(x) \hat{g}(x') \right]] \\
&= \mathbb{E}_{h_{\mathsf{LS}} \sim \mathscr{H}_{\mathsf{LS}}} [\mathbb{E}_{h_{\mathsf{PI}} \sim \mathscr{H}_{\mathsf{PI}}} \left[\mathbb{1}\{h_{\mathsf{LS}}(x) = h_{\mathsf{LS}}(x')\} \hat{g}(x) \hat{g}(x') \right] \\
&\qquad + \mathbb{1}\{h_{\mathsf{LS}}(x) \neq h_{\mathsf{LS}}(x')\} g(x) g(x')] \qquad \text{(pairwise independence)}
\end{aligned}$$

Thus the first term of the variance is

$$\begin{aligned}
&\mathbb{E}_{\substack{h_{\mathsf{LS}}\sim\mathscr{H}_{\mathsf{LS}}\\ h_{\mathsf{PI}}\sim\mathscr{H}_{\mathsf{PI}}}}\left[\left(\mathbb{E}_{x\sim\mathscr{D}}[\hat{g}(x)]\right)^2\right]\\
&=\mathbb{E}_{\substack{h_{\mathsf{LS}}\sim\mathscr{H}_{\mathsf{LS}}\\ h_{\mathsf{PI}}\sim\mathscr{H}_{\mathsf{PI}}}}\left[\mathbb{E}_{x,x'\sim\mathscr{D}}[\hat{g}(x)\hat{g}(x')]\right]\\
&=\mathbb{E}_{x,x'\sim\mathscr{D}}\left[\mathbb{E}_{\substack{h_{\mathsf{LS}}\sim\mathscr{H}_{\mathsf{LS}}\\ h_{\mathsf{PI}}\sim\mathscr{H}_{\mathsf{PI}}}}[\hat{g}(x)\hat{g}(x')]\right]\\
&=\mathbb{E}_{x,x'\sim\mathscr{D}}[\mathbb{E}_{h_{\mathsf{LS}}\sim\mathscr{H}_{\mathsf{LS}}}[\mathbb{E}_{h_{\mathsf{PI}}\sim\mathscr{H}_{\mathsf{PI}}}\left[\mathbb{1}\{h_{\mathsf{LS}}(x)=h_{\mathsf{LS}}(x')\}\hat{g}(x)\hat{g}(x')\right]\\
&\qquad+\mathbb{1}\{h_{\mathsf{LS}}(x)\neq h_{\mathsf{LS}}(x')\}g(x)g(x')]]
\end{aligned}$$

Next consider the second term:

$$\left(\mathbb{E}_{x\sim\mathscr{D}}[g(x)]\right)^2=\mathbb{E}_{x,x'\sim\mathscr{D}}[g(x)g(x')]$$

Putting these together, we have

$$
\begin{aligned}
&\mathrm{variance}(\hat{f}, f, \mathscr{D}) \\
&= \mathbb{E}_{h_{\mathsf{LS}} \sim \mathscr{H}_{\mathsf{LS}}} [\mathbb{E}_{h_{\mathsf{PI}} \sim \mathscr{H}_{\mathsf{PI}}} \left[\mathbb{E}_{x, x' \sim \mathscr{D}} [\mathbb{1}\{h_{\mathsf{LS}}(x) = h_{\mathsf{LS}}(x')\} \hat{g}(x) \hat{g}(x')] \right] \\
&\qquad - \mathbb{E}_{x, x' \sim \mathscr{D}} \left[\mathbb{1}\{h_{\mathsf{LS}}(x) = h_{\mathsf{LS}}(x')\} g(x) g(x') \right]] \\
&= \mathbb{E}_{h_{\mathsf{LS}} \sim \mathscr{H}_{\mathsf{LS}}} \left[\mathbb{E}_{x, x' \sim \mathscr{D}} \left[\mathbb{1}\{h_{\mathsf{LS}}(x) = h_{\mathsf{LS}}(x')\} \cdot \left(\mathbb{E}_{h_{\mathsf{PI}}} [\hat{g}(x) \hat{g}(x')] - g(x) g(x') \right) \right] \right] \\
&= \mathbb{E}_{h_{\mathsf{LS}} \sim \mathscr{H}_{\mathsf{LS}}} \left[\mathbb{E}_{x, x' \sim \mathscr{D}} \left[\mathbb{1}\{h_{\mathsf{LS}}(x) = h_{\mathsf{LS}}(x')\} \cdot \left(\mathbb{E}_{h_{\mathsf{PI}}} [\hat{g}(x) \hat{g}(x')] - \mathbb{E}_{h_{\mathsf{PI}}} [\hat{g}(x)] \mathbb{E}_{h_{\mathsf{PI}}} [\hat{g}(x')] \right) \right] \right] \\
&= \mathbb{E}_{h_{\mathsf{LS}} \sim \mathscr{H}_{\mathsf{LS}}} \left[\mathbb{E}_{x, x' \sim \mathscr{D}} \left[\mathbb{1}\{h_{\mathsf{LS}}(x) = h_{\mathsf{LS}}(x')\} \cdot \mathrm{Cov}_{h_{\mathsf{PI}}} \left(\hat{g}(x), \hat{g}(x') \right) \right] \right] \\
&\le \mathbb{E}_{h_{\mathsf{LS}} \sim \mathscr{H}_{\mathsf{LS}}} \left[\mathbb{E}_{x, x' \sim \mathscr{D}} \left[\mathbb{1}\{h_{\mathsf{LS}}(x) = h_{\mathsf{LS}}(x')\} \cdot \sqrt{\mathrm{Var}_{h_{\mathsf{PI}}} \left(\hat{g}(x) \right) \mathrm{Var}_{h_{\mathsf{PI}}} \left(\hat{g}(x') \right)} \right] \right] \\
&\hfill \text{(Cauchy-Schwarz inequality)} \\
&= \mathbb{E}_{h_{\mathsf{LS}} \sim \mathscr{H}_{\mathsf{LS}}} \left[\sum_{b \in B} \left(\mathbb{E}_{x \sim \mathscr{D}} \left[\mathbb{1}\{h_{\mathsf{LS}}(x) = b\} \cdot \sqrt{\mathrm{Var}_{h_{\mathsf{PI}}} \left(\hat{g}(x) \right)} \right] \right)^2 \right] \\
&= \mathbb{E}_{h_{\mathsf{LS}} \sim \mathscr{H}_{\mathsf{LS}}} \left[\sum_{b \in B} \left(\Pr_{x \sim \mathscr{D}} [h_{\mathsf{LS}}(x) = b] \cdot \mathbb{E}_{x \sim \mathscr{D}} \left[\sqrt{\mathrm{Var}_{h_{\mathsf{PI}}} \left(\hat{g}(x) \right)} \;\middle|\; h_{\mathsf{LS}}(x) = b \right] \right)^2 \right] \\
&\le \mathbb{E}_{h_{\mathsf{LS}} \sim \mathscr{H}_{\mathsf{LS}}} \left[\sum_{b \in B} \left(\Pr_{x \sim \mathscr{D}} [h_{\mathsf{LS}}(x) = b] \right)^2 \cdot \mathbb{E}_{x \sim \mathscr{D}} \left[\mathrm{Var}_{h_{\mathsf{PI}}} \left(\hat{g}(x) \right) \mid h_{\mathsf{LS}}(x) = b \right] \right] \\
&\hfill \text{(Jensen's inequality)} \\
&= \mathbb{E}_{h_{\mathsf{LS}} \sim \mathscr{H}_{\mathsf{LS}}} \left[\sum_{b \in B} \left(\Pr_{x \sim \mathscr{D}} [h_{\mathsf{LS}}(x) = b] \right)^2 \cdot \mathbb{E}_{x \sim \mathscr{D}} [g(x)(1 - g(x)) \mid h_{\mathsf{LS}}(x) = b] \right] \\
&\le \mathbb{E}_{h_{\mathsf{LS}} \sim \mathscr{H}_{\mathsf{LS}}} [\left(\max_{b \in B} \Pr_{x \sim \mathscr{D}} [h_{\mathsf{LS}}(x) = b] \right) \\
&\qquad \sum_{b \in B} \Pr_{x \sim \mathscr{D}} [h_{\mathsf{LS}}(x) = b] \cdot \mathbb{E}_{x \sim \mathscr{D}} [g(x)(1 - g(x)) \mid h_{\mathsf{LS}}(x) = b]] \\
&= \mathbb{E}_{h_{\mathsf{LS}} \sim \mathscr{H}_{\mathsf{LS}}} \left[\max_{b \in B} \Pr_{x \sim \mathscr{D}} [h_{\mathsf{LS}}(x) = b] \right] \cdot \mathbb{E}_{x \sim \mathscr{D}} [g(x)(1 - g(x))] \\
&\le \mathbb{E}_{h_{\mathsf{LS}} \sim \mathscr{H}_{\mathsf{LS}}} \left[\max_{b \in B} \Pr_{x \sim \mathscr{D}} [h_{\mathsf{LS}}(x) = b] \right] \cdot \mathbb{E}_{x \sim \mathscr{D}} \left[f(x)(1 - f(x)) + \frac{1}{k} \right] \\
&\hfill \left(\mathsf{Bias}(f, g, x) \le \tfrac{1}{k} \text{ for all } x\right)
\end{aligned}
$$

□

D.1.5 Fairness of LSH-Based Derandomization

Proof of Theorem 6.3.3. We first prove pairwise metric fairness. Consider any $x, x' \in X$, and assume without loss of generality that $f(x) \leq f(x')$. We have

$$\begin{aligned}
&\mathbb{E}_{\hat{f}\sim\mathscr{F}_{\mathsf{LS}}}\left[\left|\hat{f}(x)-\hat{f}(x')\right|\right] \\
&= \Pr_{\substack{h_{\mathsf{LS}}\sim\mathscr{H}_{\mathsf{LS}}\\ h_{\mathsf{PI}}\sim\mathscr{H}_{\mathsf{PI}}}}\left[\hat{f}(x)\neq\hat{f}(x')\right] && (\hat{f}\in\{0,1\}) \\
&= \underbrace{\Pr_{\substack{h_{\mathsf{LS}}\\ h_{\mathsf{PI}}}}\left[\hat{f}(x)\neq\hat{f}(x') \;\middle|\; h_{\mathsf{LS}}(x)=h_{\mathsf{LS}}(x')\right]}_{p_1}\cdot\Pr_{h_{\mathsf{LS}}}[h_{\mathsf{LS}}(x)=h_{\mathsf{LS}}(x')] \\
&\quad+\underbrace{\Pr_{\substack{h_{\mathsf{LS}}\\ h_{\mathsf{PI}}}}\left[\hat{f}(x)\neq\hat{f}(x') \;\middle|\; h_{\mathsf{LS}}(x)\neq h_{\mathsf{LS}}(x')\right]}_{p_2}\cdot\Pr_{h_{\mathsf{LS}}}[h_{\mathsf{LS}}(x)\neq h_{\mathsf{LS}}(x')] && \text{(D.5)}
\end{aligned}$$

We evaluate p_1 and p_2 separately. First, noting that a pairwise-independent hash family is also uniform, we have

$$\begin{aligned}
&\Pr_{h_{\mathsf{LS}},h_{\mathsf{PI}}}\left[\hat{f}(x)=0,\hat{f}(x')=1 \;\middle|\; h_{\mathsf{LS}}(x)=h_{\mathsf{LS}}(x')\right] \\
&= \Pr_{h_{\mathsf{LS}},h_{\mathsf{PI}}}\left[f(x)<\frac{h_{\mathsf{PI}}(h_{\mathsf{LS}}(x))}{k}, f(x')\geq\frac{h_{\mathsf{PI}}(h_{\mathsf{LS}}(x'))}{k} \;\middle|\; h_{\mathsf{LS}}(x)=h_{\mathsf{LS}}(x')\right] \\
&= \Pr_{h_{\mathsf{LS}},h_{\mathsf{PI}}}\left[f(x)<\frac{h_{\mathsf{PI}}(h_{\mathsf{LS}}(x))}{k}\leq f(x') \;\middle|\; h_{\mathsf{LS}}(x)=h_{\mathsf{LS}}(x')\right] \\
&= \Pr_{h_{\mathsf{LS}},h_{\mathsf{PI}}}\left[f(x)<\frac{h_{\mathsf{PI}}(h_{\mathsf{LS}}(x))}{k}\leq f(x')\right] && (h_{\mathsf{PI}}\text{ is uniform})
\end{aligned}$$

By symmetry, $\Pr_{h_{\mathsf{LS}},h_{\mathsf{PI}}}[\hat{f}(x)=1,\hat{f}(x')=0 \mid h_{\mathsf{LS}}(x)=h_{\mathsf{LS}}(x')] = \Pr_{h_{\mathsf{LS}},h_{\mathsf{PI}}}[f(x)\geq \frac{h_{\mathsf{PI}}(h_{\mathsf{LS}}(x))}{k} > f(x')]$; but this equals zero, since $f(x)\leq f(x')$. Thus

$$\begin{aligned}
p_1 &= \Pr_{h_{\mathsf{LS}},h_{\mathsf{PI}}}\left[\hat{f}(x)=1,\hat{f}(x')=0 \;\middle|\; h_{\mathsf{LS}}(x)=h_{\mathsf{LS}}(x')\right] \\
&\qquad+\Pr_{h_{\mathsf{LS}},h_{\mathsf{PI}}}\left[\hat{f}(x)=0,\hat{f}(x')=1 \;\middle|\; h_{\mathsf{LS}}(x)=h_{\mathsf{LS}}(x')\right] \\
&= \Pr_{h_{\mathsf{LS}},h_{\mathsf{PI}}}\left[f(x)<\frac{h_{\mathsf{PI}}(h_{\mathsf{LS}}(x))}{k}\leq f(x')\right] \\
&= |f(x)-f(x')|\pm\frac{2}{k} && \text{(by Equation (D.2))}
\end{aligned}$$

Next, to compute p_2, we have

$$
\begin{aligned}
&\Pr_{h_{\mathsf{LS}},h_{\mathsf{PI}}}\left[\hat{f}(x)=1,\hat{f}(x')=0 \;\middle|\; h_{\mathsf{LS}}(x)\neq h_{\mathsf{LS}}(x')\right]\\
&=\Pr_{h_{\mathsf{LS}},h_{\mathsf{PI}}}\left[f(x)\geq\frac{h_{\mathsf{PI}}(h_{\mathsf{LS}}(x))}{k}, f(x')<\frac{h_{\mathsf{PI}}(h_{\mathsf{LS}}(x'))}{k} \;\middle|\; h_{\mathsf{LS}}(x)\neq h_{\mathsf{LS}}(x')\right]\\
&=\Pr_{h_{\mathsf{LS}},h_{\mathsf{PI}}}\left[f(x)\geq\frac{h_{\mathsf{PI}}(h_{\mathsf{LS}}(x))}{k}, \;\middle|\; h_{\mathsf{LS}}(x)\neq h_{\mathsf{LS}}(x')\right]\\
&\qquad\cdot\Pr_{h_{\mathsf{LS}},h_{\mathsf{PI}}}\left[f(x')<\frac{h_{\mathsf{PI}}(h_{\mathsf{LS}}(x'))}{k} \;\middle|\; h_{\mathsf{LS}}(x)\neq h_{\mathsf{LS}}(x')\right]\\
&\hfill(h_{\mathsf{PI}}\text{ is pairwise independent})\\
&=f(x)(1-f(x'))\pm\frac{1}{k} \qquad\qquad (h_{\mathsf{PI}}\text{ is uniform})
\end{aligned}
$$

and by symmetry, $\Pr_{h_{\mathsf{LS}},h_{\mathsf{PI}}}[\hat{f}(x)=0,\hat{f}(x')=1 \mid h_{\mathsf{LS}}(x)\neq h_{\mathsf{LS}}(x')]=(1-f(x))f(x')\pm\frac{1}{k}$. Thus

$$
\begin{aligned}
p_2 &= \Pr_{h_{\mathsf{LS}},h_{\mathsf{PI}}}\left[\hat{f}(x)=1,\hat{f}(x')=0 \;\middle|\; h_{\mathsf{LS}}(x)\neq h_{\mathsf{LS}}(x')\right]\\
&\quad+\Pr_{h_{\mathsf{LS}},h_{\mathsf{PI}}}\left[\hat{f}(x)=0,\hat{f}(x')=1 \;\middle|\; h_{\mathsf{LS}}(x)\neq h_{\mathsf{LS}}(x')\right]\\
&=f(x)-2f(x')f(x)+f(x')\pm\frac{2}{k}
\end{aligned}
$$

Substituting p_1 and p_2 back into Equation (D.5) yields

$$
\begin{aligned}
&\mathbb{E}_{h_{\mathsf{LS}},h_{\mathsf{PI}}}\left[\left|\hat{f}(x)-\hat{f}(x')\right|\right] &&\text{(D.6)}\\
&=p_1\cdot\Pr_{h_{\mathsf{LS}}}[h_{\mathsf{LS}}(x)=h_{\mathsf{LS}}(x')]+p_2\cdot\Pr_{h_{\mathsf{LS}}}[h_{\mathsf{LS}}(x)\neq h_{\mathsf{LS}}(x')]\\
&=|f(x)-f(x')|\cdot(1-d(x,x'))+(f(x)-2f(x')f(x)+f(x'))\cdot d(x,x')\pm\frac{2}{k}\\
& &&(h_{\mathsf{LS}}\text{ is LSH})\\
&=|f(x)-f(x')|+2f(x)(1-f(x'))\cdot d(x,x')\pm\frac{2}{k} &&\text{(D.7)}\\
&\leq\alpha\cdot d(x,x')+\beta+2f(x)(1-f(x'))\cdot d(x,x')+\frac{2}{k} &&(f\text{ is }(\alpha,\beta,d)\text{-fair})\\
&\leq[\alpha+2f(x)(1-f(x'))]\cdot d(x,x')+\beta+\epsilon &&(k\geq 2/\epsilon)
\end{aligned}
$$

which proves the pairwise fairness bound. The aggregate fairness bound then follows from Lemma 6.4.5.

□

D.1.6 Bias-Variance Decomposition

Proof of Lemma 6.4.1. For any $c > 0$, we have

$$\begin{aligned}
&\left|\hat{f}(x) - \mathbb{1}_f(x)\right| \\
&\leq \left|\mathbb{E}_{f,\hat{f}}\left[\hat{f}(x) - \mathbb{1}_f(x)\right]\right| + \left|\hat{f}(x) - \mathbb{1}_f(x) - \mathbb{E}_{f,\hat{f}}\left[\hat{f}(x) - \mathbb{1}_f(x)\right]\right| \\
&\leq \left|\mathbb{E}_{f,\hat{f}}\left[\hat{f}(x) - \mathbb{1}_f(x)\right]\right| + c \cdot \mathrm{Var}_{f,\hat{f}}\left(\hat{f}(x) - \mathbb{1}_f(x) - \mathbb{E}_{f,\hat{f}}\left[\hat{f}(x) - \mathbb{1}_f(x)\right]\right) \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{(by Chebyshev's inequality, w.p. } 1 - 1/c^2\text{)} \\
&\leq \left|\mathbb{E}_{f,\hat{f}}\left[\hat{f}(x) - \mathbb{1}_f(x)\right]\right| + c \cdot \mathrm{Var}_{\hat{f}}\left(\hat{f}(x) - \mathbb{E}_{\hat{f}}\left[\hat{f}(x)\right]\right) + c \cdot \mathrm{Var}_f\left(\mathbb{1}_f(x) - \mathbb{E}_f[\mathbb{1}_f(x)]\right) \\
&\qquad\qquad\qquad\qquad (\hat{f}(x) - \mathbb{E}_{\hat{f}}[\hat{f}(x)] \text{ and } \mathbb{1}_f(x) - \mathbb{E}_f[\mathbb{1}_f(x)] \text{ have mean zero}) \\
&\leq \left|\mathbb{E}_{f,\hat{f}}\left[\hat{f}(x) - \mathbb{1}_f(x)\right]\right| + c \cdot \mathrm{Var}_{\hat{f}}\left(\hat{f}(x)\right) + c \cdot \mathrm{Var}_f\left(\mathbb{1}_f(x)\right)
\end{aligned}$$

The above calculation fails with probability at most $1/c^2$, in which case the left-hand side still obeys the simple bound $|\hat{f}(x) - \mathbb{1}_f(x)| \leq 1$. Thus taking expectations of both sides, we have

$$\mathbb{E}_{f,\hat{f}}\left[\left|\hat{f}(x) - \mathbb{1}_f(x)\right|\right] \leq \left|\mathbb{E}_{f,\hat{f}}\left[\hat{f}(x) - \mathbb{1}_f(x)\right]\right| + c \cdot \mathrm{Var}_{\hat{f}}\left(\hat{f}(x)\right) + c \cdot \mathrm{Var}_f\left(\mathbb{1}_f(x)\right) + \frac{1}{c^2}$$

with probability 1 for any $c > 0$. A choice of $c = (\mathrm{Var}_{\hat{f}\sim\mathscr{F}}(\hat{f}(x)) + \mathrm{Var}_f(\mathbb{1}_f(x)))^{-1/3}$ yields the result. □

D.1.7 Metric-Fair Derandomization Preserves Threshold Fairness

Proof of Lemma 6.4.3. First, fix some $\sigma \in (0,1)$ and let $X^2_{\leq\sigma} := \left\{(x, x') \in X^2 \;\middle|\; d(x, x') \leq \sigma\right\}$. Observe the following translations between metric and threshold fairness on this set:

1. If f is (σ, τ, d)-threshold fair, then for any $(x, x') \in X^2_{\leq\sigma}$,

   $$|f(x) - f(x')| \leq \tau = 0 \cdot d(x, x') + \tau$$

   So, f is also $(0, \tau, d)$-metric fair on such pairs (x, x').

2. If f is (α, β, d)-metric fair on all $(x, x') \in X^2_{\leq\sigma}$, then for such pairs,

   $$|f(x) - f(x')| \leq \alpha \cdot d(x, x') + \beta \leq \alpha\sigma + \beta$$

   So, f is also $(\sigma, \alpha\sigma + \beta, d)$-threshold fair.

Now suppose we run our derandomization procedure on a (σ, τ, d)-threshold fair stochastic classifier f. Let $\mathscr{F}$ be the deterministic classifier family from which we sample our output. Then f is $(0, \tau, d)$-metric fair over $X^2_{\leq\sigma}$ (by observation 1 above), $\mathscr{F}$ is then $(A(0), B(\tau), d)$-metric fair over $X^2_{\leq\sigma}$ (by the fairness preservation guarantee), and $\mathscr{F}$ is also $(\sigma, A(0)\cdot\sigma+B(\tau), d)$-threshold fair (by observation 2). □

Proof of Corollary 6.4.4. If f is (σ, τ, d)-threshold fair, then $\mathscr{F}_{\mathsf{LS}}$ is (σ, τ', d)-threshold fair, where

$$\begin{aligned} \tau' &= A(0) \cdot \sigma + B(\tau) && \text{(Lemma 6.4.3)} \\ &= \frac{1}{2} \cdot \sigma + \tau + \frac{2}{k} && \text{(Corollary 6.3.4)} \\ &= \sigma + \tau && \text{(choice of } k \geq 4/\sigma) \end{aligned}$$

□

D.1.8 Pairwise Fairness Implies Aggregate Fairness

Proof of Lemma 6.4.5. For all distances $\xi \in [0, 1]$, let

$X^2_\xi := \{(x, x') \in X^2 \mid d(x, x') = \xi\}$ denote the set of point pairs at distance exactly

ξ. Then, for any given $\hat{f} \in \mathscr{F}$, let

$$\rho_\xi(\hat{f}) := \Pr_{(x,x')\sim X_\xi^2}\left[\hat{f}(x) \neq \hat{f}(x')\right] \qquad \text{and} \qquad \rho_{\leq\tau}(\hat{f}) := \Pr_{(x,x')\sim X_{\leq\tau}^2}\left[\hat{f}(x) \neq \hat{f}(x')\right]$$

denote the fraction of pairs at distance ξ and within τ, respectively, to which $\hat{f}$ assigns different outputs. Treating $\rho_\xi(\hat{f})$ as a random variable of $\hat{f}$, we have

$$\mathbb{E}_{\hat{f}\sim\mathscr{F}}\left[\rho_\xi(\hat{f})\right] = \mathbb{E}_{\hat{f}\sim\mathscr{F}}\left[\Pr_{\substack{(x,x')\\ \sim X_\xi^2}}\left[\hat{f}(x) \neq \hat{f}(x')\right]\right] \tag{D.8}$$

$$= \mathbb{E}_{\hat{f}\sim\mathscr{F}}\left[\mathbb{E}_{\substack{(x,x')\\ \sim X_\xi^2}}\left[\left|\hat{f}(x) - \hat{f}(x')\right|\right]\right] \tag{D.9}$$

$$= \mathbb{E}_{\substack{(x,x')\\ \sim X_\xi^2}}\left[\mathbb{E}_{\hat{f}\sim\mathscr{F}}\left[\left|\hat{f}(x) - \hat{f}(x')\right|\right]\right] \tag{D.10}$$

Thus the fraction of separated pairs within distance τ is

$$\begin{aligned}
&\mathbb{E}_{\hat{f}\sim\mathscr{F}}\left[\rho_{\leq\tau}(\hat{f})\right] && \text{(D.11)}\\
&:= \mathbb{E}_{\hat{f}\sim\mathscr{F}}\left[\Pr_{(x,x')\sim X_{\leq\tau}^2}\left[\hat{f}(x) \neq \hat{f}(x')\right]\right]\\
&= \int_0^\tau \mathbb{E}_{\hat{f}\sim\mathscr{F}}\left[\Pr_{(x,x')\sim X_{\leq\tau}^2}\left[\hat{f}(x) \neq \hat{f}(x') \;\middle|\; d(x,x') = \xi\right] \cdot \Pr_{(x,x')\sim X_{\leq\tau}^2}[d(x,x') = \xi]\; d\xi\right]\\
&= \int_0^\tau \mathbb{E}_{\hat{f}\sim\mathscr{F}}\left[\Pr_{(x,x')\sim X_\xi^2}\left[\hat{f}(x) \neq \hat{f}(x')\right]\right] \cdot \Pr_{(x,x')\sim X_{\leq\tau}^2}[d(x,x') = \xi]\; d\xi\\
&= \int_0^\tau \mathbb{E}_{(x,x')\sim X_\xi^2}\left[\mathbb{E}_{\hat{f}\sim\mathscr{F}}\left[\left|\hat{f}(x) - \hat{f}(x')\right|\right]\right] \cdot \Pr_{(x,x')\sim X_{\leq\tau}^2}[d(x,x') = \xi]\; d\xi\\
& && \text{(by Equation (D.8))}\\
&\leq \int_0^\tau (\alpha\xi + \beta) \Pr_{(x,x')\sim X_{\leq\tau}^2}[d(x,x') = \xi]\; d\xi\\
& && \text{(by } (\alpha,\beta,d)\text{-fairness)}\\
&\leq (\alpha\tau + \beta) \int_0^\tau \Pr_{(x,x')\sim X_{\leq\tau}^2}[d(x,x') = \xi]\; d\xi\\
&= \alpha\tau + \beta && \text{(D.12)}
\end{aligned}$$

Since $\rho_{\leq\tau} \in [0,1]$, $\mathrm{Var}(\rho_{\leq\tau}) = \mathbb{E}[\rho_{\leq\tau}^2] - \mathbb{E}[\rho_{\leq\tau}]^2 \leq \mathbb{E}[\rho_{\leq\tau}]$. Thus applying

Chebyshev's inequality to Equation (D.12) yields

$$\Pr_{\hat{f}\sim\mathscr{F}}\left[\rho > \left(1+\frac{1}{\sqrt{\delta}}\right)(\alpha\tau+\beta)\right] \leq \Pr_{\hat{f}\sim\mathscr{F}}\left[\rho > \left(1+\frac{1}{\sqrt{\delta}}\right)\mathbb{E}_{\hat{f}\sim\mathscr{F}}[\rho]\right] \leq \delta$$

which proves the claim. □

D.1.9 Output Approximation and Loss Approximation

Proof of Lemma 6.4.6. For any $x \in X$ and $y \in \{0,1\}$,

$$\begin{aligned}
&\mathbb{E}_{\hat{f}\sim\mathscr{F}}\left[L(\hat{f},x,y)\right] \\
&= \mathbb{E}_{\hat{f}\sim\mathscr{F}}\left[\ell(\hat{f}(x),y)\right] && (\hat{f}(x)\in\{0,1\}) \\
&= \mathbb{E}_{\hat{f}}\left[\ell(\hat{f}(x),y) \;\middle|\; \hat{f}(x)=1\right]\cdot\Pr_{\hat{f}}\left[\hat{f}(x)=1\right] \\
&\qquad + \mathbb{E}_{\hat{f}}\left[\ell(\hat{f}(x),y) \;\middle|\; \hat{f}(x)=0\right]\cdot\Pr_{\hat{f}}\left[\hat{f}(x)=0\right] \\
&= \ell(1,y)\cdot\mathbb{E}_{\hat{f}}\left[\hat{f}(x)\right] + \ell(0,y)\cdot\left(1-\mathbb{E}_{\hat{f}}\left[\hat{f}(x)\right]\right) \\
&= \ell(1,y)f(x) + \ell(0,y)\,(1-f(x)) \pm \mathsf{Bias}(\hat{f},f,x) \\
&= f(x)\ell(1,y) + (1-f(x))\ell(0,y) \pm \mathsf{Bias}(\hat{f},f,x)
\end{aligned}$$

which proves the first inequality concerning the bias. For the variance, notice that since ℓ is binary, either $\mathrm{Var}_{\hat{f}}\left(\ell(\hat{f}(x),y)\right) = \mathrm{Var}_{\hat{f}}\left(\hat{f}(x)\right)$ or $\mathrm{Var}_{\hat{f}}\left(\ell(\hat{f}(x),y)\right) = 0$. □

D.2 Manipulation Deterrence in Strategic Classification

Fair derandomization procedures carry implications for the *strategic classification* problem, a popular framework for modeling the behavior of self-interested agents subject to classification decisions Hardt et al. [2016a], Cai et al. [2015], Chen et al. [2018], Dong et al. [2018b], Chen et al. [2020b]. Formally, strategic classification is a Stackelberg game, or a sequential game between two players:

1. First, a *decision maker* or *model designer* publishes a classifier. Traditionally, this means a stochastic classifier $f : X \to [0, 1]$, but in our setting, the model designer may publish a family of deterministic classifiers $\mathscr{F}$, and promises to select a single classifier from $\mathscr{F}$ uniformly at random.

2. Next, a *strategic agent* or *decision subject*, who is associated with some feature vector $x \in X$, decides either to present their true features x, or to change or *manipulate* their features to some $x' \in X$ to obtain the favorable outcome $\hat{f}(x') = 1$ with higher probability. However, the agent incurs a cost $c(x, x') \geq 0$ for altering their features.

Given a (stochastic or deterministic) classifier $f : X \to [0, 1]$ and cost function $c : X^2 \to [0, 1]$, the *utility* of an agent with original features x who changes to x' is defined as

$$U_f(x, x') := f(x') - c(x, x')$$

and the utility-maximizing move $\Delta_f(x) := \arg\max_{x' \in X} U_f(x, x')$ is called the *best response* of x under f and c.

In the following proposition, we observe a general connection between metric fairness and strategic manipulation; namely that the more fair a classifier is with respect to a metric cost function, the less incentive agents have to manipulate their features. The reason is intuitive: if a classifier is a smooth function, then an agent x cannot expect their outcome to change much by moving to some nearby point x'.

Proposition D.2.1 (Metric fairness implies reduced manipulation incentive)**.** *Let c be a metric cost function and let f be a (α, β, c)-metric fair classifier. Then the* *maximum utility gained by manipulating x to x' is*

$$U_f(x, x') - U_f(x, x) \leq (\alpha - 1) \cdot c(x, x') + \beta.$$

If f is a deterministic classifier drawn from a family $\mathscr{F}$, then this holds in expectation over the sampling of f.

Proof of Proposition D.2.1. Under a classifier f, an individual with original features $x \in X$ who changes to $x' \in X$ derives utility

$$\begin{aligned} U_f(x, x') &= f(x') - c(x, x') \\ &\leq f(x) + |f(x') - f(x)| - c(x, x') \\ &\leq f(x) + \alpha \cdot c(x, x') + \beta - c(x, x') && (f \text{ is } (\alpha, \beta, c)\text{-fair}) \\ &= f(x) + (\alpha - 1) \cdot c(x, x') + \beta \\ &= U_f(x, x) + (\alpha - 1) \cdot c(x, x') + \beta \end{aligned}$$

which proves the claim for stochastic classifiers. The proof for a deterministic family $\mathscr{F}$ results from taking an expectation $\mathbb{E}_{f \sim \mathscr{F}}[\cdot]$ on both sides. □

Braverman and Garg [Braverman and Garg, 2020] already observed this fact for a stochastic classifier with $\alpha = 1$ and $\beta = 0$, in which case there is no incentive to manipulate. Note that by Proposition 6.2.4, deterministic families cannot achieve such small fairness parameters; hence the upper bound of Proposition D.2.1 cannot rule out *some* incentive to manipulate. Nevertheless, it presents a nontrivial worst-case guarantee since, for a classifier without any fairness constraints, there may be individuals near the decision boundary who can flip their decision from, for example, $f(x) = 0$ to $f(x') = 1$ at near-zero cost, thereby gaining utility $U(x, x') - U(x, x) \approx 1$ through manipulation.

Cost functions studied in the strategic classification literature include the L_2 [Hardt et al., 2016a, Brückner and Scheffer, 2011] and Mahalanobis [Chen et al., 2021] distances, both of which are metrics with known LSH families Andoni and Indyk [2006], Jain et al. [2008]. Therefore, stochastic classifiers trained to be fair with respect to these costs automatically reduce incentives to manipulate features, and if such classifiers are derandomized using fairness-preserving methods, this quality is probably approximately preserved.

Bibliography

Jacob D Abernethy and Rafael M Frongillo. A characterization of scoring rules for linear properties. In *COLT*, 2012.

Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. A reductions approach to fair classification. In *International Conference on Machine Learning*, pages 60–69. PMLR, 2018.

Charu C Aggarwal and ChengXiang Zhai. A survey of text classification algorithms. In *Mining text data*, pages 163–222. Springer, 2012.

Saba Ahmadi, Hedyeh Beyhaghi, Avrim Blum, and Keziah Naggita. On classification of strategic agents who can both game and improve. *arXiv preprint arXiv:2203.00124*, 2022.

Emrah Akyol, Cedric Langbort, and Tamer Basar. Price of transparency in strategic machine learning. *arXiv preprint arXiv:1610.08210*, 2016.

Syed Mumtaz Ali and Samuel D Silvey. A general class of coefficients of divergence of one distribution from another. *Journal of the Royal Statistical Society: Series B (Methodological)*, 28(1):131–142, 1966.

Nihesh Anderson, Suman K Bera, Syamantak Das, and Yang Liu. Distributional individual fairness in clustering. *arXiv preprint arXiv:2006.12589*, 2020.

Alexandr Andoni and Piotr Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *2006 47th annual IEEE symposium on foundations of computer science (FOCS'06)*, pages 459–468. IEEE, 2006.

Alexandr Andoni and Ilya Razenshteyn. Optimal data-dependent hashing for approximate near neighbors. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 793–801, 2015.

Alexandr Andoni, Piotr Indyk, Huy L Nguyen, and Ilya Razenshteyn. Beyond locality-sensitive hashing. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 1018–1028. SIAM, 2014.

Arthur Asuncion and David Newman. Uci machine learning repository, 2007.

Kamyar Azizzadenesheli, Anqi Liu, Fanny Yang, and Animashree Anandkumar. Regularized learning for domain adaptation under label shifts. *arXiv preprint arXiv:1903.09734*, 2019.

Maria-Florina Balcan, Avrim Blum, Nika Haghtalab, and Ariel D Procaccia. Commitment without regrets: Online learning in stackelberg security games. In *Proceedings of the sixteenth ACM conference on economics and computation*, pages 61–78, 2015.

Solon Barocas, Moritz Hardt, and Arvind Narayanan. *Fairness and Machine Learning*. fairmlbook.org, 2019. `http://www.fairmlbook.org`.

Flavia Barsotti, Rüya Gökhan Koçer, and Fernando P Santos. Transparency, detection and imitation in strategic classification. In *Proceedings of the 31st International Joint Conference on Artificial Intelligence, IJCAI 2022*. International Joint Conferences on Artificial Intelligence (IJCAI), 2022.

Peter L Bartlett, Michael I Jordan, and Jon D McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006.

Yahav Bechavod, Katrina Ligett, Zhiwei Steven Wu, and Juba Ziani. Causal feature discovery through strategic modification, 2020.

Yahav Bechavod, Chara Podimata, Steven Wu, and Juba Ziani. Information discrepancy in strategic learning. In *International Conference on Machine Learning*, pages 1691–1715. PMLR, 2022.

Rachel KE Bellamy, Kuntal Dey, Michael Hind, Samuel C Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilovic, et al. Ai fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias. *arXiv preprint arXiv:1810.01943*, 2018.

Shai Ben-David, Dávid Pál, and Shai Shalev-Shwartz. agnostic online learning. In *COLT*, 2009.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Vaughan. A theory of learning from different domains. *Machine Learning*, 2010a.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1):151–175, 2010b.

Omer Ben-Porat and Moshe Tennenholtz. Best response regression. In *Proceedings* *of the 31st International Conference on Neural Information Processing Systems*, pages 1498–1507, 2017.

Umang Bhatt, Alice Xiang, Shubham Sharma, Adrian Weller, Ankur Taly, Yunhan Jia, Joydeep Ghosh, Ruchir Puri, José MF Moura, and Peter Eckersley. Explainable machine learning in deployment. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 648–657, 2020.

Daniel Björkegren, Joshua E Blumenstock, and Samsun Knight. Manipulation-proof machine learning. *arXiv preprint arXiv:2004.03865*, 2020.

Daniel Björkegren, Joshua E. Blumenstock, and Samsun Knight. Manipulation-proof machine learning. *arXiv preprint*, 2020.

Board of Governors of the Federal Reserve System (US). *Report to the congress on credit scoring and its effects on the availability and affordability of credit.* Board of Governors of the Federal Reserve System, 2007.

Stephen Boyd and Lieven Vandenberghe. *Convex optimization.* Cambridge university press, 2004.

Mark Braverman and Sumegha Garg. The role of randomness and noise in strategic classification. In *1st Symposium on Foundations of Responsible Computing (FORC 2020).* Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.

Andrei Z Broder. On the resemblance and containment of documents. In *Proceedings. Compression and Complexity of SEQUENCES 1997 (Cat. No. 97TB100171)*, pages 21–29. IEEE, 1997.

Gavin Brown, Shlomi Hod, and Iden Kalemaj. Performative prediction in a stateful world, 2020.

Michael Brückner and Tobias Scheffer. Stackelberg games for adversarial prediction problems. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 547–555, 2011.

Jeremy Buhler. Efficient large-scale sequence comparison by locality-sensitive hashing. *Bioinformatics*, 17(5):419–428, 2001.

Florentin Butaru, Qingqing Chen, Brian Clark, Sanmay Das, Andrew W Lo, and Akhtar Siddique. Risk and risk management in the credit card industry. *Journal of Banking & Finance*, 72:218–239, 2016.

Tom Bylander. Learning linear threshold functions in the presence of classification noise. In *Proceedings of the seventh annual conference on Computational learning theory*, pages 340–347. ACM, 1994.

Richard H. Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. A limited memory algorithm for bound constrained optimization. *SIAM J. Sci. Comput.*, 16(5): 1190–1208, September 1995.

Yang Cai, Constantinos Daskalakis, and Christos Papadimitriou. Optimum statistical estimation with strategic data sources. In *Conference on Learning Theory*, pages 280–296. PMLR, 2015.

Nicolo Cesa-Bianchi, Eli Dichterman, Paul Fischer, Eli Shamir, and Hans Ulrich Simon. Sample-efficient strategies for learning in the presence of noise. *Journal of the ACM (JACM)*, 46(5):684–719, 1999.

Nicolo Cesa-Bianchi, Shai Shalev-Shwartz, and Ohad Shamir. Online learning of noisy data. *IEEE Transactions on Information Theory*, 57(12):7907–7931, 2011.

Anirban Chakraborty, Manaar Alam, Vishal Dey, Anupam Chattopadhyay, and Debdeep Mukhopadhyay. Adversarial attacks and defences: A survey, 2018.

Moses S Charikar. Similarity estimation techniques from rounding algorithms. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 380–388, 2002.

Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12 (3), 2011.

Yatong Chen, Jialu Wang, and Yang Liu. Strategic recourse in linear classification. *arXiv preprint arXiv:2011.00355*, 2020a.

Yatong Chen, Jialu Wang, and Yang Liu. Linear classifiers that encourage constructive adaptation. *arXiv preprint arXiv:2011.00355*, 2021.

Yatong Chen, Reilly Raab, and Yang Liu. Bounded Fairness Transferability subject to Distribution Shift. In *Submitted to Workshop on Algorithmic Fairness through the Lens of Causality and Robustness, Conference on Neural Information Processing Systems*, 2021 (Under Review; Available upon request).

Yatong Chen, Reilly Raab, Jialu Wang, and Yang Liu. Fairness transferability subject to bounded distribution shift. *Advances in neural information processing systems*, 35:11266–11278, 2022.

Yatong Chen, Zeyu Tang, Kun Zhang, and Yang Liu. Model transferability with responsive decision subjects. In *International Conference on Machine Learning*, pages 4921–4952. PMLR, 2023.

Yatong Chen, Andrew Estronell, Yevgeniy Vorobeychik, and Yang Liu. To give or not to give? in the impacts of strategically withheld recourse. *arXiv preprint*, 2024.

Yiling Chen, Chara Podimata, Ariel D Procaccia, and Nisarg Shah. Strategyproof linear regression in high dimensions. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, pages 9–26, 2018.

Yiling Chen, Yang Liu, and Chara Podimata. Learning strategy-aware linear classifiers, 2020b.

Jiacheng Cheng, Tongliang Liu, Kotagiri Ramamohanarao, and Dacheng Tao. Learning with bounded instance-and label-dependent label noise. *ICML, arXiv:1709.03768*, 2020.

Flavio Chierichetti, Ravi Kumar, Alessandro Panconesi, and Erisa Terolli. On the distortion of locality sensitive hashing. *SIAM Journal on Computing*, 48(2): 350–372, 2019.

Alexandra Chouldechova. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big data*, 5(2):153–163, 2017.

Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*, pages 1310–1320. PMLR, 2019.

Andrew Cotter, Maya Gupta, and Harikrishna Narasimhan. On making stochastic classifiers deterministic. *Advances in Neural Information Processing Systems (NeurIPS)*, 2019a.

Andrew Cotter, Heinrich Jiang, Maya R Gupta, Serena Wang, Taman Narayan, Seungil You, and Karthik Sridharan. Optimization with non-differentiable constraints with applications to fairness, recall, churn, and other goals. *J. Mach. Learn. Res.*, 20(172):1–59, 2019b.

Koby Crammer, Michael Kearns, and Jennifer Wortman. Learning from multiple sources. *Journal of Machine Learning Research*, 9(8), 2008.

Joshua Cutler, Dmitriy Drusvyatskiy, and Zaid Harchaoui. Stochastic optimization under distributional drift, 2021.

Alexander D'Amour, Hansa Srinivasan, James Atwood, Pallavi Baljekar, D Sculley, and Yoni Halpern. Fairness is not static: deeper understanding of long term fairness via simulation studies. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 525–534, 2020.

Shai Ben David, Tyler Lu, Teresa Luu, and Dávid Pál. Impossibility theorems for domain adaptation. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 129–136. JMLR Workshop and Conference Proceedings, 2010.

Sarah Dean, Sarah Rich, and Benjamin Recht. Recommendations and user agency: the reachability of collaboratively-filtered information. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 436–445, 2020.

Ofer Dekel, Felix Fischer, and Ariel D Procaccia. Incentive compatible regression learning. *Journal of Computer and System Sciences*, 76(8):759–777, 2010.

Thomas G Dietterich. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer, 2000.

Frances Ding, Moritz Hardt, John Miller, and Ludwig Schmidt. Retiring adult: New datasets for fair machine learning. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021. URL `https://openreview.net/forum?id=bYi_2708mKK`.

Jinshuo Dong, Aaron Roth, Zachary Schutzman, Bo Waggoner, and Zhiwei Steven Wu. Strategic classification from revealed preferences. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, EC '18, New York, NY, USA, 2018a. Association for Computing Machinery.

Jinshuo Dong, Aaron Roth, Zachary Schutzman, Bo Waggoner, and Zhiwei Steven Wu. Strategic classification from revealed preferences. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, 2018b.

Roy Dong and Lillian J. Ratliff. Approximate regions of attraction in learning with decision-dependent distributions, 2021.

Dmitriy Drusvyatskiy and Lin Xiao. Stochastic optimization with decision-dependent distributions. *arXiv preprint arXiv:2011.11173*, 2020.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017a.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017b. URL `http://archive.ics.uci.edu/ml`.

Anjan Dutta, Josep Lladós, and Umapada Pal. A symbol spotting approach in graphical documents by hashing serialized graphs. *Pattern Recognition*, 46(3): 752–768, 2013.

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

Frederick Eberhardt and Richard Scheines. Interventions and causal inference. *Philosophy of Science*, 74(5):981–995, 2007. ISSN 00318248, 1539767X.

Ahmad-Reza Ehyaei, Amir-Hossein Karimi, Bernhard Schölkopf, and Setareh Maghsudi. Robustness implies fairness in causal algorithmic recourse. In *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*, pages 984–1001, 2023.

Danielle Ensign, Sorelle A Friedler, Scott Neville, Carlos Scheidegger, and Suresh Venkatasubramanian. Runaway Feedback Loops in Predictive Policing. In *Conference of Fairness, Accountability, and Transparency*, 2018.

Andrew Estornell, Yatong Chen, Sanmay Das, Yang Liu, and Yevgeniy Vorobeychik. Incentivizing recourse through auditing in strategic classification. In *IJCAI*, pages 400–408, 08 2023a.

Andrew Estornell, Sanmay Das, Yang Liu, and Yevgeniy Vorobeychik. Group-fair classification with strategic agents. In *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*, pages 389–399, 2023b.

Michael Feldman, Sorelle A Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. In *proceedings* *of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 259–268, 2015.

Abraham D Flaxman, Adam Tauman Kalai, and H Brendan McMahan. Online convex optimization in the bandit setting: Gradient descent without a gradient. In *The Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 385–394, 2005.

Hidde Fokkema, Damien Garreau, and Tim van Erven. The risks of recourse in binary classification. In *International Conference on Artificial Intelligence and Statistics*, pages 550–558. PMLR, 2024.

Benoît Frénay and Michel Verleysen. Classification in the presence of label noise: a survey. *IEEE transactions on neural networks and learning systems*, 2014.

Sorelle A Friedler, Carlos Scheidegger, and Suresh Venkatasubramanian. On the (im)possibility of fairness. *arXiv preprint arXiv:1609.07236*, 2016.

Sorelle A Friedler, Carlos Scheidegger, Suresh Venkatasubramanian, Sonam Choudhary, Evan P Hamilton, and Derek Roth. A comparative study of fairness-enhancing interventions in machine learning. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 329–338, 2019.

Daniel Friedman and Barry Sinervo. *Evolutionary games in natural, social, and virtual worlds.* Oxford University Press, 2016.

Stephen Gillen, Christopher Jung, Michael Kearns, and Aaron Roth. Online learning with an unknown fairness metric. *arXiv preprint arXiv:1802.06936*, 2018.

Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.

Gene H Golub and Charles F Van Loan. *Matrix computations*, volume 3. JHU press, 2013.

Mingming Gong, Kun Zhang, Tongliang Liu, Dacheng Tao, Clark Glymour, and Bernhard Schölkopf. Domain adaptation with conditional transferable components. In *International conference on machine learning*, pages 2839–2848. PMLR, 2016.

Bryce Goodman and Seth Flaxman. European union regulations on algorithmic decision-making and a "right to explanation". *AI Magazine*, 38(3):50–57, Oct 2017.

Arthur Gretton, Alex Smola, Jiayuan Huang, Marcel Schmittfull, Karsten Borgwardt, and Bernhard Schölkopf. Covariate shift by kernel mean matching. *Dataset shift in machine learning*, 3(4):5, 2009.

Nina Grgić-Hlača, Muhammad Bilal Zafar, Krishna Gummadi, and Adrian Weller. On fairness, diversity, and randomness in algorithmic decision making. In *4th Workshop on Fairness, Accountability, and Transparency in Machine Learning*, 2017.

Jiaxian Guo, Mingming Gong, Tongliang Liu, Kun Zhang, and Dacheng Tao. Ltf: A label transformation framework for correcting label shift. In *International Conference on Machine Learning*, pages 3843–3853. PMLR, 2020.

Vivek Gupta, Pegah Nokhiz, Chitradeep Dutta Roy, and Suresh Venkatasubramanian. Equalizing recourse across groups. *arXiv preprint arXiv:1909.03166*, 2019.

Nika Haghtalab, Nicole Immorlica, Brendan Lucier, and Jack Z. Wang. Maximizing welfare with incentive-aware evaluation mechanisms. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization, 2020.

Moritz Hardt, Nimrod Megiddo, Christos Papadimitriou, and Mary Wootters. Strategic classification. In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science*, New York, NY, USA, 2016a. Association for Computing Machinery.

Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. In *Advances in Neural Information Processing Systems*, pages 3315–3323, 2016b.

Moritz Hardt, Meena Jagadeesan, and Celestine Mendler-Dünner. Performative power. *arXiv preprint arXiv:2203.17232*, 2022.

Keegan Harris, Valerie Chen, Joon Kim, Ameet Talwalkar, Hoda Heidari, and Steven Z Wu. Bayesian persuasion for algorithmic recourse. *Advances in Neural Information Processing Systems*, 35:11131–11144, 2022.

Miguel A. Hernán, Wei Wang, and David E. Leaf. Target Trial Emulation: A Framework for Causal Inference From Observational Data. *JAMA*, 328(24):2446–2447, 12 2022. ISSN 0098-7484. doi: 10.1001/jama.2022.21383.

Lily Hu, Nicole Immorlica, and Jennifer Wortman Vaughan. The disparate effects of strategic manipulation. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 259–268, 2019.

Ling Huang, Anthony D Joseph, Blaine Nelson, Benjamin IP Rubinstein, and J Doug Tygar. Adversarial machine learning. In *ACM Workshop on Security and Artificial Intelligence*, 2011.

Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 604–613, 1998.

Rishabh Iyer, Stefanie Jegelka, and Jeff Bilmes. Fast semidifferential based submodular function optimization. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2013.

Zachary Izzo, Lexing Ying, and James Zou. How to learn when data reacts to your model: performative gradient descent. In *International Conference on Machine Learning*, pages 4641–4650. PMLR, 2021a.

Zachary Izzo, Lexing Ying, and James Zou. How to learn when data reacts to your model: Performative gradient descent. *CoRR*, 2021b.

Meena Jagadeesan, Tijana Zrnic, and Celestine Mendler-Dünner. Regret minimization with performative feedback. In *International Conference on Machine Learning*, pages 9760–9785. PMLR, 2022a.

Meena Jagadeesan, Tijana Zrnic, and Celestine Mendler-Dünner. Regret minimization with performative feedback, 2022b.

Prateek Jain, Brian Kulis, and Kristen Grauman. Fast image search for learned metrics. In *2008 IEEE Conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2008.

Jing Jiang. A literature survey on domain adaptation of statistical classifiers. *URL: http://sifaka. cs. uiuc. edu/jiang4/domainadaptation/survey*, 3:1–12, 2008.

Shalmali Joshi, Oluwasanmi Koyejo, Warut Vijitbenjaronk, Been Kim, and Joydeep Ghosh. Towards realistic individual recourse and actionable explanations in black-box decision making systems, 2019.

Christopher Jung, Michael Kearns, Seth Neel, Aaron Roth, Logan Stapleton, and Zhiwei Steven Wu. An algorithmic framework for fairness elicitation. In *2nd Symposium on Foundations of Responsible Computing*, volume 31, page 21, 2021.

Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4893–4902, 2019.

A.-H. Karimi, B. Schölkopf, and I. Valera. Algorithmic recourse: from counterfactual explanations to interventions, 2020a.

Amir-Hossein Karimi, Gilles Barthe, Bernhard Schölkopf, and Isabel Valera. A survey of algorithmic recourse: definitions, formulations, solutions, and prospects, 2020b.

Amir-Hossein Karimi, Julius von Kügelgen, Bernhard Schölkopf, and Isabel Valera. Algorithmic recourse under imperfect causal knowledge: a probabilistic approach, 2020c.

Amir-Hossein Karimi, Bernhard Schölkopf, and Isabel Valera. Algorithmic recourse: from counterfactual explanations to interventions. In *Proceedings of the 2021*

ACM conference on fairness, accountability, and transparency, pages 353–362, 2021.

Amir E Khandani, Adlar J Kim, and Andrew W Lo. Consumer credit-risk models via machine-learning algorithms. *Journal of Banking & Finance*, 34(11):2767–2787, 2010.

Niki Kilbertus, Manuel Gomez Rodriguez, Bernhard Schölkopf, Krikamol Muandet, and Isabel Valera. Fair decisions despite imperfect predictions. In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 277–287. PMLR, 2020. URL `http://proceedings.mlr.press/v108/kilbertus20a.html`.

Michael P Kim, Omer Reingold, and Guy N Rothblum. Fairness through computationally-bounded awareness. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 4847–4857, 2018.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Jon Kleinberg and Manish Raghavan. How do classifiers induce agents to invest effort strategically? *ACM Transactions on Economics and Computation (TEAC)*, 8(4):1–23, 2020.

Brian Kulis and Kristen Grauman. Kernelized locality-sensitive hashing for scalable image search. In *2009 IEEE 12th international conference on computer vision*, pages 2130–2137. IEEE, 2009.

Matt J Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. Counterfactual fairness. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4066–4076. Curran Associates, Inc., 2017. URL `http://papers.nips.cc/paper/6995-counterfactual-fairness.pdf`.

Sagi Levanon and Nir Rosenfeld. Strategic classification made practical. In *International Conference on Machine Learning*, pages 6243–6253. PMLR, 2021.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M. Hospedales. Learning to generalize: Meta-learning for domain generalization, 2017.

Qiang Li and Hoi-To Wai. State dependent performative prediction with stochastic approximation. In *International Conference on Artificial Intelligence and Statistics*, pages 3164–3186. PMLR, 2022.

Friedrich Liese and Igor Vajda. On divergences and informations in statistics and information theory. *IEEE Transactions on Information Theory*, 52(10):4394–4412, 2006.

Zachary Lipton, Yu-Xiang Wang, and Alexander Smola. Detecting and correcting for label shift with black box predictors. In *International conference on machine learning*, pages 3122–3130. PMLR, 2018.

Lydia T Liu, Sarah Dean, Esther Rolf, Max Simchowitz, and Moritz Hardt. Delayed impact of fair machine learning. In *International Conference on Machine Learning*, pages 3150–3158. PMLR, 2018.

Lydia T Liu, Ashia Wilson, Nika Haghtalab, Adam Tauman Kalai, Christian Borgs, and Jennifer Chayes. The disparate equilibria of algorithmic decision making when individuals invest rationally. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 381–391, 2020.

Tongliang Liu and Dacheng Tao. Classification with noisy labels by importance reweighting. *IEEE Transactions on pattern analysis and machine intelligence*, 38(3):447–461, 2016.

Yang Liu and Yiling Chen. Machine-learning aided peer prediction. In *Proceedings of the 2017 ACM Conference on Economics and Computation*, pages 63–80, 2017a.

Yang Liu and Yiling Chen. Machine Learning aided Peer Prediction. *ACM EC*, June 2017b.

Yang Liu and Mingyan Liu. An online learning approach to improving the quality of crowd-sourcing. *ACM SIGMETRICS Performance Evaluation Review*, 2015.

Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Unsupervised domain adaptation with residual transfer networks. *arXiv preprint arXiv:1602.04433*, 2016.

Daniel Lowd and Christopher Meek. Adversarial learning. In *ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, 2005.

Michal Lukasik, Srinadh Bhojanapalli, Aditya Menon, and Sanjiv Kumar. Does label smoothing mitigate label noise? In *International Conference on Machine Learning*, pages 6448–6458. PMLR, 2020.

Yueming Lyu and Ivor W Tsang. Curriculum loss: Robust learning and generalization against label corruption. *arXiv preprint arXiv:1905.10045*, 2019.

Xingjun Ma, Hanxun Huang, Yisen Wang, Simone Romano, Sarah Erfani, and James Bailey. Normalized loss functions for deep learning with noisy labels. In *International Conference on Machine Learning*, pages 6543–6553. PMLR, 2020.

Prasanta Chandra Mahalanobis. On the generalized distance in statistics. *Proceedings of the National Institute of Sciences (Calcutta)*, 2:49–55, 1936.

Chinmay Maheshwari, Chih-Yuan Chiu, Eric Mazumdar, S. Shankar Sastry, and Lillian J. Ratliff. Zeroth-order methods for convex-concave minmax problems: Applications to decision-dependent risk minimization, 2021.

Naresh Manwani and PS Sastry. Noise tolerance under risk minimization. *IEEE transactions on cybernetics*, 43(3):1146–1151, 2013.

Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, 54(6):1–35, 2021.

Celestine Mendler-Dünner, Juan Perdomo, Tijana Zrnic, and Moritz Hardt. Stochastic optimization for performative prediction. In *Advances in Neural Information Processing Systems*, pages 4929–4939. Curran Associates, Inc., 2020.

Celestine Mendler-Dünner, Frances Ding, and Yixin Wang. Anticipating performativity by predicting from predictions. In *Advances in Neural Information Processing Systems*, 2022.

Michele Merler, Nalini Ratha, Rogerio S Feris, and John R Smith. Diversity in faces. *arXiv preprint arXiv:1901.10436*, 2019.

John Miller, Smitha Milli, and Moritz Hardt. Strategic classification is causal modeling in disguise. In *International Conference on Machine Learning*, pages 6917–6926. PMLR, 2020.

John Miller, Juan Perdomo, and Tijana Zrnic. Outside the echo chamber: Optimizing the performative risk, 2021.

Smitha Milli, John Miller, Anca D Dragan, and Moritz Hardt. The social cost of strategic classification. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 230–239, 2019.

Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation, 2013.

Zachary Nado, Shreyas Padhy, D. Sculley, Alexander D'Amour, Balaji Lakshminarayanan, and Jasper Snoek. Evaluating prediction-time batch normalization for robustness under covariate shift, 2021.

Adhyyan Narang, Evan Faulkner, Dmitriy Drusvyatskiy, Maryam Fazel, and Lillian J Ratliff. Multiplayer performative prediction: Learning in decision-dependent games. *arXiv preprint arXiv:2201.03398*, 2022.

Harikrishna Narasimhan. Learning with complex loss functions and constraints. In *International Conference on Artificial Intelligence and Statistics*, pages 1646–1654. PMLR, 2018.

Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. Learning with noisy labels. In *Advances in neural information processing systems*, pages 1196–1204, 2013.

Curtis G. Northcutt, Lu Jiang, and Isaac L. Chuang. Confident learning: Estimating uncertainty in dataset labels. *Journal of Artificial Intelligence Research (JAIR)*, 70:1373–1411, 2021.

Matthew Olckers and Toby Walsh. Incentives to offer algorithmic recourse, 2023.

Andrew Orso, Jon Lee, and Siqian Shen. Submodular minimization in the context of modern lp and milp methods and solvers. In *Proceedings of the 14th International Symposium on Experimental Algorithms - Volume 9125*, page 193–204, Berlin, Heidelberg, 2015. Springer-Verlag. ISBN 9783319200859.

Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples, 2016.

Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1944–1952, 2017.

Judea Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, 2000.

Juan Perdomo, Tijana Zrnic, Celestine Mendler-Dünner, and Moritz Hardt. Performative prediction. In *International Conference on Machine Learning*, pages 7599–7609. PMLR, 2020.

Georgios Piliouras and Fang-Yi Yu. Multi-agent performative prediction: From global stability and optimality to chaos. *arXiv preprint arXiv:2201.10483*, 2022.

Rafael Pinot, Laurent Meunier, Alexandre Araujo, Hisashi Kashima, Florian Yger, Cedric Gouy-Pailler, and Jamal Atif. Theoretical evidence for adversarial robustness through randomization. *Advances in Neural Information Processing Systems*, 32:11838–11848, 2019.

Rafael Pinot, Raphael Ettedgui, Geovani Rizk, Yann Chevaleyre, and Jamal Atif. Randomization matters how to defend against strong adversarial attacks. In *International Conference on Machine Learning*, pages 7717–7727. PMLR, 2020.

Reilly Raab and Yang Liu. Unintended selection: Persistent qualification rate disparities and interventions. *Advances in Neural Information Processing Systems*, 2021.

Cyrus Rashtchian. Lecture 09: LSH, 2019. URL: `http://madscience.ucsd.edu/notes/lec9.pdf`.

Mark D Reid and Robert C Williamson. Information, divergence and risk for binary experiments. *Journal of Machine Learning Research*, 12(Mar):731–817, 2011.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 1135–1144, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450342322. doi: 10.1145/2939672.2939778. URL `https://doi.org/10.1145/2939672.2939778`.

Nir Rosenfeld, Sophie Hilgard, Sai Srivatsa Ravindranath, and David C. Parkes. From predictions to decisions: Using lookahead regularization, 2020.

Guy Rothblum and Gal Yona. Probably approximately metric-fair learning. In *International Conference on Machine Learning*, pages 5680–5688. PMLR, 2018.

Ronitt Rubinfeld. MIT 6.842, 2012. URL: `https://people.csail.mit.edu/ronitt/COURSE/S12/handouts/lec5.pdf`.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1 (5):206–215, 2019.

Matti Ryynanen and Anssi Klapuri. Query by humming of midi and audio using locality sensitive hashing. In *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2249–2252. IEEE, 2008.

Andreas S. Schulz and Martin Skutella. Greedy approximation algorithms for finding dense components in a graph. *SIAM Journal on Computing*, 32(4):922–936, 2003. doi: 10.1137/S0097539701398542.

Clayton Scott. A rate of convergence for mixture proportion estimation, with application to learning from noisy labels. In *AISTATS*, 2015.

Clayton Scott, Gilles Blanchard, Gregory Handy, Sara Pozzi, and Marek Flaska. Classification with asymmetric label noise: Consistency and maximal denoising. In *COLT*, 2013.

Andrew Selbst and Julia Powles. "meaningful information" and the right to explanation. In *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*, Proceedings of Machine Learning Research. PMLR, 2018.

Saeed Sharifi-Malvajerdi, Michael Kearns, and Aaron Roth. Average individual fairness: Algorithms, generalization and experiments. *Advances in Neural Information Processing Systems*, 32:8242–8251, 2019.

Yonadav Shavit, Benjamin Edelman, and Brian Axelrod. Causal strategic linear regression. *International Conference on Machine Learning*, pages 8676–8686, 2020.

Paras Sheth, Raha Moraffah, K Selçuk Candan, Adrienne Raglin, and Huan Liu. Domain generalization–a causal perspective. *arXiv preprint arXiv:2209.15177*, 2022.

Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244, 2000.

Naeem Siddiqi. *Credit Risk Scorecards: Developing And Implementing Intelligent Credit Scoring*. SAS Publishing, 2005. ISBN 1590475038.

Chuanbiao Song, Kun He, Liwei Wang, and John E. Hopcroft. Improving the generalization of adversarial training with domain adaptation, 2019.

Sadhan Sood and Dmitri Loguinov. Probabilistic near-duplicate detection using simhash. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 1117–1126, 2011.

Gabriel Stanovsky, Noah A Smith, and Luke Zettlemoyer. Evaluating gender bias in machine translation. *arXiv preprint arXiv:1906.00591*, 2019.

Ingo Steinwart, Chloé Pasin, Robert C Williamson, Siyu Zhang, et al. Elicitation and identification of properties. In *COLT*, 2014.

Guillaume Stempfel and Liva Ralaivola. Learning svms from sloppily labeled data. In *International Conference on Artificial Neural Networks*, pages 884–893. Springer, 2009.

Masashi Sugiyama, Matthias Krauledat, and Klaus-Robert Müller. Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, 8(5), 2007.

Masashi Sugiyama, Taiji Suzuki, Shinichi Nakajima, Hisashi Kashima, Paul von Bünau, and Motoaki Kawanabe. Direct importance estimation for covariate shift adaptation. *Annals of the Institute of Statistical Mathematics*, 60(4):699–746, 2008.

Sainbayar Sukhbaatar and Rob Fergus. Learning from noisy labels with deep neural networks. *arXiv preprint arXiv:1406.2080*, 2(3):4, 2014.

Zeyu Tang, Yatong Chen, Yang Liu, and Kun Zhang. Tier balancing: Towards dynamic fairness over underlying causal factors. *arXiv preprint arXiv:2301.08987*, 2023.

Peter D. Taylor and Leo B. Jonker. Evolutionary stable strategies and game dynamics. *Mathematical Biosciences*, 1978.

Stratis Tsirtsis, Behzad Tabibian, Moein Khajehnejad, Adish Singla, Bernhard Schölkopf, and Manuel Gomez-Rodriguez. Optimal Decision Making Under Strategic Behavior. *arXiv e-prints*, page arXiv:1905.09239, May 2019.

Karl Tuyls, Pieter Jan'T Hoen, and Bram Vanschoenwinkel. An evolutionary dynamical analysis of multi-agent learning in iterated games. *Autonomous Agents and Multi-Agent Systems*, 2006.

Sohini Upadhyay, Shalmali Joshi, and Himabindu Lakkaraju. Towards robust and reliable algorithmic recourse. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 16926–16937. Curran Associates, Inc., 2021. URL `https://proceedings.neurips.cc/paper/2021/file/8ccfb1140664a5fa63177fb6e07352f0-Paper.pdf`.

Berk Ustun, Alexander Spangher, and Yang Liu. Actionable recourse in linear classification. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 10–19, 2019.

Thomas Varsavsky, Mauricio Orbes-Arteaga, Carole H. Sudre, Mark S. Graham, Parashkev Nachev, and M. Jorge Cardoso. Test-time unsupervised domain adaptation, 2020.

Suresh Venkatasubramanian and Mark Alfano. The philosophical basis of algorithmic recourse. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 284–293, 2020.

Julius Von Kügelgen, Amir-Hossein Karimi, Umang Bhatt, Isabel Valera, Adrian Weller, and Bernhard Schölkopf. On the fairness of causal algorithmic recourse. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 9584–9594, 2022.

Julius von Kügelgen, Umang Bhatt, Amir-Hossein Karimi, Isabel Valera, Adrian Weller, and Bernhard Scholkopf. On the fairness of causal algorithmic recourse, 2020.

Julius von Kügelgen, Amir-Hossein Karimi, Umang Bhatt, Isabel Valera, Adrian

Weller, and Bernhard Schölkopf. On the fairness of causal algorithmic recourse, 2022.

Yevgeniy Vorobeychik and Murat Kantarcioglu. *Adversarial Machine Learning*. Morgan & Claypool Publishers, 2018.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization, 2021a.

Hao Wang, Berk Ustun, and Flavio Calmon. Repairing without retraining: Avoiding disparate impact with counterfactual distributions. In *International Conference on Machine Learning*, pages 6618–6627. PMLR, 2019.

Jialu Wang, Yang Liu, and Caleb Levy. Fair classification with group-dependent label noise. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 526–536, 2021b.

Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Tao Qin, Wang Lu, Yiqiang Chen, Wenjun Zeng, and Philip S. Yu. Generalizing to unseen domains: A survey on domain generalization, 2021c.

Jingdong Wang, Heng Tao Shen, Jingkuan Song, and Jianqiu Ji. Hashing for similarity search: A survey. *arXiv preprint arXiv:1408.2927*, 2014.

Qing Wang, S.R. Kulkarni, and S. Verdu. Divergence estimation of continuous distributions based on data-dependent partitions. *IEEE Transactions on Information Theory*, 51(9):3064–3074, 2005. doi: 10.1109/TIT.2005.853314.

L.F. Wightman and Law School Admission Council. *LSAC National Longitudinal*

Bar Passage Study. LSAC research report series. Law School Admission Council, 1998. URL https://books.google.com/books?id=O9A7AQAAIAAJ.

Jimmy Wu, Yatong Chen, and Yang Liu. Metric-fair classifier derandomization. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 23999–24016. PMLR, 17–23 Jul 2022.

Xiaobo Xia, Tongliang Liu, Bo Han, Nannan Wang, Mingming Gong, Haifeng Liu, Gang Niu, Dacheng Tao, and Masashi Sugiyama. Parts-dependent label noise: Towards instance-dependent label noise, 2020.

Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2691–2699, 2015.

Renchunzi Xie, Hongxin Wei, Lei Feng, and Bo An. Gearnet: Stepwise dual learning for weakly supervised domain adaptation. *AAAI Conference on Artificial Intelligence*, 2022.

Huan Xu and Shie Mannor. Robustness and generalization. *Machine learning*, 86 (3):391–423, 2012.

Tian Xu, Jennifer White, Sinan Kalkan, and Hatice Gunes. Investigating bias and fairness in facial expression recognition. In *European Conference on Computer Vision*, pages 506–523. Springer, 2020.

Yu Yao, Tongliang Liu, Bo Han, Mingming Gong, Jiankang Deng, Gang Niu, and Masashi Sugiyama. Dual t: Reducing estimation error for transition matrix in label-noise learning. *arXiv preprint arXiv:2006.07805*, 2020.

I-Cheng Yeh and Che-hui Lien. The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert Systems with Applications*, 36(2):2473–2480, 2009.

Bianca Zadrozny. Learning and evaluating classifiers under sample selection bias. In *Proceedings of the twenty-first international conference on Machine learning*, page 114, 2004.

Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez-Rodriguez, and Krishna P. Gummadi. Fairness constraints: A flexible approach for fair classification. *Journal of Machine Learning Research*, 20(75):1–42, 2019. URL `http://jmlr.org/papers/v20/18-262.html`.

Kai Zhang, Vincent Zheng, Qiaojun Wang, James Kwok, Qiang Yang, and Ivan Marsic. Covariate shift in hilbert space: A solution via surrogate kernels. In *International Conference on Machine Learning*, pages 388–395. PMLR, 2013a.

Kun Zhang, Bernhard Schölkopf, Krikamol Muandet, and Zhikun Wang. Domain adaptation under target and conditional shift. In *International Conference on Machine Learning*, pages 819–827. PMLR, 2013b.

Kun Zhang, Mingming Gong, Petar Stojanov, Biwei Huang, QINGSONG LIU, and Clark Glymour. Domain adaptation as a problem of inference on graphical models. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 4965–4976. Curran Associates, Inc., 2020a.

Xueru Zhang, Ruibo Tu, Yang Liu, Mingyan Liu, Hedvig Kjellström, Kun Zhang, and Cheng Zhang. How do fair decisions fare in long-term qualification? In *NeurIPS*, 2020b.

Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael Jordan. Bridging theory and algorithm for domain adaptation. In *International Conference on Machine Learning*, pages 7404–7413. PMLR, 2019.

Han Zhao, Remi Tachet Des Combes, Kun Zhang, and Geoffrey Gordon. On learning invariant representations for domain adaptation. In *International Conference on Machine Learning*, pages 7523–7532. PMLR, 2019.

Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization in vision: A survey. *arXiv preprint arXiv:2103.02503*, 2021.

Tijana Zrnic, Eric Mazumdar, Shankar Sastry, and Michael Jordan. Who leads and who follows in strategic classification? *Advances in Neural Information Processing Systems*, 34:15257–15269, 2021.