Strategic Recourse in Linear Classification

Yatong Chen
UC Santa Cruz
ychen592@ucsc.edu

Jialu Wang
UC Santa Cruz
faldict@ucsc.edu

Yang Liu
UC Santa Cruz
yangliu@ucsc.edu

Abstract

In algorithmic decision making, *recourse* refers to individuals' ability to systematically reverse an unfavorable decision made by an algorithm by altering actionable input variables. Meanwhile, individuals subjected to a classification mechanism are incentivized to behave strategically in order to gain a system's approval. However, not all strategic behavior necessarily leads to adverse results: through appropriate mechanism design, strategic behavior can induce genuine improvement in an individual's qualifications. In this paper, we explore how to design a classifier that achieves high accuracy *while* providing recourse to strategic individuals so as to incentivize them to improve their features in non-manipulative ways. We capture these dynamics using a two-stage game, and show under this model, we can provide analytical results characterizing the equilibrium strategies for both the mechanism designer and the agents. Our results provide insights for designing a machine learning model that focuses not only on the static distribution as of now, but also tries to encourage future improvement.

1 Introduction

In the context of algorithmic decision making, *recourse* refers to the ability of individuals to systematically reverse unfavorable decisions made by algorithms [1, 2]. For example, when an individual is rejected by the bank for a credit card application, the bank should provide practical suggestions to the individual on how to alter their profile to increase their chance of being approved in the future. When carefully implemented, recourse realizes important ethical decision-making principles and fosters greater trust in algorithmic systems [2]. At the same time, individuals who are subject to a classifier's decisions are *strategic*: when they have incentives to be classified in a certain way, they may behave strategically to influence their outcomes. Such behaviors, often referred to as *strategic manipulation* [3], may also lead to disparate effects between different social groups [4], or impose unnecessary social burdens on individuals [5]. Nonetheless, not all strategic behavior necessarily leads to adverse consequences. In many applications, we can leverage strategic behavior to incentivize agents to improve their qualifications [6, 7]. When taken together, the above phenomena suggest a challenging and important mechanism design problem in which the mechanism designer tries to deploy a classifier that is able to 1) classify accurately on the current data distribution and 2) provide *meaningful recourse* where agents are encouraged to to flip a negative decision, but only through improving their true qualification (improvable features). In this work, we address the following question:

Given that individuals will behave strategically, how can we design a classifier that achieves high prediction accuracy while providing individuals recourse by incentivizing improvements?

Like [8], we model the above learning and mechanism design problem as a two-stage, two-player game. The first player is the mechanism designer, who is given a set of labeled examples from some true label function h and is required to publish a classifier f. The second player is the individual or *agent*, who holds a feature x to be revealed to the classifier and is given a chance to "game" it, meaning that the agent may change their x to obtain a favorable outcome from the classifier f. At the same time, the agent incurs a cost for these changes according to a cost function that is known to both players.

Preprint. Under review.

In our setting, we distinguish between *improvable* features , *manipulated* features , and *unactionable* features .This provides us with a formal way to differentiate between honest improvement and pure manipulation. Our cost function model also departs from previous works in strategic classification, where the cost is modeled either as a separable function [8] or the L_2 norm [6]. These cost functions generally do not explicitly capture correlations between changes in the features. For this reason, we choose to model the cost using the Mahalanobis distance [9], which precisely captures such interactions between features. Our empirical results demonstates the effectiveness of our algorithms in terms of incentivizing agents' improving behaviors as well as achieving a high prediction accuracy.

Related work Our work is related to research on strategic classification [8, 10, 11, 12, 13, 14, 15], recourse [1, 2, 16, 17, 18, 19, 20, 21, 22, 23], causal modeling of features [24, 25, 26, 27, 7], as well as algorithmic fairness in machine learning [28, 29, 30, 31, 32, 33].

2 Problem Statement

Our setting involves two players. The first player is a *mechanism designer*, who publishes a binary classifier with the goal of both accurately classifying agents based on their revealed features, as well as incentivizing agents to improve certain features. The second player is a set of agents, each of whom is characterized by a *feature vector*, but may attempt to change their features to so as to obtain a favorable classification outcome. Formally, we have the following game:

Definition 2.1. [Strategic Recourse Game] The players are the mechanism designer and the agents. The agents are sampled from a population distribution $\mathcal{D}$ over a d-dimensional feature space $\mathcal{X} \subseteq \mathbb{R}^d$. Each agent holds a feature vector or profile x. Let $c : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$ be a cost function (known to all players) and let $h : \mathcal{X} \to \{-1, +1\}$ be the true label function that maps x to its true label $h(x)$. In formulating our games, we assume the designer has perfect knowledge of h. In practice, h remains unknown but the mechanism designer has access to a set of samples $\{x_n, h(x_n)\}_{n=1}^N$ drawn from the agent population to approximate her payoff. The two players take the following actions:

1. First, the mechanism designer publishes a classifier $f : \mathcal{X} \to \{-1, +1\}$ with the hope of incentivizing the agents to improve their profiles *and* achieve a high prediction accuracy.
2. Next, each agent reveals a feature vector x' with the hope of being classified as $+1$. If an agent's original feature vector is $x \in \mathcal{X}$ but chooses to reveal $x' \in \mathcal{X}$, she pays a *cost* $c(x, x')$.

The rest of this section is devoted to developing the details of the payoff functions for both players.

2.1 Key Assumptions

On top of the general setting described above, we make three key assumptions to instantiate our discussions:

Linear Threshold Classifier: we assume that the classifier f published by the mechanism designer is a *linear threshold function* of the form $f(x) = \text{sign}\left(w_f^T x - b_f\right)$ where $w_f \in \mathbb{R}^d$ and $b_f \in \mathbb{R}$ are weights, and $\text{sign}(z)$ equals -1 if $z < 0$ and $+1$ otherwise.

Features Taxonomy: we assume that the feature vector x is a concatenation of three disjoint feature sets x_I, x_M, and x_U, namely $x = [x_I \circ x_M \circ x_U] = [x_A \circ x_U]$. *Improvable* features (x_I) are those that the mechanism designer should encourage individuals to change (e.g. education level). *Manipulated* features (x_M) can be changed but shouldn't be encouraged (e.g. strategically change the loan purpose). *Unactionable* features (x_U) are those that cannot be changed (e.g. age, race). Additionally, let the *actionable* features x_A be the concatenation of the improvable and manipulated features, i.e. $x_A = x_I \circ x_M$. We use d_A to denote the dimension of x_A. let there be an analogous partition of the weights into $w_f = w_I \circ w_M \circ w_U$ and $w_A = w_I \circ w_M$.

Cost Function: as in previous works on strategic classification [8, 6], we assume that agents incur a cost for modifying their features. We choose to model this cost using the *Mahalanobis* norm of the feature changes, namely $c(x, x') = \sqrt{(x_A - x'_A)^T S^{-1} (x_A - x'_A)}$. Note that since unactionable features x_U cannot be changed as part of the agent's move, the cost function only accounts for the actionable features x_A. We call $S^{-1} \in \mathbb{R}^{d_A} \times \mathbb{R}^{d_A}$ the *cost covariance matrix*, in which each entry S_{ij}^{-1} indicates the correlation between the cost of modifying x_i and the cost of modifying x_j. In

order for $c(\cdot,\cdot)$ to be a valid norm, S^{-1} is required to be positive definite (PD), i.e. S^{-1} satisfies $x_A^T S^{-1} x_A > 0$ for all $x_A \in \mathbb{R}^{d_A}$. We also assume that S^{-1} is symmetric. We further assume that S is a diagonal block matrix of the following form, which says that there are no substantial correlations between improvable and manipulated variables:

$$S^{-1} = \begin{bmatrix} S_I^{-1} & 0 \\ 0 & S_M^{-1} \end{bmatrix} \quad \text{and} \quad S = \begin{bmatrix} S_I & 0 \\ 0 & S_M \end{bmatrix} . \tag{1}$$

2.2 Players' Payoffs

Next we motivate and define the payoff functions of the two players in details.

2.2.1 The Agent's Payoff

Given a classifier f, an agent who starts out with features x and changes them to x' derives total utility

$$U_f(x, x') = f(x') - c(x, x')$$

The default option for an agent is to change nothing about x, which results in a utility of $f(x)$ (since $c(x, x) \equiv 0$). Agents will therefore only change to some $x' \neq x$ if $U_f(x, x') \geq f(x)$. This motivates the following *best response model* for the agent:

Lemma 1 (Best-Response Agent Model)**.** *Given a classifier $f : \mathcal{X} \to \{-1, +1\}$, a cost function $c : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, and an actionable feature set $\mathcal{X}^\dagger \subseteq \mathcal{X}$ containing the feasible feature vectors to which x can move, an agent with features x has a best response to the classifier given by the following optimization problem:*

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

We provide the proof of Lemma 1 in Appendix A.1.

We will find it useful to distinguish between two types of best response: an *unconstrained* best response $\Delta(x)$ in which the agent can change both improvable and manipulated features, and an *improving* best response $\Delta_I(x)$ in which only improvable features can be changed. Later we will examine what the mechanism designer should do if their goal is to incentivize improving actions from the agents. This motivates a definition of the *improving best response* wherein agents best respond by changing only the improvable features.

Definition 2.2 (Unconstrained Best Response)**.** Let $\Delta : \mathcal{X} \to \mathcal{X}$ denote the *unconstrained best response* of an individual with feature x to f, defined as:

$$\Delta(x) = \begin{cases} \arg\max_{x' \in \mathcal{X}_A^*(x)} U_f(x, x'), & \text{if } U_f(x, x') \geq 0 \\ x, & \text{otherwise} \end{cases}$$

where $\mathcal{X}_A^*(x)$ denotes the set of feature vectors that differ from x only in the actionable features $\mathcal{X}_A$.

Definition 2.3 (Improving Best Response)**.** Let $\Delta_I : \mathcal{X} \to \mathcal{X}$ denote the *improving best response* of the agent with feature x to f, defined as:

$$\Delta_I(x) = \begin{cases} \arg\max_{x' \in \mathcal{X}_I^*(x)} U_f(x, x'), & \text{if } U_f(x, x') \geq 0 \\ x, & \text{otherwise} \end{cases}$$

where $\mathcal{X}_I^*(x)$ denotes the set of feature vectors that differ from x only in the improvable features $\mathcal{X}_I$.

2.2.2 The Mechanism Designer's Payoff

The goal of the mechanism designer is to publish a classifier f that maximizes the classification accuracy while incentivizing individuals to change their improvable features. Mathematically, we formulate the optimization problem for the mechanism designer as follows:

$$\max_{w_f, b_f} \quad \Pr_{x \sim \mathcal{D}} [f(\Delta(x)) = h(x)] + \lambda \Pr_{x \sim \mathcal{D}} [f(\Delta_I(x)) = +1]$$

$$\text{s.t.} \Delta(x) = \begin{cases} \arg\max_{x' \in \mathcal{X}_A^*(x)} U_f(x, x'), & \text{if } U_f(x, x') \geq 0 \\ x, & \text{otherwise} \end{cases}, \quad \Delta_I(x) = \begin{cases} \arg\max_{x' \in \mathcal{X}_I^*(x)} U_f(x, x'), & \text{if } U_f(x, x') \geq 0 \\ x, & \text{otherwise} \end{cases} \tag{2}$$

The first term $\Pr_{x\sim\mathcal{D}}[f(\Delta(x) = h(x))]$ is the prediction accuracy when accounting for agents' strategic behavior. Meanwhile, the mechanism designer also hopes to incentivize actual improvement from the agents and grant *meaningful recourse* by maximizing their chance of being assigned $+1$ when they choose an improving best response; this is what the second term of the objective function, $\Pr_{x\sim\mathcal{D}}[f(\Delta_I(x) = +1)]$, tries to capture. The coefficient λ between the two terms captures the trade-off between providing more recourse and ensuring the prediction accuracy of the algorithm. Notice that when $\lambda = 0$, we obtain the standard strategic classification setting [8], in which the objective function for the mechanism designer is simply to maximize prediction accuracy considering agents' strategic behavior.

3 Agents' Best-response

In this section, we derive the agents' best response equilibrium under the key assumptions that we mention in Section 2.1. we prove the following theorem characterizing the agent's unconstrained best response $\Delta(x)$ as well as the improving best response $\Delta_I(x)$:

Theorem 1. *An agent with feature x who was classified as -1 by a linear threshold function $f = sign(w_f^T x - b_f)$ has unconstrained best response $\Delta(x)$ of the form:*

$$\Delta(x) = \begin{cases} x, & \text{if } \frac{|w_f^T x - b_f|}{\sqrt{w_A^T S w_A}} \geq 2 \\ \left[x_A - \frac{w_f^T x - b_f}{w_A^T S w_A} S w_A\right] \circ x_U, & \text{otherwise} \end{cases} \tag{3}$$

with corresponding cost

$$c(x, \Delta(x)) = \begin{cases} \frac{|w_f^T x - b_f|}{\sqrt{w_A^T S w_A}}, & \text{if } \frac{|w_f^T x - b_f|}{\sqrt{w_A^T S w_A}} \geq 2 \\ 0 & \text{otherwise} \end{cases}$$

The same agent has improving best response

$$\Delta_I(x) = \begin{cases} x, & \text{if } \frac{|w_f^T x - b_f|}{\sqrt{w_I^T S_I w_I}} \geq 2 \\ \left[x_I - \frac{w_f \cdot x - b_f}{w_I^T S_I w_I} S_I w_I\right] \circ x_M \circ x_U, & \text{otherwise} \end{cases}$$

with corresponding cost

$$c(x, \Delta_I(x)) = \begin{cases} \frac{|w_f^T x - b_f|}{\sqrt{w_I^T S_I w_I}}, & \text{if } \frac{|w_f^T x - b_f|}{\sqrt{w_I^T S_I w_I}} \geq 2 \\ 0 & \text{otherwise} \end{cases}$$

We provide the proof of Theorem 1 in Appendix A.3.

A practical consideration to bring up here is that in many real-life scenarios, there are constraints on which directions some features can change towards. For example, if there is a feature called "has_phd", it can only be changed from 0 to 1. It turns out that adding such directionality constraints to the agent's best response model would make it impossible to draw a closed-form solution Instead, we incorporate such logic into the game model by adding the feature directionality constraint to the objective function of the *mechanism designer*. We discuss this approach in more details in Section 4.

4 Optimal Strategic Recourse Model

After obtaining the closed form solution of both the unconstrained and improving best response from the agents, we can further derive the objective function for the mechanism designer, and the model to deploy at equilibrium. Recall that the objective function for the mechanism designer is:

$$\max_{w_f, b_f} \quad \Pr_{x\sim\mathcal{D}}[f(\Delta(x)) = h(x)] + \lambda \Pr_{x\sim\mathcal{D}}[f(\Delta_I(x)) = +1]$$

After some mathematical transformations and remove the constants (see Appendix A.2), the objective function becomes:

$$\max_{w_f, b_f} \mathbb{E}_{x\sim\mathcal{D}}\left[\left(2 \cdot \mathbb{1}[w_f \cdot x - b_f \geq -2\sqrt{w_A^T S w_A}] - 1\right) \cdot h(x) + \lambda \cdot \mathbb{1}[w_f \cdot x - b_f \geq -2\sqrt{w_I^T S_I w_I}]\right] \tag{4}$$

Recall that here λ is a hyperparameter that captures the trade-off between prediction accuracy and incentivized improvement in the improvable features.In the empirical session, we will discuss in more details about how changes in different λ lead to difference performances. Our empirical results show that we are able to optimize Eq. (4) from samples.

4.1 Partially Actionable Features

In practice, individuals often can only modify some features towards a particular direction, and modeling the restriction on agents' side makes the problem analytically hard. Instead, we "prohibited" such moves in the mechanism designer's objective function. The idea is that if the mechanism designer is punished for encouraging an illegal action, the announced classifier will not incentivize such moves from agents. Therefore from agent's perspective, this introduces the moving constraints implicitly. In particular, we construct an array $\overline{\mathsf{dir}} \in \{-1, 0, +1\}^d$ to represent the prohibited moving direction of the feature vectors. Here, if feature x_i should not go bigger, then the i-th component of vector $\overline{\mathsf{dir}}$ should be +1; if feature x_j should not go smaller, then the j-th component of vector $\overline{\mathsf{dir}}$ should be -1; if there is no constraints on the moving direction of a feature x_k, then $\overline{\mathsf{dir}}_k = 0$. Then we add a penalty in terms of a *ramp* function after the objective Eq. (2):

$$-\eta \cdot \sum_{i=1}^{d} \max(\overline{\mathsf{dir}}_i \cdot (\Delta(x) - x)_i, 0) \tag{5}$$

where η is a positive constant to count the overall weight of this penalty term. Eq. (5) will penalize the weights of partially actionable features so that agents would prefer to move towards a certain direction. We provide more evaluation details in Section 5.2.

5 Empirical Evaluation

In this section, we present evidence for how our solution can help increase the improving recourse fraction and generate meaningful flipsets.

Dataset We conduct numeric simulation on Credit dataset [34], which contains 10,000 positive and 10,000 negative individuals. The goal of this dataset is to predict whether the individual will default on his upcoming credit payments. For each individual, there are 16 features. We show the split of improvable features (I), manipulated features (M), and unactionable features (U) in Table 2.

Cost Matrix For simplicity, we ignore the correlation among features. Let $\mathcal{I}$ denote the identity matrix. Considering the fact that making improvements usually cost more than manipulation, we set $S_I^{-1} = \mathcal{I}$ and $S_M^{-1} = \frac{1}{5}\mathcal{I}$ respectively.

Evaluation Metrics We evaluate the performance of our algorithm using two metrics: (1) *Prediction Accuracy* $\Pr[f(\Delta(x)) = h(x)]$, and (2) *Recourse fraction* $\Pr[f(\Delta_I(x)) = +1 \mid h(x) = -1]$, which represents the proportion of population who are denied by h but get approved by classifier f through improvement.

Classifiers We mainly focus on the performance of two classifiers:

- *Baseline classifier* only considers maximizing the accuracy, and optimize the objective function subject to $\lambda = 0$ in Eq. (2).
- *Recourse classifier* considers both maximizing the accuracy and incentivizing the proportion of recourse population, and optimize the objective function subject to $\lambda = 1$ in Eq. (2).

5.1 Results

Model Selection: we perform model selection and present the performance across different λ values in Fig. 1. We can observe a general trade-off trend between accuracy and recourse fraction after $\lambda > 1$. Our aim is not to show an *exact* relationship between the performance of linear classifiers and parameter λ, but to suggest mechanism designers or practitioners how to deploy a model that *carefully* incentivizes honest improvements.
Flipset: We also construct the *flipsets* for individuals in Credit dataset using the closed-form solution Eq. (3) under the trained strategic recourse classifier. Flipset is a set of actionable changes for an individual to flip the prediction of the classifier. As shown in Table 2, when we don't consider the moving direction of features ($\eta = 0$), the user who is predicted to have default next month

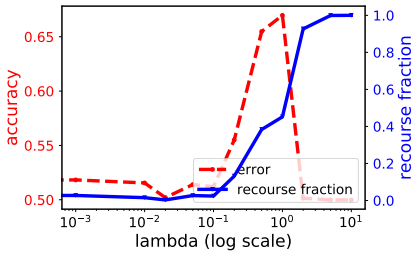

Figure 1: Accuracy vs recourse fraction across different λ values. Red dashed lines represent accuracy, while blue solid lines represent recourse fraction.

	Baseline		Recourse	
Penalty η	*Accuracy*	*Fraction*	*Accuracy*	*Fraction*
0	50.75%	1.87%	68.10%	48.25%
1	57.42%	11.10%	68.19%	32.83%
10	55.23%	7.23%	**66.77%**	**55.56%**
100	70.2%	39.77%	70.1%	39.91%
1000	70.2%	39.77%	70.21%	39.77%

Table 1: Empirical result on Credit dataset.

Feature	Type	$\overline{\text{dir}}$	Original	$\eta = 0$	$\eta = 100$
EducationLevel	I	+1	3	2 ↓	3
TotalOverdueCounts	I	0	1	1	1
TotalMonthsOverdue	I	0	1	1	0 ↓
MaxBillAmountOverLast6Months	M	0	0	0	0
MaxPaymentAmountOverLast6Months	M	0	0	0	0
MonthsWithZeroBalanceOverLast6Months	M	0	0	0	0
MonthsWithLowSpendingOverLast6Months	M	0	6	5 ↓	6
MonthsWithHighSpendingOverLast6Months	M	0	0	0	0
MostRecentBillAmount	M	0	0	0	0
MostRecentPaymentAmount	M	0	0	0	0
NoDefaultNextMonth	-	-	-1	$+1$ ↑	$+1$ ↑

Table 2: Flipsets for Credit dataset with partially actionable features.

(NoDefaultNextMonth $= -1$) is supposed to decrease his education level and decrease his MonthsWithLowSpendingOverLast6Months in order to flip his outcome of the recourse classifier with respect to the cost matrix.

5.2 Partially Actionable Features

We compare the performance of classifiers with partially actionable features on Credit dataset. Recall in this case the objective function we use for the mechanism designer is given by adding a penalty term $-\eta \cdot \sum_{i=1}^{d} \max(\overline{\text{dir}}_i \cdot (\Delta(x) - x)_i, 0)$ to Eq. (2). We show the specific $\overline{\text{dir}}$ array we used in this experiment in Table 2 where education level is prohibited from decreasing. As shown in Table 1, the direction penalty dominates the objective function when $\eta \geq 100$. In this case, increasing λ will not provide agents more recourse. We highlight the best performance the recourse classifier achieves when $\eta = 10$. We also note that for baseline classifier, large η improves both of its accuracy and recourse fraction. This fact provides the insight that direction penalty based on human experience might help the classifiers fit more to the real data.

We also build the flipsets of recourse classifier for an individual with $h(x) = -1$ when the penalty $\eta = 0$ and $\eta = 100$ respectively. As shown in Table 2, the individual will undesirably reduce his education level when the classifier is unaware of the partially actionable features. On the other hand, the individual would decrease his total overdue months instead when the direction penalty is imposed during training.

6 Concluding Remarks

In this work, we study the problem of strategic recourse. Given that agents are strategic, the goal of the mechanism designer is to simultaneously achieve high prediction accuracy and provide those agents a recourse that ultimately incentivizes them to improve their profile instead of superficially manipulation. We characterize the best response actions for both the agents and the mechanism designer, and provide useful insights for their behavior through theoretical analysis. Empirical evaluations are also provided to demonstrate that our strategic recourse classifier succeeds in achieving a better trade-off between preserving accuracy and providing as many agents an improving recourse as possible.

References

[1] Berk Ustun, Alexander Spangher, and Yang Liu. Actionable recourse in linear classification. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, FAT* '19, page 10–19, New York, NY, USA, 2019. Association for Computing Machinery.

[2] 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.

[3] J. Kleinberg and M. Raghavan. How do classifiers induce agents to invest effort strategically? *Proceedings of the 2019 ACM Conference on Economics and Computation*, 2019.

[4] Lily Hu, Nicole Immorlica, and Jennifer Wortman Vaughan. The disparate effects of strategic manipulation. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, FAT* '19, page 259–268, New York, NY, USA, 2019. Association for Computing Machinery.

[5] 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*, FAT* '19, page 230–239, New York, NY, USA, 2019. Association for Computing Machinery.

[6] Nika Haghtalab, Nicole Immorlica, Brendan Lucier, and Jack Z. Wang. Maximizing welfare with incentive-aware evaluation mechanisms. In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 160–166. International Joint Conferences on Artificial Intelligence Organization, 2020.

[7] Yonadav Shavit, B. L. Edelman, and B. Axelrod. Causal strategic linear regression. In *Proceedings of the International Conference on Machine Learning*, 2020.

[8] Moritz Hardt, Nimrod Megiddo, Christos Papadimitriou, and Mary Wootters. Strategic classification. In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science*, page 111–122, New York, NY, USA, 2016. Association for Computing Machinery.

[9] Prasanta Chandra Mahalanobis. On the generalized distance in statistics. *Proceedings of the National Institute of Sciences (Calcutta)*, 2:49–55, 1936.

[10] Yang Cai, Constantinos Daskalakis, and Christos Papadimitriou. Optimum statistical estimation with strategic data sources. volume 40 of *Proceedings of Machine Learning Research*, pages 280–296, Paris, France, 03–06 Jul 2015. PMLR.

[11] Omer Ben-Porat and Moshe Tennenholtz. Best response regression. 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 1499–1508. Curran Associates, Inc., 2017.

[12] 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*, EC '18, page 9–26, New York, NY, USA, 2018. Association for Computing Machinery.

[13] 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, page 55–70, New York, NY, USA, 2018. Association for Computing Machinery.

[14] Ofer Dekel, Felix Fischer, and Ariel D. Procaccia. Incentive compatible regression learning. *J. Comput. Syst. Sci.*, 76(8):759–777, December 2010.

[15] Yiling Chen, Yang Liu, and Chara Podimata. Learning strategy-aware linear classifiers, 2020.

[16] 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*, FAT* '20, page 436–445, New York, NY, USA, 2020. Association for Computing Machinery.

[17] 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.

[18] Vivek Gupta, Pegah Nokhiz, Chitradeep Dutta Roy, and Suresh Venkatasubramanian. Equalizing recourse across groups. *arXiv preprint arXiv:1909.03166*, 2019.

[19] 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.

[20] Julius von Kügelgen, Umang Bhatt, Amir-Hossein Karimi, Isabel Valera, Adrian Weller, and Bernhard Schölkopf. On the fairness of causal algorithmic recourse, 2020.

[21] Amir-Hossein Karimi, Julius von Kügelgen, Bernhard Schölkopf, and Isabel Valera. Algorithmic recourse under imperfect causal knowledge: a probabilistic approach, 2020.

[22] A.-H. Karimi, B. Schölkopf, and I. Valera. Algorithmic recourse: from counterfactual explanations to interventions. 2020.

[23] Amir-Hossein Karimi, Gilles Barthe, Bernhard Schölkopf, and Isabel Valera. A survey of algorithmic recourse: definitions, formulations, solutions, and prospects, 2020.

[24] Hao Wang, Berk Ustun, and Flavio Calmon. Repairing without retraining: Avoiding disparate impact with counterfactual distributions. volume 97 of *Proceedings of Machine Learning Research*, pages 6618–6627, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

[25] 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.

[26] Yahav Bechavod, Katrina Ligett, Zhiwei Steven Wu, and Juba Ziani. Causal feature discovery through strategic modification, 2020.

[27] John Miller, Smitha Milli, and Moritz Hardt. Strategic classification is causal modeling in disguise. In *International Conference on Machine Learning (ICML)*, Vienna, Austria, 2019. PMLR.

[28] 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*, KDD '15, page 259–268, New York, NY, USA, 2015. Association for Computing Machinery.

[29] A. Chouldechova. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big data*, 5 2:153–163, 2017.

[30] Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna M. Wallach. A reductions approach to fair classification. In Jennifer G. Dy and Andreas Krause, editors, *ICML*, volume 80 of *Proceedings of Machine Learning Research*, pages 60–69. PMLR, 2018.

[31] Moritz Hardt, Eric Price, and Nathan Srebro. Equality of opportunity in supervised learning. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 3323–3331, Red Hook, NY, USA, 2016. Curran Associates Inc.

[32] 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.

[33] 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*, ITCS '12, page 214–226, New York, NY, USA, 2012. Association for Computing Machinery.

[34] I. Yeh and Che hui Lien. The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert Syst. Appl.*, 36:2473–2480, 2009.

[35] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press.

A Omitted Proofs and Derivations

We present the missing proofs from the paper.

A.1 Proof of Lemma 1

Proof. Since the classifier in our game outputs a binary decision (-1 or $+1$), agents only have an incentive to change their features from x to x' when $c(x, x') \leq 2$. To see this, notice that an agent 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

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

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

A.2 Derivations of the objective function for the mechanism designer

In this section, we provide a detailed derivations of the objective funtion for the mechism designer in Section 4. Noting that the closed form of $f(\Delta(x))$ is given as follows:

$$f(\Delta(x)) = \begin{cases} +1 & \text{if } \; w_f \cdot x - b_f \geq -2\sqrt{w_A^T S w_A} \\ -1 & \text{otherwise} \end{cases}$$

which further derives as:

$$f(\Delta(x)) = 2 \cdot \mathbb{1}[w_f \cdot x - b \geq -2\sqrt{w_f^T S w_f}] - 1,$$

where $\mathbb{1}[\cdot]$ is the indicator function which equal to 1 if the specified condition is satisfied, and 0 otherwise. Similarly, the closed form for $f(\Delta_I(x))$ is given by:

$$f(\Delta_I(x)) = 2 \cdot \mathbb{1}[w \cdot x - b \geq -2\sqrt{w_I^T S_I w_I}] - 1$$

The objective function for the mechanism designer can then be re-written as follows:

$$\begin{aligned} &\Pr_{x\sim\mathcal{D}}[f(\Delta(x)) = h(x)] + \lambda \Pr_{x\sim\mathcal{D}}[f(\Delta_I(x)) = +1] \\ =&\mathbb{E}_{x\sim D}\left[\mathbb{1}[f(\Delta(x)) = h(x)] + \lambda \mathbb{1}[f(\Delta(x)) = +1]\right] \\ =&\mathbb{E}_{x\sim\mathcal{D}}\left[\frac{1}{2}(1+\lambda) + \frac{1}{2} f(\Delta(x)) \cdot h(x) + \frac{\lambda}{2} f(\Delta_I(x))\right] \end{aligned}$$

Removing the constants, the objective function becomes:

$$\max_{w_f, b_f} \mathbb{E}_{x\sim\mathcal{D}}\left[\lambda + f(\Delta(x)) \cdot h(x) + \lambda f(\Delta_I(x))\right]$$

$$\max_{w_f, b_f} \mathbb{E}_{x\sim\mathcal{D}}\left[\left(2 \cdot \mathbb{1}[w_f \cdot x - b_f \geq -2\sqrt{w_A^T S w_A}] - 1\right) \cdot h(x) + \lambda \cdot \mathbb{1}[w_f \cdot x - b_f \geq -2\sqrt{w_I^T S_I w_I}]\right]$$

A.3 Omit Proofs of Theorem 1

In this section, we provide the proofs of Theorem 1 in Section 3.

We first provide a lemma that allows us to re-formulate the optimization problem in Lemma 1:

Lemma 2. *Let $x^\star$ be an optimal solution to the following optimization problem:*

$$\begin{aligned} x^\star = \; &\arg\min_{x' \in \mathcal{X}_A^*(x)} \; c(x, x') \\ \text{s.t.} \quad &\text{sign}(w_f^T x' - b_f) = 1 \end{aligned}$$

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

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

Then $\Delta(x)$ is an optimal solution to the optimization problem in Lemma 1.

Proof. Recall the utility function of the agent is $U_f(x, x') = f(x') - c(x, x')$. And recall that the agent will only modify their features unless the utility increases, aka if $c(x, x') \leq 2$ where they achieve $f(x') = 1$ and the corresponding cost is bounded by 2 (the maximum possible cost of achieving a higher utility after modifying the features).

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

1. when x' that satisfies $c(x, x') > 2$: in this case, there are no feasible points for the optimization problem of Lemma 1.
2. when x' that satisfies $c(x, x') \leq 2$, we only need to consider those features x' that satisfies $f(x') = 1$, because if $f(x') = -1$, the agent with feature x would prefer not to change anything. Since maximizing $U_f(x, x') = f(x') - c(x, x')$ is equivalent of to minimizing $c(x, x')$ if $f(x') = 1$, and we know that when $c(x, x') \leq 2$, the solution of Lemma 1 is equivalent of the optimal solution of Lemma 2.

□

Lemma 2 provides us with an alternative way of looking at the agent's best response model: the goal of the agent is to minimize the cost of changing features such that she can cross the decision boundary of the classifier. This lemma enables us to re-formulate the objective function as follows.

Recall that $c(x, x') = \sqrt{(x_A - x'_A)^T S^{-1} (x_A - x'_A)}$ where S is a covariance matrix satisfying Eq. (1). Since S^{-1} is a symmetric positive definite matrix, it can be diagonalized into the following form, in which Q is an orthogonal matrix and Λ^{-1} is a diagonal matrix:

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

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

$$c(x, x') = \sqrt{(x_A - x'_A)^T S^{-1} (x_A - x'_A)} = \|\Lambda^{-\frac{1}{2}} Q (x_A - x'_A)\|_2$$

Meanwhile, the constraint in Lemma 2 rewrites as

$$\text{sign}(w_f \cdot x' - b_f) = \text{sign}(w_A \cdot x'_A + w_U \cdot x_U - b_f) = \text{sign}(w_A \cdot x'_A - (b_f - w_U \cdot x_U)) = 1$$

Hence the optimization problem can be reformulated into:

$$\min_{x'_A \in \mathcal{X}^*_A} \|(\Lambda^{-\frac{1}{2}} Q (x_A - x'_A))\|_2 \tag{6}$$

$$\text{s.t. } \text{sign}(w_A \cdot x'_A - (b_f - w_U \cdot x_U)) = 1 \tag{7}$$

The above optimization problem can be further simplified:

Lemma 3. *If $x_A^\mp$ is an optimal solution to Eq.* (6) *under constraint Eq.* (7)*, then it must satisfy $w_A \cdot x_A^\mp - (b_f - w_U \cdot x_U) = 0$.*

Proof. We prove by contradiction. Suppose that $x_A^\mp$ is an optimal solution to Eq. (6) and it satisfies $w_A x_A^\mp > b_f - w_U \cdot x_U$. Since the original feature x was classified as -1, we have:

$$w_A \cdot x_A^\mp > b_f - w_U \cdot x_U, \quad w_A \cdot x_A < b_f - w_U \cdot x_U$$

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

$$w_A \left(\mu \cdot x_A^\mp + (1 - \mu) x_A\right) = b_f - w_U \cdot x_U$$

Let $x''_A = \mu \cdot x_A^{\mp} + (1-\mu)x_A$, we know that $\text{sign}(w_A x''_A - (b_f - w_U \cdot x_U)) = 1$, i.e., x''_A also satisfies the constraint. Since $x_A^{\mp}$ is the optimal solution of Eq. (6), we have

$$\|\Sigma^{-\frac{1}{2}} Q(x_A^{\mp} - x_A)\| \leq \|\Sigma^{-\frac{1}{2}} Q(x''_A - x_A)\|$$

However, we also have:

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

which is contradicting to our assumption that $x_A^{\mp}$ is optimal. Therefore $x_A^{\mp}$ needs to satisfy $w_A x_A^{\mp} = b_f - w_U \cdot x_U$. □

As a result of Lemma 3, we can replace the constraint in Eq. (6) with its corresponding equality constraint, and it won't change the optimal solution. Therefore the agent's best response optimization problem in Lemma 1 is equivalent to:

$$\min_{x'_A} \|(\Lambda^{-\frac{1}{2}} Q(x_A - x'_A))\|_2 \tag{8}$$

$$\text{s.t. } w_A \cdot x'_A - (b_f - w_U \cdot x_U) = 0 \tag{9}$$

The above optimization problem satisfied a standard norm minimization with equality constraints [35] which is known to have a close-form solution. Similar arguments follow for $\Delta_I(x)$ and the rest of details can be found in the Appendix.

The following lemma gives us a closed-form solution for the above optimization problem:

Lemma 4. *The optimal solution to the optimization problem defined in Eq.* (8) *and Eq.* (9)*, has the following closed-form*

$$x_A^{\mp} = x_A - \frac{w_f^T x - b_f}{w_A^T S w_A} S w_A.$$

Proof. Notice that we can re-organize the above optimization problem defined in Eq. (8) and Eq. (9) as the following form:

$$\begin{aligned}
\min_{x'_A \in \mathcal{X}_A^*} \|Ax'_A - b\|_2 \\
\text{s.t. } Cx'_A = d
\end{aligned}$$

where $A = \Lambda^{-\frac{1}{2}} Q$, $b = \Lambda^{-\frac{1}{2}} Q x_A$, $C = w_A^T$, and $d = b_f - w_U^T x_U$. We note the following useful equalities:

$$\begin{aligned}
A^T A &= (\Lambda^{-\frac{1}{2}} Q)^T \Lambda^{-\frac{1}{2}} Q = S^{-1} \\
(A^T A)^{-1} &= S \\
A^T b &= (\Lambda^{-\frac{1}{2}} Q)^T \Lambda^{-\frac{1}{2}} Q x_A = S^{-1} x_A
\end{aligned}$$

The above is a norm minimization problem with equality constraints, whose optimum $x_A^{\mp}$ has the following closed form [35]:

$$\begin{aligned}
x_A^{\mp} &= (A^T A)^{-1} \left(A^T b - C^T (C(A^T A)^{-1} C^T)^{-1} (C(A^T A)^{-1} A^T b - d) \right) \\
&= S \left(S^{-1} x_A - w_A (w_A^T S w_A)^{-1} (w_A^T S (S^{-1} x_A) - b_f + w_U \cdot x_U) \right) \\
&= x_A - S \left(w_A (w_A^T S w_A)^{-1} (w_A^T x_A + w_U^T x_U - b_f) \right) \\
&= x_A - \frac{w_f^T x - b_f}{w_A^T S w_A} S w_A
\end{aligned}$$

□

We can now compute the cost incurred by an individual with features x who plays their best response $x^{\mp}$:

$$\begin{aligned} c(x, x\mp) &= \sqrt{(x_A - x_A^{\mp})^T S^{-1} (x_A - x_A^{\mp})} \\ &= \sqrt{\left(\frac{w_f^T x - b_f}{w_A^T S w_A} S w_A\right)^T S^{-1} \left(\frac{w_f^T x - b_f}{w_A^T S w_A} S w_A\right)} \\ &= \frac{|w_f^T x - b_f|}{\sqrt{w_A^T S w_A}} \end{aligned}$$

Hence an agent who was classified as -1 with feature vector x has the unconstrained best response

$$\Delta(x) = \begin{cases} x, & \text{if } \frac{|w_f^T x - b_f|}{\sqrt{w_A^T S w_A}} \geq 2 \\ \left[x_A - \frac{w_f^T x - b_f}{w_A^T S w_A} S w_A\right] \circ x_U, & \text{otherwise} \end{cases}$$

Similarly, finding $\Delta_I(x)$ for any x is equivalent to solving the following optimization problem:

$$\begin{aligned} \min_{x' \in \mathcal{X}_I^*(x)} \; & c(x, x') \\ \text{s.t. } & \text{sign}(w_f^T x' - b_f) = 1 \end{aligned}$$

If we let $x' = x'_I \circ x_M \circ x_U$, then this can be re-written as

$$\begin{aligned} \min_{x'_I \in \mathcal{X}_I^*} \; & \sqrt{(x_I - x'_I)^T S_I^{-1} (x_I - x'_I)} \\ \text{s.t. } & w_I \cdot x'_I = b_f - w_M \cdot x_M - w_U \cdot x_U \end{aligned}$$

By the same argument as before, we have the closed-form solution

$$x'_I = x_I - \frac{w_f \cdot x - b_f}{w_I^T S_I w_I} S_I^{-1} w_I$$

whose cost is

$$c(x, x') = \sqrt{(x_I - x'_I)^T S_I^{-1} (x_I - x'_I)} = \frac{|w_f^T x - b_f|}{\sqrt{w_I^T S_I w_I}}$$

This yields the improving best response

$$\Delta_I(x) = \begin{cases} x, & \text{if } \frac{|w_f^T x - b_f|}{\sqrt{w_I^T S_I w_I}} \geq 2 \\ \left[x_I - \frac{w_f \cdot x - b_f}{w_I^T S_I w_I} S_I w_I\right] \circ x_M \circ x_U, & \text{otherwise} \end{cases}$$

Then we finish the proof of Theorem 1.