arXiv:2011.00355v3 [cs.LG] 10 Jun 2021

Linear Classifiers that Encourage Constructive Adaptation

Yatong Chen
UC Santa Cruz
Santa Cruz, CA 95064
ychen592@ucsc.edu

Jialu Wang
UC Santa Cruz
Santa Cruz, CA 95064
faldict@ucsc.edu

Yang Liu
UC Santa Cruz
Santa Cruz, CA 95064
yangliu@ucsc.edu

Abstract

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 work, 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.

1 Introduction

Individuals subject to a classifier's predictions may act strategically to influence their predictions. Such behavior, often referred to as *strategic manipulation* [1], may lead to 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 [2, 3]. 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.

In this work, we study the following mechanism design problem: a *model designer* must 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.

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. 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. Concretely, we provide closed-form optimal

strategies for the decision subjects (Theorem 1). The implications (Section 3.3) reveal insights about the decision subjects' behaviors when the model designer uses non-causal features (features that don't 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.

1.1 Related work

Our paper builds on the strategic classification literature in machine learning [1, 4–10]. We study the interactions between a model designer and decision subjects using a a sequential two-player Stackelberg game [see e.g., 1, 11, 12, 7, 10, for similar formulations].

We consider a setting where 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., 13, 2, 3, 14]. In general, it's hard to distinguish causal features (features that affect the true outcome) from non-causal features: Miller et al. [15] show that designing an improvement-incentivizing model requires solving a non-trivial causal inference problem.

This paper also broadly relates to work on recourse [16–21] in that we aim to fit models that provide *constructive recourse*, i.e. actions that allow decision subjects to improve both their predicted *and* true outcomes. Our approach may be useful for mitigating the disparate effects of strategic adaptation [22–24] that stem from differences in the cost of manipulation (see Proposition 4). Lastly, our results may be helpful for developing robust classifiers in dynamic environments, where both decision subjects' features and the deployed models may vary across time periods [25, 3, 26].

Also relevant is the recent work on performative prediction [27–30], in which the choice of model itself affects the distribution over instances. However, this literature differs from ours in that we focus on inducing constructive adaptations from decision subjects, rather than finding a policy that incurs the minimum deployment error. In addition, our formulation arguably requires less knowledge, is more intuitive and deployable, and requires fewer assumptions on the loss function.

2 Problem statement

In this section, we describe our approach to training a classifier that encourages constructive recourse in settings with strategic adaptation.

2.1 Preliminaries

We consider a standard classification task of training a classifier $h : \mathbb{R}^d \to \{-1, +1\}$ from a dataset of n examples $(x_i, y_i)_{i=1}^n$, where example i consists of a vector of d features $x_i \in \mathbb{R}^d$ and a binary label $y_i \in \{-1, +1\}$. Example i corresponds to a person who wishes to receive a positive prediction $h(x_i) = +1$, and who will alter their features to obtain such a prediction once the model is deployed.

We formalize these dynamics as a sequential game between the following two players:

1. A model designer, who trains a classifier $h : \mathcal{X} \to \{-1, +1\}$ from a hypothesis class $\mathcal{H}$.

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* $c : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$.

We assume that each player has complete information: 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*.

We allow adaptations that alter the true outcome y. To describe these effects, we refer to the *true label function* $y : \mathcal{X} \to \{-1, +1\}$, such that $y_i = y(x_i)$. In practice, $y(\cdot)$ is unknown; however, our approach will involve assumptions about how altering a feature affects the true outcome.

2.2 Background

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

$$h^*_{\mathsf{ERM}} \in \arg\min_{h \in \mathcal{H}} R_{\mathsf{ERM}}(h)$$

where $R_{\mathsf{ERM}}(h) = \mathbb{E}_{x \sim \mathcal{D}}[\mathbb{1}(h(x) \neq y)]$. This classifier performs poorly in a setting with strategic adaptation, since the model is deployed on a population with a different distribution over $\mathcal{X}$ (as decision subjects alter their features) and y (as changes in features may alter true outcomes).

Existing approaches in strategic classification tackle these issues by training a classifier that is robust 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*:

$$h^*_{\mathsf{SC}} \in \arg\min_{h \in \mathcal{H}} R_{\mathsf{SC}}(h)$$

where $R_{\mathsf{SC}}(h) = \mathbb{E}_{x \sim \mathcal{D}}[\mathbb{1}(h(x_*) \neq y)]$, and $x_* = \Delta(x, h; c)$ denotes the features of a decision subject after adaptation. However, this classifier still has suboptimal accuracy because y changes as a result of the adaptation in x. Further, this design choice misses the opportunity to encourage a profile x to truly improve to change their y.

2.3 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.

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 (e.g. education level, which can be increased to improve the probability of repayment).
- *Manipulable* features (x_{M}), which can be altered without changing the true outcome (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.

  There may also be features that can be altered but whose effect is *unknown*. In this work, we treat them as 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 [15]. 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 adaptation 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 train a classifier that balances between robustness to manipulation and incentivizing improvement:

$$h^*_{\mathsf{CA}} = \arg\min_{h \in \mathcal{H}} [R_{\mathsf{M}}(h) + \lambda \cdot R_{\mathsf{I}}(h)], \tag{1}$$

The first term, $R_{\mathsf{M}}(h) = \mathbb{E}_{x\sim\mathcal{D}}[\mathbb{1}(h(x_*^{\mathsf{M}}) \neq y)]$, is the *manipulation risk*, which penalizes pure manipulation. The second term, $R_{\mathsf{I}}(h) = \mathbb{E}_{x\sim\mathcal{D}}[\mathbb{1}(h(x_*^{\mathsf{I}}) = +1)]$, 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 $+1$.

The two terms in the objective function can also be viewed as proxies for other familiar notions. In Section 4.1, 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.
- 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.

3 Decision subjects' best response

In this section, we characterize the decision subjects' best response function. Proofs for all results are included in Appendix B.

3.1 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:[1]

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

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, to prevent correlations between improvable and manipulable features, we assume S^{-1} is a diagonal block matrix of the form

$$S^{-1} = \begin{bmatrix} S_{\mathsf{I}}{}^{-1} & 0 \\ 0 & S_{\mathsf{M}}{}^{-1} \end{bmatrix}, \quad \text{which also implies} \quad S = \begin{bmatrix} S_{\mathsf{I}} & 0 \\ 0 & S_{\mathsf{M}} \end{bmatrix} \tag{2}$$

Otherwise, 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., 1, 2].

3.2 Decision subject's best response model

Given the assumptions of Section 3.1, 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 3.1 (F-Best Response Function). *Let* $\mathsf{F} \in \{\mathsf{I}, \mathsf{M}, \mathsf{A}\}$, *and let* $\mathcal{X}^*_{\mathsf{F}}(x)$ *denote the set of vectors that differ from* x *only in features of type* F. *Let* $\Delta_{\mathsf{F}} : \mathcal{X} \to \mathcal{X}$ *denote the* F-best response *of a decision subject with features* x *to* h*, defined as:*

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

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

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 3.3, we formalize this notion (Proposition 2).

We prove the following theorem characterizing the decision subject's different best responses:

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

$$\Delta_{\mathsf{F}}(x) = \begin{cases} \left[x_{\mathsf{F}} - \frac{w^{\mathsf{T}}x}{\Omega_{\mathsf{F}}} S_{\mathsf{F}} w_{\mathsf{F}}\right] \mid x_{\mathsf{A}\setminus\mathsf{F}} \mid x_{\mathsf{IM}}, & \textit{if } \frac{|w^{\mathsf{T}}x|}{\sqrt{\Omega_{\mathsf{F}}}} \leq 2 \\ x, & \textit{otherwise} \end{cases} \tag{3}$$

with corresponding cost

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

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

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

with corresponding cost

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

3.3 Discussion

In Proposition 1, 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 1 (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 $x \in \mathcal{X}$, $\Delta^{(m)}(x) \neq x^{(m)}$.*

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 2 (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.*

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 3 (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 22, 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 (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.*

Proposition 4 highlights the importance for a model designer to account for these differences when serving a population with heterogeneous subgroups.

4 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. Any omitted details can be found in Appendix D.

4.1 The model designer's program

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 1, we have

$$x^{\mathsf{M}}_* = \begin{cases} \left[x_{\mathsf{I}} \mid x_{\mathsf{M}} - \frac{w^\intercal x}{\Omega_{\mathsf{M}}} S_{\mathsf{M}} w_{\mathsf{M}} \mid x_{\mathsf{IM}}\right] & \text{if} \frac{|w^\intercal x|}{\sqrt{\Omega_{\mathsf{M}}}} \le 2 \\ x, & \text{otherwise} \end{cases} \tag{4}$$

$$x^{\mathsf{I}}_* = \begin{cases} \left[x_{\mathsf{I}} - \frac{w^\intercal x}{\Omega_{\mathsf{I}}} S_{\mathsf{I}} w_{\mathsf{I}} \mid x_{\mathsf{M}} \mid x_{\mathsf{IM}}\right], & \text{if } \frac{|w^\intercal x|}{\sqrt{\Omega_{\mathsf{I}}}} \le 2 \\ x, & \text{otherwise} \end{cases} \tag{5}$$

Recall from Section 2.3 that the model designer's optimization program is as follows:

$$\begin{aligned} \min_{h \in \mathcal{H}} \quad & \mathbb{E}_{x\sim\mathcal{D}}\left[\mathbb{1}(h(x^{\mathsf{M}}_*) \neq y)\right] + \lambda \mathbb{E}_{x\sim\mathcal{D}}\left[\mathbb{1}(h(x^{\mathsf{I}}_*) \neq +1)\right] \\ \text{s.t.} \quad & x^{\mathsf{M}}_* \text{ in Eq. (4)},\ x^{\mathsf{I}}_* \text{ in Eq. (5)} \end{aligned} \tag{6}$$

Interpreting the objective. The two terms in the objective function can be viewed as proxies for two other familiar objectives. The first term, $\mathbb{E}_{x\sim\mathcal{D}}\left[\mathbb{1}(h(x^{\mathsf{M}}_*) \neq y)\right]$, 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 5. *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*

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

The second term, $\mathbb{E}_{x\sim\mathcal{D}}\left[\mathbb{1}(h(x^{\mathsf{I}}_*) \neq +1)\right]$, 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 when $\Delta_{\mathsf{I}}(Y) = +1$; however, in the setting of *covariate shift*, in which 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 6. *Let $\mathcal{D}^*$ be the new distribution after decision subject's best response. Denote $\omega_h(x) = \frac{\Pr_{\mathcal{D}^*}(X=x)}{\Pr_{\mathcal{D}}(X=x)}$ 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 $\Pr(Y|X)$ is the same before and after adaptation (the covariate shift assumption). Then the following are equivalent:*

$$\Pr[h(x_*^{\prime}) = +1] > \Pr[h(x) = +1] \iff \Pr[y(x_*^{\prime}) = +1] > \Pr[y(x) = +1].$$

Proofs of Propositions 5 and 6 can be found in Appendix D.1 and D.2.

4.2 Making the program tractable

By substituting in the closed-form best responses for the decision subjects and making further mathematical steps (see Appendix D.3 for details), we can turn the model designer's *constrained* optimization problem in (6) into the following *unconstrained* problem:

$$\min_{w \in \mathbb{R}^{d+1}} \mathbb{E}_{x \sim \mathcal{D}} \left[-\left(2 \cdot \mathbb{1}\left[w^\intercal x \geq -2\sqrt{\Omega_\mathsf{M}}\right] - 1\right) \cdot y - 2\lambda \cdot \mathbb{1}\left[w^\intercal x \geq -2\sqrt{\Omega_\mathsf{I}}\right] \right] \quad (7)$$

The optimization problem in (7) 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:

$$\tilde{R}^\star_{\mathcal{D}}(h, \lambda) = \min_{w \in \mathbb{R}^{d+1}} \frac{1}{n} \sum_{i=1}^{n} \left[-\sigma\left(y_i \cdot (w^\intercal \cdot x_i + 2\sqrt{\Omega_\mathsf{M}})\right) - \lambda \cdot \sigma(w^\intercal \cdot x_i + 2\sqrt{\Omega_\mathsf{I}}) \right] \quad (8)$$

Directionally Actionable Features. An additional challenge arises when some features can be changed in either a positive or negative direction, but not both (e.g. `has_phd` can only go from *false* to *true*). In Appendix D.4, we show how to augment the above objective to enforce such constraints.

5 Experiments

In this section, we present empirical results to benchmark our method on synthetic and real-world datasets. We test the effectiveness of our approach in terms of its ability to incentivize improvement (or disincentivize manipulation) and compare its performance with other standard approaches. Our submission includes all datasets, scripts, and source code used to reproduce the results in this section.

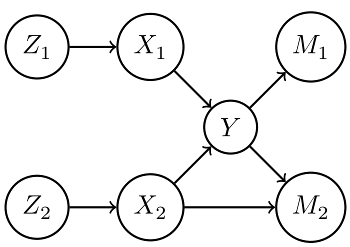


Figure 1: A causal DAG for the `toy` dataset. Z_1 and Z_2 are causal features that determine the true qualification Y, $X_1 = Z_1$, and X_2 is a noisy proxy for Z_1. We can directly observe X_1 and X_2 but not Z_1 or Z_2. M_1 and M_2 are non-causal features that correlate with Y but do not influence it.

Table 1: Performance metrics for different specifications (**Spec.**) in which features may be misspecified. ST denotes Static, DF denotes DropFeatures, MP denotes ManipulationProof, and CA denotes our method. 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	10.19
	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

5.1 Setup

Datasets. We consider five datasets: `toy`, a synthetic dataset based on the causal DAG in Fig. 1; `credit`, a dataset for predicting whether an individual will default on an upcoming credit payment [31]; `adult`, a census-based dataset for predicting adult annual incomes; `german`, a dataset to assess credit risk in loans; and `spambase`, a dataset for email spam detection. The last three are from the UCI ML Repository [32]. We provide a detailed description of each dataset along with a partitioning of features in Table 3 in the Appendix. We assume the cost of manipulation is lower than that of improvement, and that there are no correlations within the two types of adaptation; specifically, we use cost matrices ${S_{\mathsf{I}}}^{-1} = I$ and ${S_{\mathsf{M}}}^{-1} = 0.2I$. 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.

Methods. We fit linear classifiers for each dataset using the following methods:

- Static: a classifier trained using ℓ_2-logistic regression without accounting for strategic adaptation.
- DropFeatures: a classifier trained using ℓ_2-logistic regression without any manipulated features.
- ManipulationProof: a classifier that considers the agent's unconstrained best response during training, as typically done in the strategic classification literature [1].
- OurMethod: a linear logistic regression classifier that results from solving the optimization program in Eq. (8) using the BFGS algorithm [33]. This model represents our approach.

Evaluation Criteria. We run each method with 5-fold cross-validation and report the mean and standard deviation for each classifier on each of the following metrics:

- *Test Error*: the error of a classifier after training but *before* decision subjects' adaptations, i.e. $\mathbb{E}_{(x,y)\sim\mathcal{D}}\mathbb{1}[h(x) \neq y]$.
- *(Worst-Case) Deployment Error*: the test error of a classifier *after* decision subjects play their manipulating best response, i.e. $\mathbb{E}_{(x,y)\sim\mathcal{D}}\mathbb{1}[h(x_*^{\mathsf{M}}) \neq y]$.
- *(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. $\mathbb{E}_{(x,y)\sim\mathcal{D}}\mathbb{1}[h(x_*^{\mathsf{I}}) = +1 \mid y(x) = -1]$.

5.2 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 Fig. 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 1, Our method achieves the lowest deployment error (20.11%) and 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 improvable features [2, 3], 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 1, show that our classifier maintains a relatively high improvement rate in these cases, without sacrificing much deployment accuracy.

5.3 Results

We summarize the performance of each method in Table 2. Here are some key takeaways:

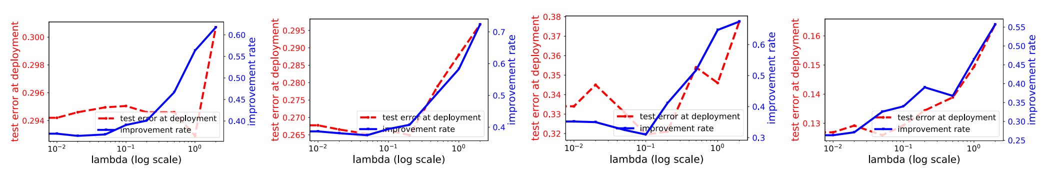
- Our method produces classifiers that achieve almost the highest deployment accuracy while providing the highest percentage of improvement across all four datasets.
- The static classifier, which does not account for adaptations, is vulnerable to strategic manipulation and consequently has the highest deployment error on every dataset.
- Naively cutting off the manipulated features may harm the accuracy at test time – DropFeatures incurs high test errors on `Adult` (33.55%) and `German` (36.10%).
- The strategic classifier ManipulationProof induces the lowest improvement rates on the `Credit` (25.26%) and `German` (29.10%) datasets.

Table 2: Performance metrics (mean $\pm$ standard deviation) for all methods on 4 data sets. ST indicates Static, DF indicates DropFeatures, MP indicates ManipulationProof, and CA indicates our method.

		METHODS			
Dataset	**Metrics**	ST	DF	MP	CA
CREDIT	*test error*	29.52 ± 0.37	29.66 ± 0.40	29.86 ± 0.52	29.60 ± 0.44
	deployment error	34.69 ± 3.23	29.66 ± 0.40	36.85 ± 1.59	29.41 ± 0.39
	improvement rate	43.70 ± 2.04	40.82 ± 2.81	34.62 ± 0.41	55.50 ± 4.03
ADULT	*test error*	23.05 ± 0.47	33.55 ± 0.73	24.94 ± 0.52	27.22 ± 0.65
	deployment error	49.15 ± 7.36	33.55 ± 0.73	28.62 ± 1.39	28.98 ± 0.68
	improvement rate	26.04 ± 2.93	61.68 ± 19.12	31.93 ± 4.13	52.07 ± 6.04
GERMAN	*test error*	30.85 ± 0.82	36.10 ± 1.97	33.25 ± 1.44	34.70 ± 2.15
	deployment error	39.30 ± 4.74	36.10 ± 1.97	37.10 ± 3.70	34.15 ± 2.64
	improvement rate	31.70 ± 5.94	34.00 ± 9.87	29.10 ± 2.85	53.00 ± 7.81
SPAMBASE	*test error*	7.11 ± 0.52	10.18 ± 0.45	11.52 ± 0.12	14.37 ± 0.24
	deployment error	38.88 ± 11.37	10.18 ± 0.45	16.07 ± 2.12	14.70 ± 0.46
	improvement rate	27.50 ± 11.24	16.88 ± 11.33	18.22 ± 6.04	39.84 ± 8.61

5.4 Effect of trade-off parameter λ

Fig. 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.



(a) `credit` (b) `adult` (c) `german` (d) `Spambase`

Figure 2: Trade-off between test error at deployment and improvement rate.

6 Conclusion remarks

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. We assume the published classifier is linear; indeed, this is ultimately what allows for a closed-form best response (Theorem 1) even with a relatively general cost function. 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 [15].

3. Our formulation of the classification setting as a two-step process gives decision subjects only one chance to adapt their features. We suspect that extending this formalism to more rounds may create more opportunities for constructive behavior in the long term, especially for agents who cannot improve their true qualification in one round.

References

[1] Moritz Hardt, Nimrod Megiddo, Christos Papadimitriou, and Mary Wootters. Strategic classification. In *Proceedings of the 2016 ACM conference on innovations in theoretical computer science*, pages 111–122, 2016.

[2] 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.

[3] Yonadav Shavit, Benjamin Edelman, and Brian Axelrod. Causal strategic linear regression. In *International Conference on Machine Learning*, pages 8676–8686. PMLR, 2020.

[4] Yang Cai, Constantinos Daskalakis, and Christos Papadimitriou. Optimum statistical estimation with strategic data sources. In *Conference on Learning Theory*, pages 280–296. PMLR, 2015.

[5] 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.

[6] 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.

[7] 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*, pages 55–70, 2018.

[8] Ofer Dekel, Felix Fischer, and Ariel D Procaccia. Incentive compatible regression learning. *Journal of Computer and System Sciences*, 76(8):759–777, 2010.

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

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] Nir Rosenfeld, Sophie Hilgard, Sai Srivatsa Ravindranath, and David C. Parkes. From predictions to decisions: Using lookahead regularization, 2020.

[15] 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.

[16] 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.

[17] 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.

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

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

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

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

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] Yang Liu and Yiling Chen. Machine-learning aided peer prediction. In *Proceedings of the 2017 ACM Conference on Economics and Computation*, pages 63–80, 2017.

[27] Juan Perdomo, Tijana Zrnic, Celestine Mendler-Dünner, and Moritz Hardt. Performative prediction. In *International Conference on Machine Learning*, pages 7599–7609. PMLR, 2020.

[28] John Miller, Juan Perdomo, and Tijana Zrnic. Outside the echo chamber: Optimizing the performative risk, 2021.

[29] Zachary Izzo, Lexing Ying, and James Zou. How to learn when data reacts to your model: Performative gradient descent. *CoRR*, 2021.

[30] Celestine Mendler-Dünner, Juan Perdomo, Tijana Zrnic, and Moritz Hardt. Stochastic optimization for performative prediction. *Advances in Neural Information Processing Systems*, 33, 2020.

[31] 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.

[32] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[33] 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.

[34] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[35] Gene H Golub and Charles F Van Loan. *Matrix computations*, volume 3. JHU press, 2013.

[36] 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.

[37] 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.

[38] 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.

[39] 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.

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

[41] 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.

[42] 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.

[43] 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.

[44] 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*, pages 3323–3331, 2016.

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

[46] 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.

[47] 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.