Actionable Recourse in Linear Classification

Berk Ustun
Harvard University
Cambridge, MA
berk@seas.harvard.edu

Alexander Spangher
Carnegie Mellon University
Pittsburgh, PA
alexander.spangher@cmu.edu

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

ABSTRACT

Classification models are often used to make decisions that affect humans: whether to approve a loan application, extend a job offer, or provide insurance. In such applications, individuals should have the ability to change the decision of the model. When a person is denied a loan by a credit scoring model, for example, they should be able to change the input variables of the model in a way that will guarantee approval. Otherwise, this person will be denied the loan so long as the model is deployed, and – more importantly – will lack agency over a decision that affects their livelihood.

In this paper, we propose to evaluate a linear classification model in terms of *recourse*, which we define as the ability of a person to change the decision of the model through *actionable* input variables (e.g., income vs. age or marital status). We present an integer programming toolkit to: (i) measure the feasibility and difficulty of recourse in a target population; and (ii) generate a list of actionable changes for a person to obtain a desired outcome. We discuss how our tools can inform different stakeholders by using them to audit recourse for credit scoring models built with real-world datasets. Our results illustrate how recourse can be significantly affected by common modeling practices, and motivate the need to evaluate recourse in algorithmic decision-making.

CCS CONCEPTS

• **Human-centered computing → Social recommendation**; • **Theory of computation → Integer programming**; • **Computing methodologies → Philosophical/theoretical foundations of artificial intelligence**; **Machine learning**;

KEYWORDS

recourse, classification, accountability, integer programming, audit, credit scoring

ACM Reference Format:
Berk Ustun, Alexander Spangher, and Yang Liu. 2019. Actionable Recourse in Linear Classification. In *FAT* '19: Conference on Fairness, Accountability, and Transparency (FAT* '19), January 29–31, 2019, Atlanta, GA, USA.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3287560.3287566

FAT '19, January 29–31, 2019, Atlanta, GA, USA*

ACM ISBN 978-1-4503-6125-5/19/01…$15.00
https://doi.org/10.1145/3287560.3287566

1 INTRODUCTION

In the context of machine learning, we define *recourse* as the ability of a person to obtain a desired outcome from afi xed prediction model. Consider, for example, a classification model used for loan approval. If the model provides recourse to someone who is denied a loan, then this person can change the input variables of the model in a way that guarantees approval. Otherwise, this person will be denied the loan so long as the model is deployed, and will lack agency in its decision-making process.

Recourse is not formally studied in machine learning. In this paper, we argue that it should be. In particular, a model should provide all individuals with recourse when it is used to allocate a good that should be universally accessible, such as credit [32], employment [2] and public services [9, 31]. Given that a lack of human agency is often perceived as a source of injustice in algorithmic decision-making [5, 11, 24], recourse should be evaluated whenever humans are subject to the predictions of a machine learning model.

The potential lack of recourse is often used to motivate calls for increased transparency and explainability in algorithmic decision-making [see e.g., 10, 14, 39]. However, transparency and explainability do not guarantee recourse. In fact, even a simple transparent model such as a linear classifier can fail to provide recourse due to seemingly innocuous modeling practices. These include:

- *Choice of Features*: A classifier could use features that are immutable (e.g., *age* ≥ 50), conditionally immutable (e.g., *has_phd*, which can only change from `FALSE` → `TRUE`), or should not be considered actionable (e.g., *married*).
- *Choice of Operating Point*: A probabilistic classifier that provides recourse at a given threshold (e.g., $\hat{y}_i = 1$ if predicted risk of default $\geq 50\%$) may fail to provide recourse at a more conservative threshold (e.g., $\hat{y}_i = 1$ if predicted risk of default $\geq 80\%$).
- *Out-of-Sample Deployment*: A feature that must be altered to obtain a desired outcome could be missing, immutable, or adversely distributed in the population on which the model is deployed (i.e., its *target population*).

In most real-world applications, an audit provides a practical mechanism to evaluate recourse. This is because an audit can examine the recourse of a model without affecting the way a model is developed and revealfi ndings that are tailored to its target population. Even when an audit suggests that a model will provide recourse within its target population, however, a model could require drastic changes that preclude certain individuals from obtaining a desired outcome. Ideally, an audit should therefore evaluate both the feasibility and *difficulty* of recourse within the target population.

In this paper, we present a practical toolkit to evaluate recourse for linear classification models (e.g., logistic regression models, linear SVMs, and rule-based models that can be expressed as linear models, such as rule sets and decision lists). Our tools allow a range

of stakeholders (e.g., practitioners, regulators, policy-makers, and decision-subjects) to answer questions, such as:

- *Does a model provide recourse to all individuals who are subject to its predictions?*
- *How does the difficulty of recourse vary across individuals in a target population?*
- *Are there disparities in recourse between subgroups in the target population?*
- *What changes can an individual make to obtain a desired prediction from the model?*

Our toolkit is based on a optimization problem that, given afi xed linear classifier, will identify changes that a person can make to flip their predicted outcome. Our problem is formulated tofind changes that are *actionable*, meaning that they will not affect immutable features, nor alter mutable features in an infeasible way (e.g., *n_credit_cards* from 5 $\longrightarrow$ 0.5 or 5 $\longrightarrow$ -1, or *has_phd* from TRUE $\longrightarrow$ FALSE). Since actionable changes for discrete features (e.g., binary, ordinal, or categorical features) can only be enforced through discrete constraints, the problem is computationally challenging. In order to allow an auditor to definitively state that a model does not provide a person with recourse, we solve our problem directly – by expressing it as an *integer program* (IP) and handing it to an IP solver (e.g., CPLEX, Gurobi, or CBC). We use this procedure to develop two tools to evaluate recourse:

1. A procedure to audit the recourse of a classifier in a target population (e.g., for model development, procurement, or impact assessments [26]). Given a sample of points (i.e., feature vectors) from the target population, we solve our problem for each point receives an undesirable prediction. Our procedure outputs an estimate of the feasibility and difficulty of recourse.

2. A method to generate a list of actionable changes for a person to obtain a desired outcome. We refer to this list as a *flipset* and provide an example in Figure 1. In the United States, for example, the Fair Credit Reporting Act [37] requires that individuals who are denied credit be sent an *adverse action notice* to explain the principal reason for the denial. It is well-known that an adverse action notice can fail to provide actionable information [see e.g., 29, 34, for a discussion]. By including afl ipset in an adverse action notice, a person would know exact changes that they can make to guarantee approval in the future.

Features toC hangeC	urrent ValuesR		equired Values
n_credit_cards	5	$\longrightarrow$	3
current_debt	\$3,250	$\longrightarrow$	\$1,000
has_savings_account	FALSE	$\longrightarrow$	TRUE
has_retirement_account	FALSE	$\longrightarrow$	TRUE

Figure 1: Illustrativefl ipset for a person who is denied credit by a classification model. Each *item* (row) shows an actionable change to a subset of features that will "flip" the prediction from $\hat{y} = -1$ to $\hat{y} = +1$. The changes guarantee that the person will be approved for credit so long as the model remains deployed and other features do not change. We describe how to buildfl ipsets in Section 3.4, and discuss their limitations in Section 5.2.

Related Work. Recourse is broadly related to several topics in machine learning. These include: *inverse classification*, which aims to determine how the inputs to a prediction model can be manipulated to obtain a desired outcome [1, 8]; *strategic classification*, which considers the converse problem of training classifiers that are robust to malicious manipulation [13, 17]; *adversarial perturbations*, which studies the robustness of predictions with respect to small changes in input [16]; and *anchors*, which are subsets of features that fully determine the prediction of a model [28].

Our tools are also broadly related to methods that explain the predictions of a machine learning model at an individual level [see e.g., 6, 21, 25, 27]. Although existing methods can produce valuable explanations of how a model outputs a specific prediction, these explanations do not necessarily correspond to actionable changes that guarantee a desired outcome. More importantly, they do not provide a principled mechanism to evaluate the feasibility and difficulty of recourse in a target population.

One notable exception are methods to generate *counterfactual explanations* [see e.g. 22, 39]. Our work is related to counterfactual explanations in that we can assess the feasibility of recourse through the existence of an actionable counterfactual explanation [1]. In a seminal paper, Wachter et al. [39] present a general method to recover counterfactual explanations from black-box models. This method cannot be used or adapted to evaluate recourse as: (i) it cannot constrain changes to be actionable; and (ii) it assumes that all feasible changes are reflected in the training data (i.e., a feasible change is defined as $\boldsymbol{a} \in \{\boldsymbol{x} - \boldsymbol{x}'\}$ where $\boldsymbol{x}, \boldsymbol{x}'$ are points in the training data)[2]. We overcome these issues by developing a fundamentally different optimization-based approach to evaluate recourse. Our approach uses integer programming, which allows users to precisely characterize the set of feasible actions, certify that a model does not provide recourse, and evaluate the difficulty of recourse using a rich class of cost functions.

Software and Workshop Paper. We provide a software implementation of our tools and scripts to reproduce our experimental results at http://github.com/ustunb/actionable-recourse. This paper extends a short workshop paper that was presented at FAT/ML 2018 [33].

2 PROBLEM STATEMENT

In this section, we define the optimization problem that we solve to evaluate recourse, and discuss formal guarantees related to the cost and feasibility of recourse. We include proofs for all technical results in Appendix A.

2.1 Optimization Framework

We consider a standard classification task where each individual is characterized by a vector of *features* $\boldsymbol{x} = [1, x_1 \dots x_d] \subseteq \mathcal{X}_0 \cup \dots \cup \mathcal{X}_d = \mathcal{X} \subseteq \mathbb{R}^{d+1}$ and a binary *label* $y \in \{-1, +1\}$.

[1] In other words, we can claim that a classifier provides recourse to a person if we can find an actionable counterfactual explanation for this person. In order to claim that a classifier does not provide recourse, however, we must prove that any actionable change will fail tofl ip this classifier's prediction for this person.

[2] To illustrate the practical consequences of (i) and (ii): the method in [39] could output an explanation stating that a person canfl ip their prediction by changing an immutable feature, due to (i). If so, we could not conclude that the model did not provide this person with recourse, as there may exist a different way tofl ip the prediction that was not reflected in the training data, due to (ii).

We wish to audit a linear classifier $f(\boldsymbol{x}) = \text{sign}(\langle \boldsymbol{w}, \boldsymbol{x} \rangle)$ where $\boldsymbol{w} = [w_0, w_1 \dots w_d] \subseteq \mathbb{R}^{d+1}$ is a vector of coefficients and w_0 is the intercept. We denote the desired outcome as $\hat{y} = 1$ and assume that $\text{sign}(0) = 1$ so that $\hat{y} = \mathbb{1}[\langle \boldsymbol{w}, \boldsymbol{x} \rangle \geq 0]$.

Given an individual whose predicted outcome is $f(\boldsymbol{x}) = -1$, we aim to determine if there exists an *action* $\boldsymbol{a}$ such that $f(\boldsymbol{x} + \boldsymbol{a}) = 1$. To this end, we solve an optimization problem of the form,

$$\begin{aligned} \min \quad & \text{cost}(\boldsymbol{a}; \boldsymbol{x}) \\ \text{s.t.} \quad & f(\boldsymbol{x} + \boldsymbol{a}) = 1 \\ & \boldsymbol{a} \in A(\boldsymbol{x}), \end{aligned} \tag{1}$$

where:

- $A(\boldsymbol{x})$ is a set of feasible actions from $\boldsymbol{x}$. Each *action* is a vector $\boldsymbol{a} = [0, a_1 \dots a_d]$ where $a_j \in A_j(x_j) \subseteq \{a_j \in \mathbb{R} \mid a_j + x_j \in \mathcal{X}_j\}$. We let $A_j(\boldsymbol{x}) = \{0\}$ if feature j is *immutable*, and say feature j is *conditionally immutable* if $A_j(\boldsymbol{x}) = \{0\}$ for some $\boldsymbol{x} \in \mathcal{X}$.
- $\text{cost}(\cdot\,; \boldsymbol{x}) : A(\boldsymbol{x}) \to \mathbb{R}_+$ is a cost function that encodes preferences between actions, or measures quantities of interest for an audit (see Section 3.2). Users can specify any cost function that satisfies two properties: (i) $\text{cost}(\boldsymbol{0}; \boldsymbol{x}) = 0$ (no action $\Leftrightarrow$ no cost); (ii) $\text{cost}(\boldsymbol{a}; \boldsymbol{x}) \leq \text{cost}(\boldsymbol{a} + \epsilon \boldsymbol{1}_j; \boldsymbol{x})$ (larger actions $\Leftrightarrow$ higher cost).

Solving the optimization problem in (1) for an individual with features $\boldsymbol{x}$ has different implications with respect to recourse:

- If (1) is *infeasible*, then no action can achieve a desired outcome from $\boldsymbol{x}$. Thus, we have certified that the model does not provide actionable recourse for an individual with features $\boldsymbol{x}$.
- If (1) is *feasible*, then its optimal solution is the minimal-cost action tofl ip the prediction of $\boldsymbol{x}$. In this case, we can use the solution to create an item in afl ipset (see Section 3.4).

Assumptions, Notation and Terminology. We denote the index sets for all features as $J = \{1, \dots, d\}$, for actionable features as $J_A(\boldsymbol{x}) = \{j \in J \mid |A_j(\boldsymbol{x})| > 1\}$ and immutable features as $J_N(\boldsymbol{x}) = \{j \in J \mid A_j(\boldsymbol{x}) = 0\}$. We drop the dependence of index sets on $\boldsymbol{x}$ when it is clear from the context.

Given a linear classifier $f(\boldsymbol{x}) = \text{sign}(\langle \boldsymbol{w}, \boldsymbol{x} \rangle)$, we express its coefficient vector as $\boldsymbol{w} = [\boldsymbol{w}_A, \boldsymbol{w}_N]$, where $\boldsymbol{w}_A$ and $\boldsymbol{w}_N$ contain the coefficients for features that are actionable and immutable, respectively. We assume that the classifier is deployed on a target population where features are bounded (i.e., for all $\boldsymbol{x} \in \mathcal{X}$, $\|\boldsymbol{x}\| \leq B$ for a sufficiently large B), and define the follow subspaces of the feature space based on the values of $f(\boldsymbol{x})$ and y:

$$H^- = \{\boldsymbol{x} \in \mathcal{X} : f(\boldsymbol{x}) = -1\} \qquad H^+ = \{\boldsymbol{x} \in \mathcal{X} : f(\boldsymbol{x}) = +1\},$$
$$D^- = \{\boldsymbol{x} \in \mathcal{X} : y = -1\} \qquad D^+ = \{\boldsymbol{x} \in \mathcal{X} : y = +1\}.$$

2.2 Feasibility Guarantees

We start with a simple sufficient condition for a linear classifier to provide recourse to all individuals in any target population.

Remark1. *A linear classifier provides recourse to all individuals if it only uses actionable features and does not trivially predict a single class.*

Remark 1 may be used to guide regulations in applications where a classifier must provide recourse, or to design screening questions for algorithmic impact assessments (e.g., "can a person change all of the inputs to the classifier?"). We observe that the converse of Remark 1 is also true: a classifier fails to provide recourse if all features are immutable, or if it trivially predicts a single class. In what follows, we therefore restrict our attention to linear classifiers with non-zero coefficients $\boldsymbol{w} \neq \boldsymbol{0}$ that do not trivially predict a single class in the target population.

Our next set of remarks show that the feasibility of recourse depends on the boundedness of the feature space.

Remark2. *If all features are unbounded, then a linear classifier with one or more actionable features will provide recourse to all individuals in any target population.*

Remark3. *If all features are bounded, then a linear classifier with one or more immutable features may not provide recourse to some individuals in a target population.*

Remarks 2 and 3 imply that when a classifier contains a combination of actionable and immutable features, then any claim regarding the feasibility of recourse depends on how we specify bounds for actionable features, and in particular actionable features with real values.[3] Given that a lack of recourse could have important ramifications, real-valued features should be bounded judiciously. In practice, we would advise setting loose bounds on real-valued features, so that the auditor does not report infeasibility due to overly restrictive bounds. This approach has a potential drawback in that loose bounds allow the classifier to provide recourse through potentially drastic changes. Although this may appear to reduce the significance of assessing feasibility, such changes will be reflected in an audit through large values of a suitable cost function (e.g., the maximum percentile shift function in (3)).

Recourse is not guaranteed when a classifier uses features that are immutable or conditionally immutable (e.g., *age* or *has_phd*). As shown in Example 2.1, a classifier with an immutable feature could achieve perfect predictive accuracy without providing a universal recourse guarantee.

Example 2.1. Consider training a linear classifier using a dataset of n examples $(\boldsymbol{x}_i, y_i)_{i=1}^n$ where $\boldsymbol{x}_i \in \{0, 1\}^d$, and the labels $y_i \in \{-1, +1\}$ are sampled from the distribution

$$\Pr(y = +1 | \boldsymbol{x}) = \frac{1}{1 + \exp(\alpha - \sum_{j=1}^d x_j)},$$

so that the Bayes optimal classifier has the form $\hat{y} = f(\boldsymbol{x}) = \mathbb{1}\left[\sum_{j=1}^d x_j > \alpha\right]$. If $\alpha \geq d - 1$, then f will not provide recourse to any individual where $x_j = 0$ for an immutable feature $j \in J_N$.

In practice, such features may be desirable to include in a model because they can improve its predictive performance or its robustness to strategic manipulation.

[3] In practice, an auditor would determine a set of feasible actions by specifying upper and lower bounds for each actionable feature (which must exist given that the target population isfi nite). Since binary, ordinal, and categorical features are bounded by definition, an auditor would only have to specify bounds for real-valued features, such as *age* or *income*. In our experiments in Section 4, we set bounds for these features as the maximum observed value in the dataset.

2.3 Cost Guarantees

In Theorem 2.3, we present a bound on the expected cost of recourse in a target population.

Definition 2.2. The *expected cost of recourse* of a classifier $f : \mathcal{X} \to \{-1, +1\}$, is defined as:

$$\overline{\text{cost}}_{H^-}(f) = \mathbb{E}_{H^-}[\text{cost}(\boldsymbol{a}^*; \boldsymbol{x})],$$

where $\boldsymbol{a}^*$ is an optimal solution to the optimization problem in (1).

Our guarantee is expressed in terms of a general cost function of the form $\text{cost}(\boldsymbol{a}; \boldsymbol{x}) = c(\boldsymbol{x}) \cdot \|\boldsymbol{a}\|$, where $c : \mathcal{X} \to (0, +\infty)$ is a positive scaling function for actions starting from $\boldsymbol{x} \in \mathcal{X}$, and $\mathcal{X}$ is a closed convex set.

THEOREM 2.3. *The expected cost of recourse of a linear classifier over a target population obeys:*

$$\overline{\text{cost}}_{H^-}(f) \leq p^+ \gamma_A^+ + p^- \gamma_A^- + 2\gamma_A^{\max} R_A(f),$$

where:

- $p^+ = \Pr_{H^-}(y = +1)$ *is the false omission rate of* f*;*
- $p^- = \Pr_{H^-}(y = -1)$ *is the negative predictive value of* f*;*
- $\gamma_A^+ = \mathbb{E}_{H^- \cap D^+}[c(\boldsymbol{x}) \cdot \frac{\boldsymbol{w}_A^\top \boldsymbol{x}_A}{||\boldsymbol{w}_A||_2^2}]$ *is the expected unit cost of actionable changes for false negatives;*
- $\gamma_A^- = \mathbb{E}_{H^- \cap D^-}[c(\boldsymbol{x}) \cdot \frac{-\boldsymbol{w}_A^\top \boldsymbol{x}_A}{||\boldsymbol{w}_A||_2^2}]$ *is the expected unit cost of actionable changes for true negatives;*
- $\gamma_A^{\max} = \max_{\boldsymbol{x} \in H^-} \left| c(\boldsymbol{x}) \cdot \frac{\boldsymbol{w}_A^\top \boldsymbol{x}_A}{||\boldsymbol{w}_A||_2^2} \right|$ *is the maximum unit cost of actionable changes for negative predictions;*
- $R_A(f) = p^+ \cdot \Pr_{H^- \cap D^+}\left(\boldsymbol{w}_A^\top \boldsymbol{x}_A \leq 0\right) + p^- \cdot \Pr_{H^- \cap D^-}\left(\boldsymbol{w}_A^\top \boldsymbol{x}_A \geq 0\right)$ *is the internal risk of actionable features.*

Theorem 2.3 implies that we can reduce the expected cost of recourse by reducing the *maximum unit cost of actionable changes* $\gamma_A^{\max}$, or the *internal risk of actionable features* $R_A(f)$. Here, $R_A(f)$ captures the calibration between true outcomes and the actionable component of the scores $\boldsymbol{w}_A^\top \boldsymbol{x}_A$ for points such that $f(\boldsymbol{x}) = -1$. If $R_A(f) = 0$, then the actionable component of the scores is perfectly aligned with true outcomes, which produces a tighter bound on the expected cost of recourse.

3 INTEGER PROGRAMMING TOOLKIT

In this section, wefi rst describe how we can solve the optimization problem in (1) using integer programming, and then discuss how we use this procedure to audit recourse, and to buildflipsets.

3.1 IP Formulation

We consider a discretized version of the optimization problem in (1), which can be expressed as an *integer program* (IP) and solved with an IP solver [see 23, for a list]. This approach has several benefits: (i) it can directly constrain actions for binary, ordinal, and categorical features; (ii) it can optimize non-linear and non-convex cost functions; (iii) it allows users to customize the set of feasible actions; and (iv) it can quickly recover a globally optimal solution or certify that actionable recourse does not exist.

We express the optimization problem in (1) as an IP of the form:

$$\begin{aligned}
\min \quad & \text{cost} \\
\text{s.t.} \quad & \text{cost} = \sum_{j \in J_A} \sum_{k=1}^{m_j} c_{jk} v_{jk} && \text{(2a)} \\
& \sum_{j \in J_A} w_j a_j \geq -\sum_{j=0}^{d} w_j x_j && \text{(2b)} \\
& a_j = \sum_{k=1}^{m_j} a_{jk} v_{jk} && j \in J_A \quad \text{(2c)} \\
& 1 = u_j + \sum_{k=1}^{m_j} v_{jk} && j \in J_A \quad \text{(2d)} \\
& a_j \in \mathbb{R} && j \in J_A \\
& u_j \in \{0, 1\} && j \in J_A \\
& v_{jk} \in \{0, 1\} && k = 1 \ldots m_j, \; j \in J_A
\end{aligned}$$

Here, constraint (2a) determines the cost of a feasible action from precomputed cost parameters $c_{jk} = \text{cost}(x_j + a_{jk}; x_j)$. Constraint (2b) requires any feasible action tofl ip the prediction of a linear classifier with coefficients $\boldsymbol{w}$. Constraints (2c) and (2d) restrict a_j to a grid of $m_j + 1$ feasible values $a_j \in \{0, a_{j1}, \ldots, a_{jm_j}\}$ via the indicator variables $u_j = 1[a_j = 0]$ and $v_{jk} = 1[a_j = a_{jk}]$. Note that the variables and constraints only depend on actions for actionable features $j \in J_A$ since $a_j = 0$ when a feature is immutable.

Modern IP solvers can quickly solve instances of (2) to optimality (i.e., in $\leq$ 0.1s with CPLEX 12.8). In practice, we can further reduce solution time (e.g., for auditing procedures where we solve (2) multiple times) by: (i) dropping the v_{jk} indicators for actions a_{jk} that do not agree in sign with w_j; and (ii) declaring $\{v_{j1}, \ldots, v_{jm_j}\}$ as a *special ordered set of type I*, which allows the solver to use a more efficient branch-and-bound algorithm [35].

Customization. Users can easily customize the set of feasible actions by adding logical constraints to (2). Many constraints can be expressed using the u_j indicators, without having to introduce new variables. To limit actions to change $\leq r$ features, we can add the constraint $\sum_{j=1}^{d}(1 - u_j) \leq r$. To ensure actions change only one feature in a subset of features $S \subseteq J$, we can add the constraint $\sum_{j \in S}^{d}(1 - u_j) \leq 1$. Such constraints are required, for example, when a linear classifier contains a subset of dummy variables to encode a categorical attribute (i.e., a one-hot encoding).

Discretization. The IP formulation in (2) requires users to discretize the actions for real-valued features over a suitable grid.

In Appendix B, we discuss how to discretize the actions for real-valued features so that discretization does not affect the feasibility or cost of recourse. In particular, we show that: (i) discretization does not affect the feasibility of recourse if we restrict the actions for real-valued features to a grid with matching upper and lower bounds; and (ii) the maximum discretization error in the cost of recourse can be bounded by refining the grid.

One can avoid discretization entirely by formulating an IP that uses continuous variables to represent the actions for real-valued features (see Appendix B.3). In light of our guarantees in Appendix B, we do not consider this approach because it unnecessarily restricts users to use linear cost functions.

3.2 Cost Functions

Cost functions should be used to encode preferences between feasible actions or to measure quantities of interest in the target population. However, they should not be used to penalize infeasible actions as infeasibility can be directly modeled by adding hard constraints to the IP formulation. The IP in (2) can optimize a large class of cost functions because it precomputes these values and encodes them in the c_{jk} parameters in constraint (2a). Although IP (2) requires costs to be specified by the values of actions in each dimension, cost functions do not need to be strictly separable since the IP can handle some kinds of non-separability by introducing additional constraints (see e.g., the cost function for auditing in (3)).

We present two off-the-shelf cost functions for auditing and buildingfl ipsets in (3) and (4). These functions can be adapted by practitioners who wish to design application-specific cost functions. Our functions measure costs in terms of the *percentiles* of x_j and $x_j + a_j$: $Q_j(x_j + a_j)$ and $Q_j(x_j)$ where $Q_j(\cdot)$ is the CDF of x_j in the target population. Unlike standard Euclidean distance metrics, cost functions based on percentiles do not depend on the scale of features, and account for the distribution of features in the target population. Our functions assign the same cost for a unit percentile change for each feature, which implicitly assumes that percentile changes along different features are equally difficult. This assumption can be relaxed by having a domain expert specify the relative difficulty of changing features relative to a baseline feature.

3.3 Auditing Recourse

We can use IP (2) to audit the recourse of a linear classifier on a target population. The auditing procedure requires: (i) the coefficient vector $\boldsymbol{w}$ of the linear classifier; and (ii) a sample of feature vectors from the target population $\{\boldsymbol{x}_i\}_{i=1}^n$ where $f(\boldsymbol{x}_i) = -1$. It solves the IP for each point in the sample to output:

- an estimate of the feasibility of recourse (i.e., the % of points for which the IP is feasible);
- an estimate of distribution of the cost of recourse (i.e., values of $\text{cost}(\boldsymbol{a}_i{}^*; \boldsymbol{x}_i)$ where $\boldsymbol{a}_i{}^*$ is the minimal-cost action).

As our cost function, we propose the *maximum percentile shift*:

$$\text{cost}(\boldsymbol{x} + \boldsymbol{a}; \boldsymbol{x}) = \max_{j \in J_A} \left| Q_j(x_j + a_j) - Q_j(x_j) \right|. \tag{3}$$

The benefit of auditing with this cost function lies in the meaning of the optimal cost. If the optimal cost is 0.25, for example, then any feasible action must change a feature by at least 25 percentiles. That is, no action canfl ip the prediction without changing a feature by less than 25 percentiles. Using (3) requires replacing constraint (2a) in IP with $|J_A|$ constraints of the form $\text{cost} \geq \sum_{k=1}^{m_j} c_{jk} v_{jk}$ for $j \in J_A$.

Minimizing the cost function in (3) is also useful when we wish to evaluate how user-defined bounds on changes affect the feasibility of recourse. Say that we wanted to measure how many individuals have recourse when actions are bounded to changes of at most 50 percentiles or changes of at most 90 percentiles. Instead of running two separate audits with action sets that restrict feasible changes to a 50 percentile shift and a 90 percentile shift, we can run a single audit that minimizes the maximum cost of an loosely bounded action set, and compare the number of individuals where the optimal cost exceeds 0.5 and 0.9.

3.4 Building Flipsets

We build a flipset such as the one in Figure 1 by using an *enumeration procedure* that repeatedly solves the IP in (2). In order to reliably provide an individual with recourse,fl ipsets should ideally include multiple actions that willfl ip the predicted outcome. This is because each action can be infeasible in a way that is only known by the individual [see e.g., 29, for an example].

In Algorithm 1, we present a procedure to enumerate $T \geq 1$ minimal-cost actions with distinct combinations of features. Each iteration solves the IP to obtain the optimal action $\boldsymbol{a}^*$, then adds a constraint to the IP to eliminate actions that use the same subset of features as $\boldsymbol{a}^*$. The procedure repeats these steps until it has recovered T actions, or the IP becomes infeasible (which means that it has enumerated a minimal-cost action for all subsets of features that canfl ip the prediction for $\boldsymbol{x}$). Each action $\boldsymbol{a}^* \in \mathcal{A}$ produced by Algorithm 1 can be used to create an *item* in aflipset by listing the current feature values x_j alongside the desired feature values $x_j + a_j^*$ for $j \in S = \{j : a_j^* \neq 0\}$.

Algorithm 1 Enumerate T Minimal Cost Actions for Flipset

Input

IP — instance of (2) for coefficients $\boldsymbol{w}$, features $\boldsymbol{x}$, and actions $A(\boldsymbol{x})$

$T \geq 1$ — number of items inflipset

Initialize

$\mathcal{A} \leftarrow \emptyset$ — actions shown inflipset

1: **repeat**

2: $\boldsymbol{a}^* \leftarrow$ optimal solution to IP

3: $\mathcal{A} \leftarrow \mathcal{A} \cup \{\boldsymbol{a}^*\}$ — *add $\boldsymbol{a}^*$ to set of optimal actions*

4: $S \leftarrow \{j : a_j^* \neq 0\}$ — *indices of features altered by $\boldsymbol{a}^*$*

5: add constraint to IP to remove actions that alter features $j \in S$:

$$\sum_{j \in S} (1 - u_j) + \sum_{j \notin S} u_j \leq d - 1.$$

6: **until** $|\mathcal{A}| = T$ or IP is infeasible

Output: $\mathcal{A}$ — actions shown inflipset

Algorithm 1 can be adapted to produce different kinds offlipsets by changing the constraint in Step 5 to enumerate other kinds of successive optima. For example, one can create afl ipset containing mutually exclusive actions by adding the constraint $u_j = 0$ for $j \in S$ to remove all features used in $\boldsymbol{a}^*$ at each iteration.

As our cost function, we propose the *total log-percentile shift*:

$$\text{cost}(\boldsymbol{x} + \boldsymbol{a}; \boldsymbol{x}) = \sum_{j \in J_A} \log \left(\frac{1 - Q_j(x_j + a_j)}{1 - Q_j(x_j)} \right). \tag{4}$$

This function aims to producefl ipsets where items reflect "easy" changes with respect to the target population. In particular, it ensures that cost of a_j increases exponentially as $Q_j(x_j) \to 1$. This aims to capture the notion that changes become harder when starting off from a higher percentile value (e.g., changing *income* from percentiles $90 \to 95$ is harder than $50 \to 55$).

4 DEMONSTRATIONS

In this section, we illustrate how our tools can be used to audit recourse in three credit scoring problems. We have two goals: (i) to show how an audit can provide useful information for different stakeholders (e.g., individuals, practitioners, and policy-makers); (ii) to show how the feasibility and difficulty of recourse can be affected by common modeling practices.

We provide a software implementation of our tools and scripts to reproduce our analyses at http://github.com/ustunb/actionable-recourse. We trained all classifiers as implemented in scikit-learn, and used a standard 10-fold cross-validation (10-CV) setup to tune free parameters and to estimate their predictive performance. We solved all IPs for auditing andfl ipset generation with the CPLEX 12.8 solver [18] on a laptop with 2.6 GHz CPU with 16 GB RAM.

4.1 Model Selection

We start with a simple experiment to show how our tools can be used to inform different stakeholders in credit scoring applications.

Setup. We consider a processed version of `credit` dataset [41]. Here, $y_i = -1$ if person i will default on their upcoming credit card payment. The dataset contains $n = 30\,000$ individuals and $d = 16$ features related to spending and payment patterns, education, credit history, age, and marital status. We assume that spending and payment patterns and education are actionable, and assume that all other variables are immutable.

We train ℓ_1-penalized logistic regression (LR) models for ℓ_1-penalties in the set $\{1, 2, 5, 10, 20, 50, 100, 500, 1000\}$. We audit the recourse of each model on the training dataset by solving (2) for each individual i such that $\hat{y}_i = -1$. Our IP includes the following constraints to ensure changes are actionable: (i) changes for discrete features must be discrete (e.g. *MonthsWithLowSpending-InPast6Months* $\in \{0, 1 \ldots 6\}$); (ii) *EducationLevel* can only increase; (iii) no changes to immutable features.

Results. We summarize the results of our audit in Figure 2, and present afl ipset for a person who is denied credit in Figure 3.

As shown in Figure 2, tuning the ℓ_1-penalty has a minor effect on test error, but a major effect on recourse. Specifically, classifiers with small ℓ_1-penalties provide all individuals with recourse. As the ℓ_1-penalty increases, however, the % of individuals with recourse decreases as the coefficients for actionable features are more heavily penalized compared to the coefficients for immutable features.

The cost of recourse provides a communicable measure of the difficulty of attaining a desired outcome (among individuals who have recourse). Since we audit using the maximum percentile shift cost function in (3), a cost of q implies a person must change a feature by at least q percentiles to attain a desired outcome. Here, we see that increasing the ℓ_1-penalty nearly doubles the median cost of recourse from 0.20 to 0.39. Thus, at a small ℓ_1-penalty, the median person with recourse can only attain a desired outcome by changing a feature by at least 20 percentiles. At a large ℓ_1-penalty, however, the median person with recourse can only attain a desired outcome by changing a feature by at least 39 percentiles.

Discussion. Our aim is not to suggest a relationship between recourse and ℓ_1-regularization, but to show how common practices such as parameter tuning can impact the feasibility and difficulty of recourse. Here, a practitioner who is primarily interested in performance could deploy a classifier that precludes individuals from achieving a desired outcome (e.g., the one that minimizes mean 10-CV test error), even as there exists a classifier that attains similar performance but provides all individuals with recourse (e.g., a classifier with a slightly lower ℓ_1-penalty). Our tools provide the necessary information for a practitioner to choose between such classifiers and incorporate the feasibility and cost of actionable recourse in their model development pipeline.

Our tools can also identify mechanisms that affect recourse in a target population by comparing the cost and feasibility of recourse for different action sets $A(\boldsymbol{x})$. For example, one can evaluate how the mutability of feature j affects recourse by running audits using: (i) an action set where feature j is immutable ($A_j(\boldsymbol{x}) = 0$ for all $\boldsymbol{x} \in \mathcal{X}$); and (ii) an action set where feature j is actionable ($A_j(\boldsymbol{x}) = \mathcal{X}_j$ for all $\boldsymbol{x} \in \mathcal{X}$). Here, such an analysis reveals that the lack of recourse is tied to an immutable feature related to credit history (i.e., a binary feature set to 1 if a person has *ever* defaulted on a loan). Given this information, a practitioner could replace this feature with a mutable variant (i.e., a binary feature set to 1 if a person has *recently* defaulted on a loan), and thereby deploy a model that provides recourse. Such changes are sometimes mandated by application-specific regulations [see e.g., policies on "forgetfulness" in 7, 15]. Our tools can support such regulations by providing policy-makers with an estimate of their impact on the feasibility and cost of recourse in a population of interest.

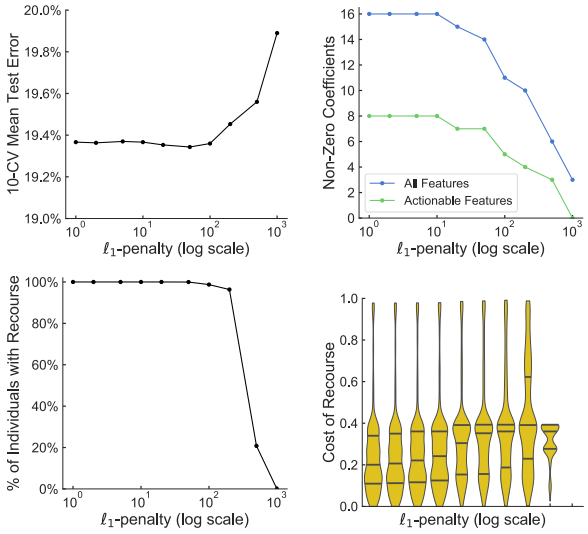

Figure 2: Overview of model performance and recourse over the training sample for ℓ_1-penalized logistic regression models. We show the mean 10-CV test error (top left), # of non-zero coefficients (top right), % of individuals with recourse (bottom left), and the distribution of the cost of recourse (bottom right) for all classifiers.

FeatureS ubsetC	urrent ValuesR		equired Values
MostRecentPaymentAmount	\$0	⟶	\$790
MostRecentPaymentAmount	\$0	⟶	\$515
MonthsWithZeroBalanceOverLast6Months	1	⟶	2
MonthsWithZeroBalanceOverLast6Months	1	⟶	4
MostRecentPaymentAmount	\$0	⟶	\$775
MonthsWithLowSpendingOverLast6Months	6	⟶	5
MostRecentPaymentAmount	\$0	⟶	\$500
MonthsWithLowSpendingOverLast6Months	6	⟶	5
MonthsWithZeroBalanceOverLast6Months	1	⟶	2

Figure 3: Flipset for a person who is denied credit by the most accurate classifier built for credit. Each item describes minimal-cost changes for the individual to attain the desired outcome. We enumerated all 5 items in ≤ 1 second using the cost function in (4) and Algorithm 1.

4.2 Out-of-Sample Deployment

We now discuss an experiment that shows how recourse is affected by out-of-sample deployment. We consider a setting where a classifier is deployed on individuals who are underrepresented in the training population. Our setup is inspired by a real-world feedback loop with credit scoring in the United States, namely: credit scoring models are built using training datasets that underrepresent young adults, since young adults lack the credit history to apply for loans and produce labeled data, thus making it harder for young adults to be approved [see e.g., 40, for a discussion].

Setup. We consider a processed version of the `givemecredit` dataset [19]. Here, $y_i = -1$ if person i will experiencefi nancial distress over the next two years. The dataset contains $n = 150\,000$ individuals and $d = 10$ features related to their age, number of dependents, and recentfi nancial history. We assume that all features are actionable except for *Age* and *NumberOfDependents*.

Our audit compares the cost of recourse for individuals in the target population for two ℓ_2-penalized logistic regression models:

1. *Baseline Classifier.* This is a baseline model that we use for the sake of comparison. It is trained using $n = 112,500$ individuals in the processed dataset, which represents our target population.
2. *Biased Classifier.* This is the model that we wish to audit. We train this model on a sample of $n = 98,120$ individuals from the baseline classifier dataset, except individuals with $Age < 35$ (14, 380 individuals) are excluded.

We present both models in Appendix C.2. We set the threshold for approval for each model to approve 10% of individuals in the target population (i.e., $\hat{y}_i = -1$ if predicted probability of repaying the loan is $< 98\%$). We compute the cost of recourse using percentile distributions of features in the target population.

Results. We present the results of our audit in Figure 4. As shown, the cost of recourse can change significantly due to dataset shift. Here, the median cost of recourse of the biased model among young adults in the target population is 0.66, which means that they can onlyfl ip their predictions by a 66 percentile shift in a given feature. In comparison, the median cost of recourse for the baseline model among young adults is 0.14, which is significantly lower. We observe that the differences in the cost of recourse are far less pronounced for other age brackets, as the median cost for individuals that are represented in both populations does not appear to change.

To illustrate the effects of out-of-sample deployment from an individual perspective, we choose a young adult from the target population who is denied credit by classifiers and show the minimal cost-action that will attain the desired outcome from each classifier in Figure 5.

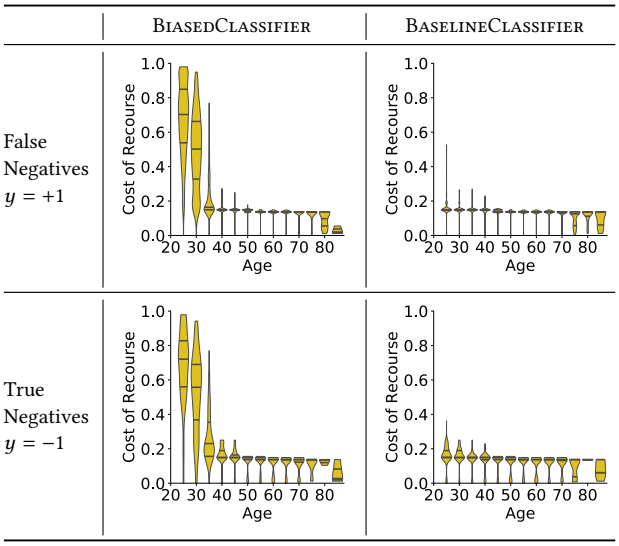

Figure 4: Distribution of the cost of recourse in the target population for each classifier split on the basis of the true outcome y. We show the cost of recourse for the biased classifier (left) and the baseline classifier (right). For each classifier, we show the cost distribution for false negative predictions (top) and true negative predictions (bottom). The baseline classifier is trained using a representative dataset from the target population, while the biased classifier is trained using an artificial dataset that undersamples young adults ($Age < 35$). The cost of recourse for young adults is significantly higher for the biased classifier, regardless of their true outcome.

Discussion. Our aim is not to suggest that out-of-sample deployment increases the cost of recourse, but that out-of-sample deployment can simply produce a significant change in the cost of recourse. Without theoretical guarantees on how recourse can change due to distributional differences in the training data and target population, such effects can only be measured by an audit using a sample of features from the target population. In practice, this procedure could be used for model procurement, where classifiers are trained using datasets that are significantly different from the target population on which they will be deployed.

There are other mechanisms by which out-of-sample deployment can affect recourse that are now shown here. In particular, models that do not allow users to adjust the threshold to a target population may result in infeasibility or higher costs for that population. Moreover, the set of feasible actions can differ significantly between populations. Both of these differences were controlled for

BiasedClassifier

FeatureC	urrent ValueR		equired Value
NumberOfTime30-59DaysPastDueNotWorse	1	⟶	0
NumberOfTime60-89DaysPastDueNotWorse	0	⟶	1

BaselineClassifier

FeatureC	urrent ValueR		equired Value
NumberOfTime3059DaysPastDueNotWorse	1	⟶	0
NumberOfTime6089DaysPastDueNotWorse	0	⟶	1
NumberRealEstateLoansOrLines	2	⟶	1
NumberOfOpenCreditLinesAndLoans	11	⟶	12
RevolvingUtilizationOfUnsecuredLines	0.358868	⟶	0.366256

Figure 5: Fullfl ipset for an individual in the baseline population with *Age* = 28. Under the baseline model, this individual has $\Pr(y_i = +1) = 97\%$. Under the biased model, the individual scores $\Pr(y_i = +1) = 93\%$. We show actions that will result in approval from the biased classifier (top) and the baseline classifier (bottom). The fullfl ipset for the biased classifier contains only a single item, whereas thefl ipset for the baseline classifier contains multiple items.

in this experiment: wefi xed the same action set, the same costs, and adjusted the threshold), so the observed effects of out-of-sample deployment only depend on distributional differences in the out-of-sample feature.

4.3 Evaluating Disparities in Recourse

We consider an experiment to demonstrate how our tools could be used to evaluate disparities in recourse across demographic groups. In particular, we wish to measure the *disparity* in recourse between males and females in a target population while controlling for potential confounders. Here, a disparity in recourse between males and females occurs if, given comparable individuals who are denied a loan in the target population, males can obtain credit by making easier changes (or vice-versa).

Setup. We consider a processed version of the `german` dataset [3]. Here, $y_i = -1$ if an individual is a "bad customer," which we assume means they have defaulted. The dataset contains $n = 1,000$ individuals and $d = 26$ features related to their loan application,financial status, and demographics. The dataset includes a feature, *gender*, which purposely drop from the training dataset.

We trained a classifier using ℓ_2-penalized logistic regression. We set the approval threshold for our classifier to approve individuals with a predicted probability of 50%. We ran an audit over all individuals who were denied the loan on the training dataset, and evaluated potential disparities by matching individuals with the same true label y and similar levels of predicted risk $\Pr(y = +1)$.

Results. As shown in Figure 6, the cost of recourse can vary between males and females in the target population. The plot in the left shows the cost for females and males among individuals with a true label of $y = 1$, while the plot in the right shows the cost for females and males among individuals with a true label of $y = -1$. These disparities in recourse can also be studied by comparing theflipsets as in Figure 7, where we show feasible actions for individuals in each subgroup with the same outcome and similar levels of predicted risk.

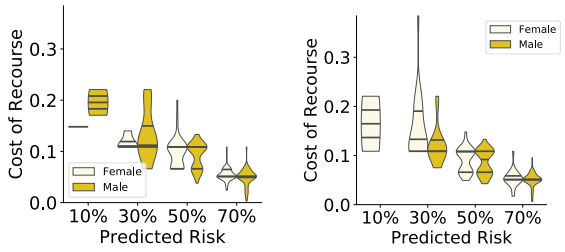

Figure 6: Overview of recourse disparities between males and females in the target population. On the top row, we plot the distribution of the cost of recourse for males and females based on their predicted risk and true label: we plot the cost for individuals where $y = +1$ (left) and $y = -1$ (right).

Female with $y_i = +1$ and $\Pr(y_i = +1) = 34.0\%$

FeatureC	urrent ValueR		equired Value
LoanAmount	$7432	⟶	$3684
LoanDuration	36 months	⟶	25 months
CheckingAccountBalance ≥ *200*	FALSE	⟶	TRUE
SavingsAccountBalance ≥ *100*	FALSE	⟶	TRUE
HasGuarantor	FALSE	⟶	TRUE
LoanAmount	$7432	⟶	$3684
LoanDuration	36 months	⟶	23 months
LoanRateAsPercentOfIncome	2.00%	⟶	1.00%
HasTelephone	FALSE	⟶	TRUE
HasGuarantor	FALSE	⟶	TRUE
LoanAmount	$7432	⟶	$912
LoanDuration	36 months	⟶	7 months
HasTelephone	FALSE	⟶	TRUE

Male with $y_i = +1$ and $\Pr(y_i = +1) = 32.1\%$

FeatureC	urrent ValueR		equired Value
LoanAmount	$15857	⟶	$7968
LoanDuration	36 months	⟶	32 months
CheckingAccountBalance ≥ *200*	FALSE	⟶	TRUE
HasCoapplicant	TRUE	⟶	FALSE
HasGuarantor	FALSE	⟶	TRUE
Unemployed	TRUE	⟶	FALSE
LoanAmount	$15857	⟶	$7086
LoanDuration	36 months	⟶	29 months
CheckingAccountBalance ≥ *200*	FALSE	⟶	TRUE
HasCoapplicant	TRUE	⟶	FALSE
HasGuarantor	FALSE	⟶	TRUE
LoanAmount	$15857	⟶	$4692
LoanDuration	36 months	⟶	29 months
CheckingAccountBalance ≥ *200*	FALSE	⟶	TRUE
SavingsAccountBalance ≥ *100*	FALSE	⟶	TRUE
LoanAmount	$15857	⟶	$3684
LoanDuration	36 months	⟶	21 months
HasTelephone	FALSE	⟶	TRUE

Figure 7: Flipsets for a matched pair of individuals from each subgroup. Individuals are matched on the basis of their true label y_i and their predicted risk $\Pr(y_i = +1)$.

5 CONCLUDING REMARKS

We have presented new tools to evaluate the recourse of a linear classifier in a population of interest and shown how they can inform various stakeholders, including: practitioners, who may unknowingly affect recourse through seemingly harmless modeling decisions; regulators, who may be interested in certifying that a model provides recourse over a target population; and decision-subjects, who may wish to learn changes that let them attain a desired predicted outcome.

5.1 Extensions

Non-Linear Classifiers. We are currently extending our tools to evaluate recourse for non-linear classifiers. One could immediately apply our tools to this setting (albeit heuristically) by solving our IP with a local linear classifier that approximates the local decision boundary in actionable space [see e.g., the technique used to estimate the local decision boundary in 27]. This approach could be useful to buildfl ipsets, but would not provide a proof of infeasibility required to assess the feasibility and difficulty of recourse.

Evaluating Strategic Incentives. Our tools can price incentives of a model in a target population by comparing the cost of recourse for different action sets [see e.g., 20]. Consider a case where a credit score contains features that are causally related to creditworthiness (e.g., income) as well as ancillary features that have predictive value, but are prone to manipulation (e.g., social media presence). In this case, we could evaluate incentive structures in a target population by comparing the cost of recourse using actions on: (i) only the causal features; or (ii) using causal features and at least one ancillary feature. If actions using (i) are less costly than actions using (ii), then individuals in the target population may not be incentivized to manipulate the model.

Measuring Flexibility. An interesting extension of our work is to run an audit where, for each individual in our target sample, we enumerate all distinct minimal-cost actions that will attain a desired outcome (i.e., by running the enumeration procedure in Algorithm 1 until the IP becomes infeasible). This produces a collection of minimal-cost actions that fully characterizes all of the ways in which an individual can attain a desired outcome. The size of this collection reflects thefl exibility of recourse for an individual, which could be used to quantitatively evaluate other properties of the recourse set (e.g., if a classifier provides 16 types of changes that provide recourse, 15 of which are legally contestable, then the model may be deemed contestable). This audit would be computationally intensive, but not necessarily intractable given that enumerating all actions can still be achieved relatively quickly based on the dimensions of the action set (i.e., $\leq$ 10 seconds).

5.2 Limitations

Misleading Flipsets. Flipsets do not necessarily reveal the principle reasons for a decision and may not present legally contestable information when it exists. Since thefl ipsets in this work only show features that must be altered, a person may fail tofl ip their prediction after making the suggested changes if they were to unintentionally change other features used by the classifier. In practice, this limitation can be overcome by providing users with clear guidelines (e.g., a complete list of features, or the signs of their coefficients). Alternatively, one could produce afl ipset of "worst-case" actions that would allow a person tofl ip their prediction while providing a buffer for potential changes on omitted features.

Cost Functions. Our off-the-shelf cost functions depend on percentile distributions, which may not correctly reflect the difficulty of recourse (e.g., if there are not enough samples from the target population, or the sample does not reflect the target population). In practice, we would expect an auditor to choose cost functions carefully using our guidelines in Section 3.2.

Manipulation and Model Theft. Providing individuals withflipsets has the drawback in that it could lead individuals to attain a desired outcome by making superficial changes. This kind of manipulation may be avoided by releasingfl ipsets with actions pertaining to features that are causally related to the outcome, or by training a model that only uses such features in thefi rst place [see e.g., 30].

Releasingfl ipsets could also lead to model theft (see e.g. [36], and efforts to reverse-engineer the Schufa credit score in Germany by crowdsourcing [12]). In light of potential model theft, it would be interesting to study how many actions must be collected to faithfully reconstruct a proprietary model, and whether model theft could be mitigated by producingfl ipsets with actions that have weaker guarantees.

5.3 Discussion

Machine Learning. Atfi rst glance, the goal of building a model that provides recourse may seem antithetical to the goal of building a model that is robust to manipulation. However, this is not the case if the model uses features that are causally related to the outcome. A model could be built so that a person can attain a desired outcome by only making constructive changes (e.g., a person who is denied credit can only be approved by changing features that improve their creditworthiness, such as income). A model could also be designed so that a person could attain a desired outcome by making "antagonistic" changes, but is incentivized to make constructive changes (see Section 5.1 for a discussion on how our tools can evaluate such incentives).

As shown in Section 4, recourse and predictive accuracy are not necessarily incompatible: it may be possible to train a model that provides recourse which is just as accurate as a model that fails to provides recourse. As illustrated in Example 2.1, however, there may exist classification problems in which there is a trade-off between recourse and accuracy. For example, a credit score could include immutable features that improve accuracy but reduce recourse. Such a trade-off would lead to a difficult decision regarding deployment: should we deploy a credit score with perfect predictive accuracy, but that precludes some individuals from receiving loans? Or should we deploy a model that provides all individuals with recourse but that may allocate loans inefficiently?

Policy Implications. Individual rights with respect to algorithmic decision-making are often motivated by the need for agency over machine-made decisions [e.g., 38, argues that autonomy is one of the core motivations for data protection laws]. Recourse reflects a precise notion of agency, namely the ability to meaningfully influence a decision-making process.

A lack of recourse may be contestable in applications where we would expect individuals to have agency over their predicted outcome (e.g., loan approval or hiring). It is not clear, however, if a "right to recourse" would extend to other areas where machine learning is used. In an application such as recidivism prediction, for example, one could argue that a model should provide defendants who are predicted to recidivate with the ability to change this prediction by altering certain kinds of attributes. For example, a defendant who is predicted to recidivate due to their age and prior criminal history should be able to alter this prediction by clearing their criminal history.

While regulations for algorithmic decision-making are still in their infancy, the vast majority of existing efforts have sought to ensure this kind of agency *indirectly*, through laws that focus on transparency and explanation [see e.g., regulations for credit scores in the United States such as 37].

In light of past efforts, we argue that recourse should be treated as a standalone policy objective in applications where it is desirable. This is because recourse not only represents a well-defined and measurable notion (c.f. explainability), but also because there exist multiple ways in which it can be effectively regulated. For example, one could mandate that a classification model that makes predictions on individuals must be paired with a recourse audit for its target population, or mandate that individuals who attain an undesirable prediction are provided with a set of actions that guarantee a desired outcome.

ACKNOWLEDGMENTS

We thank the following individuals for helpful discussions: Solon Barocas, Flavio Calmon, Suresh Venkatasubramanian, Yaron Singer, Ben Green, Hao Wang, Jesse Engreitz, and Margaret Haffey.

REFERENCES

[1] Charu C Aggarwal, Chen Chen, and Jiawei Han. 2010. The Inverse Classification Problem. *Journal of Computer Science and Technology* 25, 3 (2010), 458–468.
[2] Ifeoma Ajunwa, Sorelle Friedler, Carlos E Scheidegger, and Suresh Venkatasubramanian. 2016. Hiring by Algorithm: Predicting and Preventing Disparate Impact. *Available at SSRN* (2016).
[3] Kevin Bache and Moshe Lichman. 2013. UCI Machine Learning Repository.
[4] Pietro Belotti, Pierre Bonami, Matteo Fischetti, Andrea Lodi, Michele Monaci, Amaya Nogales-Gómez, and Domenico Salvagnin. 2016. On handling indicator constraints in mixed integer programming. *Computational Optimization and Applications* 65, 3 (2016), 545–566.
[5] Reuben Binns, Max Van Kleek, Michael Veale, Ulrik Lyngs, Jun Zhao, and Nigel Shadbolt. 2018. 'It's Reducing a Human Being to a Percentage': Perceptions of Justice in Algorithmic Decisions. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 377.
[6] Or Biran and Kathleen McKeown. 2014. Justification narratives for individual classifications. In *Proceedings of the AutoML workshop at ICML*, Vol. 2014.
[7] Jean-François Blanchette and Deborah G Johnson. 2002. Data retention and the panoptic society: The social benefits of forgetfulness. *The Information Society* 18, 1 (2002), 33–45.
[8] Allison Chang, Cynthia Rudin, Michael Cavaretta, Robert Thomas, and Gloria Chou. 2012. How to reverse-engineer quality rankings. *Machine Learning* 88, 3 (2012), 369–398.
[9] Alexandra Chouldechova, Diana Benavides-Prado, Oleksandr Fialko, and Rhema Vaithianathan. 2018. A Case Study of Algorithm-Assisted Decision Making in Child Maltreatment Hotline Screening Decisions. In *Conference on Fairness, Accountability and Transparency*. 134–148.
[10] Danielle Keats Citron and Frank Pasquale. 2014. The Scored Society: Due Process for Automated Predictions. *Washington Law Review* 89 (2014), 1.
[11] Kate Crawford and Jason Schultz. 2014. Big data and due process: Toward a framework to redress predictive privacy harms. *BCL Rev.* 55 (2014), 93.
[12] Open Knowledge Foundation Deutschland. 2018. Get Involved: We Crack the Schufa! https://okfn.de/blog/2018/02/openschufa-english/.
[13] Jinshuo Dong, Aaron Roth, Zachary Schutzman, Bo Waggoner, and Zhiwei Steven Wu. 2018. Strategic Classification from Revealed Preferences. In *Proceedings of the 2018 ACM Conference on Economics and Computation*. ACM, 55–70.
[14] Finale Doshi-Velez, Mason Kortz, Ryan Budish, Chris Bavitz, Sam Gershman, David O'Brien, Stuart Schieber, James Waldo, David Weinberger, and Alexandra Wood. 2017. Accountability of AI Under the Law: The Role of Explanation. *ArXiv e-prints*, Article arXiv:1711.01134 (Nov. 2017). arXiv:1711.01134
[15] Lilian Edwards and Michael Veale. 2017. Slave to the Algorithm: Why a Right to an Explanation Is Probably Not the Remedy You Are Looking for. *Duke L. & Tech. Rev.* 16 (2017), 18.
[16] Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. 2018. Analysis of classifiers' robustness to adversarial perturbations. *Machine Learning* 107, 3 (2018), 481–508.
[17] Moritz Hardt, Nimrod Megiddo, Christos Papadimitriou, and Mary Wootters. 2016. Strategic classification. In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science*. ACM, 111–122.
[18] IBM ILOG. 2018. CPLEX Optimizer 12.8. https://www.ibm.com/analytics/cplex-optimizer.
[19] Kaggle. 2011. Give Me Some Credit. http://www.kaggle.com/c/GiveMeSomeCredit/.
[20] Jon Kleinberg and Manish Raghavan. 2018. How Do Classifiers Induce Agents To Invest Effort Strategically? *ArXiv e-prints*, Article arXiv:1807.05307 (July 2018), arXiv:1807.05307 pages. arXiv:cs.CY/1807.05307
[21] Brian Y Lim and Anind K Dey. 2009. Assessing demand for intelligibility in context-aware applications. In *Proceedings of the 11th international conference on Ubiquitous computing*. ACM, 195–204.
[22] David Martens and Foster Provost. 2014. Explaining data-driven document classifications. *MIS Quarterly* 38, 1 (2014), 73–100.
[23] Hans Mittleman. 2018. Mixed Integer Linear Programming Benchmarks (MIPLIB 2010). http://plato.asu.edu/ftp/milpc.html.
[24] Cathy O'Neil. 2016. *Weapons of math destruction: How big data increases inequality and threatens democracy*. Broadway Books.
[25] Brett Poulin, Roman Eisner, Duane Szafron, Paul Lu, Russell Greiner, David S Wishart, Alona Fyshe, Brandon Pearcy, Cam MacDonell, and John Anvik. 2006. Visual explanation of evidence with additive classifiers. In *Proceedings Of The National Conference On Artificial Intelligence*, Vol. 21. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 1822.
[26] Dillon Reisman, Jason Schultz, Kate Crawford, and Meredith Whittaker. 2018. Algorithmic Impact Assessments: A Practical Framework for Public Agency Accountability. AI Now Technical Report.
[27] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. 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*. ACM, 1135–1144.
[28] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Anchors: High-precision model-agnostic explanations. In *AAAI Conference on Artificial Intelligence*.
[29] Andrew D Selbst and Solon Barocas. 2018. The Intuitive Appeal of Explainable Machines. *Fordham Law Review, Forthcoming* (2018).
[30] Shayak Sen, Piotr Mardziel, Anupam Datta, and Matthew Fredrikson. 2018. Supervising Feature Influence. *arXiv preprint arXiv:1803.10815* (2018).
[31] Ravi Shroff. 2017. Predictive Analytics for City Agencies: Lessons from Children's Services. *Big data* 5, 3 (2017), 189–196.
[32] Naeem Siddiqi. 2012. *Credit Risk Scorecards: Developing and Implementing Intelligent Credit Scoring*. Vol. 3. John Wiley & Sons.
[33] Alexander Spangher and Berk Ustun. 2018. Actionable Recourse in Linear Classification. In *Proceedings of the 5th Workshop on Fairness, Accountability and Transparency in Machine Learning*. https://econcs.seas.harvard.edu/files/econcs/files/spangher_fatml18.pdf
[34] Winnie F Taylor. 1980. Meeting the Equal Credit Opportunity Act's Specificity Requirement: Judgmental and Statistical Scoring Systems. *Buff. L. Rev.* 29 (1980), 73.
[35] John A Tomlin. 1988. Special ordered sets and an application to gas supply operations planning. *Mathematical programming* 42, 1-3 (1988), 69–84.
[36] Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. 2016. Stealing Machine Learning Models via Prediction APIs.. In *USENIX Security Symposium*. 601–618.
[37] United States Congress. 2003. The Fair and Accurate Credit Transactions Act.
[38] Sandra Wachter and Brent Mittelstadt. 2018. A Right to Reasonable Inferences: Re-thinking Data Protection Law in the Age of Big Data and AI. *Columbia Business Law Review, Forthcoming* (2018).
[39] Sandra Wachter, Brent Mittelstadt, and Chris Russell. 2017. Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GDPR. (2017).
[40] Colin Wilhelm. 2018. Big Data and the Credit Gap. https://www.politico.com/agenda/story/2018/02/07/big-data-credit-gap-000630.
[41] I-Cheng Yeh and Che-hui Lien. 2009. The Comparisons of Data Mining Techniques for the Predictive Accuracy of Probability of Default of Credit Card Clients. *Expert Systems with Applications* 36, 2 (2009), 2473–2480.