Classification with Strategically Withheld Data*

Anilesh K. Krishnaswamy,[1] Haoming Li,[2] David Rein,[1] Hanrui Zhang,[1] Vincent Conitzer.[1]

[1] Duke University, [2] University of Southern California.

anilesh@cs.duke.edu, haoming.li@usc.edu, irving.rein@duke.edu, hrzhang@cs.duke.edu, conitzer@cs.duke.edu

Abstract

Machine learning techniques can be useful in applications such as credit approval and college admission. However, to be classified more favorably in such contexts, an agent may decide to strategically withhold some of her features, such as bad test scores. This is a missing data problem with a twist: which data is missing *depends on the chosen classifier*, because the specific classifier is what may create the incentive to withhold certain feature values. We address the problem of training classifiers that are robust to this behavior.

We design three classification methods: MINCUT, HILL-CLIMBING (HC) and Incentive-Compatible Logistic Regression (IC-LR). We show that MINCUT is optimal when the true distribution of data is fully known. However, it can produce complex decision boundaries, and hence be prone to overfitting in some cases. Based on a characterization of truthful classifiers (i.e., those that give no incentive to strategically hide features), we devise a simpler alternative called HC which consists of a hierarchical ensemble of out-of-the-box classifiers, trained using a specialized hill-climbing procedure which we show to be convergent. For several reasons, MINCUT and HC are not effective in utilizing a large number of complementarily informative features. To this end, we present IC-LR, a modification of Logistic Regression that removes the incentive to strategically drop features. We also show that our algorithms perform well in experiments on real-world data sets, and present insights into their relative performance in different settings.

1 Introduction

Applicants to most colleges in the US are required to submit their scores for at least one of the SAT and the ACT. Both tests are more or less equally popular, with close to two million taking each in 2016 (Adams 2017). Applicants usually take one of these two tests – whichever works to their advantage.[1] However, given the growing competitiveness of college admissions, many applicants now take both tests and then strategically decide whether to drop one of the scores (if they think it will hurt their application) or report both.[2] The key issue here is that it is impossible to distinguish between an applicant who takes both tests but reports only one, and an applicant that takes only one test—for example because the applicant simply took the one required by her school, the dates for the other test did not work with her schedule, or for other reasons that are not strategic in nature.[3]

Say a college wants to take a principled machine learning approach to making admission decisions based on the scores from these two tests. For simplicity, assume no other information is available. Assume that the college has enough historical examples that contain the scores of individuals (on whichever tests are taken, truthfully reported) along with the corresponding ideal (binary) admission decisions.[4] Based on this data, the college has to choose a decision function that determines which future applicants are accepted. If this function is known to the applicants, they are bound to strategize and use their knowledge of the decision function to decide the scores they report.[4] How can the classifier be trained to handle strategic reporting of scores at prediction time?

To see the intricacies of this problem, let us consider a simple example.

Example 1. *Say the scores for each of the two tests (SAT and ACT) take one of two values: h (for high) or l (for low). Let $*$ denote a missing value. Then there are eight possible inputs (excluding $(*,*)$ since at least one score is required): (h,h), (h,l), (l,h), (l,l), $(h,*)$, $(*,h)$, $(l,*)$ and $(*,l)$. Assume the natural distribution (without any withholding) over these inputs is known, and so are the conditional probabilities of the label $Y \in \{0,1\}$, as shown below:*

Table 1: True distribution of inputs and targets:

X	(h,h)	(h,l)	(l,h)	(l,l)	$(h,*)$	$(*,h)$	$(l,*)$	$(*,l)$
$Pr(X)$	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
$Pr(Y=1 \mid X)$	0.9	0.7	0.3	0.1	0.6	0.6	0.2	0.2
$Pr(Y=0 \mid X)$	0.1	0.3	0.7	0.9	0.4	0.4	0.8	0.8

Assume $Y = 1$ is the more desirable "accept" decision. Then, ideally, we would like to predict $\widehat{Y} = 1$ whenever $X \in \{(h,h),(h,l),(h,*),(*,h)\}$. However, the strategic reporting of scores at prediction time effectively means, for

*A version of the paper including the Supplement is available at https://users.cs.duke.edu/~anilesh/clas-full.pdf

[1] https://www.princetonreview.com/college/sat-act

[2] https://blog.collegevine.com/should-you-submit-your-sat-act-scores/

[3] https://blog.prepscholar.com/do-you-need-to-take-both-the-act-and-sat

[4] We make these assumptions more generally throughout the paper.

example, that an input $(*, h)$ cannot be assigned the accept decision of $\widehat{Y} = 1$ unless the same is done for (l, h) as well; otherwise, someone with (l, h) would simply not report the first test, thereby misreporting $(*, h)$ and being accepted. Taking this into account, the classifier with minimum error is given by $\widehat{Y} = 1$ whenever $X \in \{(h, h), (h, l), (h, *)\}$.

There are many other settings where a similar problem arises. Many law schools now allow applicants to choose between the GRE and the traditional LSAT.[5] Recently, as a result of the COVID-19 pandemic, universities have implemented optional pass/fail policies, where students can choose to take some or all of their courses for pass/fail credit, as opposed to a standard letter grade that influences their GPA. They are often able to decide the status after already knowing their performance in the course. For credit scoring, some individuals might not report some of their information, especially if it is not mandatory by law (Florez-Lopez 2010).

The ability of strategic agents to withhold some of their features at prediction time poses a challenge only when the data used to train the classifier has some naturally missing components to begin with. For if not, the *principal* – e.g., the entity deciding on admissions – can reject all agents that withhold any of their features, thereby forcing them to reveal all features. We focus on how a principal can best train classifiers that are robust even when there is strategic withholding of data by agents. Our methods produce classifiers that eliminate the incentive for agents to withhold data.

Our contributions We now describe the key questions we are facing, and how we answer them. Our model is described formally in *Section 2*. All proofs are in the Supplement.

If the true input distribution is known, can we compute the optimal classifier? (Section 3) We answer this question in the affirmative by showing that the problem of computing the optimal classifier (Theorem 1) in this setting reduces to the classical Min-cut problem (Cormen et al. 2009). This analysis gives us the MINCUT classifier, which can be computed on the empirical distribution, estimated using whatever data is available. However, since it can potentially give complex decision boundaries, it might not generalize well.

Are there simpler classifiers that are robust to strategic withholding of features? (Section 4) We first characterize the structure of classifiers that are "truthful", i.e., give no incentive to strategically hide features at prediction time (Theorem 2). Using this characterization, we devise a hill-climbing procedure (HC) to train a hierarchical ensemble of out-of-the-box classifiers and show that the procedure converges (Theorem 4) as long as we have black-box access to an agnostic learning oracle. We also analytically bound the generalization error of HC (Theorem 3). The ensemble of HC can be populated with any of the commonly used classifiers such as logistic regression, ANNs, etc.

Another truthful classifier we present is a modification of Logistic Regression. This method, called IC-LR (Incentive Compatible Logistic Regression), works by encoding all features with positive values, and using positive regression coefficients – whereby it is in every agent's best interest to report all features truthfully. IC-LR uses Projected Gradient Descent for its training. The advantage of this method is that it can be directly to a large number of features.

How do our methods perform on real data sets? (Section 6) We conduct experiments on several real-world data sets to test the performance of our methods, comparing them to each other, as well as to other methods that handle missing data but ignore the strategic aspect of the problem. We see that our methods perform well overall, and uncover some interesting insights on their relative performance:

1. When the number of features is small, HC is the most reliable across the board.
2. When the number of features is small, and many of them are discrete/categorical (or suitably discretized), MINCUT and IC-LR perform better.
3. If a large number of features must be used, IC-LR gives the best performance, although HC performs reasonably well with some simple feature selection techniques.

Related work Our work falls broadly in the area of *strategic machine learning*, wherein a common assumption is that strategic agents can modify their features (i.e., misreport) in certain ways (normally at some cost), either to improve outcomes based on the classifier chosen by the principal (Hardt et al. 2016) or to influence which classifier is chosen in the first place (Dekel, Fischer, and Procaccia 2010). The main challenge in strategic machine learning, as in this paper, is the potential misalignment between the interests of the agents and the principal. Existing results in this line of work (Chen et al. 2018; Kleinberg and Raghavan 2019; Haghtalab et al. 2020), often mainly theoretical, consider classifiers of a specific form, say linear, and ways of misreporting or modifying features in that context. Our results are different in that we focus on a specific type of strategic misreporting, i.e., withholding parts of the data, and devise general methods that are robust to this behavior that, in addition to having theoretical guarantees, can be tested practically. Some experimental results (Hardt et al. 2016) do exist – but our work is quite different; for instance, we do not need to invent a cost function (as in Hardt et al. (2016)). Another major difference is that we consider generalization in the presence of strategic behavior, while most previous work does not (except for a concurrent paper (Zhang and Conitzer 2021)), which studies the sample complexity of PAC learning in the presence of strategic behavior).

Our problem can also be viewed as an instance of *automated mechanism design with partial verification* (Green and Laffont 1986; Yu 2011; Kephart and Conitzer 2015, 2016) where it is typically assumed that the feature space (usually called type space in mechanism design) is discrete and has reasonably small cardinality, and a prior distribution is known over the feature space. In contrast, the feature spaces considered in this paper consist of all possible combinations of potentially continuous feature values. Moreover, the population distribution can only be accessed by observing examples. Thus, common methodologies in automated mechanism design do not suffice for our setting.

[5] https://www.ets.org/gre/revised_general/about/law/

A set of closely related (in particular, to Theorem 1) theoretical results are those of Zhang, Cheng, and Conitzer (2019b,a, 2021b) on the problem of distinguishing "good" agents from "bad" (where each produces a different distribution over a sample space, and the agent can misreport the set of n samples that she has drawn). However, our work is different in that we consider the standard classification problem, we focus more on practical aspects, and we do not rely on the full knowledge of the input distribution.

Our work also finds a happy intersection between strategic machine learning and the literature on classification with missing data (Marlin 2008). The problem we study is also connected to *adversarial classification* (Dalvi et al. 2004; Dekel, Shamir, and Xiao 2010). We discuss these connections in more detail in the Supplement.

2 Preliminaries

We now describe our model and the requisite notation.

Model with strategically withheld features: We have an input space $\mathcal{X}$, a label space $\mathcal{Y} = \{0, 1\}$, and a distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$ which models the population. A classifier $f : \mathcal{X} \to \mathcal{Y}$ maps a combination of features to a label. Let $F = [k] = \{1, \ldots, k\}$ be the set of features, each of which a data point may or may not have. For $x \in \mathcal{X}$, let x_i denote the value of its i-th feature ($x_i = *$ if x does not have feature $i \in [k]$). For any $S \subseteq [k]$, define $x|_S$ to be the projection of x onto S (i.e., retain features in S and drop those not in S):

$$(x|_S)_i = \begin{cases} x_i, & \text{if } i \in S \\ *, & \text{otherwise.} \end{cases}$$

We assume that data can be strategically manipulated at prediction (test) time in the following way: an agent whose true data point is x can report any other data point x' such that $x|_S = x'$ for some $S \subseteq [k]$. We use $\to$ to denote the relation between any such pair x, x' ($x \to x' \iff \exists S \subseteq [k] : x|_S = x'$). Note that $\to$ is transitive, i.e., for any $x_1, x_2, x_3 \in \mathcal{X}$, $x_1 \to x_2$ and $x_2 \to x_3 \implies x_1 \to x_3$.

We assume agents prefer label 1 to 0: in response to a classifier f, an agent with data point x will always withhold [6] features to receive label 1 if possible, i.e., the agent will report $x' \in \operatorname{argmax}_{x'':x \to x''} f(x'')$. Incorporating such strategic behavior into the loss of a classifier f, we get

$$\ell_{\mathcal{D}}(f) = \Pr_{(x,y)\sim\mathcal{D}} \left[y \neq \max_{x':x \to x'} f(x') \right].$$

Truthful classifiers We will also be interested in *truthful* classifiers, which provably eliminate incentives for such strategic manipulation. A classifier f is *truthful* if for any $x, x' \in \mathcal{X}$ where $x \to x'$, $f(x) \geq f(x')$. In other words, not withholding any features is always an optimal way to respond to a truthful classifier. As a result, the loss of any truthful classifier f in the presence of strategically withheld features has the standard form: $\ell_{\mathcal{D}}(f) = \Pr_{(x,y)\sim\mathcal{D}}[f(x) \neq y]$.

[6] In practice, f might not be perfectly known, and agents might not be able to best respond. This problem does not arise for our methods, since they are truthul. For other classifiers, their accuracy may go up or down if agents fail to best-respond; but the assumption that agents best-respond is common in many such contexts.

Note that the so-called Revelation Principle – which states that in the presence of strategic behavior, any classifier f is equivalent to a truthful classifier f' – holds in this case because the reporting structure is transitive.[7] In other words, we are guaranteed that, for any classifier f, there exists a truthful classifier f', such that for any $x \in \mathcal{X}$, $\max_{x':x \to x'} f(x') = f'(x)$. Therefore, we focus on truthful classifiers in our model, without loss of generality.

3 The MINCUT Classifier

We first present a method for computing an optimal classifier *when the input distribution is fully known*.[8] Assuming $\mathcal{X}$ is finite, our goal is to characterize a classifier f^* which minimizes the loss $\ell_{\mathcal{D}}(.)$, for a known input distribution $\mathcal{D}$. As shorthand, define, for all $x \in \mathcal{X}$,

Definition 1. $\mathcal{D}^+(x) \coloneqq \Pr_{(x',y')\sim\mathcal{D}}[x' = x \wedge y' = 1]$, $\mathcal{D}^-(x) \coloneqq \Pr_{(x',y')\sim\mathcal{D}}[x' = x \wedge y' = 0]$.

The basic idea here is simple: to partition $\mathcal{X}$ into two sides, one labeled 1 and the other 0, where the error accrued for each $x \in \mathcal{X}$ is given by $\mathcal{D}^-(x)$ or $\mathcal{D}^+(x)$, according as x is labeled 1 or 0. Such a partition should crucially respect the constraints imposed by the strategic behavior of agents : if $x \to x'$, then either x is labeled 1 or x' is labeled 0.

Definition 2. Given $\mathcal{X}$ and $\mathcal{D}$, let $G(\mathcal{D}, \mathcal{X})$ be a directed capacitated graph with vertices $V = \mathcal{X} \cup \{s, t\}$, where the edges E and edge capacities u are defined as follows:

- For each $x \in \mathcal{X}$, there are edges (s, x) and (x, t) in E, with capacities $u(s, x) = \mathcal{D}^-(x)$ and $u(x, t) = \mathcal{D}^+(x)$.
- For all pairs $x, x' \in \mathcal{X}$ such that $x \to x'$, there is an edge $(x, x') \in E$ with capacity $u(x, x') = \infty$.

In terms of the graph defined above, computing the optimal classifier f^* we seek is equivalent to finding a minimum s-t cut on $G(\mathcal{D}, \mathcal{X})$. The intuition is that the edges from s and to t reflect the value gained from labeling an example 0 or 1, respectively; one of the edges must be cut, reflecting the loss of not assigning it to the corresponding side. Moreover, if $x \to x'$, then the corresponding edge with infinite capacity prevents the assigning of 0 to x and 1 to x'.

Theorem 1. *If $(S, \bar{S})$ is a minimum s-t cut of $G(\mathcal{D}, \mathcal{X})$ (where S is on the same side as s), then for the classifier $f^*(x) \coloneqq \mathbb{1}(x \in \bar{S})$, we have $\ell_{\mathcal{D}}(f^*) = \min_f \ell_{\mathcal{D}}(f)$.*

We note that, consequently, the optimal classifier can be computed in $\operatorname{poly}(|\mathcal{X}|)$ time. In practice, it is natural to expect that we do not know $\mathcal{D}$ exactly, but have a finite number of samples from it. A more practical option is to apply Theorem 1 to the empirical distribution induced by the samples observed, and hope for the classifier computed from that to generalize to the true population distribution $\mathcal{D}$.

[7] More details, including a formal proof, are in the Supplement.

[8] A theoretical companion paper (Zhang, Cheng, and Conitzer 2021a) contains a more general version of the mincut-based algorithm. There, the goal is to compute an optimal classifier with possibly more than 2 outcomes given perfect knowledge of the entire population distribution. In this paper, we investigate the special case with only 2 outcomes (i.e., accept and reject), but do not assume prior knowledge about the population distribution.

Implementing MINCUT Given a set $\widehat{\mathcal{X}}$ of m i.i.d. samples from $\mathcal{D}$, let $\widehat{\mathcal{D}}$ be the corresponding empirical distribution over $\widehat{\mathcal{X}}$, and $\bar{\mathcal{X}} := \widehat{\mathcal{X}} \cup \{x' : x' \to x, \exists x \in \widehat{\mathcal{X}}\}$. The MINCUT classifier is then obtained by applying Theorem 1 to $G(\widehat{\mathcal{D}}, \widehat{\mathcal{X}})$, and extending it to $\bar{\mathcal{X}}$ as and when required. Here, note that MINCUT runs in time $\text{poly}(m)$ (and not $\text{poly}(|\mathcal{X}|)$), since $G(\widehat{\mathcal{D}}, \widehat{\mathcal{X}})$ has m nodes, and checking if a test point is in $\bar{\mathcal{X}}$ takes $\text{poly}(m)$ time.

In light of traditional wisdom, the smaller m is relative to $\mathcal{X}$, the larger the generalization error of MINCUT will be. We do not attempt a theoretical analysis in this regard, but note that when $\mathcal{X}$ is large, the generalization error can be extremely large (see Example 2 in the Supplement). The reason for this is two-fold:

1. MINCUT can give complicated decision boundaries.
2. MINCUT is indecisive on samples not in $\bar{\mathcal{X}}$.[9]

Therefore, a suitable discretization of features is sometimes useful (see Section 6). Note that MINCUT is truthful, by virtue of the infinite capacity edges in Definition 2.

4 Truthful classifiers and HILL-CLIMBING

The other drawback of MINCUT, related to the issue of generalization just discussed, is that it can be hard to interpret meaningfully in a practical setting. In this section, we devise a simpler alternative called HILL-CLIMBING. To help introduce this algorithm, we first present a characterization of truthful classifiers in our setting, since we can limit our focus to them without loss of generality (as discussed in Section 2). For shorthand, we use the following definition:

Definition 3 (F'-classifier)**.** For a subset of features $F' \subseteq F$, a classifier f is said to be an F'-classifier if for all $x \in \mathcal{X}$, we have $f(x) = f(x|_{F'})$, and if there exists $i \in F'$ such that $x_i = *$, then $f(x) = 0$.

In other words, an F'-classifier depends only on the values of the features in F', rejecting any x where any of these is empty. We can collect many such classifiers into an ensemble as follows:

Definition 4 (MAX Ensemble)**.** For a collection of classifiers $\mathcal{C} = \{f_j\}$, its MAX Ensemble classifier is given by $\text{MAX}_{\mathcal{C}}(.) \coloneqq \max_j f_j(.)$.

This is equivalent to getting each agent to pick the most favorable classifier from among those in $\{f_j\}$. Now using the above definitions we have the following characterization of truthful classifiers:

Theorem 2. *A classifier f is truthful iff $f(.) = \text{MAX}_{\mathcal{C}}(.)$ for a collection of classifiers $\mathcal{C} = \{f_j\}$ such that, for some $\{F_j\} \subseteq 2^F$, each f_j is an F_j-classifier .*

Now, for any truthful classifier f, we can bound the gap between its population loss $\ell_{\mathcal{D}}(f)$ and its empirical loss on a set of samples $\widehat{\mathcal{X}}$ denoted by $\ell_{\widehat{\mathcal{X}}}(f) \coloneqq \frac{1}{m}\sum_{i\in[m]} |f(x_i) - y_i|$. Before stating a theorem to this end, we define the following entities: Let $\mathcal{H}$ be a base hypothesis space over $\mathcal{X}$, and $n \in \{1, \ldots, 2^k\}$ be a parameter. Define $d \coloneqq d_{\text{VC}}(\mathcal{H})$ to be the VC dimension of $\mathcal{H}$. Define $\bar{\mathcal{H}}$ as the set of all classifiers that can be written as the MAX Ensemble of n classifiers in $\mathcal{H}$.

Theorem 3. *Let $\widehat{\mathcal{X}} = \{(x_i, y_i)\}_{i\in[m]}$ be m i.i.d. samples from $\mathcal{D}$. For any $f \in \bar{\mathcal{H}}$, for any $\delta > 0$, with probability at least $1 - \delta$, we have $\ell_{\mathcal{D}}(f) \le \ell_{\widehat{\mathcal{X}}}(f) + O\left(\sqrt{\frac{dn\cdot\log dn\cdot\log m + \log(1/\delta)}{m}}\right)$.*

It is easy to see that for any of the commonly used hypothesis spaces – say $\mathcal{H}$ consists of linear hypotheses – if a truthful classifier f is in $\mathcal{H}$, then so are the components of the MAX Ensemble version of f as in Theorem 2. We have, however, stated Theorem 3 in slightly more general terms.

The HILL-CLIMBING **classifier** We now present a hill-climbing approach with provable convergence and generalization guarantees. The HILL-CLIMBING classifier (henceforth HC) is of the same form as given by the characterization of truthful classifiers in Theorem 2.[10] Intuitively, the approach works by considering a hierarchy of classifiers, organized by the features involved. For example, consider a setting with $k = 3$ features. We make a choice as to what classifiers we use — say f_1 for input of the form $(x_1, *, *)$, f_2 for input of the form $(x_1, x_2, *)$, and f_3 for input of the form (x_1, x_2, x_3). Any agent with features 1 and 2 (but not 3), for example, should be able to report both features so as to be classified by f_2, or feature 2 to be classified by f_1 instead. So in effect, assuming full knowledge of the classifiers, each agent can check all of the classifiers and choose the most favorable one. Without loss of generality, we assume that when a data point does not have all the features required by a classifier, it is automatically rejected.

Algorithm 1 HILL-CLIMBING (HC) Classifier

Input: data set $\widehat{\mathcal{X}} = \{(x_i, y_i)\}_{i\in[m]}$, n subsets $F_1, F_2, \ldots, F_n$ of F.
Initialize: $t \leftarrow 0$, $\{f_1^0, \ldots, f_n^0\}$.
while $\Delta > 0$ **do**
  for $i = 1, 2, \ldots, n$ **do**
    $S_i \leftarrow \{(x, y) \in \widehat{\mathcal{X}} : f_j^t(x|_{F_j}) = 0, \forall j \neq i\}$.
    $f_i^{t+1} = \text{argmin}_{f\in\mathcal{H}} \sum_{(x,y)\in S_i} |f(x|_{F_i}) - y|$.
  end for
  $f^* \leftarrow \text{MAX}_{\{f_1^{t+1}, \ldots, f_n^{t+1}\}}$; $\ell_t = \ell_{\widehat{\mathcal{X}}}(f^*)$
  $\Delta \leftarrow \ell_t - \ell_{t-1}$; $t \leftarrow t + 1$
end while
Return: f^*.

In short, HC (defined formally in Algorithm 1) works as follows: first choose a hypothesis space $\mathcal{H}$, in order for Theorem 3 to apply. Then select n subsets of F (where n is a parameter), say $F_1, F_2, \ldots, F_n$. For each F_j, we learn a F_j-classifier, say f_j, from among those in H. Start by initializing these classifiers to any suitable $\{f_1^0, \ldots, f_n^0\}$. In each

[9] This is more likely to happen when using a large number of features.

[10] And, therefore, is truthful, and inherits Theorem 3.

iterative step, each of the subclassifiers is updated to minimize the empirical loss on the samples that are rejected by all other classifiers. We next show that such an update procedure always converges. To do so, as far as our theoretical analysis goes, we assume we have black-box access to an agnostic learning oracle (Line 6 in Algorithm 1). After convergence, the HC classifier is obtained as the MAX Ensemble of these classifiers. The generalization guarantee of Theorem 3 applies directly to the HC classifier.

Theorem 4. *Algorithm 1 converges.*

Connection with MINCUT: The HC formulation given above can be thought of as a less complicated version of MINCUT: some of the edge constraints are ignored with respect to learning the individual classifiers, and are instead factored in via the MAX function. Say $F_1 \subset F_2$. For some x, it is possible that $f_1(x|_{F_1}) = 1$ and $f_2(x|_{F_1}) = 0$. In other words, the individual classifiers could violate the MINCUT constraints, in order to learn classification functions that generalize well individually, and also collectively thanks to the combined HC training procedure.

Implementing HC: In practice, the classifiers $\{f_1, f_2, \ldots, f_n\}$ in HC can be populated with any standard out-of-the-box methods such as logistic regression or neural networks, the choice of which can influence the performance of f. In Section 6, we test HC with a few such options. The assumption of having access to an agnostic learning oracle does not play a crucial role in practice, with standard training methods performing well enough to ensure convergence. Also, HC will converge in at most m (number of training examples) iterations, because in each iteration the number of correctly classified examples increases by at least one. (An iteration may need to train n individual classifiers.) This also means there is no difference between checking whether $\Delta > 0$ or $\Delta \geq 1/m$. In our experiments, we run HC using $\Delta \geq 10^{-4}$, and convergence is achieved pretty quickly (see the Supplement for exact details).

Choosing subsets: Note that we are free to choose any $F_1, F_2, \ldots, F_n$ to define HC. Its generalization (via Theorem 3), will depend on the choice of n. As more and more subsets of features are included (and further binning them based on their values), HC starts behaving more and more like MINCUT. In addition, using a large number of subsets increases the computational complexity of HC. In practice, therefore, the number of subsets must be limited somehow – we find that some simple strategies like the following work reasonably well: (a) selecting a few valuable features and taking all subsets of those features, (b) taking all subsets of size smaller than a fixed number k, say $k = 2$. In many practical situations, a few features (possibly putting their values in just a few bins) are often enough to get close to optimal accuracy, also improving interpretability (e.g., see Wang and Rudin (2015) or Jung et al. (2017)) The question of devising a more nuanced algorithm for selecting these subsets merits a separate investigation, and we leave this to future work.

5 Incentive-Compatible Logistic Regression

As we just mentioned, it is challenging to directly apply HC and MINCUT to a large number of features. As we will see, we can address this challenge in various ways to still get very strong performance with HC. Moreover, HC enjoys remarkable generality, generalization and convergence guarantees. Nevertheless, we would like to have an algorithm that tries to make use of all the available features, while still being truthful. In this section, we present such an approach, which, as we show later in Section 6, indeed performs comparably to – and in some cases better than – MINCUT and HC.

Below we present a simple and truthful learning algorithm, Incentive-Compatible Logistic Regression (IC-LR), which is a truthful variant of classical gradient-based algorithms for logistic regression. Recall that in logistic regression, the goal is to learn a set of coefficients $\{\beta_i\}$, one for each feature $i \in F$, as well as an intercept β_0, such that for each data point (x, y), the predicted label $\hat{y}$ given by

$$\hat{y} = \mathbb{1}\left[\sigma(\beta_0 + \sum_{i \in F} x_i \cdot \beta_i) \geq 0.5\right]$$

fits y as well as possible, where $\sigma(t) = 1/(1 + e^{-t})$ is the logistic function. Roughly speaking, IC-LR. (formally defined in Algorithm 2) works by restricting the coefficients $\{\beta_i\}$ in such a way that dropping a feature (i.e., setting x_i to 0) can never make the predicted label larger. If, without loss of generality, all feature values x_i are nonnegative[11], then this is equivalent to: for each feature $i \in F$, the coefficient $\beta_i \geq 0$. IC-LR. enforces this nonnegativity constraint throughout the training procedure, by requiring a projection step after each gradient step, which projects the coefficients to the feasible nonnegative region by setting any negative coefficient to 0 (equivalently, an ℓ_1 projection).

Algorithm 2 Incentive-Compatible Logistic Regression

Input: data set $\widehat{\mathcal{X}} = \{(x, y)\}$, learning rate $\{\eta_t\}$, $\delta \geq 0$.
Initialize: $t \leftarrow 0$, $\{\beta_0, \beta_1, \ldots, \beta_k\}$.
while $\Delta > \delta$ **do**
  $g_i \leftarrow 0$ for all $i \in \{0, 1, \ldots, k\}$
  for $(x, y) \in \widehat{\mathcal{X}}$ **do**
    $g_0 \leftarrow g_0 + \sigma\left(\beta_0 + \sum_{i \in F} x_i \cdot \beta_i\right) - y$
    for $i \in F$ **do**
      $g_i \leftarrow g_i + \left(\sigma\left(\beta_0 + \sum_{i \in F} x_i \cdot \beta_i\right) - y\right) \cdot x_i$
    end for
  end for
  $\forall i \in \{0, 1, \ldots, k\}$, $\beta_i \leftarrow \max\{\beta_i - \eta_t \cdot g_i, 0\}$
  $f^*(x) \coloneqq \mathbb{1}\left(\sigma\left(\beta_0 + \sum_{i \in F} \beta_i \cdot x_i\right) \geq 0.5\right)$
  $\ell_t = \ell_{\widehat{\mathcal{X}}}(f^*)$; $\Delta \leftarrow \ell_t - \ell_{t-1}$; $t \leftarrow t + 1$
end while
Return: f^*.

One potential issue with IC-LR. is the following: if a certain feature $x_i \geq 0$ is negatively correlated with the positive classification label, then IC-LR is forced to ignore it

[11] If not, they can be suitably translated.

Table 2: Data set summary statistics (num. = numerical, cat. = categorical)

Data set	Size	Total # of features	Size after balancing	Features after restriction
Australia	690	15	614	2 num., 2 cat.
Germany	1000	20	600	1 num., 3 cat.
Poland	5910	64	820	4 num.
Taiwan	30,000	23	13,272	4 ordinal

(since it is constrained to use positive coefficients). To make good use of this feature, we can include an inverted copy $x_i' = \lambda - x_i$ (where λ is chosen such that $x_i' \geq 0$). We could also choose an apt discretization of such features (using cross-validation) and translate the discretized bins into separate binary variables. Such a discretization can account for more complex forms of correlation, e.g., a certain feature's being too high or too low me makes the positive label likelier. In practice, we find that the latter method does better. If such transformations are undesirable, perhaps for reasons of complexity or interpretability, HC methods are a safer bet.

6 Evaluation

In this section, we show that, when strategic withholding is at play, MINCUT, HC and IC-LR perform well and provide a significant advantage over several out-of-the-box counterparts (that do not account for strategic behavior).

Datasets Four credit approval datasets are obtained from the UCI repository (Dua and Graff 2017), one each from Australia, Germany, Poland and Taiwan. As is common for credit approval datasets, they are imbalanced to various degrees. In order to demonstrate the performance of classifiers in a standard, controlled setting, we balance them by random undersampling. There is a dedicated community (Chawla, Japkowicz, and Kotcz 2004) that looks at the issue of imbalanced learning. We do not delve into these issues in our paper, and evaluate our methods on both balanced and imbalanced datasets (see the Supplement for the latter). In addition, to demonstrate the challenge of high-dimensional data imposed on some of the classification methods, the experiments are run on the datasets (a) restricted to 4 features,[12] and (b) with all available features. The basic characteristics of the datasets are summarized in Table 2 – note that there is enough variation in terms of the types of features present. We then randomly remove a fraction $\epsilon = 0, 0.1, \ldots, 0.5$ of all feature values in each dataset to simulate data that is missing "naturally" – i.e., not due to strategic withholding.

Testing We test all methods under two ways of reporting: "truthful", i.e., all features are reported as is, and "strategic", i.e., some features might be withheld if it leads to a better outcome. We measure the test accuracy of each classifier, averaged over N=100 runs, with randomness over the undersampling and the data that is randomly chosen to be missing, to simulate data missing for non-strategic reasons. Other metrics, and details about implementing and training the classifiers, are discussed in the Supplement. It is important to note that for testing any method, we have to, in effect, compute the best response of each data point toward the classifier. Since the methods we propose are truthful, this is a trivial task. But for other methods, this might not be easy, thereby limiting what baselines can be used.

Classifiers We evaluate our proposed methods, MINCUT, HC with logistic regression (HC (LR)) and neural networks (HC (ANN)) as subclassifiers, and incentive-compatible logistic regression (IC-LR), against several baseline methods.

First, they will be compared against three out-of-the-box baseline classifiers: logistic regression (LR), neural networks (ANN) and random forest (RF). We select LR for its popularity in credit approval applications; we select ANN for it being the best-performing individual classifier on some credit approval datasets (Lessmann et al. 2015); we select RF for it being the best-performing homogeneous ensemble on some credit approval datasets (Lessmann et al. 2015), as HC can be viewed as a homogeneous ensemble method. For the sake of exposition, we present numbers just for baselines based on LR, as they perform relatively better.

Second, for the purposes of comparison, we include MAJ – predict the *majority* label if examples with the exact same feature values appeared in the training set, and reject if not – which can be thought of as a non-strategic counterpart of MINCUT. We also include k-nearest neighbors (KNN) as a baseline, since it is closely related to MAJ.

These out-of-the-box classifiers need help dealing with missing data, whether they are missing naturally at training and test time or strategically at test time, and to this end, we employ (a) IMP: mean/mode imputation (Lessmann et al. 2015), and (b) R-F: reduced-feature modeling (Saar-Tsechansky and Provost 2007), for each of them.

When the dataset has a large number of features, MINCUT and IC-LR can be directly applied. For HC, we assist it in two ways: (a) by selecting 4 features based on the training data, denoted by FS (feature selection),[13] and (b) by choosing a limited number of small subsets (30 with 1 feature and 30 with 2 features), denoted by APP (approximation). Note that since our proposed methods are truthful, we can assume that features are reported as is. However, for all out-of-the-box classifiers, except IMP(LR), it is infeasible to simulate strategic withholding of feature values, due to the enormous number of combinations of features.

Last but not least, we test all methods with the discretization of continuous features (into categorical ones) (Fayyad and Irani 1993), for reasons given in earlier sections.

6.1 Results

For want of space, we report results only for $\epsilon = 0.2$. We also limit our exposition of HC, IMP and R-F methods to those based on logistic regression, as these perform better

[12] According to ANOVA F-value evaluated before dropping any feature values.

[13] Such a technique can be applied to other methods too – the results (see the Supplement) are not very different from those in Tables 4.

Table 3: Our methods vs. the rest: mean classifier accuracy for $\epsilon = 0.2$, balanced datasets, 4 features

Classifier	Australia		Germany		Poland		Taiwan	
	Tru.	Str.	Tru.	Str.	Tru.	Str.	Tru.	Str.
HC(LR)	.792	**.792**	.639	.639	.659	.659	.648	.648
MINCUT	.770	.770	.580	.580	.501	.501	.652	**.652**
IC-LR	.788	.788	.654	**.654**	.639	.639	.499	.499
IMP(LR)	.796	.791	**.663**	.580	**.714**	**.660**	**.670**	.618
R-F(LR)	**.808**	.545	.631	.508	.670	.511	.665	.590

Table 4: Our methods vs. the rest: mean classifier accuracy for $\epsilon = 0.2$, balanced datasets, 4 features ("w/ disc." stands for "with discretization of features")

Classifier	Australia		Germany		Poland		Taiwan	
	Tru.	Str.	Tru.	Str.	Tru.	Str.	Tru.	Str.
HC(LR) w/ disc.	.794	.794	.641	.641	.692	.692	.650	**.650**
MINCUT w/ disc.	.789	.789	.629	.629	.692	.692	.649	.649
IC-LR w/ disc.	**.800**	**.800**	.651	**.651**	.698	**.698**	.646	.646
IMP(LR) w/ disc.	.799	.762	**.652**	.577	**.719**	.631	**.686**	.541
R-F(LR) w/ disc.	.796	.542	.633	.516	.708	.522	.684	.587

than their ANN/RF/KNN counterparts. For a comprehensive compilation of all results, along with standard deviation numbers, please refer to the Supplement.

With a small number of features (Table 3): As expected, the out-of-the-box baselines perform well under truthful reporting, but not with strategic reporting. Our methods are robust to strategic withholding, and in line with the earlier discussion on the potential issues faced by MINCUT and IC-LR (in Sections 3 and 5), we see that **(a)** HC(LR) performs most consistently, and **(b)** in some cases, MINCUT (e.g., Poland) and IC-LR (e.g., Taiwan) do not do well.

With discretization (Table 4): As expected, discretization of numerical features into binary categories improves the performance of MINCUT and IC-LR, for reasons explained in Sections 3 and 5 respectively. We also see some benefit from discretization for HC(LR) when the features are mostly continuous (e.g., Poland), and less so when they are already discrete (e.g., Taiwan).

With a large number of features (Table 5): We see broadly similar trends here, except that in the case with discretization, IC-LR performs much better than before (e.g., Poland). The reason for this is that IC-LR is able to use all the available features once they are discretized into binary categories. However, without discretization, HC methods are more reliable (e.g., Poland and Taiwan).

On the out-of-the-box baselines: • *Imputation-based methods* are sensitive vis-á-vis the mean/mode values used. There is incentive to drop a certain feature if the imputed value is a positive signal. If there are many such features, then these methods perform poorly, as seen in Table 5 (cf. Table 3, Australia). If the imputed values do not give a clear signal (e.g., when the distribution of each feature value is not skewed), there is a high variance in the performance of these methods (see the Supplement). In some cases, the benchmarks perform as well as, or slightly better than, our incentive-compatible classifiers. For example, in Table 3, for the Australia and Poland data sets, the accuracy of IMP(LR) and that of HC(LR) are within 0.001 of each other. This happens because the imputed values are, in these cases (but not in most of our other cases), negative indicators of the positive label, and therefore there is generally no incentive to strategically drop features. • *Reduced-Feature modeling*, despite performing well under truthful reporting, allows too many examples to be accepted under strategic reporting, which hurts its performance. This is true especially for smaller ϵ, as each subclassifier has fewer examples to train on, giving several viable options for strategic withholding.

Table 5: Our methods vs. the rest: mean classifier accuracy for $\epsilon = 0.2$, balanced datasets, all features

Classifier	Australia		Germany		Poland		Taiwan	
	Tru.	Str.	Tru.	Str.	Tru.	Str.	Tru.	Str.
HCFS(LR)	.795	.795	.625	.625	.678	.678	.648	.648
HCAPP(LR)	.777	.777	.617	.617	.658	.658	.638	.638
MINCUT	.496	.496	.499	.499	.499	.499	.499	.499
IC-LR	.798	.798	.654	**.654**	.607	.607	.588	.588
HCFS(LR) w/ disc.	.794	.794	.632	.632	.694	.694	.649	.649
HCAPP(LR) w/ disc.	.782	.782	.620	.620	.724	.724	.644	.644
MINCUT w/ disc.	.534	.534	.503	.503	.499	.499	.550	.550
IC-LR w/ disc.	.805	**.805**	.653	.653	.773	**.773**	.667	**.667**
IMP(LR)	.802	.701	**.663**	.523	.729	.507	.657	.501
IMP(LR) w/ disc.	**.809**	.723	.659	.554	**.783**	.503	**.697**	.501

We note here that the variance (in the accuracy achieved) produced by our methods, since they are robust to strategic withholding, is much smaller than that of the baseline methods (exact numbers are deferred to the Supplement).

7 Conclusion

In this paper, we studied the problem of classification when each agent at prediction time can strategically withhold some of its features to obtain a more favorable outcome. We devised classification methods (MINCUT, HC and IC-LR) that are robust to this behavior, and in addition, characterized the space of all possible truthful classifiers in our setting. We tested our methods on real-world data sets, showing that they outperform out-of-the-box methods that do not account for the aforementioned strategic behavior.

An immediate question that follows is relaxing the assumption of having access to truthful training data – for example, one could ask what the best incentive-compatible classifier is given that the training data consists of best responses to a known classifier f; or, one could consider an online learning model where the goal is to bound the overall loss over time. A much broader question for future work is to develop a more general theory of robustness to missing data that naturally includes the case of strategic withholding.

Acknowledgements

We are thankful for support from NSF under award IIS-1814056.

Ethics Statement

The methods presented in this paper are geared towards preventing the strategic withholding of data when machine learning methods are used in real-world applications. This will increase the robustness of ML techniques in these contexts: without taking this issue into account, deployment of these techniques will generally result in a rapid change in the distribution of submitted data due to the new incentives faced, causing techniques to work much more poorly than expected at training time. Thus, there is an AI safety (Amodei et al. 2016) benefit to our work. The lack of strategic withholding also enables the collection of (truthful) quality data. Of course, there can be a downside to this as well if the data is not used responsibly, which could be the case especially if the features that (without our techniques) would have been withheld are sensitive or private in nature.

The other issues to consider in our context are those of transparency and fairness. We assume that the classifier is public knowledge, and therefore, agents can appropriately best-respond. In practice, this might not be the case; however, agents may learn how to best-respond over time if similar decisions are made repeatedly (e.g., in the case of college admissions or loan applications). While US college admission is often a black box, it need not be; many countries have transparent public criteria for university admissions (e.g., the Indian IIT admission system), and the same is true in many other contexts (e.g., Canadian immigration). Of course, transparency goes hand in hand with interpretability, i.e., the classifier must be easily explainable as well, and there could be a trade-off, in principle, between how easy the classifier is to interpret and the accuracy it can achieve. It is also possible that our methods hurt the chances of those with more missing data (similarly to how immigrants without credit history may not be able to get a credit card). This is to some extent inevitable, because if one can get in without any feature, everyone could get in by dropping all features. Therefore, the issue of fairness might arise in the case where some groups systematically tend to have more missing data.

References

Adams, C. J. 2017. *In Race for Test-Takers, ACT Outscores SAT—for Now*. URL https://www.edweek.org/ew/articles/2017/05/24/in-race-for-test-takers-act-outscores-sat--for.html.

Amodei, D.; Olah, C.; Steinhardt, J.; Christiano, P. F.; Schulman, J.; and Mané, D. 2016. Concrete Problems in AI Safety. *ArXiv* abs/1606.06565.

Chawla, N. V.; Japkowicz, N.; and Kotcz, A. 2004. Special issue on learning from imbalanced data sets. *ACM SIGKDD explorations newsletter* 6(1): 1–6.

Chen, Y.; Podimata, C.; Procaccia, A. D.; and Shah, N. 2018. Strategyproof linear regression in high dimensions. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, 9–26.

Cormen, T. H.; Leiserson, C. E.; Rivest, R. L.; and Stein, C. 2009. *Introduction to algorithms*. MIT press.

Dalvi, N.; Domingos, P.; Sanghai, S.; and Verma, D. 2004. Adversarial classification. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, 99–108.

Dekel, O.; Fischer, F.; and Procaccia, A. D. 2010. Incentive compatible regression learning. *Journal of Computer and System Sciences* 76(8): 759–777.

Dekel, O.; Shamir, O.; and Xiao, L. 2010. Learning to classify with missing and corrupted features. *Machine learning* 81(2): 149–178.

Dua, D.; and Graff, C. 2017. UCI Machine Learning Repository. URL http://archive.ics.uci.edu/ml.

Fayyad, U. M.; and Irani, K. B. 1993. Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning. In Bajcsy, R., ed., *Proceedings of the 13th International Joint Conference on Artificial Intelligence. Chambéry, France, August 28 - September 3, 1993*, 1022–1029. Morgan Kaufmann. URL http://ijcai.org/Proceedings/93-2/Papers/022.pdf.

Florez-Lopez, R. 2010. Effects of Missing Data in Credit Risk Scoring. A Comparative Analysis of Methods to Achieve Robustness in the Absence of Sufficient Data. *The Journal of the Operational Research Society* 61(3): 486–501.

Green, J. R.; and Laffont, J.-J. 1986. Partially verifiable information and mechanism design. *The Review of Economic Studies* 53(3): 447–456.

Haghtalab, N.; Immorlica, N.; Lucier, B.; and Wang, J. 2020. Maximizing Welfare with Incentive-Aware Evaluation Mechanisms. In *29th International Joint Conference on Artificial Intelligence*.

Hardt, M.; Megiddo, N.; Papadimitriou, C.; and Wootters, M. 2016. Strategic classification. In *Proceedings of the 2016 ACM conference on innovations in theoretical computer science*, 111–122.

Jung, J.; Concannon, C.; Shroff, R.; Goel, S.; and Goldstein, D. G. 2017. Simple rules for complex decisions. *Available at SSRN 2919024* .

Kephart, A.; and Conitzer, V. 2015. Complexity of mechanism design with signaling costs. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, 357–365.

Kephart, A.; and Conitzer, V. 2016. The revelation principle for mechanism design with reporting costs. In *Proceedings of the 2016 ACM Conference on Economics and Computation*, 85–102.

Kleinberg, J.; and Raghavan, M. 2019. How Do Classifiers Induce Agents to Invest Effort Strategically? In *Proceedings of the 2019 ACM Conference on Economics and Computation*, 825–844.

Lessmann, S.; Baesens, B.; Seow, H.-V.; and Thomas, L. C. 2015. Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. *European Journal of Operational Research* 247(1): 124–136.

Marlin, B. 2008. *Missing data problems in machine learning*. Ph.D. thesis, Citeseer.

Saar-Tsechansky, M.; and Provost, F. 2007. Handling missing values when applying classification models. *Journal of machine learning research* 8(Jul): 1623–1657.

Wang, F.; and Rudin, C. 2015. Falling rule lists. In *Artificial Intelligence and Statistics*, 1013–1022.

Yu, L. 2011. Mechanism design with partial verification and revelation principle. *Autonomous Agents and Multi-Agent Systems* 22(1): 217–223.

Zhang, H.; Cheng, Y.; and Conitzer, V. 2019a. Distinguishing Distributions When Samples Are Strategically Transformed. In *Advances in Neural Information Processing Systems*, 3187–3195.

Zhang, H.; Cheng, Y.; and Conitzer, V. 2019b. When samples are strategically selected. In *International Conference on Machine Learning*, 7345–7353.

Zhang, H.; Cheng, Y.; and Conitzer, V. 2021a. Automated Mechanism Design for Classification with Partial Verification. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Zhang, H.; Cheng, Y.; and Conitzer, V. 2021b. Classification with Few Tests through Self-Selection. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Zhang, H.; and Conitzer, V. 2021. Incentive-Aware PAC Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*.