On the Limits of Dictatorial Classification

Reshef Meir
School of Computer Science and Engineering
The Hebrew University of Jerusalem
reshef.meir@mail.huji.ac.il

Ariel D. Procaccia
Harvard SEAS
arielpro@seas.harvard.edu

Jeffrey S. Rosenschein
School of Computer Science and Engineering
The Hebrew University of Jerusalem
jeff@cs.huji.ac.il

ABSTRACT

In the strategyproof classification setting, a set of labeled examples is partitioned among multiple agents. Given the reported labels, an optimal classification mechanism returns a classifier that minimizes the number of mislabeled examples. However, each agent is interested in the accuracy of the returned classifier on its own examples, and may misreport its labels in order to achieve a better classifier, thus contaminating the dataset. The goal is to design strategyproof mechanisms that correctly label as many examples as possible.

Previous work has investigated the foregoing setting under limiting assumptions, or with respect to very restricted classes of classifiers. In this paper, we study the strategyproof classification setting with respect to prominent classes of classifiers—boolean conjunctions and linear separators—and without any assumptions on the input. On the negative side, we show that strategyproof mechanisms cannot achieve a constant approximation ratio, by showing that such mechanisms must be dictatorial on a subdomain, in the sense that the outcome is selected according to the preferences of a single agent. On the positive side, we present a randomized mechanism—Iterative Random Dictator—and demonstrate both that it is strategyproof and that its approximation ratio does not increase with the number of agents. Interestingly, the notion of dictatorship is prominently featured in all our results, helping to establish both upper and lower bounds.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent Systems*;
J.4 [**Computer Applications**]: Social and Behavioral Sciences—*Economics*

General Terms

Algorithms, Theory, Economics

Keywords

Mechanism design, Classification, Social choice, Game theory

Cite as: On the Limits of Dictatorial Classification, Reshef Meir, Ariel D. Procaccia and Jeffrey S. Rosenschein, *Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, van der Hoek, Kaminka, Lespérance, Luck and Sen (eds.), May, 10–14, 2010, Toronto, Canada, pp. 609-616

1. INTRODUCTION

A classification setting consists of an input space and a class of classifiers (known as the *concept class*), that is, functions from the input space to the set of *labels* $\{+,-\}$. A classification mechanism receives as input a *dataset*—a set of input points and their labels, where each such labeled input point is known as an *example*—and must return a classifier from the given class that classifies the given examples as well as possible. For instance, the input space might be images (represented as matrices of pixels), and the dataset might label different images as showing a human face (positive label) or not showing a face (negative label).

In our setting the labels are reported by strategic agents. Each agent controls a subset of the dataset, i.e., its own subset of (labeled) examples. The input points controlled by each agent are known, but their labels are private information. Given the reported labels, the classification mechanism selects a classifier. However, by lying the agents may achieve a classifier that better reflects their own labels, at the expense of overall accuracy.

The cost of an agent, known as its *risk*, is the portion of its dataset that is misclassified by the classifier. The social cost, or the *global risk*, is the portion of misclassified examples with respect to the complete dataset. The performance of a classifier is measured by its approximation ratio: the ratio between the number of examples it misclassifies, and the number of examples misclassified by the optimal classifier from the concept class. A mechanism is said to be an *α-approximation mechanism* if it yields an approximation ratio of α with respect to any dataset. In addition, we say that a mechanism is *strategyproof* (SP) if no agent can lower its own risk by lying. In this paper we study classification mechanisms that, at the same time, are SP and yield a good approximation ratio.

A Motivating Example. Consider a company, Nestor, conducting market research for a new line of products, e.g., candies. Nestor's research department, always eager to exploit new technologies, decides to use state-of-the-art classification algorithms to classify a wide variety of existing candies as either *Tasty* or *Nasty*. Sweetness, size, crunchiness, and other features are selected by the company to represent the input space. Nestor's customer relations department holds tasting events nationwide, where people taste and classify some popular candies. Individuals can only taste several candies each, so some candies receive more attention than others, because they have been served to more people. Nestor will then apply these answers to generalize over the entire feature space, creating the "perfect candy".

Naturally, our tasters have different preferences, so their answers may vary. Moreover, a person might try to influence the creation of the new treat; he might classify a candy that he likes as Nasty (or vice versa), if by doing so he can bias the final classification to better match his opinion on most candies. Other tasters may act in the same manner, if only to "counter the mistakes" of others. Such manipulation becomes even more powerful if carried out by one of Nestor's employees who is in charge, say, of reporting the results of all tasting events held in Boston.

While our candy example may have an artificial flavor to it, data collected from customers, retailers, salespeople, and others is increasingly being mined and analyzed to learn purchasing patterns, effectiveness of ads, and more. As the "data providers" are being affected by the outcome of such analyses, it is important to study the impact of their strategic behavior.

Previous work on SP learning mechanisms. The agenda of studying incentives in the context of classification was introduced by Meir et al. in [9]. They studied a very restricted setting, where the concept class contains exactly two classifiers: the constant positive classifier, which labels the entire input space positively, and the constant negative classifier. They put forward a deterministic SP 3-approximation mechanism, and a randomized SP 2-approximation mechanism, and proved that no mechanisms can do better.

In more recent work, the same authors studied a much richer set of concept classes, under the restricting assumption of *shared inputs* [10]. This means that all agents label the same set of data points, but may disagree on the "correct" label of each data point. They show that under this simplifying assumption, there is a randomized SP 3-approximation mechanism for any concept class, whereas the approximation ratio in the deterministic case depends linearly on the number of agents.

The model described above is the classification analog of an earlier model introduced by Dekel et al. [4] in the context of regression learning, i.e., where labels are real numbers. Dekel et al. devised a deterministic SP mechanism that provides good results with respect to restricted concept classes.

Approximate mechanism design without money. A crucial point with respect to all the previous papers on SP classification and regression [4, 9, 10], and this paper as well, is that the mechanisms under investigation do not employ payments. This is important since in the setting that we study, if payments were allowed and preferences of the agents are quasi-linear, truthfulness could be obtained while minimizing the social cost, using the well-known VCG mechanism (see, e.g., [11]).

However, there are many domains where payments cannot be made due to ethical or legal considerations (see, e.g., [16]). Moreover, in internet settings payments are particularly hard to implement, mainly due to security issues. The goal is then to obtain truthfulness by sacrificing optimality without resorting to money, that is, to design SP approximation mechanisms without payments. This agenda of *approximate mechanism design without money* was implicitly introduced in [4], and was recently made explicit by Procaccia and Tennenholtz [14].

Our results. The question repeated throughout this paper regards the minimal classification error that can be guaranteed using SP mechanisms in an unrestricted environment. More precisely, we are interested in the worst-case approximation ratio, when the output of the mechanism is compared to the optimal classifier.

Our main results are negative. We show that when the shared inputs assumption is dropped, deterministic mechanisms become utterly useless. Indeed, we present a concept class for which no deterministic SP mechanism can guarantee a nontrivial approximation: the approximation ratio must linearly increase with the size of the complete dataset. We consequently show that this negative result holds also for the widely used concept classes of Linear Separators and Boolean Conjunctions. We supply another negative result, albeit weaker, regarding randomized mechanisms, suggesting that their approximation ratio cannot be smaller than the size of the largest dataset controlled by a single agent.

To establish a clear distinction between the deterministic and the randomized cases, we put forward a randomized SP mechanism, namely the Iterative Random Dictator mechanism, which is capable of achieving an error that is close to the randomized lower bound on many concept classes.

Other related work. SP regression mechanisms have been studied by Perote-Peña and Perote [13]. Using simulations, they compared such mechanisms to simple empirical risk minimization under some complex assumptions on agent behavior. They were able to show that their SP mechanisms do perform better, albeit without supplying analytical bounds on the approximation ratio.

The clustering problem is closely related to classification. Perote and Perote-Peña [12] study the case of SP clustering. In their setting, each agent controls a point in $\mathbb{R}^2$. Given the reported locations, the clustering algorithm outputs a set of centroids. The utility of an agent is its distance from the closest centroid. The authors establish a strong impossibility result: they show that there are no reasonable deterministic clustering mechanisms that are SP. We essentially establish a result along the same lines in our model, but also complement it with positive and negative results regarding randomized mechanisms.

In learning theory there is some work on learning in noisy settings (see, e.g., [7, 2, 3]). These papers investigate situations where the noise is random or adversarial. While this line of work is related to the learning-theoretic interpretation of the current paper, our assumption is that false labels are reported so as to increase the utility of the liar, and not in an adversarial way. This assumption allows us to use game-theoretic tools to study how such "noise" can be discouraged in the first place.

For additional references, we refer the reader to earlier papers on SP regression and classification [4, 9, 10].

Structure of the paper. In Section 2 we formally describe the SP classification model. In Section 3 we prove lower bounds on the approximation ratio of deterministic and randomized SP mechanisms with respect to a toy problem. In Section 4 we show how to match these bounds by presenting the Iterative Random Dictator mechanism, and in Section 5 we consider the implications of our results with respect to prominent concept classes. Some proofs are omitted due to their length, but can be found online in [8].

2. MODEL

A *classifier* or *concept* c is a function from some input space $\mathcal{X}$ (either a finite set or some subset of $\mathbb{R}^d$) to *labels*

$\{+,-\}$. A *concept class* $\mathcal{C}$ is a set of concepts.

Let $I = \{1,\ldots,n\}$ be the set of agents, where $n \geq 2$. For each agent $i \in I$, let $X_i = \{x_{i,1},\ldots,x_{i,m_i}\} \in \mathcal{X}^{m_i}$ be the set of input points that agent i *controls*, and let $Y_i : X_i \to \{-,+\}$ be a function from input points to labels. Informally, the label $y_{i,j} = Y_i(x_{i,j})$ reflects whether agent i believes that the input point $x_{i,j}$ should be labeled as positive or negative. We assume that the input points in X_i are *public* information, whereas their labels Y_i are *private*. In game-theoretic terms, the labels Y_i are the *type* of agent i.

We refer to the pair $s_{i,j} = \langle x_{i,j}, y_{i,j}\rangle$ as an *example*. We denote all examples that are controlled by agent i by $S_i = \{s_{i,j}\}_{j=1}^{m_i}$, or alternatively (and slightly abusing notation), $S_i = \langle X_i, Y_i\rangle$. We emphasize that each example is controlled by exactly one agent, but there may be several examples in the same place, possibly with different labels (i.e., with $x_{i,j} = x_{i',j'}$ but $y_{i,j} \neq y_{i',j'}$).[1]

A *Classification Problem* is defined by the pair $\langle \mathcal{X}, \mathcal{C}\rangle$. An *instance* of the problem is given by a complete dataset $S = \langle S_1,\ldots,S_n\rangle$, specifying the number of agents n, as well as the exact examples of each agent. We sometimes also use the notation S to refer to the *multiset* containing all examples, i.e., $S = \bigcup_{i\in I} S_i$. For a specific instance we also denote $m = |S| = \sum_i m_i$, and $k = \max_i m_i$. It clearly holds that $\max\{n,k\} \leq m \leq n \cdot k$.

We use the prominent 0–1 loss function (also employed by Meir et al. [9, 10]) to measure the error of a classifier. The *risk*, or cost, of agent i with respect to concept c is the relative number of errors that c makes on S_i.[2] Formally

$$\mathrm{R}_i(c,S) = \frac{1}{m_i} \sum_{\langle x,y\rangle \in S_i} [\![c(x) \neq y]\!] \quad ,$$

where $[\![A]\!]$ denotes the indicator function of the expression A. The *global risk* is defined as

$$\mathrm{R}_I(c,S) = \sum_{i\in I} \frac{m_i}{m} \cdot \mathrm{R}_i(c,S) = \frac{1}{m} \sum_{\langle x,y\rangle \in S} [\![c(x) \neq y]\!] \quad .$$

In other words, the global risk is proportional to the *social cost*, that is, the weighted sum of the costs of the agents, where the weights are $w_i = m_i$. The goal is to find a classifier that is good on average, namely, to minimize the global risk.

A *deterministic mechanism* $\mathcal{M}$ is a function from input datasets S to classifiers $c \in \mathcal{C}$; recall that we do not allow a mechanism to make payments. Furthermore, we remark that $\mathrm{R}_i(\mathcal{M}(S),S)$ for all $i \in I$ and $\mathrm{R}_I(\mathcal{M}(S),S)$ are well-defined. A *randomized mechanism*, given a dataset, returns a distribution over concepts, i.e., the output of a randomized mechanism is a random variable $\hat{c}$ taken from $\mathcal{C}$, and we would like to minimize the *expected risk*. Formally, for a randomized mechanism $\mathcal{M}$, define

$$\mathrm{R}_i(\mathcal{M}(S),S) = \mathbb{E}\left[\mathrm{R}_i(\hat{c},S)|S\right] \quad ,$$
$$\text{and} \quad \mathrm{R}_I(\mathcal{M}(S),S) = \mathbb{E}\left[\mathrm{R}_I(\hat{c},S)|S\right] \quad .$$

We denote by $\mathrm{OPT}(S)$ (or simply by OPT if S is clear from the context) the optimal risk that can be attained on the dataset S, i.e.,

$$\mathrm{OPT}(S) = \min_{c\in\mathcal{C}} \mathrm{R}_I(c,S) \quad .$$

If $\mathrm{OPT}(S) = 0$, we say that S is *separable*. The quality of the outcome of a mechanism is measured using the common notion of *approximation*. Formally, a mechanism $\mathcal{M}$ is said to be an α-*approximation* mechanism if for every dataset S,

$$\mathrm{R}_I(\mathcal{M}(S),S) \leq \alpha \cdot \mathrm{OPT}(S) \quad .$$

A classifier c with the lowest risk in $\mathcal{C}$ with respect to S is known as an *Empirical Risk Minimizer* (ERM), that is,

$$\mathrm{ERM}(S) = \mathrm{argmin}_{c\in\mathcal{C}} \mathrm{R}_I(c,S) \quad .$$

Note that even if $\mathcal{C}$ contains an infinite number of concepts, S is finite and therefore can only be classified in a finite number of ways. Thus there is always at least one ERM, although it may not be unique.

In our game-theoretic model, the agents may lie by reporting labels that are different than the ones given by Y_i. We denote by $\overline{Y}_i : X_i \to \{+,-\}$ the reported labels of agent i. We also denote by

$$\overline{S}_i = \{\langle x, \overline{Y}_i(x)\rangle \ : \ x \in X_i\}$$

the reported partial dataset of agent i, and the reported dataset is denoted by $\overline{S} = \langle \overline{S}_1,\ldots,\overline{S}_n\rangle$.

A *strategyproof* mechanism has the property that agents can never benefit by lying, regardless of the behavior of the other agents. Formally, for a dataset S and $i \in I$, let S_{-i} be the complete dataset without the partial dataset of agent i. A (deterministic or randomized) mechanism $\mathcal{M}$ is *strategyproof* (SP) if for every dataset S, for every $i \in I$, and for every $\overline{S}_i$,

$$\mathrm{R}_i(\mathcal{M}(S),S) \leq \mathrm{R}_i(\mathcal{M}(\overline{S}_i,S_{-i}),S) \quad .$$

We would like to find good truthful approximation mechanisms, i.e., mechanisms that are SP and also yield an α-approximation ratio for a small α.

3. LOWER BOUNDS FOR A TOY PROBLEM

In this section we study the limitations of deterministic SP mechanisms with respect to a toy classification problem. We find that well-known impossibility results from social choice theory can be leveraged to obtain a powerful lower bound. The purpose of the toy problem is twofold:

1. It allows a clear presentation of our technique.
2. In Section 5, lower bounds for other prominent classification problems are obtained using the toy problem.

Our toy problem is defined as follows. The input space is $\mathcal{X}_{ab} = \{a,b\}$, i.e., there are only two possible input points. There are three possible classifiers: $\mathcal{C}_{ab} = \{c_a, c_b, c_{ab}\}$. These classifiers are defined as follows:

	c_a	c_b	c_{ab}
a	$+$	$-$	$+$
b	$-$	$+$	$+$

We denote the toy problem by $\mathrm{TP} = \langle \mathcal{X}_{ab}, \mathcal{C}_{ab}\rangle$.

We will demonstrate that every deterministic SP mechanism cannot guarantee *any* approximation ratio other than the most trivial one. Indeed, notice that the maximum number of errors a mechanism can make on a given dataset is m

[1] This multiplicity occurs not due to lack of decisiveness on the part of the agent, but since the coordinates of the space $\mathcal{X}$ may not be able to separate different data points. For example, two different candies may have the same sweetness, size, crunchiness, etc.

[2] *Risk*, rather than *cost*, is the term usually employed in the classification literature.

(since any classifier makes at most $|S| = m$ errors); therefore, it is not hard to obtain an SP m-approximation mechanism (we discuss some subtleties in Section 4). Our negative result states that with respect to the toy problem above, no deterministic mechanism can do better, up to a constant.

THEOREM 3.1. *There is no deterministic SP $o(m)$-approximation mechanism for TP.*

Crucially, even when k (the maximum number of points controlled by an agent) is bounded, the approximation ratio still grows linearly with the number of agents n.

The remainder of this section is devoted to proving Theorem 3.1. To this end, we first put forward some key notions from social choice theory, and present the Gibbard-Satterthwaite impossibility result [5, 15] regarding voting rules. We then show how every voting setting can be embedded into a classification setting, where concepts and private labels replace the candidates and voters' preferences. The main steps of the reduction show (roughly) that: first, every SP classification mechanism induces an SP voting rule, which must be dictatorial (due to Gibbard-Satterthwaite); second, the proposed mechanism must also be dictatorial; finally, any dictatorial mechanism cannot have a good approximation ratio.

Voting rules and manipulation. Let C be a finite set of *candidates*, and, as before, let $I = \{1, \ldots, n\}$ be a set of agents. Each agent $i \in I$ holds a (private) strict linear order $\succ_i$ over C, that is, a strict ranking of the candidates. We denote the set of all linear orders over C by $\mathcal{L} = \mathcal{L}(C)$. The preferences of agent i are denoted by $\succ_i \in \mathcal{L}$, i.e., $c_1 \succ_i c_2$ means that agent i prefers c_1 to c_2.

The collection of the preferences of all agents is called a *preference profile*, and denoted by $\succ = \langle \succ_1, \ldots, \succ_n \rangle$. We denote by $\succ_{-i}$ the preference profile of all agents except i.

A *voting rule* is a function $f : \mathcal{L}^n \to C$ from preference profiles to candidates, which designates the winning candidate given the preferences of the voters. A voting rule f is *manipulable* if there is a profile $\succ \in \mathcal{L}$ and some preference $\succ_i'$ of agent i, such that i strictly gains (according to $\succ_i$) by voting $\succ_i'$ instead of $\succ_i$, i.e.,

$$f(\succ_{-i}, \succ_i') \succ_i f(\succ) \quad .$$

If f is not manipulable, it is said to be *strategyproof*. Lastly, a voting rule f is *dictatorial* if there is some agent i such that for every preference profile $\succ \in \mathcal{L}$, the top-ranked candidate in $\succ_i$ is the winner $f(\succ)$. The Gibbard-Satterthwaite Theorem [5, 15] essentially implies that strategyproofness can only be obtained via dictatorship.

THEOREM 3.2 (GIBBARD-SATTERTHWAITE). *Let $|C| \geq 3$. If f is onto C and SP, then f is dictatorial.*

Reducing classification to voting. The crux of the proof of Theorem 3.1 is the fact that the Gibbard-Satterthwaite Theorem basically holds in our toy classification problem. This is not obvious, since the latter theorem implicitly assumes that the agents can hold any ranking of the alternatives. In contrast, in our setting the preferences of an agent over the classifiers are determined by its private labels, but *a priori* the labels cannot induce every ranking. Let us presently turn to the theorem's proof.

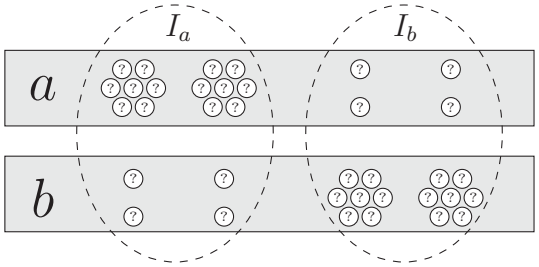

Figure 1: The fixed dataset X of TP, for $n = 4$ and $k' = 3$. Private labels are not shown, as they are determined as part of the reduction.

PROOF OF THEOREM 3.1. In our scenario, the collection of input points $X = \langle X_1, \ldots, X_n \rangle$ is fixed, and defined as follows. Assume for ease of exposition that the number of agents n is even, and that there exists $k' \in \mathbb{N}$ such that $k = 2k' + 3$. There are two blocks of agents in I, each of size $n/2$. Agents from the first block hold $2k' + 1 = k - 2$ input points on a and two more input points on b; we denote these agents by I_a. Agents from the second block, denoted I_b, are symmetric with respect to a and b. The structure of the dataset is illustrated in Figure 1.

As the input points X are common knowledge, the behavior of the mechanism is fully determined by the reported labels Y. Note that any deterministic classification mechanism in our scenario is a function $\mathcal{M}$ from datasets into the set of classifiers $\mathcal{C}_{ab} = \{c_a, c_b, c_{ab}\}$.

We define our set of candidates to be $C = \{c_a, c_b, c_{ab}\} = \mathcal{C}_{ab}$ (thus it is of size 3). When agent i labels its set of input points X_i with the labeling Y_i, it induces a preference ranking over all possible classifiers (possibly with ties). Given Y_i and $\succ_i \in \mathcal{L}$, we say that $\succ_i$ *fits* Y_i if for all $c_1, c_2 \in C$,

$$\mathrm{R}_i(c_1, S) < \mathrm{R}_i(c_2, S) \iff c_1 \succ_i c_2 \quad . \tag{1}$$

That is, the order of preference over the three possible classifiers that is naturally induced by the labeling Y_i is exactly $\succ_i$. Note that at most one order fits a given labeling. Hence, it is clear that there is a natural mapping from Y to $\succ$. This is not enough though, as our reduction requires a (one-to-one but not onto) mapping in the other direction as well.

With respect to the first block of agents, I_a, we define a mapping g_a from preferences over C to labelings by explicitly setting the private labels of the input points of X_i for each of the six possible orders on C. This mapping is shown in Table 1.

The reader is encouraged to verify, using the leftmost three columns of the table, that each order $\succ_i$ indeed fits the labeling $Y_i = g_a(\succ_i)$. Since there are only 6 possible orders on C, g_a is well-defined. For agents in I_b, g_b is defined in a symmetric way, with the roles of a and b switched. To conclude the point, the full mapping is naturally defined by taking $g(\succ) = \langle (g_a(\succ_i))_{i \in I_a}, (g_b(\succ_i))_{i \in I_b} \rangle$. Hence, every classification mechanism $\mathcal{M}$ induces a valid voting rule $(\mathcal{M} \circ g) : \mathcal{L} \to C$.

LEMMA 3.3. *Denote $f = \mathcal{M} \circ g$. If $\mathcal{M}$ is SP and guarantees a bounded approximation ratio, then f is dictatorial.*

PROOF. By the Gibbard-Satterthwaite Theorem (Theorem 3.2), in order to show that f is dictatorial it suffices to demonstrate that f is SP and onto.

	Value of mapping $g_a(\succ_i)$			Number of errors on S_i		
$\succ_i$	$Y_i(A_1)$	$Y_i(A_2)$	$Y_i(B)$	c_a	c_b	c_{ab}
$c_a \succ_i c_{ab} \succ_i c_b$	+	+	−	0	$2k'+3$	2
$c_a \succ_i c_b \succ_i c_{ab}$	+	−	−	$k'+1$	$k'+2$	$k'+3$
$c_{ab} \succ_i c_a \succ_i c_b$	+	+	+	2	$2k'+1$	0
$c_{ab} \succ_i c_b \succ_i c_a$	−	+	+	$k'+2$	$k'+1$	k'
$c_b \succ_i c_a \succ_i c_{ab}$	−	−	−	$2k'+1$	2	$2k'+3$
$c_b \succ_i c_{ab} \succ_i c_a$	−	−	+	$2k'+3$	0	$2k'+1$

Table 1: The leftmost column enumerates all six possible orders over classifiers. The next three columns define the label of each input point according to $Y_i = g_a(\succ_i)$, as follows: B denotes the 2 input points on b; the input points on a are divided into two sets, where A_1 denotes the first (arbitrary) k' input points, and A_2 denotes the other $k'+1$. The last three columns show the risk of each classifier with respect to the labels Y_i.

We first argue that the onto property of f follows from the fact that $\mathcal{M}$ has a bounded approximation ratio. Indeed, let $c \in \mathcal{C}_{ab}$. By Table 1, for each agent $i \in I$ there is some preference order $\succ_c$ such that the risk of i with respect to c, when the labels of i are set according to g, is zero. Now, assume that all agents have the labels that induce $\succ_c$ (i.e., $\forall i \in I_a,\ Y_i = g_a(\succ_c)$, and likewise for I_b); then we must have that for the labels $Y_c = \langle Y_1, \ldots, Y_n \rangle$ the output of $\mathcal{M}$ is c, since this concept has zero risk and the others have nonzero risk. Indeed, otherwise the approximation ratio is unbounded. Therefore,

$$f(\succ_c, \ldots, \succ_c) = \mathcal{M}(X, g(\succ_c, \ldots, \succ_c)) = \mathcal{M}(X, Y_c) = c \quad .$$

We now prove that f is SP. Indeed, assume for the purpose of contradiction that f is manipulable. Then there are $\succ \in \mathcal{L}^n,\ i \in I$, and $\succ'_i \in \mathcal{L}$ such that

$$f(\succ_{-i}, \succ'_i) \succ_i f(\succ) \quad .$$

Assume without loss of generality that $i \in I_a$. From the definition of f,

$$\mathcal{M}(X, g(\succ_{-i}, \succ'_i)) \succ_i \mathcal{M}(X, g(\succ)) \quad .$$

Let $Y = g(\succ),\ Y'_i = g_a(\succ'_i)$. Thus

$$\mathcal{M}(X, \langle Y_{-i}, Y'_i \rangle) \succ_i \mathcal{M}(X, Y) \quad .$$

From the definition of g, Y_i fits $\succ_i$; thus, by using Equation (1) we get that

$$\mathrm{R}_i(\mathcal{M}(X, \langle Y_{-i}, Y'_i \rangle), S) < \mathrm{R}_i(\mathcal{M}(X, Y), S) \quad .$$

Therefore agent i strictly gains by misreporting its true labels, in contradiction to the assumption that $\mathcal{M}$ is SP. □

We are now in a position to bound the approximation ratio of $\mathcal{M}$. Define f as in Lemma 3.3; by the lemma, f is dictatorial. Without loss of generality let agent 1 be the dictator, with $1 \in I_a$; that is, agent 1 holds 2 input points on b, and $k-2$ input points on a. We construct a dataset S, defining a labeling for the input points in X, as follows. We label the input points of agent 1 that are on b as negative, and label all the other input points of all agents as positive. It holds that

$$\mathrm{OPT}(S) = \mathrm{R}_I(c_{ab}, S) = \frac{2}{|S|} \quad .$$

On the other hand, the labeling that we constructed for agent 1 is the image under g of the order $\succ_{c_a}$ where agent 1 favors c_a. Since f is dictatorial, $f(\succ_{c_a}, \succ_{-1})$ must be c_a, hence the image under $\mathcal{M}$ is c_a as well, since

$$\mathcal{M}(S_1, S_{-1}) = \mathcal{M}(X, g(\succ_{c_a}, \succ_{-1})) = f(\succ_{c_a}, \succ_{-1}) = c_a \quad .$$

We have that $\mathrm{R}_I(\mathcal{M}(S), S) = \mathrm{R}_I(c_a, S) = \frac{1}{2} - \frac{2}{|S|}$. Therefore, the approximation ratio is bounded from below by

$$\frac{\mathrm{R}_I(\mathcal{M}(S), S)}{\mathrm{OPT}(S)} = \frac{\frac{1}{2} - \frac{2}{|S|}}{\frac{2}{|S|}} = \frac{m}{4} - 1 \quad ,$$

as required. □

3.1 Lower bounds of randomized mechanisms

A natural question concerns the lower approximation bound when randomization is allowed. Ideally, we would take our toy problem TP (or a similar one), and prove that the expected risk of any *randomized* SP mechanism is also bounded from below by some function of the size of the dataset.

We prove a somewhat weaker result by adding *private weights* to the dataset of each agent, in addition to the private labels. These weights affect the risk, but cannot be taken into account by the mechanism. We define the *weighted risk* as follows:

$$\widetilde{R}_i(c, S) = \sum_{j=1}^{m_i} w_{i,j} [\![c(x_{i,j}) \neq y_{i,j}]\!] \quad , \qquad (2)$$

where $\sum_{j=1}^{m_i} w_{i,j} = 1$. A mechanism is said to be SP in this setting if for any set of weights no agent has an incentive to submit false labels. We construct a new randomized toy problem, RTP, with two agents and three classifiers, and show the following.

THEOREM 3.4. *Assume that the risk is computed according to Equation* (2)*. Then there is no* randomized *SP $o(k)$-approximation mechanism.*

The proof idea is in principle similar to the proof of Theorem 3.1, but instead of using the Gibbard-Satterthwaite Theorem (which is limited to deterministic voting rules) we use a later, more general result by Gibbard [6].

4. THE ITERATIVE RANDOM DICTATOR MECHANISM

In this section we present a randomized mechanism that beats the deterministic lower bound given by Theorem 3.1, by guaranteeing an approximation ratio that does not grow with the number of agents n. Rather, it depends only on k—the size of the largest partial dataset. This result holds with respect to a number of prominent concept classes. Our mechanism is based on the simple idea of sequential, or iterative, dictatorship.

Let us first describe this idea in its deterministic interpretation. A naïve dictatorship would simply return a classifier in $\mathrm{ERM}(S_i)$, where $i \in I$ is the dictator, for any given dataset S. However, a dictatorship does not provide

a bounded approximation ratio. Indeed, consider a situation where agent 1 controls one positive example, and agent 2 controls one negative example. Agent 1 might choose a classifier that labels the input point of agent 2 as positive, even if the concept class contains a perfect classifier with zero global risk.

This is remedied by considering an iterative (or sequential) dictatorship. The deterministic mechanism considers the classifiers that are optimal with respect to agent 1, that is, the set $\mathrm{ERM}(S_1)$. Next, the mechanism restricts its attention to the subset of classifiers, among $\mathrm{ERM}(S_1)$, that are optimal with respect to agent 2. We iteratively proceed in this way, until we have enumerated all the agents. If there are several classifiers left at the end of the process, we choose one of them arbitrarily. In other words, agent 1 is a dictator, but since ties between classifiers may occur, they are sequentially broken according to the labels of agents $2, \ldots, n$, in this order. Similar sequential dictatorships have been proposed in the literature for other domains, such as voting [6] and resource matching [1].

Now, it is straightforward that this mechanism is SP. Furthermore, the mechanism trivially yields an approximation ratio of $|S| = m$. Indeed, if there is a classifier with global risk zero, this classifier is chosen by the mechanism. Otherwise, the optimal classifier has a global risk of at least $\frac{1}{|S|}$, whereas the maximum global risk is 1. Thus, this simple mechanism yields an SP upper bound that matches, up to a constant, the one given in Theorem 3.1.

The randomized version of the iterative dictator notion is quite natural as well. The mechanism first chooses a random permutation of the agents. Then, the mechanism restricts its attention to the optimal classifiers according to the first agent, and iteratively breaks ties according to the next agent in the permutation. We refer to this mechanism as *Iterative Random Dictator* (IRD), and denote it by $\mathcal{M}_{\mathrm{IRD}}$.

Since there are some technical nuances with the implementation of the mechanism, we give a more formal description in Algorithm 1. Assume each input point $x \in X$ has a boolean field, $x.marked$. Intuitively, the field is set to T (for true) when our mechanism determines the final label of x.

We wish to quickly verify that our mechanism is indeed SP, regardless of the classification problem. Indeed, first note that the order in which agents are selected as dictators is independent of the labeling and thus cannot be affected by it. Consider any agent j that is selected in iteration t. The input points of j that are not in $S_{t,j}$ are already classified (i.e., marked). Furthermore, the mechanism minimizes the risk of agent j with respect to $S_{t,j}$, hence agent j cannot benefit by lying. We next establish that the IRD mechanism can be implemented efficiently.

THEOREM 4.1. *Let $|S| = m$. Suppose that the time required to find an ERM on S with respect to the concept class $\mathcal{C}$, is polynomial in m. Then it is possible to implement the IRD mechanism such that the runtime of the algorithm is also polynomial in m.*

The key idea of this efficient implementation is to never explicitly represent the current set of concepts $\mathcal{C}_t$. Instead, we employ the marked examples as support vectors, using them to determine the status of each unmarked example in the next stage of the algorithm. The exact details of the implementation are omitted.

The following result establishes an upper bound on the approximation ratio provided by the IRD mechanism. This bound holds for any finite input space, regardless of the concept class.

Algorithm 1 ITERATIVE RANDOM DICTATOR ($\mathcal{M}_{IRD}$)

```
Initialize the given concept class as C_0 = C
Initialize x.marked ← F for each x ∈ X
Generate a random permutation that maps iterations to
    agents π : {1,...,n} → {1,...,n}
for iteration t = 1,...,n do
    Select agent j = π(t)
    S_{t,j} ← {⟨x, Y_j(x)⟩ : x ∈ X_j ∧ ¬x.marked}
    // Consider all the examples of agent j
    // that are not marked at time t
    Let c̃ ∈ argmin_{c∈C_{t-1}} R_j(c, S_{t,j})
    // c̃ is an ERM with respect to C_{t-1} and S_{t,j}
    C_t ← {c ∈ C_{t-1} : ∀⟨x, y⟩ ∈ S_{t,j} (c(x) = c̃(x))}
    // Remove concepts that disagree with c̃ on
    // some example in S_{t,j}
    for each input point x ∈ X do
        if ∀c, c' ∈ C_t (c(x) = c'(x)) then
            x.marked ← T
        end if
    end for
end for
Return an arbitrary concept from C_n
```

THEOREM 4.2. *Let $\mathcal{X}$ be a finite space of size s. For any class $\mathcal{C}$, the IRD mechanism is an SP $(s{\cdot}k{+}1)$-approximation mechanism for the classification problem $\langle \mathcal{X}, \mathcal{C} \rangle$.*

PROOF SKETCH. Let $\mathcal{X} = \{a_1, \ldots, a_s\}$ be a finite input space, and let $c^* \in \mathrm{ERM}(S)$. Denote the set of "good" agents by I_G; these are the agents that completely agree with c^* (i.e., $\mathrm{R}_i(c^*, S) = 0$). Furthermore, let $G = \bigcup_{i \in I_G} S_i$ be the set of all "good" examples. Denote by B all the examples that are inconsistent with c^*. Let I_B be the set of "bad" agents that control examples in B, i.e., $I_B = I \setminus I_G$. Clearly it holds that $\mathrm{OPT} = |B|/m$. For each $j \leq s$, we denote by G_j the set of all good examples that are located in a_j, that is, $G_j = \{\langle x, y \rangle \in G \ : \ x = a_j\}$.

Let Z_j be a random variable that reflects the number of examples in G_j that are labeled correctly. Furthermore, denote by $Z = \sum_{j=1}^{s} Z_j$ the total number of good examples that are labeled correctly.

LEMMA 4.3. $\mathbb{E}[Z] \geq |G| - k|B| \cdot s$.

The proof of the lemma relies on the fact that if a good agent is selected prior to all bad agents, then all the examples controlled by this agent will be classified correctly (as well as all other examples in the same location).

The total risk is composed of the misclassified bad examples (at most $|B|$), and the misclassified good examples ($|G| - Z$). Thus

$$\mathrm{R}_I(\mathcal{M}_{IRD}(S), S) \leq \mathbb{E}\left[\frac{|B| + (|G| - Z)}{|S|}\right] = \frac{|B|}{|S|} + \frac{|G| - \mathbb{E}[Z]}{|S|}$$

$$\leq \frac{|B|}{|S|} + \frac{k|B| \cdot s}{|S|} = (1 + k \cdot s)\frac{|B|}{|S|} = (s \cdot k + 1) \cdot \mathrm{OPT} \ ,$$

as announced. □

Theorem 4.2 implies that the IRD mechanism breaks the deterministic lower bound given for our toy problem TP. Indeed, recall that in TP we have that $s{=}2$, thus the mechanism provides an approximation ratio of $2k + 1 = o(m)$.

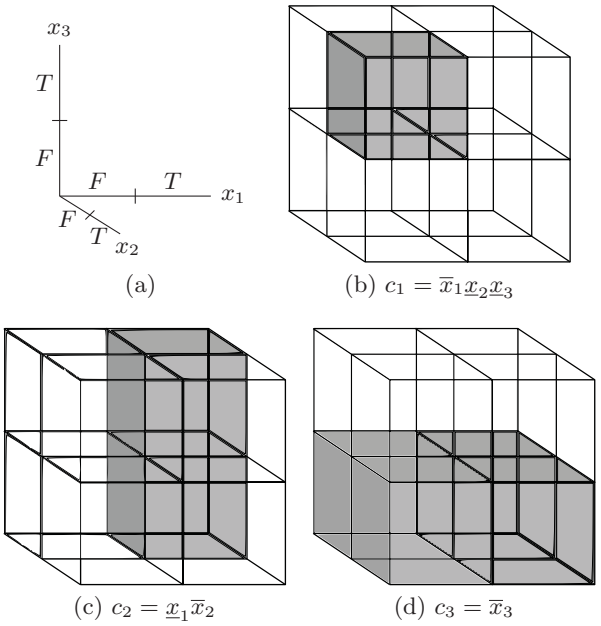

Figure 2: Conjunctions over the input space $\mathcal{X} = \{T, F\}^3$, which contains 8 different input points. Figure (a) shows the axes of the space. Figures (b)–(d) are examples of different classifiers. The grayed area is the positive part of space.

5. SPECIFIC CONCEPT CLASSES

At this point it remains unclear to what degree the lower bound of Theorem 3.1 also applies to common concept classes that are in wide use, and whether the IRD mechanism provides any guarantees with respect to such concept classes. In this section we show how our results apply to two highly useful concept classes: boolean conjunction formulas, and linear separators.

5.1 Boolean Conjunctions

A common concept class used over $\mathcal{X} = \{T, F\}^d$ is the set $\mathcal{C}_d$ of all *literal conjunctions* over d boolean variables.

We denote the events $x_i = T$ and $x_i = F$ by $\underline{x}_i, \overline{x}_i$, respectively. A literal conjunction $c : \{T, F\}^d \to \{+, -\}$ is defined by two sets $pos_c, neg_c \subseteq \{1, \ldots, d\}$. For any input vector $x \in \{T, F\}^d$, $c(x) = -$ if x contradicts some literal in c, i.e., if there is $i \in \{1, \ldots, d\}$ such that either $\underline{x}_i$ and $i \in neg_c$, or $\overline{x}_i$ and $i \in pos_c$. Otherwise, $c(x) = +$. Each classifier c classifies a hypercube in $\{T, F\}^d$ as positive. Figure 2 demonstrates some conjunctions and their spatial interpretation. We refer to the problem of learning a conjunction in $\{T, F\}^d$ as CONJ_d.

REMARK 5.1. *Although all our results in this section deal with conjunctions, similar results hold for* disjunctions.

We have the following dichotomy, according to the dimension of the problem.

THEOREM 5.2. *For $CONJ_1$, there is a 3-approximation deterministic SP mechanism, and this bound is tight. For any $d \geq 2$, there is no deterministic SP $o(\sqrt{m})$-approximation mechanism for $CONJ_d$.*

The 1-dimensional case can be shown to be equivalent to the classification problem presented in [9], which employs only two constant functions (it can be shown in the same way that there is a randomized 2-approximation mechanism for CONJ_1, and that this bound is also tight).

As for the higher dimensional case, the proof uses a reduction to the toy problem TP that was presented in Section 3. We omit the full proof due to space constraints, but the reader can grasp its key ideas from the proof sketch of Theorem 5.3 in the next section.

We now recall that for a fixed dimension d, it holds that $s = |\mathcal{X}| = 2^d$ is also fixed, and the IRD mechanism is an SP $2^d \cdot \mathcal{O}(k)$-approximation mechanism for CONJ_d (Theorem 4.2). This result breaks the deterministic bound of $\sqrt{m}$, which depends also on the number of agents.[3]

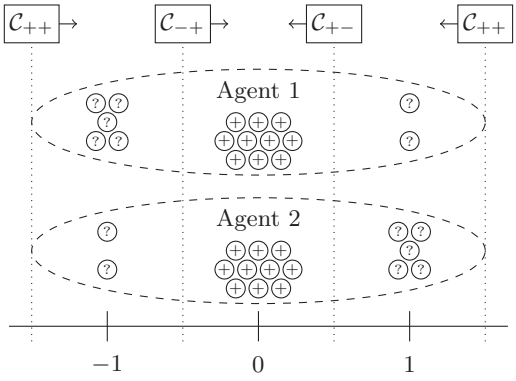

Figure 3: Illustration of the proof of Theorem 5.3, with $n = 2$. Agents 1 and 2 are obtained from agents with more examples on a, and more examples on b, respectively. The arrows indicate the positive halfspace of each class of classifiers.

5.2 Linear Separators

The class of linear separators over $\mathbb{R}^d$ is the set of classifiers that are defined by the parameters $\mathbf{a} \in \mathbb{R}^d$ and $\mathbf{b} \in \mathbb{R}$, and map a point $\mathbf{x} \in \mathbb{R}^d$ to $+$ if and only if $\mathbf{a} \cdot \mathbf{x} + \mathbf{b} \geq 0$. We refer to the problem of learning a linear separator over $\mathbb{R}^d$ as LINEAR_d.

As in the previous settings, a deterministic SP mechanism cannot guarantee a good approximation ratio.

THEOREM 5.3. *For any $d \geq 1$, there is no deterministic SP $o(\sqrt{m})$-approximation mechanism for $LINEAR_d$.*

PROOF SKETCH. As in the proof of Theorem 5.2, we reduce the problem to TP. We sketch the reduction for $d = 1$.

Both scenarios contain n agents. Originally each agent controls $2k + 3$ input points in TP, whereas it controls $k' = n(2k + 3)^2$ input points in the new instance of LINEAR_1.

The total number of examples in the original instance is $m = n(2k+3) = \Theta(nk)$. The total number of new examples is thus $m' = nk' = (n(2k + 3))^2 = \Theta((nk)^2) = \Theta(m^2)$. We translate any example originally on a to the location -1, and each example on b to the location 1 on the real line. The remainder of examples ($k' - (2k + 3)$ for each agent) are all placed on 0 with a *positive label*. Figure 3 shows the spatial arrangement of the input points in the new instance.

The remainder of the proof is straightforward, with one subtlety. Since the class of linear separators (even in $\mathbb{R}^1$) is continuous, we cannot create a case where only 3 classifiers

[3]It is possible to slightly alter the reduction so that the lower bound becomes $\Omega(\max\{\sqrt{m}, k\})$.

are admissible. Hence we partition the class $\mathcal{C}_1$ of linear separators over $\mathbb{R}$ into the following four sets: $\mathcal{C}_{+-} = \{c \in \mathcal{C}_1 : c(-1) = +, c(1) = -\}$, and we define $\mathcal{C}_{--}$, $\mathcal{C}_{-+}$ and $\mathcal{C}_{++}$ in the same way. These classes also appear in Figure 3.

Note that every concept in $\mathcal{C}_{--}$ classifies 0 as negative, and hence cannot give an approximation ratio of $o(m') = o(\sqrt{m})$. The three other sets can be mapped to c_a, c_b and c_{ab}. This observation allows us to show that any deterministic SP $o(\sqrt{m})$-approximation mechanism for LINEAR_1 induces an $o(m)$-approximation mechanism for TP. □

We next show, once again, that the IRD mechanism beats the deterministic lower bound, this time with respect to the problem LINEAR_1.

THEOREM 5.4. *The IRD mechanism is an SP $\mathcal{O}(k^2)$ -approximation mechanism for LINEAR_1.*

This result is somewhat weaker than the one for literal conjunctions, since our bound is quadratic in k (instead of linear), and for higher dimensions we have no bound at all. Nevertheless, the proof is significantly more involved, due to the continuous nature of the problem.

The proof outline is as follows. Let c^* be an optimal classifier. As in the proof of Theorem 4.2, we divide the agents into "good" and "bad". In every iteration a single agent sets the final labels for its examples, but also enforces a label on other examples in the process. If the agent is good, then all the labels set in this iteration must agree with c^* as well, but if this agent is bad then some examples might be "ruined" in this iteration, that is, labeled in a way that disagrees with c^*. The heart of the proof is in bounding the expected number of examples that are ruined in each iteration; indeed, we show that this number decreases exponentially fast.

We conjecture that the bound for LINEAR_1 can be tightened, making it linear in k. More importantly, we believe that a similar bound can be obtained for linear separators in higher dimension; we leave this issue as a very challenging open question for future research.

6. DISCUSSION

We have studied a broad framework for strategyproof classification. We have shown that the design of useful deterministic strategyproof mechanisms is impossible without further assumptions (such as shared inputs). Our results further suggest that the classification quality of randomized SP mechanisms strongly depends on the size of the largest dataset (rather than on the number of agents). While the results of Section 3.1 use some additional assumptions, they still serve as an indirect demonstration of this principle; we further conjecture that Theorem 3.4 holds even if we drop the assumption of private weights.

An important observation is called for regarding the issue of *generalization*, as discussed in previous research on strategyproof classification and regression: is there an algorithm capable of learning an approximately optimal concept with respect to some distribution over the input space, by generalizing from a bounded number of examples? To improve generalization, more samples have to be taken. However, this means that we increase the size of the dataset, thereby worsening the approximation ratio. Our negative results in the decision making setting thus become acute in a machine-learning setting that involves generalization from samples.

Future research may follow several directions. It may be interesting to address our technical open questions by improving some of the bounds or relaxing underlying assumptions. Our results can also be extended to more concept classes. Alternatively, different assumptions and restrictions may be applied to the input to guarantee a constant upper bound. It may also be interesting to explore other loss functions (instead of 0–1 loss), or investigate altogether different formulations of the strategyproof classification setting.

7. REFERENCES

[1] A. Abdulkadiroğlu and T. Sönmez. Random serial dictatorship and the core from random endowments in house allocation problems. *Econometrica*, 66(3):689–701, 1998.

[2] N. H. Bshouty, N. Eiron, and E. Kushilevitz. PAC learning with nasty noise. *Theoretical Computer Science*, 288(2):255–275, 2002.

[3] N. Dalvi, P. Domingos, Mausam, S. Sanghai, and D. Verma. Adversarial classification. In *Proc. of 10th KDD*, pages 99–108, 2004.

[4] O. Dekel, F. Fischer, and A. D. Procaccia. Incentive compatible regression learning. In *Proc. of 19th SODA*, pages 277–286, 2008.

[5] A. Gibbard. Manipulation of voting schemes. *Econometrica*, 41:587–602, 1973.

[6] A. Gibbard. Manipulation of schemes that mix voting with chance. *Econometrica*, 45:665–681, 1977.

[7] S. A. Goldman and R. H. Sloan. Can PAC learning algorithms tolerate random attribute noise? *Algorithmica*, 14(1):70–84, 1995.

[8] R. Meir. Strategy proof classification. Master's thesis, The Hebrew University of Jerusalem, 2008. Available from: http://www.cs.huji.ac.il/∼reshef24/spc.thesis.pdf.

[9] R. Meir, A. D. Procaccia, and J. S. Rosenschein. Strategyproof classification under constant hypotheses: A tale of two functions. In *Proc. of 23rd AAAI*, pages 126–131, 2008.

[10] R. Meir, A. D. Procaccia, and J. S. Rosenschein. Strategyproof classification with shared inputs. In *Proc. of 21st IJCAI*, pages 220–225, 2009.

[11] N. Nisan. Introduction to mechanism design (for computer scientists). In N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani, editors, *Algorithmic Game Theory*, chapter 9. Cambridge University Press, 2007.

[12] J. Perote and J. Perote-Peña. The impossibility of strategy-proof clustering. *Economics Bulletin*, 4(23):1–9, 2003.

[13] J. Perote and J. Perote-Peña. Strategy-proof estimators for simple regression. *Mathematical Social Sciences*, 47:153–176, 2004.

[14] A. D. Procaccia and M. Tennenholtz. Approximate mechanism design without money. In *Proc. of 10th EC*, pages 177–186, 2009.

[15] M. Satterthwaite. Strategy-proofness and Arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory*, 10:187–217, 1975.

[16] J. Schummer and R. V. Vohra. Mechanism design without money. In N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani, editors, *Algorithmic Game Theory*, chapter 10. Cambridge University Press, 2007.