Strategyproof Classification with Shared Inputs

Reshef Meir
The School of Engineering
and Computer Science
The Hebrew University of Jerusalem
Jerusalem, Israel
reshef24@cs.huji.ac.il

Ariel D. Procaccia
Microsoft Israel R&D Center
Herzeliya, Israel
arielpro@gmail.com

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

Abstract

Strategyproof classification deals with a setting where a decision-maker must classify a set of input points with binary labels, while minimizing the expected error. The labels of the input points are reported by self-interested agents, who might lie in order to obtain a classifier that more closely matches their own labels, thus creating a bias in the data; this motivates the design of *truthful* mechanisms that discourage false reports. Previous work [Meir *et al.*, 2008] investigated both decision-theoretic and learning-theoretic variations of the setting, but only considered classifiers that belong to a degenerate class.

In this paper we assume that the agents are interested in a *shared* set of input points. We show that this plausible assumption leads to powerful results. In particular, we demonstrate that variations of a truthful random dictator mechanism can guarantee approximately optimal outcomes with respect to *any* class of classifiers.

1 Introduction

There are many settings in which one has to make a decision based on empirical information arriving from multiple sources. When the data sources are rational agents that are affected by the final decision, the agents may act in a strategic, non-cooperative manner in an attempt to increase their own utility at the expense of the social good.

We assume that the final decision assumes the form of a *binary classifier*, which assigns a positive or negative label to each point of the input space. The choice of classifier may be limited, due to external constraints, to a fixed class of classifiers that we refer to as the *concept class*, e.g., linear separators over the input space.

We consider two interrelated settings. The first setting is *decision-theoretic*; a decision must be made based on data reported by multiple self-interested agents. The agents are concerned with the binary labels of a set of input points. The utility of an agent with respect to a given decision (i.e., a given classifier) is the number of points on which the label provided by the classifier agrees with the agent's own label. The goal of the decision-maker is to choose a classifier that maximizes the social welfare—the sum of utilities.

The second setting is *learning-theoretic*, a variation of the standard Supervised Classification problem. Samples are drawn from an unknown distribution over the input space, and are then labeled by experts. A classification mechanism receives the sampled data as input, and outputs a classifier. Unlike the standard setting in machine learning (but similarly to our first setting), the experts are assumed to be self-interested agents, and may lie in order to increase their utility.

In both settings the decision-maker (or mechanism, or learning algorithm) aims to find a classifier that classifies the available data as well as possible. However, the agents may misreport their labels in an attempt to influence the final decision in their favor. The result of a decision-making process based on such biased data may be completely unexpected and difficult to analyze. A truthful learning mechanism eliminates any such bias and allows the decision-maker to select a classifier that best fits the reported data, without having to take into account the hidden interests of the agents.

Previous work on strategyproof classification. The foregoing model of strategyproof classification was recently presented by Meir et al. [2008]. Their paper can be seen as only a preliminary step towards an understanding of incentives in classification, as they investigate a degenerate concept class that is restricted to exactly two classifiers: the one that classifies *all* the points as positive, and the one that classifies *all* the points as negative. Put another way, the decision-maker has only two possible decisions. In contrast, in most classification settings the concept class—the set of decisions that can be made—is far richer.

The assumption of shared inputs. Our main conceptual contribution in this paper is the assumption of *shared inputs*. In the decision-theoretic setting, this means that the agents share the same set of input points, and only disagree on the labels of these points. In the learning-theoretic setting, the assumption implies that the agents are interested in a common distribution over the input space, but, once again, differ with respect to the labels.

The model of Meir et al. [2008] did not address the issue of shared inputs. However, as the two possible classifiers were constant, the identity of the input points (i.e., their location) was irrelevant—only their labels mattered. Hence, the setting of Meir et al. is in fact a very special case of our setting, even though we assume shared inputs. Furthermore, this assumption allows us to obtain inclusive results with respect to *any concept class*. We feel that in many environments the requirement of shared inputs is satisfied; below we give one such example.

Shared inputs: A motivating example. Let us consider the following example which involves learning in a non-cooperative environment under the shared input assumption. A large organization is trying to fight the congestion in an internal email system by designing a smart spam filter. In order to train the system, managers are asked to review the last 1000 emails sent to the "all employees" mailing list (hence, shared inputs) and classify them as either "work-related" (positive label) or "spam" (negative label). Whereas the managers will likely agree on the classification of some of the messages (e.g., "Buy Viagra now!!!" or "Christmas Bonus for all employees"), it is likely that others (e.g., "Joe from the Sales department goes on a lunch break") would not be unanimously classified. Moreover, as each manager would like to filter most of what he sees as spam, a manager might try to compensate for the "mistakes" of his colleagues by misreporting his real opinion in some cases. For example, the manager of the R&D department, believing that about 90% of the Sales messages are utterly unimportant, might classify *all* of them as spam in order to reduce the congestion. The manager of Sales, suspecting the general opinion of her department, might do the exact opposite to prevent her emails from being filtered.

Overview of our results. As in [Meir *et al.*, 2008], we wish to design classification mechanisms that achieve a good outcome in the face of strategic behavior. By "good outcome" we mean that the output of the mechanism provides an approximation of the optimal solution. We would also like our mechanisms to be *strategyproof* (SP), that is, the agents must not be able to benefit from lying.

We begin by investigating mechanisms for the decision-theoretic setting (Section 2). We first show that, even under the shared input assumption, SP deterministic mechanisms cannot guarantee a sublinear approximation ratio. We then consider randomized mechanisms in the weighted case, in which the decision mechanism may value some agents more than others. Surprisingly, and in contrast to the above, we show that choosing a dictator at random according to agents' weights provides an approximation ratio of three in expectation. If all weights are equal, then the approximation is shown to be slightly better. We emphasize that these results hold with respect to any concept class.

In the learning-theoretic setting (Section 3), designing strategyproof mechanisms is virtually impossible, since there is an additional element of randomness introduced by sampling the input space. We therefore relax the strategyproofness requirements, and instead investigate each of two incomparable strategic assumptions: that agents do not lie if they cannot gain more than ϵ, and that agents always use a dominant strategy if one exists with respect to a specific sample. We show that under either assumption, our randomized mechanism of Section 2 can be run directly on sampled data, while maintaining a bounded expected error. Our theorems give a connection between the number of samples and the expected error of the mechanism.

An important remark is that in the strategyproof classification setting, standard economic money-based mechanisms such as the Vickrey-Clarke-Groves mechanism (see, e.g., [Nisan, 2007]) can be used to obtain good results. However, this setting admits strategyproof mechanisms that do well *even without assuming money*. Achieving our goals without resorting to payments is highly desirable, since often payments cannot be made due to legal or ethical considerations. Moreover, in internet environments payments are notoriously difficult to implement, due to banking and security issues. Hence, we consider approximation mechanisms that do not require payments.

Due to their length, most proofs are omitted, but can be found online in [Meir, 2008, Chapter 5].

Related work. Apart from [Meir *et al.*, 2008], which was discussed above, the work most closely related to ours is the paper by Dekel et al. [2008]. Their work focused on regression learning, where the labels are real numbers and one is interested in the *distances* between the mechanism's outputs and the labels. Except for this very significant difference, the settings that we study and our goals are very similar to theirs. Dekel et al. provided upper and lower bounds on the approximation ratio achieved by supervised regression mechanisms in this model. Notably, some of our bounds resemble the bounds in their regression setting. Moreover, similar intuitions sometimes apply to both settings, although it seems the results of one setting cannot be analytically mapped to the other. Dekel et al. also concentrate on mechanisms without payments, but their results hold only with respect to very specific function classes (as they do not assume shared inputs; see, e.g., Theorems 4.1 and 4.2 of [Dekel *et al.*, 2008]).

Another rather closely related work has results of a negative flavor. Perote and Perote-Peña [2003] put forward a model of unsupervised *clustering*, where each agent controls a single point in $\mathbb{R}^2$ (i.e., its reported location). A clustering mechanism aggregates these locations and outputs a partition and a set of centroids. They show that if every agent wants to be close to some centroid, then under very weak restrictions on the clustering mechanism there *always* exists a beneficial manipulation, that is, there are no reasonable (deterministic) clustering mechanisms that are SP. The same authors have also investigated linear regression in a strategic setting [Perote and Perote-Peña, 2004].

There is a significant body of work on learning in the face of noise, where the noise can be either random or adversarial (see, e.g., [Bshouty *et al.*, 2002; Dalvi *et al.*, 2004]). However, in that research the goal is to do well in the face of noise, rather than provide incentives in a way that prevents the dataset from being manipulated in the first place.

For additional relevant references, the reader is encour-

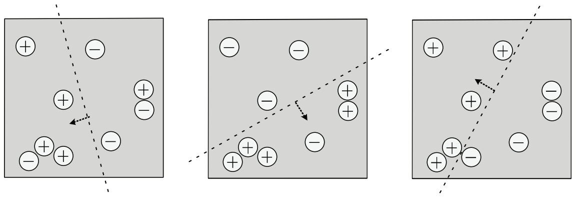

Figure 1: An instance with shared inputs. Here, $\mathcal{X} = \mathbb{R}^2$, $\mathcal{C}$ is the class of linear separators over $\mathbb{R}^2$, and $n = 3$. The input points X of all three agents are identical, but the labels, i.e., their types, are different. The best classifier from $\mathcal{C}$ with respect to each S_i is also shown (the arrow marks the positive halfspace of the separator). Only the rightmost dataset is realizable.

aged to consult previous papers on strategyproof learning settings [Dekel *et al.*, 2008; Meir *et al.*, 2008].

2 The Decision-Theoretic Setting

In this section we analyze our decision-theoretic setting, where the dataset is fixed and no generalization takes place. We start by introducing this section's model and notations.

Let $\mathcal{X}$ be an input space, which we assume to be either a finite set or some subset of $\mathbb{R}^d$. A *classifier* or *concept* c is a function $c : \mathcal{X} \to \{+, -\}$ from the input space to the *labels* $\{+, -\}$. A *concept class* $\mathcal{C}$ is a set of such concepts. For example, the class of linear separators over $\mathbb{R}^d$ is the set of concepts that are defined by the parameters $\mathbf{a} \in \mathbb{R}^d$ and $b \in \mathbb{R}$, and map a point $\mathbf{x} \in \mathbb{R}^d$ to $+$ if and only if $\mathbf{a}{\cdot}\mathbf{x}{+}b \geq 0$.

Denote the set of *agents* by $I = \{1, \ldots, n\}$, $n \geq 2$. The agents are interested in a (finite) set of m input points $X \in \mathcal{X}^m$. In this paper we assume that X is *shared* among the agents, that is, all the agents are equally interested in each input point in X. This plausible assumption, as we shall see, allows us to obtain surprisingly strong results. Naturally, the points in X are common knowledge.

Each agent has a private *type*: its labels for the points in X. Specifically, agent $i \in I$ holds a function $Y_i : X \to \{+, -\}$, which maps every point $x \in X$ to the label $Y_i(x)$ that i attributes to x. Each agent $i \in I$ is also assigned a *weight* w_i, which reflects its relative importance; by normalizing the weights we can assume that $\sum_{i \in I} w_i = 1$. Let

$$S_i = \{\langle x, Y_i(x)\rangle \ : \ x \in X\}$$

be the partial *dataset* of agent i, and let $S = \langle S_1, \ldots, S_n\rangle$ denote the complete *dataset*. S_i is said to be *realizable* w.r.t. a concept class $\mathcal{C}$ if there is $c \in \mathcal{C}$ which perfectly separates the positive samples from the negative ones. If S_i is realizable for all $i \in I$, then S is said to be *individually realizable*. Figure 1 shows an example of a dataset with a shared set of points X.

We use the common 0-1 loss function (also employed by Meir et al. [2008]) to measure the error. The *risk*, or negative utility, of agent $i \in I$ with respect to a concept c is simply the relative number of errors that c makes on its dataset. Formally,

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

where $[\![A]\!]$ denotes the indicator function of the boolean expression A. Note that S_i is realizable if and only if $\min_{c \in \mathcal{C}} \mathrm{R}_i(c, S) = 0$. The *global risk* is defined as

$$\begin{aligned} \mathrm{R}_I(c, S) &= \sum_{i \in I} w_i \cdot \mathrm{R}_i(c, S) \\ &= \frac{1}{m}\sum_{i \in I}\sum_{x \in X} w_i \cdot [\![c(x) \neq Y_i(x)]\!] \ . \end{aligned}$$

A *deterministic mechanism* $\mathcal{M}$ receives as input a dataset S (and the weights of the agents), and outputs a classifier $c \in \mathcal{C}$. Note 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* returns a random variable $\hat{c}$ taken from $\mathcal{C}$, and we are interested in the *expected risk*. Formally,

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

and the global risk is defined analogously.

We measure the quality of the outcome of a mechanism using the notion of *approximation*. A mechanism is said to be an *α-approximation* mechanism if for every dataset S,

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

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

We emphasize that the real labels of the input points are private information, and an agent may report different labels than the ones indicated by Y_i. We denote by $\overline{Y}_i : X \to \{+, -\}$ the reported labels of agent i. We also denote by $\overline{S}_i = \{\langle x, \overline{Y}_i(x)\rangle \ : \ x \in X\}$ the reported partial dataset of agent i, and by $\overline{S} = \langle \overline{S}_1, \ldots, \overline{S}_n\rangle$ the reported dataset.

Strategyproofness implies that reporting the truthful types is a dominant strategy for all 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 $\overline{S}_i$,

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

We remark that for randomized mechanisms, this is strategyproofness *in expectation*. Interestingly, this notion of strategyproofness is sufficient for our lower bounds, but our upper bounds also hold with respect to strategyproofness *in dominant strategies*, that is, an agent cannot gain from lying regardless of the random outcome of the mechanism.

Notice that we do not allow mechanisms to make payments. Since we are essentially interested in maximizing the social welfare, an optimal truthful mechanism can be obtained using VCG payments (see, e.g., [Nisan, 2007]). However, achieving strategyproofness without payments is far more desirable (for the reasons outlined in the introduction). Therefore, we adopt the approach of previous work on strategyproof learning [Dekel *et al.*, 2008; Meir *et al.*, 2008], and sacrifice the optimality of the solution in order to achieve strategyproofness *without payments*.

2.1 Deterministic Mechanisms

We start by examining an extremely simple deterministic mechanism. However, despite its simplicity, its analysis is nontrivial.

For any dataset S, we define by $\text{ERM}(S) \in \mathcal{C}$ the *Empirical Risk Minimizer* of S (following the conventions of the learning theory literature), i.e., the concept that achieves OPT, the minimum risk on S. Formally,

$$\text{ERM}(S) = \text{argmin}_{c \in \mathcal{C}} \sum_{\langle x,y \rangle \in S} [\![c(x) \neq y]\!] \ .$$

Our mechanism simply lets the heaviest agent dictate which concept is chosen.

Mechanism 1 (Heaviest Dictator). *Let $h \in I$ be a heaviest agent, $h \in argmax_{i \in I} w_i$. Return ERM$(S_h)$.*

If more than one ERM exists, return one of them arbitrarily. The mechanism is clearly SP: the heaviest dictator h has no interest in lying, since its best concept is selected; all other agents are simply ignored, and therefore have no reason to lie either. We have the following result.

Theorem 2.1. *Let $|I| = n$. For every concept class $\mathcal{C}$, Mechanism 1 is an SP $(2n-1)$-approximation mechanism.*

Unfortunately, this bound is tight, i.e., there is an example in which the ratio is exactly $2n-1$. An approximation ratio that increases linearly with the number of agents is not very appealing. However, it turns out that using deterministic mechanisms we cannot do better with respect to every concept class, as the following theorem states.

Theorem 2.2. *Let $|I| = n$. There exist concept classes for which any deterministic SP mechanism has an approximation ratio of at least $\Omega(n)$, even if all the weights are equal.*

The proof of this theorem is quite nontrivial. The key ingredient is an application of the Gibbard-Satterthwaite impossibility theorem (see, e.g., [Nisan, 2007]), but since the theorem requires that agents be able to report any ranking of the alternatives, an elaborate mapping between our setting and the theorem's setting is required.

Theorem 2.2 implies that Mechanism 1 is asymptotically optimal as a generic mechanism that applies to any concept class. However, for specific concept classes one can obviously do much better. For example, recall that the paper of Meir et al. [2008] focuses on the concept class $\mathcal{C} = \{c_+, c_-\}$, which contains only the constant positive concept and the constant negative concept. With respect to this concept class, Meir et al. provided a deterministic SP 3-approximation mechanism (and also showed that this bound is tight).

2.2 Randomized Mechanisms

In order to break the lower bound given by Theorem 2.2, we employ a simple randomization. Strikingly, we will see that this randomization yields a constant approximation ratio *with respect to any concept class* (under our assumption of shared inputs, of course).

Mechanism 2 (Weighted Random Dictator). *For each $i \in I$, select agent i with probability w_i. Return ERM(S_i).*

This mechanism is clearly SP. Our main results are the following two theorems.

Theorem 2.3. *For every concept class $\mathcal{C}$, Mechanism 2 is an SP 3-approximation mechanism. Moreover, if S is individually realizable, then 2-approximation is guaranteed.*

We give a very rough proof sketch, which in particular assumes that the set of input points X does not contain two copies of an input point $x \in \mathcal{X}$, but this assumption can be relaxed. For the detailed proof, see [Meir, 2008].

Proof sketch of Theorem 2.3. Let X be a fixed set of input points, and let H the set of all functions $h : X \to \{-,+\}$. We define the *distance* between two functions $h, h' \in H$ as the number of input points that they label differently; formally:

$$d(h, h') = \frac{1}{m} \sum_{x \in X} [\![h(x) \neq h'(x)]\!] \ .$$

For every $c \in \mathcal{C}$, there is a single function $h_c \in H$ such that $\forall x \in X\ (h_c(x) = c(x))$. For simplicity, we slightly abuse notation by using c instead of h_c. We denote by c_i the best concept with respect to agent i, i.e., $c_i = \text{ERM}(S_i)$. We also denote $\text{ERM}(S)$ by c^*.

We show that d is reflexive, non-negative, symmetric and satisfies the triangle inequality. Further, the following properties also hold for the distance d:

$$\forall c \in \mathcal{C}, \forall i \in I\ (d(Y_i, c) = \text{R}_i(c, S)) \ . \tag{1}$$

In particular, it follows that $d(Y_i, c_i) = 0$ if S_i is realizable, and that $\forall i, j \in I\ (d(c_i, Y_j) = \text{R}_j(c_i, S))$.

$$\forall i \in I\ \left(c_i = \text{argmin}_{c \in \mathcal{C}} d(c, Y_i)\right) \ . \tag{2}$$

$$\sum_{i \in I} w_i \text{R}_I(c_i, S) = \sum_i \sum_j w_i w_j d(c_i, Y_j) \ . \tag{3}$$

$$\sum_i \sum_j w_i w_j d(Y_i, Y_j) \leq 2 \cdot \text{OPT} \ . \tag{4}$$

Using these properties, we analyze the risk of the mechanism, which randomizes the dictator:

$$\begin{aligned} \text{R}_I(\mathcal{M}(S), S) &= \sum_{i \in I} w_i \text{R}_I(c_i, S) = \sum_i \sum_j w_i w_j d(Y_i, c_j) \\ &\leq \sum_i \sum_j w_i w_j (d(Y_i, Y_j) + d(Y_j, c_j)) \ . \end{aligned}$$

In the individually realizable case, the second term equals 0, and hence

$$\text{R}_I(\mathcal{M}(S), S) \leq \sum_i \sum_j w_i w_j d(Y_i, Y_j) \leq 2 \cdot \text{OPT} \ .$$

Otherwise,

$$\begin{aligned} \text{R}_I(\mathcal{M}(S), S) &\leq \sum_i \sum_j w_i w_j (d(Y_i, Y_j) + d(Y_j, c^*)) \\ &= \sum_i \sum_j w_i w_j d(Y_i, Y_j) + \sum_j w_j d(Y_j, c^*) \sum_i w_i \\ &\leq 2 \cdot \text{OPT} + \sum_j w_j d(Y_j, c^*) \\ &= 2 \cdot \text{OPT} + \sum_j w_j \text{R}_j(c^*, S) \\ &= 2 \cdot \text{OPT} + \text{R}_I(c^*, S) = 3 \cdot \text{OPT} \ . \end{aligned}$$ □

Theorem 2.4. *Let $|I| = n$, and assume all agents have equal weights. For every concept class $\mathcal{C}$, Mechanism 2 is an SP $(3 - \frac{2}{n})$-approximation mechanism ($2 - \frac{2}{n}$ when S is individually realizable).*

The proof is similar to the proof of Theorem 2.3, but is somewhat more involved, since a careful analysis is required to tighten the bound in Equation (4). Similar intuition also accounts for Theorem 2.1.

It is possible to show that the analysis of Mechanism 2 is tight. Indeed, for every concept class of size at least two, there is an example where the approximation ratio yielded by the mechanism is exactly $3 - \frac{2}{n}$, even when the agents have equal weights. If weights are allowed, for every $\epsilon > 0$ an example that provides a lower bound of $3 - \epsilon$ can be constructed using only two agents.

It is natural to ask whether better SP mechanisms exist. For specific concept classes, the answer to this question is positive. For example, Meir et al. [2008] designed a randomized SP 2-approximation mechanism for the concept class $\mathcal{C} = \{c_+, c_-\}$. They further showed the following theorem.

Theorem 2.5 (Meir, Procaccia and Rosenschein [2008]). *For all $\epsilon > 0$, there are no randomized SP $(2 - \epsilon)$-approximation mechanisms for $\mathcal{C} = \{c_+, c_-\}$, even if there are only 2 agents with equal weight.*

This negative result can be easily extended to *any* concept class of size at least two (even with shared inputs). We conclude that for any nontrivial concept class $\mathcal{C}$ and for any dataset with shared inputs S, the worst-case approximation ratio of the best randomized SP mechanism has to lie between 2 and 3. The exact value may depend on the characteristics of $\mathcal{C}$ and S.

3 The Learning-Theoretic Setting

In this section we leverage the upper bounds that were attained in the decision-theoretic setting to obtain results in a machine-learning framework. That is, we present a learning mechanism that guarantees a constant approximation of the optimal risk in expectation, even in the face of strategic behavior.

In contrast to the previous setting where the input was a fixed dataset, in instances of the learning problem the type of an agent $i \in I$ is defined by a function $Y_i : \mathcal{X} \to \{+, -\}$ that assigns a label to *every point* of the input space.[1] Reinterpreting our shared input assumption in the learning-theoretic setting, we assume that all agents have *the same* probability distribution $\mathcal{D}$ over $\mathcal{X}$, which reflects the relative importance that the agents attribute to different input points; the distribution $\mathcal{D}$ is common knowledge.

Let us now redefine the notion of risk. The risk of a concept is computed with respect to $\mathcal{D}$, as the *expected* relative number of errors. Specifically,

$$\mathrm{R}_i(c) = \mathbb{E}_{x \sim \mathcal{D}} \left[[\![c(x) \neq Y_i(x)]\!] \right] \ ,$$

and

$$\mathrm{R}_I(c) = \sum_{i \in I} w_i \mathrm{R}_i(c) \ .$$

Following the standard assumption in machine learning, we have no direct access to $\mathcal{D}$, nor can agents report the function Y_i; our mechanisms can only sample from $\mathcal{D}$ and ask the agents for their labels. Put another way, whereas in Section 2 we had a set of shared inputs X, in our current setting this shared set of inputs is *sampled* from $\mathcal{D}$.

Our goal is, once again, to design mechanisms with low risk. However, constructing an SP mechanism that learns from sampled data is nearly impossible (see [Dekel *et al.*, 2008; Meir *et al.*, 2008] for further discussion). Hence, we weaken the SP requirement, and analyze the performance of our mechanisms under each of the following two assumptions.

1. *The ϵ-truthfulness assumption*: Agents do not lie if they gain at most ϵ from lying.
2. *The rationality assumption*: Agents will always use a strategy that is guaranteed to minimize their risk in situations where such a strategy exists.

The former approach was taken by Dekel et al. [2008], whereas a variation on the latter approach was adopted by Meir et al. [2008]. Notice that the two assumptions are *incomparable*. The latter assumption may seem to be weaker than the former, but the latter assumption implies that an agent will definitely lie if this proves beneficial. Hence, we study both assumptions under our setting of shared inputs.

3.1 The ϵ-Truthfulness Assumption

An ϵ-strategyproof mechanism is one where agents cannot gain more than ϵ by lying. We show below that, similarly to Dekel et al. [2008], the results of Section 2 can be employed to obtain a mechanism that is "usually" ϵ-strategyproof. We focus on the following mechanism.

Mechanism 3.

1. *Sample m input points i.i.d. from $\mathcal{D}$ (denote the sampled points by X).*
2. *Ask each agent $i \in I$ to label X according to Y_i; this produces a dataset S.*
3. *Run Mechanism 2 on S (using given weights), and return the output.*

We wish to formulate a theorem that asserts that, given enough samples, the expected risk of Mechanism 3 is relatively small under the ϵ-truthfulness assumption. The exact number of samples needed depends on the combinatorial richness of the function class; this is usually measured using some notion of dimension, such as the VC dimension (see, e.g., [Kearns and Vazirani, 1994]). For instance, the VC dimension of the class of linear separators over $\mathbb{R}^d$ is $d + 1$. We do not dwell on this point too much, and instead assume that the dimension is bounded.

Theorem 3.1. *Let $|I| = n$, and let $\mathcal{C}$ be a concept class with bounded dimension. Let $\epsilon > 0$, and assume that agents are truthful when they cannot gain more than ϵ by lying. Then*

[1] Our results also hold in a more general model, where agents have distributions over the labels, but we use this simpler formulation for ease of exposition.

given any distribution $\mathcal{D}$, the expected risk of Mechanism 3 is at most $3 \cdot OPT + \epsilon$, where

$$OPT = \inf_{c \in \mathcal{C}} R_I(c) \ .$$

Under these assumptions, the number of samples required by the mechanism is polynomial in $\frac{1}{\epsilon}$ and $\log n$.

The expectation is taken over the randomness of sampling and the randomness of Mechanism 2. In order to prove the theorem, one must establish a result in the spirit of Theorem 5.1 of [Dekel *et al.*, 2008]: given δ and enough samples (whose number m also depends on δ), with probability at least $1 - \delta$ *none of the agents* can gain more than ϵ by lying, and also (assuming the agents are truthful) the mechanism yields an approximation ratio close to three. The theorem then follows by taking a small enough δ.

3.2 The Rationality Assumption

We presently state an alternative assumption regarding the strategic behavior of the agents. Consider a mechanism that samples a set of input points X and then executes a mechanism $\mathcal{M}$ on the labeled dataset (e.g., Mechanism 3). Informally, we assume that each agent uses a dominant strategy, if one exists. We emphasize that although a dominant strategy is a specific labeling of the dataset, it minimizes the agent's private risk with respect to the *entire distribution*, rather than the number of errors on its sampled dataset. We make no assumptions regarding the agent's action in the case where there are no dominant strategies.

More formally, our rationality assumption states the following: for each agent $i \in I$, if there is a labeling $\overline{Y}_i$ of X, such that for any S_{-i}, $\mathrm{R}_i(\mathcal{M}(\overline{S}_i, S_{-i}))$ is minimized (where $\overline{S}_i$ is X labeled by $\overline{Y}_i$), then agent i would report $\overline{Y}_i$. We once again consider the performance of Mechanism 3.

Theorem 3.2. *Let $|I| = n$, and let $\mathcal{C}$ be a concept class with bounded dimension. Let $\epsilon > 0$, and assume that agents always use a dominant strategy when one exists. Then given any distribution $\mathcal{D}$, the expected risk of Mechanism 3 is at most $3 \cdot OPT + \epsilon$, where*

$$OPT = \min_{c \in \mathcal{C}} R_I(c) \ .$$

Under these assumptions, the number of samples required by the mechanism is polynomial (only) in $\frac{1}{\epsilon}$.

Interestingly, the alternative assumption improved the sample complexity: the number of required samples no longer depends on n, only on $\frac{1}{\epsilon}$. In a somewhat counter-intuitive way, the rationality assumption provides us with better bounds without using the notion of truthfulness at all. This can be explained by the fact that a *rational* (i.e., self-interested) labeling of the dataset is a better proxy to an agent's real type than a truthful labeling. Indeed, this strange claim is true since the sampling process might produce a set of points X that represents the agent's distribution in an inaccurate way.[2]

4 Discussion

The focus of this paper has been the design of strategyproof mechanisms that yield an approximation ratio, without allowing payments. This approach is part of an emerging agenda which we call *approximate mechanism design without money* [Procaccia and Tennenholtz, 2009], and stands in contrast to most existing work on algorithmic mechanism design, where payments are ubiquitous. The second author and colleagues are currently working on several other instances of this agenda.

There are two main avenues for expanding our understanding of strategyproof classification. One is to drop the assumption of shared inputs and observe how it affects the general case and particular concept classes (e.g., linear separators), and we are already taking some steps in this direction. The other is to investigate alternative formulations of the strategyproof classification setting, such as different loss functions.

5 Acknowledgment

This research was supported in part by Israel Science Foundation grant #898/05.

References

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

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

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

[Kearns and Vazirani, 1994] M. J. Kearns and U. V. Vazirani. *An Introduction to Computational Learning Theory*. MIT Press, 1994.

[Meir *et al.*, 2008] 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.

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

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

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

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

[Procaccia and Tennenholtz, 2009] A. D. Procaccia and M. Tennenholtz. Approximate mechanism design without money. In *Proc. of 10th ACM-EC*, 2009. To appear.

[2] Note that the revelation principle does not apply here, since the agents do not report their full preferences.