Incentive Compatible Regression Learning (Extended Abstract)

Ofer Dekel* Felix Fischer† Ariel D. Procaccia‡

Abstract

We initiate the study of incentives in a general machine learning framework. We focus on a game-theoretic regression learning setting where private information is elicited from multiple agents, which are interested in different distributions over the sample space. This conflict potentially gives rise to untruthfulness on the part of the agents. In the restricted but important case when distributions are degenerate, and under mild assumptions, we show that agents are motivated to tell the truth. In a more general setting, we study the power and limitations of mechanisms without payments. We finally establish that, in the general setting, the VCG mechanism goes a long way in guaranteeing truthfulness and efficiency.

Keywords: machine learning; regression; algorithmic mechanism design

*School of Computer Science and Engineering, The Hebrew University of Jerusalem, Jerusalem 91904, Israel, email: oferd@cs.huji.ac.il

†Institut für Informatik, Ludwig-Maximilians-Universität München, 80538 München, Germany, email: fischerf@tcs.ifi.lmu.de. The work was done while the author was visiting the School of Computer Science and Engineering, The Hebrew University of Jerusalem. This visit was supported by the School of Computer Science and Engineering and the Leibniz Center for Research in Computer Science at The Hebrew University of Jerusalem, and by the Deutsche Forschungsgemeinschaft under grant BR 2312/3-1.

‡School of Computer Science and Engineering, The Hebrew University of Jerusalem, Jerusalem 91904, Israel, email: arielpro@cs.huji.ac.il

Dagstuhl Seminar Proceedings 07271
Computational Social Systems and the Internet
http://drops.dagstuhl.de/opus/volltexte/2007/1162

1 Introduction

Machine learning is the area of computer science concerned with the design and analysis of algorithms that can learn from experience. A learning algorithm observes a training set of labeled examples, and attempts to learn a prediction rule that accurately predicts the labels of new examples. Following the rise of the Internet as a computational platform, machine learning problems have become increasingly dispersed, in the sense that different parts of the training data may be controlled by different computational or economic entities.

Motivation Consider an Internet search company trying to improve the performance of their search engine by learning a ranking function from examples. The ranking function is the heart of a modern search engine, and can be thought of as a mapping that assigns a real-valued score to every pair of a query and a URL. Some of the large Internet search companies currently hire Internet users (which we hereinafter refer to as "experts") to manually rank such pairs. These rankings may then be pooled and used to train a ranking function. Moreover, the experts are chosen in a way such that averaging over the experts' opinions and interests presumably pleases the average Internet user.

However, different experts may have different interests and a different idea of the results a good search engine should return. For instance, take the ambiguous query "Jaguar", which has become folklore in search engine designer circles. The top answer given by most search engines for this query is the website of the luxury car manufacturer. Knowing this, an animal-loving expert may decide to give this pair a disproportionately low score, hoping to improve the relative rank of websites dedicated to the Panthera Onca. An expert who is an automobile enthusiast may counter this measure by giving automotive websites a higher-than-appropriate score. From the search company's perspective, this type of strategic manipulation introduces an undesired bias in the training data.

Setting and Goals Our problem setting falls within the general boundaries of statistical regression learning. *Regression learning* is the task of constructing a real-valued function f based on a training set of examples, where each example consists of an input to the function and its corresponding output. In particular, the example $(\mathbf{x}, y)$ suggests that $f(\mathbf{x})$ should be equal to y. The accuracy of a function f on a given input-output pair $(\mathbf{x}, y)$ is defined by a loss function ℓ. Popular choices of the loss function are the squared loss, $\ell(f(\mathbf{x}), y) = (f(\mathbf{x}) - y)^2$, and the absolute loss, $\ell(f(\mathbf{x}), y) = |f(\mathbf{x}) - y|$. We typically assume that the training set is obtained by sampling i.i.d. from an underlying distribution over the product space of inputs and outputs. The overall quality of the function constructed by the learning algorithm is defined to be its expected loss, with respect to this same distribution.

We augment this well-studied setting by introducing a set of *strategic agents*. Each agent holds as private information an individual distribution over the set of examples, and measures the quality of a regression function with respect to this distribution. The global goal, on the other hand, is to do well with respect to the average of the individual distributions. A training set is obtained by eliciting private information from the agents, who may reveal this information untruthfully in order to favorably influence the result of the learning process.

Mechanism design is a subfield of economics that is concerned with the question of how to incentivize agents to truthfully report their private information, also known as their *type*. Given potentially non-truthful reports from the agents, a mechanism determines a global solution, and possibly additional monetary transfers to and from the agents. A mechanism is said to be *incentive compatible* if it is always in the agents' best interest to report their true types, and *efficient* if the solution maximizes social welfare, *i.e.*, minimizes the overall loss. Our goal in this paper will be to design and analyze incentive compatible and efficient mechanisms for the regression learning setting.

Results We begin our investigation by considering a restricted setting where each agent is only interested in a single point of the input space. Quite surprisingly, it turns out that a specific choice of ℓ, namely the absolute loss function, leads to excellent game-theoretic properties: the algorithm which simply finds a empirical risk minimizer on the training set is group incentive compatible, *i.e.*, no coalition of agents is motivated to lie. All of our incentive compatibility results are obtained with respect to dominant strategies: truthfulness holds regardless of the other agents' actions. In a sense, this is the strongest incentive compatibility result that could possibly be obtained. We also show that even basic truthfulness cannot be obtained for a wide range of other loss functions, including the popular squared loss.

In the more general case, where agents are interested in non-degenerate distributions, achieving incentive compatibility requires more sophisticated mechanisms. We show that the well-known VCG mechanism does very well: with probability $1 - \delta$, no agent can gain more than ϵ by lying, where both ϵ and δ can be made arbitrarily small by increasing the size of the training set. This result holds for any choice of loss function ℓ.

We also study what happens when payments are disallowed. In this setting, we obtain limited positive results for the absolute loss function and for restricted yet interesting function classes. In particular, we present a mechanism which is approximately *group* incentive compatible as above and 3-efficient in the sense that the solution provides a 3-approximation to social welfare. We complement these results with a matching lower bound and provide strong evidence that no approximately incentive compatible and approximately efficient mechanism exists for more expressive functions classes.

Related Work To the best of our knowledge, this paper is the first to study incentives in a general machine learning framework. Previous work has focused on the related problem of learning in the presence of inconsistent and noisy training data, where the noise can be either random (Littlestone, 1991; Goldman and Sloan, 1995) or adversarial (Kearns and Li, 1993; Bshouty et al., 2002). Barreno et al. (2006) consider a specific situation where machine learning is used as a component of a computer security system, and account for the possibility that the training data is subject to a strategic attack intended to infiltrate the secured system. In contrast to these approaches, we do not attempt to design algorithms that can tolerate noise, but instead focus on designing algorithms that discourage the strategic addition of noise.

Closely related to our work is the area of *algorithmic mechanism design*, introduced in the seminal work of Nisan and Ronen (1999). Algorithmic mechanism design studies algorithmic problems in a game-theoretic setting where the different participants cannot be assumed to follow the algorithm but rather act in a selfish way. It has turned out that the main challenge of algorithmic mechanism design is the inherent incompatibility of generic truthful mechanisms with approximation schemes for hard algorithmic problems. As a consequence, most of the current work in algorithmic mechanism design focuses on dedicated mechanisms for hard problems (see, *e.g.*, Lehmann et al., 1999; Archer et al., 2003; Dobzinski et al., 2006). What distinguishes our setting from that of algorithmic mechanism design is the need for *generalization* to achieve globally satisfactory results on the basis of a small number of samples. Due to the dynamic and uncertain nature of the domain, inputs are usually assumed to be drawn from some underlying fixed distribution. The goal then is to design algorithms that, with high probability, perform well on samples drawn from the same distribution.

More distantly related to our work is research which applies machine learning techniques in game theory and mechanism design. Balcan et al. (2005), for instance, use techniques from sample complexity to reduce mechanism design problems to standard algorithmic problems. Another line of research puts forward that machine learning can be used to predict consumer behavior, or find an efficient description for collective decision making. Works along this line studied the learnability of choice sets (Kalai, 2003) and of social choice functions (Procaccia et al., 2006, 2007).

Structure of the Paper In the following section, we give a general exposition of regression learning and introduce our model of regression learning with multiple agents. We then examine three levels of generality: in Section 3, we discuss the setting where the distribution of each agent puts all of the weight on a single point of the sample space; in Section 4, we consider a more general case, where the distribution of each agent is a discrete distribution supported on a finite set of points; we finally investigate arbitrary distributions in Section 5. In Section 6, we discuss our results and give some directions for future research.

2 The Model

In this section we formalize the regression learning problem described in the introduction and cast it in the framework of game theory. Some of the definitions are illustrated by relating them to the Internet search example presented in the previous section. We focus on the task of learning a real-valued function over an *input space* $\mathcal{X}$. In the Internet search example, $\mathcal{X}$ would be the set of all query-URL pairs, and our task would be to learn the ranking function of a search engine. Let $N = \{1, \ldots, n\}$ be a set of agents, which in our running example would be the set of all experts. For each agent $i \in N$, let o_i be a function from $\mathcal{X}$ to $\mathbb{R}$ and let ρ_i be a probability distribution over $\mathcal{X}$. Intuitively, o_i is what agent i thinks to be the correct real-valued function, while ρ_i captures the relative importance that agent i gives to the different regions of $\mathcal{X}$. In the Internet search example, o_i would be the optimal ranking function according to agent i, and ρ_i would be a distribution over query-URL pairs that assigns higher weight to queries from that agent's areas of interest.

Let $\mathcal{F}$ be a class of functions, where every $f \in \mathcal{F}$ is a function from $\mathcal{X}$ to the real line. We call $\mathcal{F}$ the *hypothesis space* of our problem, and restrict the output of the learning algorithm to functions in $\mathcal{F}$. We evaluate the accuracy of each $f \in \mathcal{F}$ using the *loss function* $\ell : \mathbb{R} \times \mathbb{R} \to \mathbb{R}_+$. For any concrete input-output pair $(\mathbf{x}, y)$, we interpret $\ell(f(\mathbf{x}), y)$ as the penalty associated with predicting the output value $f(\mathbf{x})$ when the true output is known to be y. As mentioned in the introduction, common choices of ℓ are the squared loss, $\ell(\alpha, \beta) = (\alpha - \beta)^2$, and the absolute loss, $\ell(\alpha, \beta) = |\alpha - \beta|$. The accuracy of a hypothesis $f \in \mathcal{F}$ is defined to be the average loss of f over the entire input space. Formally, define the *risk* associated by agent i with the function f as

$$R_i(f) \;=\; \mathbb{E}_{\mathbf{x} \sim \rho_i}\big[\ell(f(\mathbf{x}), o_i(x))\big] \;.$$

Clearly, this subjective definition of hypothesis accuracy allows for different agents to have significantly different valuations of different functions in $\mathcal{F}$, and it is quite possible that we will not be able to please all of the agents simultaneously. Instead, our goal is to satisfy the agents in N on average. Define J to be a random variable distributed uniformly over the elements of N. Now define the *global risk* of of a function f to be the average risk with respect to all of the agents, namely

$$R_N(f) \;=\; \mathbb{E}_J\left[R_J(f)\right] \;.$$

We are now ready to define our formal learning-theoretic goal: we would like to find a hypothesis in $\mathcal{F}$ that attains a global risk as close as possible to $\inf_{f \in \mathcal{F}} R_N(F)$.

Even if N is small, we still have no explicit way of calculating $R_N(f)$. Instead, we use an empirical estimate of the risk as a proxy to the risk itself. For each $i \in N$, we randomly sample m points independently from the distribution ρ_i and request their respective labels from agent i. In this way, we obtain the labeled training set $\tilde{S}_i = \{(\mathbf{x}_{i,j}, \tilde{y}_{i,j})\}_{j=1}^m$. Agent i may label the points in $\tilde{S}_i$ however he sees fit, and we therefore say that agent i *controls* (the labels of) these points. We usually denote agent i's "true" training set by $S_i = \{(\mathbf{x}_{i,j}, y_{i,j})\}_{j=1}^m$, where $y_{ij} = o_i(x_{ij})$. After receiving labels from all agents in N, we define the *global training set* to be the multiset $\tilde{S} = \biguplus_{i \in N} \tilde{S}_i$.

The elicited training set $\tilde{S}$ is presented to a regression learning algorithm, which in return constructs a *hypothesis* $\tilde{f} \in \mathcal{F}$. Each agent can influence $\tilde{f}$ by modifying the labels he controls. This observation brings

us to the game-theoretic aspect of our setting. For all $i \in N$, agent i's private information is a vector of true labels $y_{ij} = o_i(x_{ij})$, $j = 1, \ldots, m$; the sampled points x_{ij}, $j = 1, \ldots, m$, are exogenously given and assumed to be common knowledge. The *strategy space* of each agent then consists of all possible *values* for the labels he controls. In other words, each agent reports a labeled training set $\tilde{S}_i$. We sometimes use $\tilde{S}_{-i}$ as a shorthand for $\tilde{S} \setminus \tilde{S}_i$, the strategy profile of all agents except agent i. The space of possible outcomes is the hypothesis space $\mathcal{F}$, and the utility of agent i for an outcome $\tilde{f}$ is determined by his risk $R_i(\tilde{f})$. More precisely, agent i chooses $\tilde{y}_{i,1}, \ldots, \tilde{y}_{i,m}$ so as to minimize $R_i(f)$. We follow the usual game-theoretic assumption that he does this with full knowledge of the inner workings of our regression learning algorithm. We name this game the *learning game*.

One of the simplest and most popular regression learning techniques is *empirical risk minimization* (ERM). The *empirical risk* associated with a hypothesis f, with respect to a sample S, is denoted by $\hat{R}(f, S)$ and defined to be the average loss attained by f on the examples in S, *i.e.*,

$$\hat{R}(f, S) = \frac{1}{|S|} \sum_{(\mathbf{x}, y) \in S} \ell(f(\mathbf{x}), y) \ .$$

An ERM algorithm finds the empirical risk minimizer $\hat{f}$ within $\mathcal{F}$. In other words, the ERM algorithm calculates

$$\hat{f} \ = \ \operatorname*{argmin}_{f \in \mathcal{F}} \ \hat{R}(f, S) \ .$$

For some choices of loss function and hypothesis class, it may occur that the global minimizer of the empirical risk is not unique, and we must define an appropriate tie-breaking mechanism. A large part of this paper will be dedicated to ERM algorithms.

Since our strategy is to use $\hat{R}(f, \tilde{S})$ as a surrogate for $R_N(f)$, we need $\hat{R}(f, \tilde{S})$ to be an unbiased estimator of $R_N(f)$. A particular situation in which this can be achieved is when all agents $i \in N$ truthfully report $\tilde{y}_{i,j} = o_i(\mathbf{x}_{i,j})$ for all j. Another argument for truthfulness regards the quality of the overall solution and can be obtained by a variation of the well-known revelation principle. Assume that for a given mechanism and given true inputs there is an equilibrium in which some agents report their inputs untruthfully, and which leads to an outcome that is strictly better than any outcome achievable by an incentive compatible mechanism. Then we can design a new mechanism that, given the true inputs, simulates the agents' lies and yields the exact same output in equilibrium. To summarize, truthful mechanisms will allow us to obtain an unbiased estimator of the true risk, and this need not come at the expense of the overall solution quality.

3 Degenerate Distributions

We begin our study by focusing on a special case, where each agent is only interested in a single point of the sample space. Even this simple setting has interesting applications. For example, consider the problem of allocating tasks among service providers, *e.g.*, messages to routers, jobs to remote processors, or reservations of bandwidth to Internet providers. A provider's private information is its capacity; machine learning techniques are used to obtain a global picture of providers' capabilities. Analysis of regression learning is appropriate in this context, as each provider is interested in an allocation which is as close as possible to its capacity: more tasks mean more money, but an overload is clearly undesirable.

Formally, the distribution ρ_i of agent i is degenerate, and the sample S_i becomes a singleton. Let $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ denote the truthful set of input-output pairs, where now $y_i = o_i(\mathbf{x}_i)$ and $S_i = \{(\mathbf{x}_i, y_i)\}$ is the single example controlled by agent i. Each agent selects an output value $\tilde{y}_i$, and the reported (possibly untruthful) training set $\tilde{S} = \{(\mathbf{x}_i, \tilde{y}_i)\}_{i=1}^n$ is presented to a regression learning algorithm. The algorithm constructs a hypothesis $\tilde{f}$ and each agent i's cost is the loss

$$R_i(\tilde{f}) \ = \ \mathbb{E}_{\mathbf{x} \sim \rho_i}[\ell(\tilde{f}(\mathbf{x}), o_i(\mathbf{x}))] = \ell(\tilde{f}(\mathbf{x}_i), y_i)$$

on the point he controls, where ℓ is a predefined loss function. Within this setting, we examine the game-theoretic properties of ERM algorithms.

As noted above, an ERM algorithm takes as input a loss function ℓ and a training set S, and outputs the hypothesis that minimizes the empirical risk over S according to ℓ. Throughout this section, we write $\hat{f} = \texttt{ERM}(\mathcal{F}, \ell, S)$ as shorthand for $\arg\min_{f \in \mathcal{F}} \hat{R}(f, \ell, S)$. We restrict our discussion to loss functions of the form $\ell(\alpha, \beta) = \mu(|\alpha - \beta|)$, where $\mu : \mathbb{R}_+ \to \mathbb{R}$ is a monotonically increasing convex function, and to the case where $\mathcal{F}$ is a convex set of functions. These assumptions enable us to cast the ERM algorithm as a convex optimization problem, which typically has an efficient solution. Most choices of ℓ and $\mathcal{F}$ that do not satisfy the above constraints may not allow for efficient learnability, and are therefore less interesting.

We prove two main theorems: First, we show that if μ is a linear function, then the ERM algorithm is group incentive compatible. Second, we prove that if μ grows faster than any linear function and if $\mathcal{F}$ contains more than one function, then ERM is not incentive compatible.

3.1 ERM with Absolute Loss

In this subsection, we focus on the absolute loss function. Indeed, let ℓ denote the absolute loss, $\ell(a, b) = |a - b|$, and let $\mathcal{F}$ be a convex hypothesis class. Because ℓ is only weakly convex, there may be multiple hypotheses in $\mathcal{F}$ that globally minimize the empirical risk and we must add a tie-breaking step to our ERM algorithm. Concretely, consider the following two-step procedure:

1. Empirical risk minimization: calculate $r = \min_{f \in \mathcal{F}} \hat{R}(f, S)$.
2. Tie-breaking: return $\tilde{f} = \operatorname*{argmin}_{f \in \mathcal{F} : \hat{R}(f,S) = r} \|f\| \quad \left(\text{where } \|f\|^2 = \int f^2(\mathbf{x})\, d\mathbf{x}\right)$.

Our assumption that $\mathcal{F}$ is a convex set implies that the set of empirical risk minimizers $\{f \in \mathcal{F} : \hat{R}(f, S) = r\}$ is also convex. The function $\|f\|$ is a strictly convex function and therefore the output of the tie-breaking step is uniquely defined. In our analysis, we only use the fact that $\|f\|$ is a strictly convex function of f. Any other strictly convex function can be used in its place in the tie-breaking step.

The following theorem states that, when the absolute loss function is used, the ERM algorithm has excellent game-theoretic properties. More precisely, the algorithm is *group incentive compatible*: for any possible combination of lies by the members of an arbitrary coalition, and regardless of the strategies of the other agents, at least one of the agents in the coalition does not gain. The proof of the theorem will be given in the full version of this paper.

Theorem 3.1. *Let N be a set of agents, $S = \uplus_{i \in N} S_i$ a dataset such that $S_i = \{\mathbf{x}_i, y_i\}$ for all $i \in N$, and let ρ_i be degenerate at $\mathbf{x}_i$. Let ℓ denote the absolute loss, $\ell(a, b) = |a - b|$, and let $\mathcal{F}$ be a convex hypothesis class. Then, the ERM algorithm that minimizes ℓ over $\mathcal{F}$ with respect to S is group incentive compatible.*

3.2 ERM with Other Convex Loss Functions

We have shown that performing ERM with absolute loss is incentive compatible. We now show that this is not the case for most other convex loss functions. Specifically, we examine loss functions of the form $\ell(\alpha, \beta) = \mu(|\alpha - \beta|)$, where $\mu : \mathbb{R}_+ \to \mathbb{R}$ is a monotonically increasing strictly convex function with unbounded subderivatives. Put another way, μ is any monotonically increasing strictly convex function that cannot be bounded from above by any linear function.

For example, μ can be the function $\mu(\alpha) = \alpha^d$, where d is a real number strictly greater than 1. A popular choice is $d = 2$, which induces the squared loss, $\ell(\alpha, \beta) = (\alpha - \beta)^2$. It is straightforward to construct an example showing that ERM with squared loss is not incentive compatible. An example for a function that does not fall within our definition is given by $\nu(\alpha) = \ln(1 + \exp(\alpha))$, which is both monotonic and strictly

convex, but its derivative is bounded from above by 1. Our definition uses the subderivatives of μ, rather than its derivatives, since we do not require μ to be differentiable.

For every $\mathbf{x} \in \mathcal{X}$, let $\mathcal{F}(\mathbf{x})$ denote the *set of feasible values* of $\mathbf{x}$, formally defined as $\mathcal{F}(\mathbf{x}) = \{f(\mathbf{x}) : f \in \mathcal{F}\}$. Since $\mathcal{F}$ is a convex set, it follows that $\mathcal{F}(\mathbf{x})$ is either an interval on the real line, a ray, or the entire real line. Similarly, for a multiset $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \in \mathcal{X}^n$, denote

$$\mathcal{F}(X) = \{\langle f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n)\rangle : f \in \mathcal{F}\} \subseteq \mathbb{R}^n \ .$$

We then say that $\mathcal{F}$ is *full* on a multiset $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \in \mathcal{X}^n$ if $\mathcal{F}(X) = \mathcal{F}(\mathbf{x}_1) \times \cdots \times \mathcal{F}(\mathbf{x}_n)$. Clearly, requiring that $\mathcal{F}$ is not full on X is a necessary condition for the existence of a training set with points X where one of the agents gains by lying. Otherwise, whatever the agents' values, they always get the best thing possible, with respect to $\mathcal{F}$, from the ERM hypothesis. Note, for example, that if $|\mathcal{F}| \geq 2$ there is some point $\mathbf{x}_0 \in \mathcal{X}$ such that $f_1(\mathbf{x}_0) \neq f_2(\mathbf{x}_0)$; in this case, $\mathcal{F}$ is not full on any X which contains $\mathbf{x}_0$ twice.

In addition, if $\mathcal{F}$ contained only one function, any algorithm would trivially be incentive compatible irrespective of the loss function. In the following theorem we therefore consider hypothesis classes $\mathcal{F}$ of size at least two which are *not* full on the set X of points of the training set. The proof of the theorem will be given in the full version of this paper.

Theorem 3.2. *Let $\mu : \mathbb{R}_+ \to \mathbb{R}$ be a monotonically increasing strictly convex function with unbounded subderivatives, and define the loss function $\ell(\alpha, \beta) = \mu(|\alpha - \beta|)$. Let $\mathcal{F}$ be a convex hypothesis class that contains at least two functions, and let $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \in \mathcal{X}^n$ be a multiset such that $\mathcal{F}$ is not full on X. Then there exist $y_1, \ldots, y_n \in \mathbb{R}$ such that, if $S = \uplus_{i \in N} S_i$ with $S_i = \{(\mathbf{x}_i, y_i)\}$, ρ_i is degenerate at $\mathbf{x}_i$, and the ERM algorithm is used, there is an agent who has an incentive to lie.*

It is natural to ask what happens for loss functions that are sublinear in the sense that they cannot be bounded from below by any linear function with strictly positive derivative. A characteristic of such loss functions, and the reason why they are rarely used in practice, is that the set of empirical risk minimizers need no longer be convex. It is thus unclear how tie-breaking should be defined in order to find a unique empirical risk minimizer. Furthermore, ERM with sublinear loss is not in general incentive compatible.

4 Uniform Distribution Over the Sample

We now turn to settings where a single agent holds a (possibly) nondegenerate distribution over the input space. However, we still do not wish to jump to the full level of generality. Rather, we will concentrate on a setting where for each agent i, ρ_i is the uniform distribution over the points x_{ij}, $j = 1, \ldots, m$. While this setting is equivalent to curve fitting with multiple agents and may be interesting in its own right, we primarily engage in this sort of analysis as a stepping stone in our quest to understand the learning game in its entire generality. The results in this section will thus function as building blocks for the theorems of Section 5.

Since each agent has a uniform distribution over his sample, we can simply assume that each agent's cost is his average empirical loss on the sample, $\hat{R}(\tilde{f}, S_i) = \frac{1}{m} \sum_{j=1}^{m} \ell(\tilde{f}(\mathbf{x}_{ij}), y_{ij})$. The designer's goal is to minimize $\hat{R}(\tilde{f}, S)$. We stress at this point that the results in this section also hold if the agents' samples differ in size (this is of course true for the negative results, but also for the positive results).

As we move to this intermediate level of generality, truthfulness of ERM immediately becomes a thorny issue even under absolute loss. Indeed, the following example indicates that more sophisticated mechanisms must be used to achieve incentive compatibility.

Example 4.1. Let $\mathcal{F}$ be the class of constant functions over $\mathbb{R}^k$, and let $N = \{1, 2\}$; assume the absolute loss function is used. Let $S_1 = \{(1, 1), (2, 1), (3, 0)\}$ and $S_2 = \{(4, 0), (5, 0), (6, 1)\}$. The global empirical

risk minimizer (according to our tie-breaking rule) is the constant function $f_1(x) \equiv 0$ with $\hat{R}(f_1, S_1) = 2/3$. However, if agent 1 declares $\tilde{S}_1 = \{(1,1),(2,1),(3,1)\}$, then the empirical risk minimizer becomes $f_2(x) \equiv 1$, which is the optimal fit for agent 1 since $\hat{R}(f_2, S_1) = 1/3$.

4.1 Mechanisms with Payments

One possibility to overcome the issue manifested in Example 4.1 is to consider mechanisms that not only return an allocation, but can also transfer payments to and from the agents based on the inputs they provide. A famous example for such a payment rule is the Vickrey-Clarke-Groves (VCG) Mechanism (Vickrey, 1961; Clarke, 1971; Groves, 1973). This mechanism starts from an efficient allocation , and computes each agent's payment according to the utility of the other agents, thus aligning the individual interests of each agent with that of society.

In our setting, where social welfare equals the total empirical risk, ERM is a function which maximizes social welfare and can therefore be directly augmented with VCG payments. Given an outcome $\hat{f}$, each agent i has to pay an amount of $\hat{R}(\hat{f}, \tilde{S}_{-i})$. In turn, the agent can receive some amount $h_i(\tilde{S}_{-i})$ that does *not* depend on the values he has reported (but possibly on the values reported by the other agents). It is well known (Groves, 1973), and also easily verified, that this family of mechanisms is incentive compatible: no agent is motivated to lie regardless of the others agents' actions. Furthermore, this result holds for any loss function, and may thus be an excellent solution for some settings.

In many other settings, however, especially in the world of the Internet, transferring payments to and from users can pose serious problems, up to the extent that it might become completely infeasible. The practicality of VCG payments in particular has recently been disputed for various other reasons (Rothkopf, 2007). Perhaps most relevant to our work is the fact that VCG mechanisms are in general not group incentive compatible. It is therefore worthwhile to explore which results can be obtained when payments are disallowed. This will be the subject of the following subsection.

4.2 Mechanisms without Payments

In this subsection, we henceforth restrict ourselves to the absolute loss function. When the ERM algorithm is used, and for the special case covered in Section 3, this function was shown to possess properties far superior to any other loss function. This encourages the ambition that similar results (*i.e.*, group incentive compatibility) can be obtained in our current setting (uniform distributions over the samples), even when payments are disallowed. This does not necessarily mean that good mechanisms (without payments) cannot be designed for other loss functions, even in the more general setting of this section. We leave the study of such mechanisms for future work.

ERM is *efficient* in the sense that it minimizes the overall loss, and maximizes social welfare. In light of Example 4.1, we shall now sacrifice efficiency for incentive compatibility. More precisely, we seek incentive compatible mechanisms which are *approximately efficient* in the sense that for all samples S, the ratio $\hat{R}(f, S)/\hat{R}(\hat{f}, S)$ between the empirical risk of the solution f returned by the mechanism and that of the optimal solution $\hat{f}$ is bounded. We say that a regression learning mechanism is α-efficient if for all S, $\hat{R}(f, S)/\hat{R}(\hat{f}, S) \leq \alpha$. We should stress that the reason we resort to approximation is not to achieve computational tractability, but to achieve incentive compatibility without payments (like we had in Section 3).

Example 4.1, despite its simplicity, is surprisingly robust against many conceivably truthful mechanisms. The reader may have noticed, however, that the values of the agents in this example are not "individually realizable": in particular, there is no constant function which *realizes* agent 1's values, *i.e.*, fits them with a loss of zero. In fact, agent 1 benefits from revealing values which are consistent with his individual empirical risk minimizer. This insight leads us to design a simple but useful mechanism, which

we will term "project-and-fit".

Input: A hypothesis class $\mathcal{F}$ and a sample $S = \uplus S_i$, $S_i \subseteq \mathcal{X} \times \mathbb{R}$
Output: A function $f \in \mathcal{F}$.
Mechanism:

1. For each $i \in N$, let $f_i = \texttt{ERM}(\mathcal{F}, S_i)$.
2. Define $\tilde{S}_i = \{(\mathbf{x}_{i1}, f_i(x_{i1})), \ldots, (\mathbf{x}_{im}, f_i(x_{im}))\}$.
3. Return $f = \texttt{ERM}(\tilde{S})$, where $\tilde{S} = \uplus_{i=1}^{n} \tilde{S}_i$.

To put it differently, this mechanism projects each agent's values to his individual empirical risk minimizer, and then returns the global empirical risk minimizer based on these values. Notice that, at least with respect to Example 4.1, this mechanism is group incentive compatible.

More generally, it can be shown that the mechanism is group incentive compatible when $\mathcal{F}$ is the class of constant functions over $\mathbb{R}^k$ or the class of homogenous linear functions over $\mathbb{R}$, *i.e.*, functions of the form $f(\mathbf{x}) = \mathbf{a} \cdot \mathbf{x}$. The class of homogeneous linear functions, in particular, is very significant in machine learning, for instance in the context of Support Vector Machines (Shawe-Taylor and Cristianini, 2000). Quite surprisingly, project-and-fit is not only truthful but also provides a constant efficiency ratio in these cases. The proof of the following theorem will be given in the full version of this paper.

Theorem 4.2. *Assume that $\mathcal{F}$ is the class of constant functions over $\mathbb{R}^k$, $k \in \mathbb{N}$ or the class of homogeneous linear functions over $\mathbb{R}$. Then project-and-fit is group incentive compatible and* 3*-efficient.*

A simple example shows that the 3-efficiency analysis given in the proof is tight. We generalize this observation by proving that, for the class of constant or homogeneous linear functions and irrespective of the dimension of $\mathcal{X}$, no (individually) truthful mechanism without payments can achieve an efficiency ratio better than 3. It should be noted that this lower bound holds for any choice of points x_{ij}. The proof of the theorem will be given in the full version of this paper.

Theorem 4.3. *Let $\mathcal{F}$ be the class of constant functions over $\mathbb{R}^k$ or the class of homogeneous linear functions over $\mathbb{R}^k$, $k \in \mathbb{N}$. Then there exists* no *incentive compatible mechanism without payments that is $(3 - \epsilon)$-efficient for any $\epsilon > 0$, even when $|N| = 2$.*

Let us recapitulate. We have found a truthful 3-efficient mechanism for the class of constant functions over $\mathbb{R}^k$ and for the class of homogeneous linear functions over $\mathbb{R}$. We have also seen that this result cannot be improved upon for these classes. The lower bound also holds for multi-dimensional homogeneous linear functions. It is natural to inquire at this point if the same mechanism is also incentive compatible when considering more complex hypothesis classes, such as homogeneous linear functions over $\mathbb{R}^k$, $k \geq 2$, or linear functions. The answer to this question is negative.

Is there some other mechanism which deals with more complex hypothesis classes and provides a truthful approximation? We believe that the answer is negative even for homogeneous linear functions over $\mathbb{R}^k$ for $k \geq 2$. The following conjecture formalizes this statement.

Conjecture 4.4. Let $\mathcal{F}$ be the class of homogeneous linear functions over $\mathbb{R}^k$, $k \geq 2$, and assume that $m = |S_i| \geq 3$. Then any incentive compatible and surjective mechanism is a dictatorship.

Conceivably, dictatorship would be a decent solution if it could guarantee an approximately efficient solution. Unfortunately this is not the case.

5 Generalization

In Section 4 we established several positive results in the setting where each agent cares about a uniform distribution on his portion of the global training set. In this section we extend these results to the general regression learning setting outlined in the Introduction. More formally, the extent to which agent $i \in N$ cares about each point in $\mathcal{X}$ will now be defined by the distribution function ρ_i, and agent i controls the labels of a finite set of points sampled according to ρ_i. Our strategy in this section will consist of two steps. First, we want to show that under standard assumptions on the hypothesis class $\mathcal{F}$ and the number m of samples, each agent's empirical risk on the training set S_i reflects his real risk according to ρ_i. Second, we intend to establish that, as a consequence, our incentive compatibility results are not significantly weakened when moving to the general setting.

Abstractly, let $\mathcal{D}$ be a probability distribution and let $\mathcal{G}$ be a class of real-valued functions, bounded in $[0, C]$. We would like to prove that for any $\epsilon > 0$ and $\delta > 0$ there exists m such that, if $X_1, \ldots, X_m$ are sampled i.i.d. according to $\mathcal{D}$,

$$\Pr\left(\text{for all } g \in \mathcal{G},\ \left|\mathbb{E}_{X\sim\mathcal{D}}[g(X)] \ -\ \frac{1}{m}\sum_{i=1}^{m} g(X_i)\right| \ \leq\ \epsilon\right) \ \geq\ 1-\delta\ . \tag{1}$$

To prove this bound, we use standard *uniform convergence of empirical means* arguments. A specific technique is to show that the hypothesis class $\mathcal{G}$ has bounded complexity. The complexity of $\mathcal{G}$ can be measured in various different ways, for example using the pseudo-dimension (Pollard, 1984; Haussler, 1992), an extension of the well-known VC-dimension to real valued hypothesis classes, or the Rademacher complexity (Bartlett and Mendelson, 2003). If the pseudo-dimension of $\mathcal{G}$ is bounded by a constant, or if the Rademacher complexity of $\mathcal{G}$ with respect to an m-point sample is $O(\sqrt{m})$, then there indeed exists m such that Equation (1) holds.

More formally, assume that the hypothesis class $\mathcal{F}$ has bounded complexity, choose $\epsilon > 0$, $\delta > 0$, and consider a sample S_i of size $m = \Theta(\log(1/\delta)/\epsilon^2)$ drawn i.i.d. from the distribution ρ_i of any agent $i \in N$. Then we have that

$$\Pr\left(\text{for all } f \in \mathcal{F}\ \ \left|R_i(f) - \hat{R}(f, S_i)\right| \ \leq\ \epsilon\right) \ \geq\ 1-\delta\ . \tag{2}$$

In particular, we want the events in Equation (2) to hold simultaneously for all $i \in N$, *i.e.*,

$$\text{for all } f \in \mathcal{F},\ \ \left|R_N(f) - \hat{R}(f, S)\right| \ \leq\ \epsilon\ . \tag{3}$$

Using the union bound, this holds with probability at least $1 - n\delta$.

We now turn to incentive compatibility. The following theorem implies that mechanisms which do well in the setting of Section 4 are also good, but slight less so, when arbitrary distributions are allowed. Specifically, given a training set satisfying Equation (2) for all agents, a mechanism that is incentive compatible in the setting of Section 4 becomes ϵ-incentive compatible , *i.e.*, no agent can gain more than ϵ by lying, no matter what the other agents do. Analogously, a group incentive compatible mechanism for the setting of Section 4 becomes ϵ-group incentive compatible, *i.e.*, there exists an agent in the coalition that does not gain more than ϵ. Furthermore, efficiency is preserved up to an additive factor of ϵ. We wish to point out that ϵ-equilibrium is a well-established solution concept; the underlying assumption is that agents wouldn't bother to lie if they were to gain an amount as small as ϵ. This concept is particularly appealing when one recalls that ϵ can be chosen to be arbitrarily small. The proof of this theorem will be given in the full version of this paper.

Theorem 5.1. *Let $\mathcal{F}$ be a hypothesis class, ℓ some loss function, and $S = \uplus S_i$ a dataset such that for all $f \in \mathcal{F}$ and $i \in N$, $|R_i(f) - \hat{R}(f, S_i)| \leq \epsilon/2$, and $|R_N(f) - \hat{R}(f, S)| \leq \epsilon/2$. Let M be a mechanism with or without payments.*

1. *If M is incentive compatible (group incentive compatible, respectively) under the assumption that each agent's cost is $\hat{R}(\tilde{f}, S_i)$, then M is ϵ-incentive compatible (ϵ-group incentive compatible, respectively) in the general regression setting.*

2. *If M is α-efficient under the assumption that the designer's goal is to minimize $\hat{R}(\tilde{f}, S)$, $M(S) = \tilde{f}$, then $R_N(\tilde{f}) \leq \alpha \cdot argmin_{f \in \mathcal{F}} R_N(f) + \epsilon$.*

The connection between the properties of $\mathcal{F}$ and the size of m needed to achieve the conditions of Theorem 5.1 is well-studied in the machine learning literature, and we will therefore not fully elaborate this question. As a corollary of the above discussion, the conditions of this theorem are satisfied with probability $1 - \delta$ when $\mathcal{F}$ has bounded dimension and $m = \Theta(\log(1/\delta)/\epsilon)$. As the latter expression depends logarithmically on $1/\delta$, the sample size only needs to be increased by a factor of $\log n$ to achieve the stronger requirement of Equation (3).

Let us examine how Theorem 5.1 applies to our positive results. Since ERM with VCG payments is incentive compatible and efficient under uniform distributions on the samples, ERM with VCG payments is ϵ-incentive compatible in general and we obtain efficiency up to an additive factor of ϵ when it is used in the learning game. This holds for any loss function ℓ. The project-and-fit mechanism is ϵ-group incentive compatible in the learning game when $\mathcal{F}$ is the class of constant functions or of homogeneous linear functions over $\mathbb{R}$, and is 3-efficient up to an additive factor of ϵ. This is true only for the absolute loss function.

6 Discussion

In this paper, we have studied mechanisms for a general regression learning framework involving multiple strategic agents. In the case where each agent controls one point, we have obtained a strong and surprising characterization of the truthfulness of ERM. When the absolute loss function is used, ERM is group incentive compatible. On the other hand, ERM is not incentive compatible for any loss function that is superlinear in a certain well-defined way. This particularly holds for the popular squared loss function. In the general learning setting, we have established the following result: For any $\epsilon, \delta > 0$, given a large enough training set, and with probability $1 - \delta$, ERM with VCG payments is efficient up to an additive factor of ϵ, and ϵ-incentive compatible. We have also obtained limited positive results for the case when payments are disallowed, namely an algorithm that is ϵ-group incentive compatible and 3-efficient (up to an additive factor of ϵ) for constant functions over $\mathbb{R}^k$, $k \in \mathbb{N}$, and for homogeneous linear functions over $\mathbb{R}$. We also gave a matching lower bound, which also applies to multidimensional homogeneous linear functions. The number of samples required by the aforementioned algorithms depends on the combinatorial richness of the hypothesis space $\mathcal{F}$, but differs only by a factor of $\log n$ from that in the traditional regression learning setting without strategic agents. Since $\mathcal{F}$ can be assumed to be learnable in general, this factor is not very significant.

Since at the moment there is virtually no other work on incentives in machine learning, many exciting directions for future work exist. While regression learning constitutes an important area of machine learning with numerous applications, adapting our framework for studying incentives in classification or in unsupervised settings will certainly prove interesting as well. In classification, functions label points in the input space as either positive or negative. It is readily appreciated that ERM is trivially incentive compatible in classification when each agent controls only a single point. The situation again becomes complicated when agents control multiple points. In addition, we have not considered settings where ERM is computationally intractable. Just like in general algorithmic mechanism design, VCG is bound to fail. It is an open question whether one can simultaneously achieve computational tractability, approximate efficiency, and (approximate) incentive compatibility. Several interesting questions follow directly from our work. The one we are most interested in is settling Conjecture 4.4: are there incentive compatible and approximately efficient

mechanisms without payments for homogeneous linear functions? Do such mechanisms exist for other interesting hypothesis classes? These questions are closely related to general questions about the existence of incentive compatible and non-dictatorial mechanisms, and are interesting well beyond the scope of machine learning and computer science.

Acknowledgements

We would like to thank Bezalel Peleg for discussions regarding Conjecture 4.4.

References

A. Archer, C. Papadimitriou, K. Talwar, and E. Tardos. An approximate truthful mechanism for combinatorial auctions with single parameter agents. In *Proceedings of the 14th Annual ACM Symposium on Discrete Algorithms (SODA)*, 2003.

M.-F. Balcan, A. Blum, J. D. Hartline, and Y. Mansour. Mechanism design via machine learning. In *Proceedings of the 46th Symposium on Foundations of Computer Science (FOCS)*, pages 605–614. IEEE Computer Society Press, 2005.

M. Barreno, B. Nelson, R. Sears, A. D. Joseph, and J. D. Tygar. Can machine learning be secure? In *Proceedings of the 2006 ACM Symposium on Information, Computer and Communications Security*, pages 16–25, 2006.

P. L. Bartlett and S. Mendelson. Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2003.

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

E. H. Clarke. Multipart pricing of public goods. *Public Choice*, 11:17–33, 1971.

S. Dobzinski, N. Nisan, and M. Schapira. Truthful randomized mechanisms for combinatorial auctions. In *Proceedings of the 38th ACM Symposium on Theory of Computing*, 2006.

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

T. Groves. Incentives in teams. *Econometrica*, 41:617–631, 1973.

D. Haussler. Decision theoretic generalization of the pac model for neural net and other learning applications. *Information and Computation*, 100(1):78–150, 1992.

G. Kalai. Learnability and rationality of choice. *Journal of Economic Theory*, 113(1):104–117, 2003.

M. Kearns and M. Li. Learning in the presence of malicious errors. *SIAM Journal on Computing*, 22(4): 807–837, 1993.

D. Lehmann, L. I. O'Callaghan, and Y. Shoham. Truth revelation in rapid, approximately efficient combinatorial auctions. In *Proceedings of the 1st ACM conference on electronic commerce*, 1999.

N. Littlestone. Redundant noisy attributes, attribute errors, and linear-threshold learning using Winnow. In *Proceedings of the Fourth Annual Workshop on Computational Learning Theory (COLT)*, pages 147–156, 1991.

N. Nisan and A. Ronen. Algorithmic mechanism design. In *Proceedings of the 31st Annual ACM Symposium on the Theory of Computing (STOC)*, pages 129–140. ACM Press, 1999.

D. Pollard. *Convergence of Stochastic Processes*. Springer-Verlag, 1984.

A. D. Procaccia, A. Zohar, and J. S. Rosenschein. Automated design of voting rules by learning from examples. In *Proceedings of the 1st International Workshop on Computational Social Choice (COMSOC)*, pages 436–449, 2006.

A. D. Procaccia, A. Zohar, Y. Peleg, and J. S. Rosenschein. Learning voting trees. In *Proceedings of the 22nd Conference on Artificial Intelligence (AAAI)*, 2007. To appear.

M. Rothkopf. Thirteen reasons the Vickrey-Clarke-Groves process is not practical. *Operations Research*, 55(2):191–197, 2007.

J. Shawe-Taylor and N. Cristianini. *Support Vector Machines and other Kernel Based Learning Methods*. Cambridge University Press, 2000.

W. Vickrey. Counter speculation, auctions, and competitive sealed tenders. *Journal of Finance*, 16(1):8–37, 1961.