Adversarial Regression with Multiple Learners

Liang Tong [*1] **Sixie Yu** [*1] **Scott Alfeld** [2] **Yevgeniy Vorobeychik** [1]

Abstract

Despite the considerable success enjoyed by machine learning techniques in practice, numerous studies demonstrated that many approaches are vulnerable to attacks. An important class of such attacks involves adversaries changing features at test time to cause incorrect predictions. Previous investigations of this problem pit a single learner against an adversary. However, in many situations an adversary's decision is aimed at a collection of learners, rather than specifically targeted at each independently. We study the problem of adversarial linear regression with multiple learners. We approximate the resulting game by exhibiting an upper bound on learner loss functions, and show that the resulting game has a unique symmetric equilibrium. We present an algorithm for computing this equilibrium, and show through extensive experiments that equilibrium models are significantly more robust than conventional regularized linear regression.

1. Introduction

Increasing use of machine learning in adversarial settings has motivated a series of efforts investigating the extent to which learning approaches can be subverted by malicious parties. An important class of such attacks involves adversaries changing their behaviors, or features of the environment, to effect an incorrect prediction. Most previous efforts study this problem as an interaction between a single learner and a single attacker (Brückner & Scheffer, 2011; Dalvi et al., 2004; Li & Vorobeychik, 2014; Zhou et al., 2012). However, in reality attackers often target a broad array of potential victim organizations. For example, they craft generic spam templates and generic malware, and then disseminate these widely to maximize impact. The resulting ecology of attack targets reflects not a single learner, but many such learners, all making autonomous decisions about how to detect malicious content, although these decisions often rely on similar training datasets.

We model the resulting game as an interaction between multiple learners, who simultaneously learn linear regression models, and an attacker, who observes the learned models (as in white-box attacks (Šrndic & Laskov, 2014)), and modifies the original feature vectors at test time in order to induce incorrect predictions. Crucially, rather than customizing the attack to each learner (as in typical models), the attacker chooses a single attack for *all* learners. We term the resulting game a *Multi-Learner Stackelberg Game*, to allude to its two stages, with learners jointly acting as Stackelberg leaders, and the attacker being the follower. Our first contribution is the formal model of this game. Our second contribution is to approximate this game by deriving upper bounds on the learner loss functions. The resulting approximation yields a game in which there always exists a symmetric equilibrium, and this equilibrium is unique. In addition, we prove that this unique equilibrium can be computed by solving a convex optimization problem. Our third contribution is to show that the equilibrium of the approximate game is robust, both theoretically (by showing it to be equivalent to a particular robust optimization problem), and through extensive experiments, which demonstrate it to be much more robust to attacks than standard regularization approaches.

Related Work Both attacks on and defenses of machine learning approaches have been studied within the literature on *adversarial machine learning* (Brückner & Scheffer, 2011; Dalvi et al., 2004; Li & Vorobeychik, 2014; Zhou et al., 2012; Lowd & Meek, 2005). These approaches commonly assume a single learner, and consider either the problem of finding evasions against a fixed model (Dalvi et al., 2004; Lowd & Meek, 2005; Šrndic & Laskov, 2014), or algorithmic approaches for making learning more robust to attacks (Russu et al., 2016; Brückner & Scheffer, 2011; Dalvi et al., 2004; Li & Vorobeychik, 2014; 2015). Most of these efforts deal specifically with classification learning, but several consider adversarial tampering with regression models (Alfeld et al., 2016; Grosshans et al., 2013), although still within a single-learner and single-attacker framework. Stevens & Lowd (2013) study the algorithmic problem of

[*]Equal contribution [1]Department of EECS, Vanderbilt University, Nashville, TN, USA [2]Computer Science Department, Amherst College, Amherst, MA, USA. Correspondence to: Yevgeniy Vorobeychik <eug.vorobey@gmail.com>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018.

attacking multiple linear classifiers, but did not consider the associated game among classifiers.

Our work also has a connection to the literature on security games with multiple defenders (Laszka et al., 2016; Smith et al., 2017; Vorobeychik et al., 2011). The key distinction with our paper is that in multi-learner games, the learner strategy space is the space of possible models in a given model class, whereas prior research has focused on significantly simpler strategies (such as protecting a finite collection of attack targets).

2. Model

We investigate the interactions between a collection of learners $\mathcal{N} = \{1, 2, ..., n\}$ and an attacker in regression problems, modeled as a *Multi-Learner Stackelberg Game (MLSG)*. At the high level, this game involves two stages: first, all learners choose (train) their models from data, and second, the attacker transforms test data (such as features of the environment, at prediction time) to achieve malicious goals. Below, we first formalize the model of the learners and the attacker, and then formally describe the full game.

2.1. Modeling the Players

At training time, a set of training data $(\mathbf{X}, \mathbf{y})$ is drawn from an unknown distribution $\mathcal{D}$. $\mathbf{X} \in \mathbb{R}^{m\times d}$ is the training sample and $\mathbf{y} \in \mathbb{R}^{m\times 1}$ is a vector of values of each data in $\mathbf{X}$. We let $\mathbf{x}_j \in \mathbb{R}^{d\times 1}$ denote the jth instance in the training sample, associated with a corresponding value $y_j \in \mathbb{R}$ from $\mathbf{y}$. Hence, $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_m]^\top$ and $\mathbf{y} = [y_1, y_2, ..., y_m]^\top$. On the other hand, test data can be generated either from $\mathcal{D}$, the same distribution as the training data, or from $\mathcal{D}^{'}$, a modification of $\mathcal{D}$ generated by an attacker. The nature of such malicious modifications is described below. We let β $(0 \leq \beta \leq 1)$ represent the probability that a test instance is drawn from $\mathcal{D}^{'}$ (i.e., the malicious distribution), and $1-\beta$ be the probability that it is generated from $\mathcal{D}$.

The action of the ith learner is to select a $d \times 1$ vector $\boldsymbol{\theta}_i$ as the parameter of the linear regression function $\hat{\mathbf{y}}_i = \mathbf{X}\boldsymbol{\theta}_i$, where $\hat{\mathbf{y}}_i$ is the predicted values for data $\mathbf{X}$. The expected cost function of the ith learner at test time is then

$$\begin{aligned} c_i(\boldsymbol{\theta}_i, \mathcal{D}^{'}) = \beta\mathbb{E}_{(\mathbf{X}^{'},\mathbf{y})\sim\mathcal{D}^{'}}[\ell(\mathbf{X}^{'}\boldsymbol{\theta}_i, \mathbf{y})] \\ + (1-\beta)\mathbb{E}_{(\mathbf{X},\mathbf{y})\sim\mathcal{D}}[\ell(\mathbf{X}\boldsymbol{\theta}_i, \mathbf{y})]. \end{aligned} \tag{1}$$

where $\ell(\hat{\mathbf{y}}, \mathbf{y}) = ||\hat{\mathbf{y}} - \mathbf{y}||_2^2$. That is, the cost function of a learner i is a combination of its expected cost from both the attacker and the honest source.

Every instance $(\mathbf{x}, y)$ generated according to $\mathcal{D}$ is, with probability β, maliciously modified by the attacker into another, $(\mathbf{x}', y)$, as follows. We assume that the attacker has an instance-specific target $z(\mathbf{x})$, and wishes that the prediction made by each learner i on the modified instance, $\hat{y} = \boldsymbol{\theta}_i^\top \mathbf{x}^{'}$, is close to this target. We measure this objective for the attacker by $\ell(\hat{\mathbf{y}}, \mathbf{z}) = ||\hat{\mathbf{y}} - \mathbf{z}||_2^2$ for a vector of predicted and target values $\hat{\mathbf{y}}$ and $\mathbf{z}$, respectively. In addition, the attacker incurs a cost of transforming a distribution $\mathcal{D}$ into $\mathcal{D}^{'}$, denoted by $R(\mathcal{D}^{'}, \mathcal{D})$.

After a dataset $(\mathbf{X}^{'}, \mathbf{y})$ is generated in this way by the attacker, it is used simultaneously against all the learners. This is natural in most real attacks: for example, spam templates are commonly generated to be used broadly, against many individuals and organizations, and, similarly, malware executables are often produced to be generally effective, rather than custom made for each target. The expected cost function of the attacker is then a sum of its total expected cost for all learners plus the cost of transforming $\mathcal{D}$ into $\mathcal{D}^{'}$ with coefficient $\lambda > 0$:

$$c_a(\{\boldsymbol{\theta}_i\}_{i=1}^n, \mathcal{D}^{'}) = \sum_{i=1}^{n} \mathbb{E}_{(\mathbf{X}^{'},\mathbf{y})\sim\mathcal{D}^{'}}[\ell(\mathbf{X}^{'}\boldsymbol{\theta}_i, \mathbf{z})] + \lambda R(\mathcal{D}^{'}, \mathcal{D}). \tag{2}$$

As is typical, we estimate the cost functions of the learners and the attacker using training data $(\mathbf{X}, \mathbf{y})$, which is also used to simulate attacks. Consequently, the cost functions of each learner and the attacker are estimated by

$$c_i(\boldsymbol{\theta}_i, \mathbf{X}^{'}) = \beta\ell(\mathbf{X}^{'}\boldsymbol{\theta}_i, \mathbf{y}) + (1-\beta)\ell(\mathbf{X}\boldsymbol{\theta}_i, \mathbf{y}) \tag{3}$$

and

$$c_a(\{\boldsymbol{\theta}_i\}_{i=1}^n, \mathbf{X}^{'}) = \sum_{i=1}^{n} \ell(\mathbf{X}^{'}\boldsymbol{\theta}_i, \mathbf{z}) + \lambda R(\mathbf{X}^{'}, \mathbf{X}) \tag{4}$$

where the attacker's modification cost is measured by $R(\mathbf{X}^{'}, \mathbf{X}) = ||\mathbf{X}^{'} - \mathbf{X}||_F^2$, the squared Frobenius norm.

2.2. The Multi-Learner Stackerlberg Game

We are now ready to formally define the game between the n learners and the attacker. The MLSG has two stages: in the first stage, learners simultaneously select their model parameters $\boldsymbol{\theta}_i$, and in the second stage, the attacker makes its decision (manipulating $\mathbf{X}^{'}$) after observing the learners' model choices $\{\boldsymbol{\theta}_i\}_{i=1}^n$. We assume that the proposed game satisfies the following assumptions:

1. The learners have complete information about parameters β, λ and $\mathbf{z}$. This is a strong assumption, and we relax it in our experimental evaluation (Section 6), providing guidance on how to deal with uncertainty about these parameters.
2. Each learner has the same action (model parameter) space $\boldsymbol{\Theta} \subseteq \mathbb{R}^{d\times 1}$ which is nonempty, compact and convex. The action space of the attacker is $\mathbb{R}^{m\times d}$.

3. The columns of the training data $\mathbf{X}$ are linearly independent.

We use *Multi-Learner Stackelberg Equilibrium* (MLSE) as the solution for the MLSG, defined as follows.

Definition 1 (Multi-Learner Stackelberg Equilibrium (MLSE))**.** *An action profile* $(\{\boldsymbol{\theta}_i^*\}_{i=1}^n, \mathbf{X}^*)$ *is an MLSE if it satisfies*

$$\begin{aligned} \boldsymbol{\theta}_i^* &= \arg\min_{\boldsymbol{\theta}_i \in \boldsymbol{\Theta}} c_i(\boldsymbol{\theta}_i, \mathbf{X}^*(\boldsymbol{\theta})), \forall i \in \mathcal{N} \\ s.t. \quad \mathbf{X}^*(\boldsymbol{\theta}) &= \arg\min_{\mathbf{X}' \in \mathbb{R}^{m \times d}} c_a(\{\boldsymbol{\theta}_i\}_{i=1}^n, \mathbf{X}'). \end{aligned} \tag{5}$$

where $\boldsymbol{\theta} = \{\boldsymbol{\theta}_i\}_{i=1}^n$ *constitutes the joint actions of the learners.*

At the high level, the MLSE is a blend between a Nash equilibrium (among all learners) and a Stackelberg equilibrium (between the learners and the attacker), in which the attacker plays a best response to the *observed* models $\boldsymbol{\theta}$ chosen by the learners, and given this behavior by the attacker, all learners' models $\boldsymbol{\theta}_i$ are mutually optimal.

The following lemma characterizes the best response of the attacker to arbitrary model choices $\{\boldsymbol{\theta}_i\}_{i=1}^n$ by the learners.

Lemma 1 (Best Response of the Attacker)**.** *Given* $\{\boldsymbol{\theta}_i\}_{i=1}^n$*, the best response of the attacker is*

$$\mathbf{X}^* = (\lambda \mathbf{X} + \mathbf{z} \sum_{i=1}^n \boldsymbol{\theta}_i^\top)(\lambda \mathbf{I} + \sum_{i=1}^n \boldsymbol{\theta}_i \boldsymbol{\theta}_i^\top)^{-1}. \tag{6}$$

Proof. We derive the best response of the attacker by using the first order condition. The details are included in the supplementary material. □

Lemma 1 shows that the best response of the attacker, $\mathbf{X}^*$, has a closed form solution, as a function of learner model parameters $\{\boldsymbol{\theta}_i\}_{i=1}^n$. Let $\boldsymbol{\theta}_{-i} = \{\boldsymbol{\theta}_j\}_{j \neq i}$, then $c_i(\boldsymbol{\theta}_i, \mathbf{X}^*)$ in Eq. (5) can be rewritten as

$$c_i(\boldsymbol{\theta}_i, \boldsymbol{\theta}_{-i}) = \beta \ell(\mathbf{X}^*(\boldsymbol{\theta}_i, \boldsymbol{\theta}_{-i})\boldsymbol{\theta}_i, \mathbf{y}) + (1-\beta)\ell(\mathbf{X}\boldsymbol{\theta}_i, \mathbf{y}). \tag{7}$$

Using Eq. (7), we can then define a *Multi-Learner Nash Game (MLNG)*:

Definition 2 (Multi-Learner Nash Game (MLNG))**.** *A static game, denoted as* $\langle \mathcal{N}, \boldsymbol{\Theta}, (c_i) \rangle$ *is a* Multi-Learner Nash Game *if*

1. *The set of players is the set of learners* $\mathcal{N}$*,*
2. *the cost function of each learner* i *is* $c_i(\boldsymbol{\theta}_i, \boldsymbol{\theta}_{-i})$ *defined in Eq.* (7)*,*
3. *all learners simultaneously select* $\boldsymbol{\theta}_i \in \boldsymbol{\Theta}$*.*

We can then define *Multi-Learner Nash Equilibrium (MLNE)* of the game $\langle \mathcal{N}, \boldsymbol{\Theta}, (c_i) \rangle$:

Definition 3 (Multi-Learner Nash Equilibrium (MLNE))**.** *An action profile* $\boldsymbol{\theta}^* = \{\boldsymbol{\theta}_i^*\}_{i=1}^n$ *is a* Multi-Learner Nash Equilibrium *of the MLNG* $\langle \mathcal{N}, \boldsymbol{\Theta}, (c_i) \rangle$ *if it is the solution of the following set of coupled optimization problem:*

$$\min_{\boldsymbol{\theta}_i \in \boldsymbol{\Theta}} c_i(\boldsymbol{\theta}_i, \boldsymbol{\theta}_{-i}), \forall i \in \mathcal{N}. \tag{8}$$

Combining the results above, the following result is immediate.

Theorem 1. *An action profile* $(\{\boldsymbol{\theta}_i^*\}_{i=1}^n, \mathbf{X}^*)$ *is an MLSE of the multi-learner Stackelberg game if and only if* $\{\boldsymbol{\theta}_i^*\}_{i=1}^n$ *is a MLNE of the multi-learner Nash game* $\langle \mathcal{N}, \boldsymbol{\Theta}, (c_i) \rangle$*, with* $\mathbf{X}^*$ *defined in Eq.* (6) *for* $\boldsymbol{\theta}_i = \boldsymbol{\theta}_i^*, \forall i \in \mathcal{N}$*.*

Theorem 1 shows that we can reduce the original $(n+1)$-player Stackelberg game to an n-player simultaneous-move game $\langle \mathcal{N}, \boldsymbol{\Theta}, (c_i) \rangle$. In the remaining sections, we focus on analyzing the Nash equilibrium of this multi-learner Nash game.

3. Theoretical Analysis

In this section, we analyze the game $\langle \mathcal{N}, \boldsymbol{\Theta}, (c_i) \rangle$. As presented in Eq. (6), there is an inverse of a complicated matrix to compute the best response of the attacker. Hence, the cost function $c_i(\boldsymbol{\theta}_i, \boldsymbol{\theta}_{-i})$ shown in Eq. (7) is intractable. To address this challenge, we first derive a new game, $\langle \mathcal{N}, \boldsymbol{\Theta}, (\widetilde{c}_i) \rangle$ with tractable cost function for its players, to approximate $\langle \mathcal{N}, \boldsymbol{\Theta}, (c_i) \rangle$. Afterward, we analyze existence and uniqueness of the *Nash Equilibirum* of $\langle \mathcal{N}, \boldsymbol{\Theta}, (\widetilde{c}_i) \rangle$.

3.1. Approximation of $\langle \mathcal{N}, \boldsymbol{\Theta}, (c_i) \rangle$

We start our analysis by computing $(\lambda \mathbf{I} + \sum_{i=1}^n \boldsymbol{\theta}_i \boldsymbol{\theta}_i^\top)^{-1}$ presented in Eq. (6). Let matrix $\mathbf{A}_n = \lambda \mathbf{I} + \sum_{i=1}^n \boldsymbol{\theta}_i \boldsymbol{\theta}_i^\top$, and $\mathbf{A}_{-i} = \lambda \mathbf{I} + \sum_{j \neq i} \boldsymbol{\theta}_j \boldsymbol{\theta}_j^\top$. Then, $\mathbf{A}_n = \mathbf{A}_{-i} + \boldsymbol{\theta}_i \boldsymbol{\theta}_i^\top$. Similarly, let matrix $\mathbf{B}_n = \lambda \mathbf{X} + \mathbf{z} \sum_{i=1}^n \boldsymbol{\theta}_i^\top$, and $\mathbf{B}_{-i} = \lambda \mathbf{X} + \mathbf{z} \sum_{j \neq i} \boldsymbol{\theta}_j^\top$, which implies that $\mathbf{B}_n = \mathbf{B}_{-i} + \mathbf{z}\boldsymbol{\theta}_i^\top$ The best response of the attacker can then be rewritten as $\mathbf{X}^* = \mathbf{B}_n \mathbf{A}_n^{-1}$. We then obtain the following results.

Lemma 2. $\mathbf{A}_n$ *and* $\mathbf{A}_{-i}$ *satisfy*

1. $\mathbf{A}_n$ *and* $\mathbf{A}_{-i}$ *are invertible, and the corresponding invertible matrices,* $\mathbf{A}_n^{-1}$ *and* $\mathbf{A}_{-i}^{-1}$*, are positive definite.*
2. $\mathbf{A}_n^{-1} = \mathbf{A}_{-i}^{-1} - \frac{\mathbf{A}_{-i}^{-1} \boldsymbol{\theta}_i \boldsymbol{\theta}_i^\top \mathbf{A}_{-i}^{-1}}{1 + \boldsymbol{\theta}_i^\top \mathbf{A}_{-i}^{-1} \boldsymbol{\theta}_i}$.
3. $\boldsymbol{\theta}_i^\top \mathbf{A}_{-i}^{-1} \boldsymbol{\theta}_i \leq \frac{1}{\lambda} \boldsymbol{\theta}_i^\top \boldsymbol{\theta}_i$.

Proof. The proof is included in the supplementary document. □

Lemma 2 allows us to relax $\ell(\mathbf{X}^*(\boldsymbol{\theta}_i, \boldsymbol{\theta}_{-i})\boldsymbol{\theta}_i, \mathbf{y})$ as follows:

Lemma 3.

$$\ell(\mathbf{X}^*(\boldsymbol{\theta}_i, \boldsymbol{\theta}_{-i})\boldsymbol{\theta}_i, \mathbf{y}) \leq \ell(\mathbf{B}_{-i}\mathbf{A}_{-i}^{-1}\boldsymbol{\theta}_i, \mathbf{y}) + \frac{1}{\lambda^2}||\mathbf{z}-\mathbf{y}||_2^2(\boldsymbol{\theta}_i^\top \boldsymbol{\theta}_i)^2. \quad (9)$$

Proof. Firstly, by using *Sherman-Morrison formula* we have

$$\begin{aligned}
&\mathbf{X}^*\boldsymbol{\theta}_i = \mathbf{B}_n\mathbf{A}_n^{-1}\boldsymbol{\theta}_i \\
&= (\mathbf{B}_{-i} + \mathbf{z}\boldsymbol{\theta}_i^\top)(\mathbf{A}_{-i}^{-1} - \frac{\mathbf{A}_{-i}^{-1}\boldsymbol{\theta}_i\boldsymbol{\theta}_i^\top\mathbf{A}_{-i}^{-1}}{1+\boldsymbol{\theta}_i^\top\mathbf{A}_{-i}^{-1}\boldsymbol{\theta}_i})\boldsymbol{\theta}_i \\
&= \mathbf{B}_{-i}\mathbf{A}_{-i}^{-1}\boldsymbol{\theta}_i + \frac{\mathbf{z}\boldsymbol{\theta}_i^\top\mathbf{A}_{-i}^{-1}\boldsymbol{\theta}_i - \mathbf{B}_{-i}\mathbf{A}_{-i}^{-1}\boldsymbol{\theta}_i\boldsymbol{\theta}_i^\top\mathbf{A}_{-i}^{-1}\boldsymbol{\theta}_i}{1+\boldsymbol{\theta}_i^\top\mathbf{A}_{-i}^{-1}\boldsymbol{\theta}_i} \\
&= \frac{(\mathbf{B}_{-i} + \mathbf{z}\boldsymbol{\theta}_i^\top)\mathbf{A}_{-i}^{-1}\boldsymbol{\theta}_i}{1+\boldsymbol{\theta}_i^\top\mathbf{A}_{-i}^{-1}\boldsymbol{\theta}_i} \\
&= \frac{\mathbf{B}_n\mathbf{A}_{-i}^{-1}\boldsymbol{\theta}_i}{1+\boldsymbol{\theta}_i^\top\mathbf{A}_{-i}^{-1}\boldsymbol{\theta}_i}.
\end{aligned}$$

Then,

$$\begin{aligned}
&\ell(\mathbf{X}^*\boldsymbol{\theta}_i, \mathbf{y}) = ||\frac{\mathbf{B}_n\mathbf{A}_{-i}^{-1}\boldsymbol{\theta}_i}{1+\boldsymbol{\theta}_i^\top\mathbf{A}_{-i}^{-1}\boldsymbol{\theta}_i} - \mathbf{y}||_2^2 \\
&= ||\frac{\mathbf{B}_n\mathbf{A}_{-i}^{-1}\boldsymbol{\theta}_i - \mathbf{y} - \boldsymbol{\theta}_i^\top\mathbf{A}_{-i}^{-1}\boldsymbol{\theta}_i\mathbf{y}}{1+\boldsymbol{\theta}_i^\top\mathbf{A}_{-i}^{-1}\boldsymbol{\theta}_i}||_2^2 \\
&\leq ||\mathbf{B}_n\mathbf{A}_{-i}^{-1}\boldsymbol{\theta}_i - \mathbf{y} - \boldsymbol{\theta}_i^\top\mathbf{A}_{-i}^{-1}\boldsymbol{\theta}_i\mathbf{y}||_2^2 \\
&= ||(\mathbf{B}_{-i} + \mathbf{z}\boldsymbol{\theta}_i^\top)\mathbf{A}_{-i}^{-1}\boldsymbol{\theta}_i - \mathbf{y} - \boldsymbol{\theta}_i^\top\mathbf{A}_{-i}^{-1}\boldsymbol{\theta}_i\mathbf{y}||_2^2 \\
&= ||\mathbf{B}_{-i}\mathbf{A}_{-i}^{-1}\boldsymbol{\theta}_i - \mathbf{y} + (\mathbf{z}-\mathbf{y})\boldsymbol{\theta}_i^\top\mathbf{A}_{-i}^{-1}\boldsymbol{\theta}_i||_2^2 \\
&\leq \ell(\mathbf{B}_{-i}\mathbf{A}_{-i}^{-1}\boldsymbol{\theta}_i, \mathbf{y}) + ||\mathbf{z}-\mathbf{y}||_2^2(\boldsymbol{\theta}_i^\top\mathbf{A}_{-i}^{-1}\boldsymbol{\theta}_i)^2
\end{aligned}$$

By using Lemma 2, we have $(\boldsymbol{\theta}_i^\top\mathbf{A}_{-i}^{-1}\boldsymbol{\theta}_i)^2 \leq \frac{1}{\lambda^2}(\boldsymbol{\theta}_i^\top\boldsymbol{\theta}_i)^2$ which completes the proof. □

Note that in Eq. (9), $\mathbf{B}_{-i}$ and $\mathbf{A}_{-i}$ only depend on $\{\boldsymbol{\theta}_j\}_{j\neq i}$. Hence, the RHS of Eq. (9) is a strictly convex function with respect to $\boldsymbol{\theta}_i$. Lemma 3 shows that $\ell(\mathbf{X}^*(\boldsymbol{\theta}_i, \boldsymbol{\theta}_{-i})\boldsymbol{\theta}_i, \mathbf{y})$ can be relaxed by moving $\boldsymbol{\theta}_i$ out of $\mathbf{X}^*(\boldsymbol{\theta}_i, \boldsymbol{\theta}_{-i})$ and adding a regularizer $(\boldsymbol{\theta}_i^\top\boldsymbol{\theta}_i)^2$ with its coefficient $\frac{||\mathbf{z}-\mathbf{y}||_2^2}{\lambda^2}$. Motivated by this method, we iteratively relax $\ell(\mathbf{X}^*(\boldsymbol{\theta}_i, \boldsymbol{\theta}_{-i})\boldsymbol{\theta}_i, \mathbf{y})$ by adding corresponding regularizers. We now identify a tractable upper bound function for $c_i(\boldsymbol{\theta}_i, \boldsymbol{\theta}_{-i})$.

Theorem 2.

$$\begin{aligned}
c_i(\boldsymbol{\theta}_i, \boldsymbol{\theta}_{-i}) &\leq \bar{c}_i(\boldsymbol{\theta}_i, \boldsymbol{\theta}_{-i}) \\
&= \ell(\mathbf{X}\boldsymbol{\theta}_i, \mathbf{y}) + \frac{\beta}{\lambda^2}||\mathbf{z}-\mathbf{y}||_2^2\sum_{j=1}^{n}(\boldsymbol{\theta}_j^\top\boldsymbol{\theta}_i)^2 + \epsilon,
\end{aligned} \quad (10)$$

where ϵ is a positive constant and $\epsilon < +\infty$.

Proof. We prove by extending the results in Lemma 3 and iteratively relaxing the cost function. The details are included in the supplementary material. □

As represented in Eq. (10), $\bar{c}_i(\boldsymbol{\theta}_i, \boldsymbol{\theta}_{-i})$ is strictly convex with respect to $\boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_j(\forall j \neq i)$. We then use the game $\langle \mathcal{N}, \boldsymbol{\Theta}, (\bar{c}_i)\rangle$ as an approximation of $\langle \mathcal{N}, \boldsymbol{\Theta}, (c_i)\rangle$. Let

$$\begin{aligned}
\widetilde{c}_i(\boldsymbol{\theta}_i, \boldsymbol{\theta}_{-i}) &= \bar{c}_i(\boldsymbol{\theta}_i, \boldsymbol{\theta}_{-i}) - \epsilon \\
&= \ell(\mathbf{X}\boldsymbol{\theta}_i, \mathbf{y}) + \frac{\beta}{\lambda^2}||\mathbf{z}-\mathbf{y}||_2^2\sum_{j=1}^{n}(\boldsymbol{\theta}_j^\top\boldsymbol{\theta}_i)^2,
\end{aligned} \quad (11)$$

then $\langle \mathcal{N}, \boldsymbol{\Theta}, (\widetilde{c}_i)\rangle$ has the same Nash equilibrium with $\langle \mathcal{N}, \boldsymbol{\Theta}, (\bar{c}_i)\rangle$ if one exists, as adding or deleting a constant term does not affect the optimal solution. Hence, we use $\langle \mathcal{N}, \boldsymbol{\Theta}, (\widetilde{c}_i)\rangle$ to approximate $\langle \mathcal{N}, \boldsymbol{\Theta}, (c_i)\rangle$, and analyze the Nash equilibrium of $\langle \mathcal{N}, \boldsymbol{\Theta}, (\widetilde{c}_i)\rangle$ in the remaining sections.

3.2. Existence of Nash Equilibrium

As introduced in Section 2, each learner has identical action spaces, and they are trained with the same dataset. We exploit this symmetry to analyze the existence of a Nash equilibrium of the approximation game $\langle \mathcal{N}, \boldsymbol{\Theta}, (\widetilde{c}_i)\rangle$.

We first define a *Symmetric Game* (Cheng et al., 2004):

Definition 4 (Symmetric Game)**.** *An n-player game is symmetric if the players have the same action space, and their cost functions $c_i(\boldsymbol{\theta}_i, \boldsymbol{\theta}_{-i})$ satisfy*

$$c_i(\boldsymbol{\theta}_i, \boldsymbol{\theta}_{-i}) = c_j(\boldsymbol{\theta}_j, \boldsymbol{\theta}_{-j}), \forall i, j \in \mathcal{N} \quad (12)$$

if $\boldsymbol{\theta}_i = \boldsymbol{\theta}_j$ and $\boldsymbol{\theta}_{-i} = \boldsymbol{\theta}_{-j}$.

In a symmetric game $\langle \mathcal{N}, \boldsymbol{\Theta}, (\widetilde{c}_i)\rangle$ it is natural to consider a *Symmetric Equilibrium*:

Definition 5 (Symmetric Equilibrium)**.** *An action profile $\{\boldsymbol{\theta}_i^*\}_{i=1}^n$ of $\langle \mathcal{N}, \boldsymbol{\Theta}, (\widetilde{c}_i)\rangle$ is a symmetric equilibrium if it is a Nash equilibrium and $\boldsymbol{\theta}_i^* = \boldsymbol{\theta}_j^*, \forall i, j \in \mathcal{N}$.*

We now show that our approximate game is symmetric, and always has a symmetric Nash equilibrium.

Theorem 3 (Existence of Nash Equilibrium)**.** *$\langle \mathcal{N}, \boldsymbol{\Theta}, (\widetilde{c}_i)\rangle$ is a symmetric game and it has at least one symmetric equilibrium.*

Proof. As described above, the players of $\langle \mathcal{N}, \boldsymbol{\Theta}, (\widetilde{c}_i)\rangle$ use the same action space and complete information of others. Hence, the cost function c_i is symmetric, making $\langle \mathcal{N}, \boldsymbol{\Theta}, (\widetilde{c}_i)\rangle$ a symmetric game. As $\langle \mathcal{N}, \boldsymbol{\Theta}, (\widetilde{c}_i)\rangle$ has nonempty, compact and convex action space, and the cost function $\widetilde{c}_i$ is continuous in $\{\boldsymbol{\theta}_i\}_{i=1}^n$ and convex in $\boldsymbol{\theta}_i$, according to Theorem 3 in Cheng et al. (2004), $\langle \mathcal{N}, \boldsymbol{\Theta}, (\widetilde{c}_i)\rangle$ has at least one symmetric Nash equilibrium. □

3.3. Uniqueness of Nash Equilibrium

While we showed that the approximate game always admits a symmetric Nash equilibrium, it leaves open the possibility that there may be multiple symmetric equilibria, as well as equilibria which are not symmetric. We now demonstrate that this game in fact has a *unique* equilibrium (which must therefore be symmetric).

Theorem 4 (Uniqueness of Nash Equilibrium)**.** *$\langle \mathcal{N}, \boldsymbol{\Theta}, (\widetilde{c}_i)\rangle$ has a unique Nash equilibrium, and this unique NE is symmetric.*

Proof. We have known that $\langle \mathcal{N}, \boldsymbol{\Theta}, (\widetilde{c}_i)\rangle$ has at least one NE, and each learner has an nonempty, compact and convex action space $\boldsymbol{\Theta}$. Hence, we can apply Theorem 2 and Theorem 6 of Rosen (1965). That is, for some fixed $\{r_i\}_i^n (0 < r_i < 1, \sum_{i=1}^n r_i = 1)$, if the matrix in Eq. (13) is positive definite, then $\langle \mathcal{N}, \boldsymbol{\Theta}, (\widetilde{c}_i)\rangle$ has a unique NE.

$$Jr(\boldsymbol{\theta}) = \begin{bmatrix} r_1\nabla_{\boldsymbol{\theta}_1,\boldsymbol{\theta}_1}\widetilde{c}_1(\boldsymbol{\theta}) & \dots & r_1\nabla_{\boldsymbol{\theta}_1,\boldsymbol{\theta}_n}\widetilde{c}_1(\boldsymbol{\theta}) \\ \vdots & & \vdots \\ r_n\nabla_{\boldsymbol{\theta}_n,\boldsymbol{\theta}_1}\widetilde{c}_n(\boldsymbol{\theta}) & \dots & r_n\nabla_{\boldsymbol{\theta}_n,\boldsymbol{\theta}_n}\widetilde{c}_n(\boldsymbol{\theta}) \end{bmatrix} \tag{13}$$

We first let $r_1 = r_2 = ... = r_n = \frac{1}{n}$ and decompose $Jr(\boldsymbol{\theta})$ as follows,

$$Jr(\boldsymbol{\theta}) = \frac{2}{n}\mathbf{P} + \frac{2\beta||\mathbf{z}-\mathbf{y}||_2^2}{\lambda^2 n}(\mathbf{Q}+\mathbf{S}+\mathbf{T}), \tag{14}$$

where $\mathbf{P}$ and $\mathbf{Q}$ are *block diagonal matrices* such that $\mathbf{P}_{ii} = \mathbf{X}^\top\mathbf{X}$, $\mathbf{P}_{ij} = \mathbf{0}$, $\mathbf{Q}_{ii} = 4\boldsymbol{\theta}_i\boldsymbol{\theta}_i^\top + \boldsymbol{\theta}_i^\top\boldsymbol{\theta}_i\mathbf{I}$ and $\mathbf{Q}_{ij} = \mathbf{0}$, $\forall i, j \in \mathcal{N}, j \neq i$. $\mathbf{S}$ and $\mathbf{T}$ are *block symmetric matrices* such that $\mathbf{S}_{ii} = \boldsymbol{\theta}_i^\top\boldsymbol{\theta}_i\mathbf{I}$, $\mathbf{S}_{ij} = \boldsymbol{\theta}_i^\top\boldsymbol{\theta}_j\mathbf{I}$, $\mathbf{T}_{ii} = \sum_{j\neq i}\boldsymbol{\theta}_j\boldsymbol{\theta}_j^\top$ and $\mathbf{T}_{ij} = \boldsymbol{\theta}_j\boldsymbol{\theta}_i^\top, \forall i, j \in \mathcal{N}, j \neq i$.

Next, we prove that $\mathbf{P}$ is *positive definite*, and $\mathbf{Q}$, $\mathbf{S}$ and $\mathbf{T}$ are *positive semi-definite*. Hence, $Jr(\boldsymbol{\theta})$ is positive definite, which indicates that $\langle \mathcal{N}, \boldsymbol{\Theta}, (\widetilde{c}_i)\rangle$ has a unique NE. As Theorem 3 points out, the game has at least one symmetric NE. Therefore, the NE is unique and must be symmetric. Due to space limitation the details of this proof are included in the supplementary material. □

4. Computing the Equilibrium

Having shown that $\langle \mathcal{N}, \boldsymbol{\Theta}, (\widetilde{c}_i)\rangle$ has a unique symmetric Nash equilibrium, we now consider computing its solution. We exploit the symmetry of the game which enables to reduce the search space of the game to only symmetric solutions. Particularly, we derive the symmetric Nash equilibrium of $\langle \mathcal{N}, \boldsymbol{\Theta}, (\widetilde{c}_i)\rangle$ by solving a single convex optimization problem. We obtain the following result.

Theorem 5. *Let*

$$f(\boldsymbol{\theta}) = \ell(\mathbf{X}\boldsymbol{\theta}, \mathbf{y}) + \frac{\beta(n+1)}{2\lambda^2}||\mathbf{z}-\mathbf{y}||_2^2(\boldsymbol{\theta}^\top\boldsymbol{\theta})^2, \tag{15}$$

Then, the unique symmetric NE of $\langle \mathcal{N}, \boldsymbol{\Theta}, (\widetilde{c}_i)\rangle$, $\{\boldsymbol{\theta}_i^*\}_{i=1}^n$, can be derived by solving the following convex optimization problem

$$\min_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} f(\boldsymbol{\theta}) \tag{16}$$

and then letting $\boldsymbol{\theta}_i^* = \boldsymbol{\theta}^*, \forall i \in \mathcal{N}$, where $\boldsymbol{\theta}^*$ is the solution of Eq. (16).

Proof. We prove this theorem by characterizing the first-order optimality conditions of each learner's minimization problem in Eq. (8) with c_i being replaced with its approximation $\widetilde{c}_i$. Let $\{\boldsymbol{\theta}_i^*\}_{i=1}^n$ be the NE, then it satisfies

$$(\boldsymbol{\eta} - \boldsymbol{\theta}_i^*)^\top\nabla_{\boldsymbol{\theta}_i}\widetilde{c}_i(\boldsymbol{\theta}_i^*, \boldsymbol{\theta}_{-i}^*) \geq 0, \forall\boldsymbol{\eta} \in \boldsymbol{\Theta}, \forall i \in \mathcal{N} \tag{17}$$

where $\nabla_{\boldsymbol{\theta}_i}\widetilde{c}_i(\boldsymbol{\theta}_i^*, \boldsymbol{\theta}_{-i}^*)$ is the gradient of $\widetilde{c}_i(\boldsymbol{\theta}_i, \boldsymbol{\theta}_{-i})$ with respect to $\boldsymbol{\theta}_i$ and is evaluated at $\{\boldsymbol{\theta}_i^*\}_{i=1}^n$. Then, Eq. (17) is equivalent to the equations as follows:

$$\begin{cases} (\boldsymbol{\eta} - \boldsymbol{\theta}_1^*)^\top\nabla_{\boldsymbol{\theta}_1}\widetilde{c}_1(\boldsymbol{\theta}_1^*, \boldsymbol{\theta}_{-1}^*) \geq 0, \forall\boldsymbol{\eta} \in \boldsymbol{\Theta}, \\ \boldsymbol{\theta}_1^* = \boldsymbol{\theta}_j^*, \forall j \in \mathcal{N}\setminus\{1\} \end{cases} \tag{18}$$

The reasons are: first, any solution of Eq. (17) satisfies Eq. (18), as $\{\boldsymbol{\theta}_i^*\}_{i=1}^n$ is symmetric; Second, any solution of Eq. (18) also satisfies Eq. (17). By definition of symmetric game, if $\boldsymbol{\theta}_1^* = \boldsymbol{\theta}_j^*$, then $\nabla_{\boldsymbol{\theta}_1}\widetilde{c}_1(\boldsymbol{\theta}_1^*, \boldsymbol{\theta}_{-1}^*) = \nabla_{\boldsymbol{\theta}_j}\widetilde{c}_j(\boldsymbol{\theta}_j^*, \boldsymbol{\theta}_{-j}^*)$, and we have

$$(\boldsymbol{\eta} - \boldsymbol{\theta}_j^*)^\top\nabla_{\boldsymbol{\theta}_j}\widetilde{c}_j(\boldsymbol{\theta}_j^*, \boldsymbol{\theta}_{-j}^*), \forall\boldsymbol{\eta} \in \boldsymbol{\Theta}, \forall j \in \mathcal{N}\setminus\{1\}$$

Hence, Eq. (17) and Eq. (18) are equivalent. Eq. (18) can be further rewritten as

$$(\boldsymbol{\eta} - \boldsymbol{\theta}_1^*)^\top\nabla_{\boldsymbol{\theta}_1}\widetilde{c}_1(\boldsymbol{\theta}_1^*, \boldsymbol{\theta}_{-1}^*)|_{\boldsymbol{\theta}_1^*=...=\boldsymbol{\theta}_n^*} \geq 0, \forall\boldsymbol{\eta} \in \boldsymbol{\Theta}. \tag{19}$$

We then let

$$\begin{aligned} F(\boldsymbol{\theta}_1^*) &= \nabla_{\boldsymbol{\theta}_1}\widetilde{c}_1(\boldsymbol{\theta}_1^*, \boldsymbol{\theta}_{-1}^*)|_{\boldsymbol{\theta}_1^*=...=\boldsymbol{\theta}_n^*} \\ &= 2\mathbf{X}^\top(\mathbf{X}\boldsymbol{\theta}_1^* - \mathbf{y}) + \frac{2\beta(n+1)}{\lambda^2}||\mathbf{z}-\mathbf{y}||_2^2\boldsymbol{\theta}_1^{*\top}\boldsymbol{\theta}_1^*\boldsymbol{\theta}_1^*. \end{aligned} \tag{20}$$

Then, $F(\boldsymbol{\theta}_1^*) = \nabla_{\boldsymbol{\theta}_1}f(\boldsymbol{\theta}_1^*)$ where $f(\cdot)$ is defined in Eq. (15). Hence, we have

$$(\boldsymbol{\eta} - \boldsymbol{\theta}_1^*)^\top\nabla_{\boldsymbol{\theta}_1}f(\boldsymbol{\theta}_1^*) \geq 0, \forall\boldsymbol{\eta} \in \boldsymbol{\Theta}, \tag{21}$$

This means that $\boldsymbol{\theta}_1^*$ is the solution of the optimization problem in Eq.(16) which finally completes the proof. □

A deeper look at Eq. (15) reveals that the Nash equilibrium can be obtained by each learner independently, without knowing others' actions. This means that the Nash equilibrium can be computed in a distributed manner while the convergence is still guaranteed. Hence, our proposed approach is highly scalable, as increasing the number of learners does not impact the complexity of finding the Nash equilibrium. We investigate the robustness of this equilibrium both using theoretical analysis and experiments in the remaining sections.

5. Robustness Analysis

We now draw a connection between the multi-learner equilibrium in the adversarial setting, derived above, and robustness, in the spirit of the analysis by Xu et al. (2009). Specifically, we prove the equivalence between Eq. (16) and a robust linear regression problem where data is maliciously corrupted by some disturbance $\triangle$. Formally, a robust linear regression solves the following problem:

$$\min_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \max_{\triangle\in\mathcal{U}} ||\mathbf{y} - (\mathbf{X} + \triangle)\boldsymbol{\theta}||_2^2, \tag{22}$$

where the uncertainty set $\mathcal{U} = \{\triangle \in \mathbb{R}^{m\times d} \mid \triangle^T\triangle = \mathbf{G} : |\mathbf{G}_{ij}| \leq c|\theta_i\theta_j| \ \forall i,j\}$, with $c = \frac{\beta(n+1)}{2\lambda^2}||\mathbf{z}-\mathbf{y}||_2^2$. Note that $\boldsymbol{\theta}$ is a vector and θ_i is the i-th element of $\boldsymbol{\theta}$.

From a game-theoretic point of view, in training phase the defender is simulating an attacker. The attacker maximizes the training error by adding disturbance to $\mathbf{X}$. The magnitude of the disturbance is controlled by a parameter $c = \frac{\beta(n+1)}{2\lambda^2}||\mathbf{z}-\mathbf{y}||_2^2$. Consequently, the robustness of Eq. (22) is guaranteed if and only if the magnitude reflects the uncertainty interval. This sheds some light on how to choose λ, β and $\mathbf{z}$ in practice. One strategy is to overestimate the attacker's strength, which amounts to choosing small values of λ, large values of β and exaggerated target $\mathbf{z}$. The intuition of this strategy is to enlarge the uncertainty set so as to cover potential adversarial behavior. In Experiments section we will show this strategy works well in practice. Another insight from Eq. (22) is that the fundamental reason *MLSG* is robust is because it proactively takes adversarial behavior into account.

Theorem 6. *The optimal solution $\boldsymbol{\theta}^*$ of the problem in Eq. (16) is an optimal solution to the robust optimization problem in Eq. (22).*

Proof. Fix $\boldsymbol{\theta}^*$, we show that

$$\max_{\triangle\in\mathcal{U}} ||\mathbf{y} - (\mathbf{X} + \triangle)\boldsymbol{\theta}^*||_2^2 = ||\mathbf{y} - \mathbf{X}\boldsymbol{\theta}^*||_2^2 + c\left(\boldsymbol{\theta}^{*T}\boldsymbol{\theta}^*\right)^2.$$

The left-hand side can be expanded as:

$$\begin{aligned}
&\max_{\triangle\in\mathcal{U}} ||\mathbf{y} - (\mathbf{X} + \triangle)\boldsymbol{\theta}^*||_2^2 \\
=&\max_{\triangle\in\mathcal{U}} ||\mathbf{y} - \mathbf{X}\boldsymbol{\theta}^* - \triangle\boldsymbol{\theta}^*||_2^2 \\
\leq&\max_{\triangle\in\mathcal{U}} ||\mathbf{y} - \mathbf{X}\boldsymbol{\theta}^*||_2^2 + \max_{\triangle\in\mathcal{U}} ||\triangle\boldsymbol{\theta}^*||_2^2 \\
=&\max_{\triangle\in\mathcal{U}} ||\mathbf{y} - \mathbf{X}\boldsymbol{\theta}^*||_2^2 + \max_{\triangle\in\mathcal{U}} \boldsymbol{\theta}^{*T}\triangle^T\triangle\boldsymbol{\theta}^* \\
&\text{(substitute } \triangle^T\triangle = \mathbf{G}) \\
=&||\mathbf{y} - \mathbf{X}\boldsymbol{\theta}^*||_2^2 + \max_{\mathbf{G}} \boldsymbol{\theta}^{*T}\mathbf{G}\boldsymbol{\theta}^* \\
=&||\mathbf{y} - \mathbf{X}\boldsymbol{\theta}^*||_2^2 + \max_{\mathbf{G}} \sum_{i=1}^{d} |\theta_i^*|^2\mathbf{G}_{ii} + 2\sum_{j=1}^{d}\sum_{i=1}^{j-1}\theta_i^*\theta_j^*\mathbf{G}_{ij} \\
\leq&||\mathbf{y} - \mathbf{X}\boldsymbol{\theta}^*||_2^2 + c\sum_{i=1}^{d}|\theta_i^*|^4 + 2c\sum_{j=1}^{d}\sum_{i=1}^{j-1}(\theta_i^*\theta_j^*)^2 \\
=&||\mathbf{y} - \mathbf{X}\boldsymbol{\theta}^*||_2^2 + c\Big(\sum_{i=1}^{d}|\theta_i^*|^2\Big)^2 \\
=&||\mathbf{y} - \mathbf{X}\boldsymbol{\theta}^*||_2^2 + c\left(\boldsymbol{\theta}^{*T}\boldsymbol{\theta}^*\right)^2.
\end{aligned}$$

Now we define $\triangle^* = [\sqrt{c}\theta_1^*\mathbf{u}, \cdots, \sqrt{c}\theta_n^*\mathbf{u}]$, where θ_i^* is the i-th element of $\boldsymbol{\theta}^*$ and $\mathbf{u}$ is defined as:

$$\mathbf{u} \triangleq \begin{cases} \frac{\mathbf{y}-\mathbf{X}\boldsymbol{\theta}^*}{||\mathbf{y}-\mathbf{X}\boldsymbol{\theta}^*||_2}, & \textit{if } \mathbf{y} \neq \mathbf{X}\boldsymbol{\theta}^* \\ \textit{any vector with unit } L_2 \textit{ norm}, & \text{otherwise} \end{cases} \tag{23}$$

Then we have:

$$\begin{aligned}
&\max_{\triangle\in\mathcal{U}} \|\mathbf{y} - (\mathbf{X} + \triangle)\boldsymbol{\theta}^*\|_2^2 \\
\geq&||\mathbf{y} - (\mathbf{X} + \triangle^*)\boldsymbol{\theta}^*||_2^2 \\
=&||\mathbf{y} - \mathbf{X}\boldsymbol{\theta}^* - \triangle^*\boldsymbol{\theta}^*||_2^2 \\
=&||\mathbf{y} - \mathbf{X}\boldsymbol{\theta}^* - \sum_{i=1}^{d}\sqrt{c}|\theta_i^*|^2\mathbf{u}||_2^2 \\
&(\mathbf{u} \text{ is in the same direction as } \mathbf{y} - \mathbf{X}\boldsymbol{\theta}^*) \\
=&||\mathbf{y} - \mathbf{X}\boldsymbol{\theta}^*||_2^2 + ||\sum_{i=1}^{d}\sqrt{c}|\theta_i^*|^2\mathbf{u}||_2^2 \\
=&||\mathbf{y} - \mathbf{X}\boldsymbol{\theta}^*||_2^2 + c\left(\boldsymbol{\theta}^{*T}\boldsymbol{\theta}^*\right)^2
\end{aligned} \tag{24}$$

□

6. Experiments

As previously discussed, a dataset is represented by $(\mathbf{X}, \mathbf{y})$, where $\mathbf{X}$ is the feature matrix and $\mathbf{y}$ is the vector of labels. We use $(\mathbf{x}_j, \mathbf{y}_j)$ to denote the j-th instance and its corresponding label. The dataset is equally divided into a training set $(\mathbf{X}_{\text{train}}, \mathbf{y}_{\text{train}})$ and a testing set $(\mathbf{X}_{\text{test}}, \mathbf{y}_{\text{test}})$. We conducted experiments on three datasets: Wine Quality (redwine),PDF malware (PDF), and Boston Housing Market (boston). The number of learners is set to 5. Due to space limitation the experimental results for the boston dataset are included in supplement.

The Wine Quality dataset (Cortez et al., 2009) contains 1599 instances and each instance has 11 features. Those features are physicochemical and sensory measurements for wine. The response variables are quality scores ranging from 0 to 10, where 10 represents for best quality and 0 for least quality. The PDF malware dataset consists of 18658 PDF files collected from the internet. We employed an open-sourced tool *mimicus*[1] to extract 135 real-valued features

[1]https://github.com/srndic/mimicus

from PDF files (Šrndic & Laskov, 2014). We then applied *peepdf*[2] to score each PDF between 0 and 10, with a higher score indicating greater likelihood of being malicious.

Throughout, we abbreviate our proposed approach as *MLSG*, and compare it to three other algorithms: ordinary least squares (*OLS*) regression, as well as *Lasso*, and Ridge regression (*Ridge*). *Lasso* and *Ridge* are ordinary least square with L_1 and L_2 regularizations. In our evaluation, we simulate the attacker for different values of β (the probability that a given instance is maliciously manipulated). The specific attack targets $\mathbf{z}$ vary depending on the dataset; we discuss these below. For our evaluation, we compute model parameters (for the equilibrium, in the case of *MLSG*) on training data. We then use test data to compute optimal attacks, characterized by Eq. (6). Let $\mathbf{X}'_{\text{test}}$ be the test feature matrix after adversarial manipulation, $\hat{\mathbf{y}}^A_{\text{test}}$ the associated predicted labels on manipulated test data, $\hat{\mathbf{y}}_{\text{test}}$ predicted labels on untainted test data, and $\mathbf{y}_{\text{test}}$ the ground truth labels for test data. We use root expected mean square error (RMSE) as an evaluation metric, where the expectation is with respect to the probability β of a particular instance being maliciously manipulated: $\sqrt{\frac{\beta(\hat{\mathbf{y}}^A_{\text{test}}-\mathbf{y}_{\text{test}})^T(\hat{\mathbf{y}}^A_{\text{test}}-\mathbf{y}_{\text{test}})+(1-\beta)(\hat{\mathbf{y}}_{\text{test}}-\mathbf{y}_{\text{test}})^T(\hat{\mathbf{y}}_{\text{test}}-\mathbf{y}_{\text{test}})}{N}}$, where N is the size of the test data.

The redwine dataset: Recall that the response variables in redwine dataset are quality scores ranging from 0 to 10. We simulated an attacker whose target is to increase the overall scores of testing data. In practice this could correspond to the scenario that wine sellers try to manipulate the evaluation of third-party organizations. We formally define the attacker's target as $\mathbf{z} = \mathbf{y} + \Delta$, where $\mathbf{y}$ is the ground-truth response variables and Δ is a real-valued vector representing the difference between the attacker's target and the ground-truth. Since the maximum score is 10, any element of $\mathbf{z}$ that is greater than 10 is clipped to 10. We define Δ to be homogeneous (all elements are the same); generalization to heterogeneous values is direct. The mean and standard deviation of $\mathbf{y}$ are $\mu_r = 5.64$ and $\sigma_r = 0.81$. We let $\Delta = 5\sigma_r \times \mathbf{1}$, where $\mathbf{1}$ is a vector with all elements equal to one. The intuition for this definition is to simulate the generating process of adversarial data. Specifically, by setting the attacker's target to an unrealistic value (i.e. in current case outside the $3\sigma_r$ of μ_r), the generated adversarial data $\mathbf{X}'$ is supposed to be intrinsically different from $\mathbf{X}$. For ease of exposition we use the term *defender* to refer to *MLSG*.

Remember that in Eq.(11) there are three hyper-parameters in the defender's loss function: λ, β, and $\mathbf{z}$. λ is the regularization coefficient in the attacker's loss function shown in Eq.(4). It is negatively proportional to the attacker's strength. β is the probability of a test data being malicious. $\mathbf{z}$ is the predication targets of the attacker. In practice these three hyper-parameters are externally set by the attacker. In the first experiment below we assume the defender knows the values of these three hyper-parameters, which corresponds to the *best case*. The result is shown in Figure 1. Each bar is averaged over 50 runs, where at each run we randomly sampled training and test data. The regularization parameters of *Lasso* and *Ridge* were selected by cross-validation. Figure 1 demonstrates that *MLSG* approximate equilibrium solution is significantly more robust than conventional linear regression learning, with and without regularization.

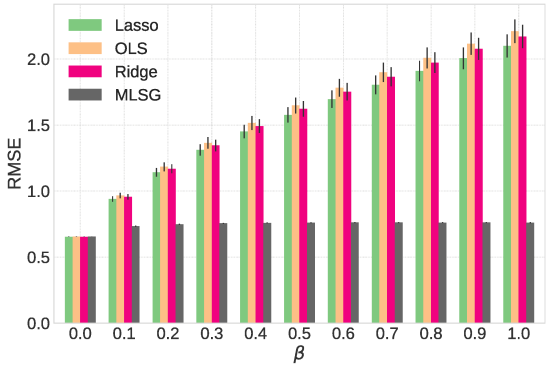

Figure 1. RMSE of $\mathbf{y}'$ and $\mathbf{y}$ on redwine dataset. The defender knows λ, β, and $\mathbf{z}$.

In the second experiment we relaxed the assumption that the defender knows λ, β and $\mathbf{z}$, and instead simulated the practical scenario that the defender obtains estimates for these (for example, from historical attack data), but the estimates have error. We denote by $\hat{\lambda} = 0.5$ and $\hat{\beta} = 0.8$ the defender's estimates of the true λ and β.[3] Remember that β is the probability of an instance being malicious and λ is negatively proportional to the attacker's strength. So the estimation characterizes a *pessimistic* defender that is expecting very strong attacks. We experimented with two kinds of estimation about $\mathbf{z}$: 1) the defender overestimates $\mathbf{z}$: $\hat{\mathbf{z}} = \mathbf{y} + t\mathbf{1}$, where t is a random variable sampled from a uniform distribution over $[5\sigma_r, 10]$; and 2) the defender underestimates $\mathbf{z}$: $\hat{\mathbf{z}} = \mathbf{y} + t\mathbf{1}$, where t is sampled from $[0, 5\sigma_r]$. Due to space limitations we only present the results for the latter; the former can be found in the supplementary materials. In Figure 2 the y-axis represents the actual values of λ, and the x-axis represents the actual values of β. The color bar on the right of each figure visualizes the average RMSE. Each cell is averaged over 50 runs. The result shows that even if there is a discrepancy between the defender's

[2]https://github.com/rohit-dua/peePDF

[3]We tried alternative values of $\hat{\lambda}$ and $\hat{\beta}$, and the results are consistent. Due to space limitations we include them in supplemental materials.

estimation and the actual adversarial behavior, *MLSG* is consistently more robust than the other approaches.

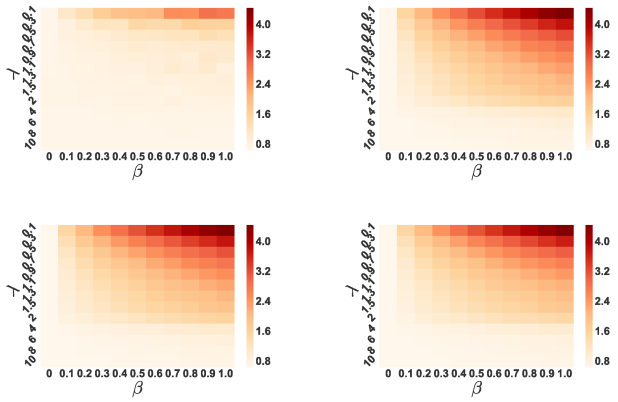

Figure 2. The average RMSE across different values of actual λ and β on redwine dataset. Upper Left: *MLSG*; Upper Right: *Lasso*; Lower Left: *Ridge*; Lower Right: *OLS*.

The PDF dataset: The response variables of this dataset are malicious scores ranging between 0 and 10. The mean and standard deviation of $\mathbf{y}$ are $\mu_p = 5.56$ and $\sigma_p = 2.66$. Instead of letting the Δ be non-negative as in previous two datasets, the attacker's target is to descrease the scores of malicious PDFs. Consequently, we define $\Delta = -2\sigma_p \times \mathbf{1}_{\mathcal{M}}$, where $\mathcal{M}$ is the set of indices of malicious PDF and $\mathbf{1}_{\mathcal{M}}$ is a vector with only those elements indexed by $\mathcal{M}$ being one and others being zero. Our experiments were conducted on a subset (3000 malicious PDF and 3000 benign PDF) randomly sampled from the original dataset. We evenly divided the subset into training and testing sets. We applied PCA to reduce dimensionality of the data and selected the top-10 principal components as features. The result for *best case* is displayed in Figure 3. Notice that when $\beta = 0$, *MLSG* is less robust than *Lasso*. This is to be expected, as $\beta = 0$ corresponds to non-adversarial data.

Similarly as before we relaxed the assumption that the defender knows λ, β and $\mathbf{z}$ and let the defender's estimation of the true λ and β be $\hat{\lambda} = 1.5$ and $\hat{\beta} = 0.5$. We also experimented with both overestimation and underestimation of $\mathbf{z}$. The defender's estimation is $\hat{\mathbf{z}} = \mathbf{y} - t\mathbf{1}_{\mathcal{M}}$. For overestimation setting t is sampled from $[2\sigma_p, 3\sigma_p]$, and for underestimation setting it is sampled from $[\sigma_p, 2\sigma_p]$. The result for underestimated $\hat{\mathbf{z}}$ is showed in Figure 4. Notice that in the upper left plot of Figure 4 the area inside the blue rectangle corresponds to those cases where $\hat{\lambda}$ and $\hat{\beta}$ are overestimated and they are more robust than the remaining underestimated cases. Similar patterns can be observed in Figure 2. This further supports our claim that it is advantageous to overestimate adversarial behavior.

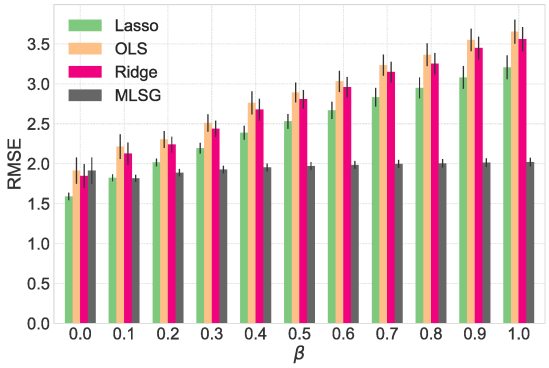

Figure 3. RMSE of $\mathbf{y}'$ and $\mathbf{y}$ on PDF dataset. The defender knows λ, β, and $\mathbf{z}$.

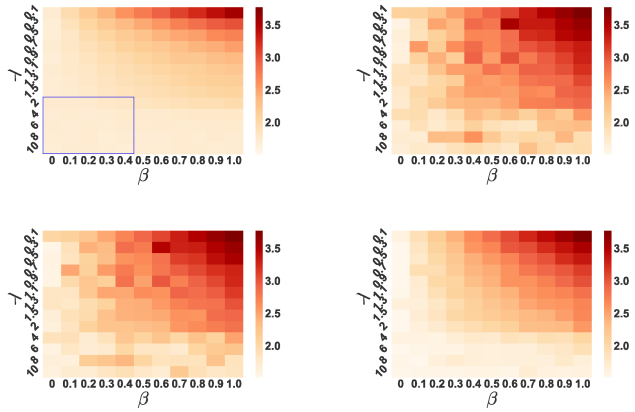

Figure 4. The average RMSE across different values of actual λ and β on PDF dataset. Upper Left: *MLSG*; Upper Right: *Lasso*; Lower Left: *Ridge*; Lower Right: *OLS*.

7. Conclusion

We study the problem of linear regression in adversarial settings involving multiple learners learning from the same or similar data. In our model, learners first simultaneously decide on their models (i.e., learn), and an attacker then modifies test instances to cause predictions to err towards the attacker's target. We first derive an upper bound on the cost functions of all learners, and the resulting approximate game. We then show that this game has a unique symmetric equilibrium, and present an approach for computing this equilibrium by solving a convex optimization problem. Finally, we show that the equilibrium is robust, both theoretically, and through an extensive experimental evaluation.

Acknowledgements

We would like to thank Shiying Li in the Department of Mathematics at Vanderbilt University for her valuable and constructive suggestions for the proof of Theorem 4. This work was partially supported by the National Science Foundation (CNS-1640624, IIS-1526860, IIS-1649972), Office of Naval Research (N00014-15-1-2621), Army Research Office (W911NF-16-1-0069), and National Institutes of Health (R01HG006844-05).

References

Alfeld, S., Zhu, X., and Barford, P. Data poisoning attacks against autoregressive models. In *AAAI Conference on Artificial Intelligence*, 2016.

Brückner, M. and Scheffer, T. Stackelberg games for adversarial prediction problems. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 547–555, 2011.

Cheng, S.-F., Reeves, D. M., Vorobeychik, Y., and Wellman, M. P. Notes on equilibria in symmetric games. In *International Workshop on Game Theoretic and Decision Theoretic Agents*, pp. 71–78, 2004.

Cortez, P., Cerdeira, A., Almeida, F., Matos, T., and Reis, J. Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, 47(4): 547–553, 2009.

Dalvi, N., Domingos, P., Mausam, Sanghai, S., and Verma, D. Adversarial classification. In *SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 99–108, 2004.

Grosshans, M., Sawade, C., Brückner, M., and Scheffer, T. Bayesian games for adversarial regression problems. In *International Conference on International Conference on Machine Learning*, pp. 55–63, 2013.

Laszka, A., Lou, J., and Vorobeychik, Y. Multi-defender strategic filtering against spear-phishing attacks. In *AAAI Conference on Artificial Intelligence*, 2016.

Li, B. and Vorobeychik, Y. Feature cross-substitution in adversarial classification. In *Advances in Neural Information Processing Systems*, pp. 2087–2095, 2014.

Li, B. and Vorobeychik, Y. Scalable optimization of randomized operational decisions in adversarial classification settings. In *Conference on Artificial Intelligence and Statistics*, 2015.

Lowd, D. and Meek, C. Adversarial learning. In *ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, pp. 641–647. ACM, 2005.

Rosen, J. B. Existence and uniqueness of equilibrium points for concave n-person games. *Econometrica*, pp. 520–534, 1965.

Russu, P., Demontis, A., Biggio, B., Fumera, G., and Roli, F. Secure kernel machines against evasion attacks. In *ACM Workshop on Artificial Intelligence and Security*, pp. 59–69, 2016.

Smith, A., Lou, J., and Vorobeychik, Y. Multidefender security games. *IEEE Intelligent Systems*, 32(1):50–60, 2017.

Šrndic, N. and Laskov, P. Practical evasion of a learning-based classifier: A case study. In *2014 IEEE Symposium on Security and Privacy*, pp. 197–211, 2014.

Stevens, D. and Lowd, D. On the hardness of evading combinations of linear classifiers. In *ACM Workshop on Artificial Intelligence and Security*, 2013.

Vorobeychik, Y., Mayo, J. R., Armstrong, R. C., and Ruthruff, J. Noncooperatively optimized tolerance: Decentralized strategic optimization in complex systems. *Physical Review Letters*, 107(10):108702, 2011.

Xu, H., Caramanis, C., and Mannor, S. Robust regression and lasso. In *Advances in Neural Information Processing Systems*, pp. 1801–1808, 2009.

Zhou, Y., Kantarcioglu, M., Thuraisingham, B. M., and Xi, B. Adversarial support vector machine learning. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1059–1067, 2012.