Explicit Defense Actions Against Test-Set Attacks

Scott Alfeld[*] and **Xiaojin Zhu**[*] and **Paul Barford**[*†]

[*]Department of Computer Sciences
University of Wisconsin–Madison
Madison WI 53706, USA

[†]comScore, Inc.
11950 Democracy Drive, Suite 600
Reston, VA 20190, USA.

{salfeld, jerryzhu, pb}@cs.wisc.edu

Abstract

Automated learning and decision making systems in public-facing applications are vulnerable to malicious attacks. Examples of such systems include spam detectors, credit card fraud detectors, and network intrusion detection systems. These systems are at further risk of attack when money is directly involved, such as market forecasters or decision systems used in determining insurance or loan rates. In this paper, we consider the setting where a predictor Bob has a fixed model, and an unknown attacker Alice aims to perturb (or *poison*) future test instances so as to alter Bob's prediction to her benefit. We focus specifically on Bob's optimal defense actions to limit Alice's effectiveness. We define a general framework for determining Bob's optimal defense action against Alice's worst-case attack. We then demonstrate our framework by considering linear predictors, where we provide tractable methods of determining the optimal defense action. Using these methods, we perform an empirical investigation of optimal defense actions for a particular class of linear models — autoregressive forecasters — and find that for ten real world futures markets, the optimal defense action reduces the Bob's loss by between 78 and 97%.

Introduction

Systems in domains such as finance, energy, medicine, entertainment, security, advertising, etc., increasingly rely on diverse input data. If decisions in these systems are based on the output of automated learning systems, then some actors may have incentive to alter (or *poison*) input data so as to affect the learned system. Thus, any such system should be robust to these threats.

The study of effective strategies in the presence of adversaries has long been of interest (Tzu Circa 500 BCE). Broadly speaking, there are two forms of data poisoning attacks against a learning system — altering the data while the model is being trained, and altering the data fed into an already learned model. In this work, we focus on the latter setting.

We consider the setting where a predictor, Bob, has a fixed, publicly known prediction function. For example, Bob may be an insurance company estimating the expected future cost of an applicant to determine terms (*e.g.,* the monthly premium) of the insurance plan offered. An actor, the adversary Alice with motivation unknown to Bob, asserts her limited control over the features (*e.g.*, to lie about her age, credit history, etc.) fed into Bob's prediction function, to attempt to pull his prediction toward her goal. An attacker is defined by her target and her loss function (measuring the distance of Bob's resulting prediction to her target), both of which we assume are unknown to Bob. Bob may select some action from a set of available *defense actions*, and aims to limit the effectiveness of any potential attacker. In our insurance example, perhaps Bob can verify certain aspects of her application with third parties.

We assume no security through obscurity. That is, Bob publishes his model and selected defense action, and Alice studies both to plan her attack. Under this setting, Bob aims to best defend himself. In this work, we answer the question: What action should Bob take to best defend against an unknown attacker, assuming worst case initial values, attacker target, and attacker loss function?

A primary goal of adversarial learning research is defense. That is, to augment or create learners (or models) so as to harden them against attackers. We address the issue of defense explicitly, by framing the interaction between attacker and predictor as a two player, non-zero sum, Stackelberg game. Specifically, in this work we make three primary contributions: (i) We define a general framework for a predictor's explicit defense strategy against intelligent, unknown adversaries. (ii) We utilize the framework to provide tractably computable optimal actions for linear predictors. (iii) We empirically demonstrate our methods on real world data sets, and perform an investigation of their properties on synthetic data.

Defense Framework

An agent Bob is a predictor with a fixed function mapping instances in an input space $\mathcal{X}$ to target values in an output space $\mathcal{Y}$. We denote Bob's fixed, presumably learned, prediction function as $f : \mathcal{X} \to \mathcal{Y}$. Our framework is applicable to general prediction. That is, Bob's task may be, *e.g.,* (binary) classification ($f : \mathbb{R}^d \to \{0, 1\}$) regression ($f : \mathbb{R}^d \to \mathbb{R}$), clustering (hard or soft) ($f : \mathbb{R}^d \to [1, \ldots, k]$ or $f : \mathbb{R}^d \to \mathcal{S}_k$), ranking ($\mathbb{R}^{d\times n} \to \binom{\{1,\ldots,k\}}{k}$) or other forms of prediction. For ease of notation we assume $\mathcal{X} \subseteq \mathbb{R}^d$

and $\mathcal{Y} \subseteq \mathbb{R}^m$, where $d, m \in \mathbb{Z}_+$.

We illustrate the concepts described in this work by use of a running example. Bob takes as input daily stock prices for Wednesday (x_W) and Thursday (x_T) and predicts values for $\hat{x}_F = 1.5x_T - 1.0x_W$ for Friday, and $\hat{x}_S = 1.5\hat{x}_F - 1.0x_T$ for Saturday.

Alice is an adversary with limited ability to perturb or *poison* test instances before Bob observes them. She aims to perform an *attractive* (Alfeld, Zhu, and Barford 2016) attack, moving Bob's prediction towards some target. After observing a test instance[1] $\boldsymbol{x} \in \mathcal{X}$, Alice will select a *poison* vector $\boldsymbol{\alpha}^{\text{atr}}$ and supply Bob with the poisoned instance $\boldsymbol{x} + \boldsymbol{\alpha}^{\text{atr}}$. We define Alice in terms of: (i) Her target $\boldsymbol{t} \in \mathcal{Y}$, which she aims to pull Bob's prediction toward[2]. (ii) Her loss function $\|\cdot\|_A$, where $\|\mathbf{0}\|_A = 0$ and $\|\boldsymbol{a}\|_A > 0 \; \forall \boldsymbol{a} \neq \mathbf{0}$. (iii) Her set of feasible attacks $\mathcal{A}$. (iv) Her effort function $g(\cdot) : \mathcal{A} \to \mathbb{R}$, defining the costs she incurs for a given attack ($g(\boldsymbol{\alpha}) \geq 0 \; \forall \boldsymbol{\alpha}$).

We assume a powerful attacker. Namely, Alice has full knowledge of Bob's model, and will select the attack which minimizes the sum of her loss and effort. Formally, Alice selects the optimal attack by solving:

$$\boldsymbol{\alpha}^{\text{atr}}\left(\mathcal{A}, \boldsymbol{x}, \boldsymbol{t}, \|\cdot\|_A, g(\cdot)\right) \stackrel{\text{def}}{=} \arg\min_{\boldsymbol{\alpha} \in \mathcal{A}} \|f(\boldsymbol{x} + \boldsymbol{\alpha}) - \boldsymbol{t}\|_A + g(\boldsymbol{\alpha}) \quad (1)$$

For a variety of settings, there are known, tractable methods for computing Alice's optimal attack (Alfeld, Zhu, and Barford 2016). We instead focus on Bob, defining a framework for determining his optimal method of defending against an unknown adversary Alice.

We phrase the interplay between Alice and Bob as a one-shot, two-player, non-zero-sum, Stackelberg game. For brevity, we restrict our attention to settings where Bob considers only pure (as opposed to mixed) strategies, and his actions are to further restrict Alice. Our methods, however, extend beyond this. In further interest of clarity, we make the order of events explicit: (1) Bob selects action β from his set of potential actions $\mathcal{B}$. (2) Alice observes f, β, and $\boldsymbol{x}$. (3) Alice selects her poison vector $\boldsymbol{\alpha}^{\text{atr}}$ (from $\mathcal{A}$ constrained by β). (4) Bob observes $\boldsymbol{x} + \boldsymbol{\alpha}^{\text{atr}}$, and suffers loss $\|f(\boldsymbol{x} + \boldsymbol{\alpha}^{\text{atr}}) - f(\boldsymbol{x})\|_B$. Note that Bob does not observe his loss, as he never observes the unpoisoned $\boldsymbol{x}$. We argue that this framework and order of events is both relevant and realistic, as an attacker may determine Bob's action and model via outside channels.

In keeping with the assumption of a powerful attacker, we assume that Bob does not know Alice's target $\boldsymbol{t}$, loss function $\|\cdot\|_A$, or effort function $g(\cdot)$, but he does know her constraints (defining $\mathcal{A}$). This allows our methods to be used in evaluating the robustness of a system against bounded attackers — Bob can evaluate the worth of limiting an attacker's abilities through *e.g.,* legal or technological means.

[1]To avoid cluttered notation we assume only one test instance. All methods described herein, however, extend easily to the case where Bob receives a test set of more than one point.

[2]Note that the attacker's target $\boldsymbol{t}$ may be absorbed into her loss function, rather than remaining a separate variable. For the sake of clarity, however, we keep both as explicit variables.

We further assume that Bob does not know the unpoisoned value $\boldsymbol{x}$ (if he did, he could simply undo Alice's attack). Bob aims to minimize the deviation, as defined by his loss function $\|\cdot\|_B$, between his prediction on the poisoned test set and what he would have predicted on the unpoisoned set. To do this, Bob selects some action $\beta \in \mathcal{B}$. As an example: by selecting action $\beta \in \mathcal{B}$, Bob reduces Alice's feasible set of attacks to $\mathcal{A}_\beta$. See later section (Extensions) for a further discussion of possible types of defense actions. Formally, Bob seeks to solve the bi-level optimization problem:

$$\arg\min_{\beta \in \mathcal{B}} \max_{\boldsymbol{x}, \|\cdot\|_A, g(\cdot), \boldsymbol{t}, \boldsymbol{\alpha}^{\text{atr}}} \left\|f(\boldsymbol{x}) - f(\boldsymbol{x} + \boldsymbol{\alpha}^{\text{atr}})\right\|_B \quad (2)$$
$$\text{s.t. } \boldsymbol{\alpha}^{\text{atr}} = \arg\min_{\boldsymbol{\alpha} \in \mathcal{A}_\beta} \|f(\boldsymbol{x} + \boldsymbol{\alpha}) - \boldsymbol{t}\|_A + g(\boldsymbol{\alpha})$$

To simplify, we exploit the duality between considering all possible attractive attacks, and considering the one *repulsive* attack — the attack which explicitly aims to maximize Bob's loss. We define the following single-level optimization problem, equivalent to (2):

$$\arg\min_{\beta \in \mathcal{B}} \max_{\boldsymbol{x} \in \mathbb{R}^d, \boldsymbol{\alpha} \in \mathcal{A}_\beta} \|f(\boldsymbol{x}) - f(\boldsymbol{x} + \boldsymbol{\alpha})\|_B \quad (3)$$

In essence, we are creating a phantom adversary performing the optimal repulsive attack. We then have Bob defend against this phantom Alice, thus limiting the potential effect of any attacker. In doing so, we formulate the problem in standard minimax form (a zero-sum game) rather than a bi-level optimization problem. Rather than maximizing over all possible targets, effort, and loss functions, we now maximize over only the potential initial values $\boldsymbol{x}$ and Alice's attack $\boldsymbol{\alpha}$.

We note that Bob and any particular attacker Alice are engaged in a non-zero sum game. However, because Bob defends against an *unknown* attacker, he considers the worst-case attacker, against which he plays a zero-sum game. We express this formally in the following theorem.

Theorem 1. *The bi-level optimization problem (2) is equivalent to standard minimax problem (3).*

Proof. For a fixed $\beta, \boldsymbol{x}$, let:

$$e_1 \stackrel{\text{def}}{=} \max_{\|\cdot\|_A, g(\cdot), \boldsymbol{t}, \boldsymbol{\alpha}^{\text{atr}}} \left\|f(\boldsymbol{x}) - f(\boldsymbol{x} + \boldsymbol{\alpha}^{\text{atr}})\right\|_B \quad (4)$$
$$\text{s.t. } \boldsymbol{\alpha}^{\text{atr}} = \arg\min_{\boldsymbol{\alpha} \in \mathcal{A}_\beta} \|f(\boldsymbol{x} + \boldsymbol{\alpha}) - \boldsymbol{t}\|_A + g(\boldsymbol{\alpha})$$

$$e_2 \stackrel{\text{def}}{=} \max_{\boldsymbol{\alpha} \in \mathcal{A}_\beta} \|f(\boldsymbol{x}) - f(\boldsymbol{x} + \boldsymbol{\alpha})\|_B \quad (5)$$

We prove the statement by showing that $e_1 = e_2 \; \forall \beta, \boldsymbol{x}$.

$e_1 \leq e_2$: Note that e_2 is the maximum over a relaxed version of the maximization problem of e_1. That is, removing the constraint on $\boldsymbol{\alpha}^{\text{atr}}$ in (4) yields (5). Therefore $e_1 \leq e_2$.

$e_1 \geq e_2$: It is here that we construct the phantom attacker. Let $\tilde{g}(\cdot) \stackrel{\text{def}}{=} 0$, $\tilde{\boldsymbol{t}} \stackrel{\text{def}}{=} f(\boldsymbol{x} + \boldsymbol{\alpha}^{\text{rep}})$, $\|\cdot\|_{\tilde{A}} \stackrel{\text{def}}{=} \|\cdot\|_B$ where $\boldsymbol{\alpha}^{\text{rep}}$ is the optimal repulsive attack:

$$\boldsymbol{\alpha}^{\text{rep}} \stackrel{\text{def}}{=} \arg\max_{\boldsymbol{\alpha} \in \mathcal{A}_\beta} \|f(\boldsymbol{x}) - f(\boldsymbol{x} + \boldsymbol{\alpha})\|_B \quad (6)$$

Pluggin in terms we have:

$$\arg\min_{\boldsymbol{\alpha}\in\mathcal{A}_\beta} \left\|f(\boldsymbol{x}+\boldsymbol{\alpha})-\tilde{\boldsymbol{t}}\right\|_{\tilde{A}}+\tilde{g}(\boldsymbol{\alpha}) \tag{7}$$

$$=\arg\min_{\boldsymbol{\alpha}\in\mathcal{A}_\beta} \|f(\boldsymbol{x}+\boldsymbol{\alpha})-f(\boldsymbol{x}+\boldsymbol{\alpha}^{\text{rep}})\|_B+0 \tag{8}$$

$$=\boldsymbol{\alpha}^{\text{rep}} \tag{9}$$

where the second equality holds by the fact that $\|0\|_B=0$ and $\|\cdot\|_B\geq 0$. Let

$$\tilde{e}_1 \overset{\text{def}}{=} \max_{\boldsymbol{\alpha}^{\text{atr}}} \left\|f(\boldsymbol{x})-f(\boldsymbol{x}+\boldsymbol{\alpha}^{\text{atr}})\right\|_B \tag{10}$$

$$\text{s.t. } \boldsymbol{\alpha}^{\text{atr}}=\arg\min_{\boldsymbol{\alpha}\in\mathcal{A}_\beta} \left\|f(\boldsymbol{x}+\boldsymbol{\alpha})-\tilde{\boldsymbol{t}}\right\|_{\tilde{A}}+\tilde{g}(\boldsymbol{\alpha})$$

$$=\|f(\boldsymbol{x})-f(\boldsymbol{x}+\boldsymbol{\alpha}^{\text{rep}})\|_B \tag{11}$$

$$=e_2 \tag{12}$$

Note that in (4), we are maximizing over all possible $\|\cdot\|_A, g(\cdot), \boldsymbol{t}, \boldsymbol{\alpha}^{\text{atr}}$ whereas in (10) we use specific instantiations. Therefore $e_1\geq\tilde{e}_1=e_2$. $\square$

Linear Predictors

Thus far we have defined a framework (in (3)) for selecting an optimal defense action for a general predictor. In what follows, we utilize this framework, and describe instantiations of Alice and Bob inspired by real-world settings. These instantiations result in tractable methods for determining Bob's optimal defense action. For simplicity, we assume that $\mathcal{B}$ is a finite set. In this setting, the task of determining Bob's optimal defense action reduces to computing the optimal repulsive attack — we evaluate the optimal repulsive attack for each $\beta\in\mathcal{B}$. Cases where Bob has a continuous or countable infinite set of actions are left as future work.

When Bob's loss is convex, it is often tractable to compute Alice's optimal attractive attack — that is, minimizing a quadratic function subject to hard constraints. However, even when Bob's loss is convex, the task of computing an optimal *repulsive* attack — *maximizing* a quadratic function subject to hard constraints — is NP-Hard in general (*e.g.,* under box constraints) (Nocedal and Wright 2006). We consider a subset of all possible predictors (Bobs) and attackers (Alices) so as to yield tractable methods for solving (3).

We consider the case where $\mathcal{Y}$ is continuous (*e.g.,* regression) and let Bob be a (homogeneous) linear predictor. That is, his prediction function may be written as:

$$f(\boldsymbol{x}) \overset{\text{def}}{=} M\boldsymbol{x} \tag{13}$$

for some matrix M. In our running example of forecasting Friday and Saturday's stock price,

$$M^{(\text{RE})}=\begin{bmatrix}-1 & 1.5\\ -1.5 & 1.25\end{bmatrix}, \boldsymbol{x}=\begin{bmatrix}x_W\\ x_T\end{bmatrix} \tag{14}$$

We note that by linearity of f,

$$\boldsymbol{\alpha}^{\text{rep}}(\mathcal{A},\boldsymbol{x},\|\cdot\|_B) \overset{\text{def}}{=} \arg\max_{\boldsymbol{\alpha}\in\mathcal{A}} \|f(\boldsymbol{x}+\boldsymbol{\alpha})-f(\boldsymbol{x})\|_B$$

$$=\arg\max_{\boldsymbol{\alpha}\in\mathcal{A}} \|f(\boldsymbol{\alpha})\|_B \tag{15}$$

$$\overset{\text{def}}{=} \boldsymbol{\alpha}^{\text{rep}}(\mathcal{A},\|\cdot\|_B) \tag{16}$$

That is, *the optimal repulsive attack is independent of the test instance*. This results in (3) being equivalent to:

$$\arg\min_{\beta\in\mathcal{B}}\max_{\boldsymbol{\alpha}\in\mathcal{A}_\beta} \|f(\boldsymbol{\alpha})\|_B$$

$$=\arg\min_{\beta\in\mathcal{B}} \|f(\boldsymbol{\alpha}^{\text{rep}}(\mathcal{A}_\beta,\|\cdot\|_B))\|_B \tag{17}$$

We let Bob's loss be a generalization of mean squared error — the squared Mahalanobis norm:

$$\|f(\boldsymbol{\alpha})\|_B \overset{\text{def}}{=} \|f(\boldsymbol{\alpha})\|_W^2 = f(\boldsymbol{\alpha})^\top W f(\boldsymbol{\alpha}) \tag{18}$$

where $W=V^\top V$ is a positive-definite matrix. Note that we do not require V to be unique. This both yields mathematical benefits, and is applicable in many real world settings. Similarly, we consider the case where each of Bob's actions restricts Alice to select an attack from some ellipsoid. One such setting where this is natural is when Alice is already restricted to an ellipsoid (as in (Alfeld, Zhu, and Barford 2016) or when she has a constraint on the ℓ_2 norm of her poison vector) and Bob's actions each reweight the constraints along some features. That is, each of Bob's actions $\beta\in\mathcal{B}$ defines Alice's feasible attacks as:

$$\mathcal{A}_\beta \overset{\text{def}}{=} \{\boldsymbol{\alpha} \,:\, \|\boldsymbol{\alpha}\|_{C_\beta}\leq c\} \tag{19}$$

where $\|\boldsymbol{\alpha}\|_{C_\beta}=\sqrt{(G_\beta\boldsymbol{\alpha})^\top G_\beta\boldsymbol{\alpha}}$, $C_\beta=G_\beta^\top G_\beta$ is a positive-definite matrix and $c\in\mathbb{R}_+$.

Under these conditions, we may leverage the alignment of Bob's loss with Alice's constraints (in that they are both quadratic) to convert the optimization problem to that of computing an induced matrix norm. By Theorem 2, we have:

$$\boldsymbol{\alpha}^{\text{rep}} \overset{\text{def}}{=} cG^{-1}\boldsymbol{s}_1 \tag{20}$$

where $\boldsymbol{s}_1$ is the right-singular vector corresponding to the largest singular value of VMG_β^{-1}. Therefore $\boldsymbol{\alpha}^{\text{rep}}$ may be found by computing G^{-1} and the SVD of VMG^{-1}. We note that for large matrices, $\boldsymbol{\alpha}^{\text{rep}}$ may be approximated by applying the power method to $(VMG^{-1})^\top(VMG^{-1})$.

Theorem 2. *Let $\boldsymbol{\alpha}^{rep}$ be the optimal repulsive attack against a linear predictor Bob with prediction function $f(\boldsymbol{x})=M\boldsymbol{x}$ and (squared Mahalanobis) loss $\|f(\boldsymbol{\alpha})\|_B=\|f(\boldsymbol{\alpha})\|_W^2=f(\boldsymbol{\alpha})^\top Wf(\boldsymbol{\alpha})$, where $W=V^\top V$ is a positive definite matrix, under the general ball constraints $\mathcal{A}=\{\boldsymbol{\alpha}\,:\,\|\boldsymbol{\alpha}\|_C\leq c\}$ for the attacker, where $C=G^\top G$ is a positive-definite matrix and $c\in\mathbb{R}_+$ is a constant. Then the repulsive attack as defined in (20) $\boldsymbol{\alpha}^{rep}$ is equal to $cG^{-1}\boldsymbol{s}_1$ where $\boldsymbol{s}_1$ is the right-singular vector corresponding to the largest singular value of VMG^{-1}. Further, Bob's loss induced by $\boldsymbol{\alpha}^{rep}$ is the squared spectral norm of $cVMG^{-1}$.*

Proof. Recall that the optimal repulsive attack is given by:

$$\boldsymbol{\alpha}^{\text{rep}}=\arg\max_{\boldsymbol{\alpha}} \|f(\boldsymbol{\alpha})\|_B \tag{21}$$

$$\text{s.t. } \|\boldsymbol{\alpha}\|_C\leq c \tag{22}$$

Note that:

$$\|f(\boldsymbol{\alpha})\|_B=\|M\boldsymbol{\alpha}\|_B=\|M\boldsymbol{\alpha}\|_W^2 \tag{23}$$

$$=\boldsymbol{\alpha}^\top M^\top WM\boldsymbol{\alpha}=\boldsymbol{\alpha}^\top M^\top V^\top VM\boldsymbol{\alpha} \tag{24}$$

$$=\|VM\boldsymbol{\alpha}\|_2^2=\|VM\underbrace{G^{-1}G}_{I}\boldsymbol{\alpha}\|_2^2 \tag{25}$$

Similarly:

$$\|\boldsymbol{\alpha}\|_C = \|G\boldsymbol{\alpha}\|_2 \tag{26}$$

This yields the following optimization problem:

$$\boldsymbol{\alpha}^{\text{rep}} = \arg\max_{\boldsymbol{\alpha}} \|VMG^{-1}G\boldsymbol{\alpha}\|_2^2 \tag{27}$$

$$\text{s.t. } \|G\boldsymbol{\alpha}/c\|_2 \leq 1 \tag{28}$$

We then perform a change of variables letting $\tilde{\boldsymbol{\alpha}} = G\boldsymbol{\alpha}/c$, yielding[3]:

$$\tilde{\boldsymbol{\alpha}}^{\text{rep}} = \arg\max_{\tilde{\boldsymbol{\alpha}}} \|VMG^{-1}\tilde{\boldsymbol{\alpha}}\|_2^2 \tag{29}$$

$$\text{s.t. } \|\tilde{\boldsymbol{\alpha}}\|_2 \leq 1 \tag{30}$$

The spectral norm (Horn and Johnson 2012) of a matrix Z is defined as the induced 2-norm:

$$\max_{\boldsymbol{x}} \|Z\boldsymbol{x}\|_2 \tag{31}$$

$$\text{s.t.} \|\boldsymbol{x}\|_2 \leq 1 \tag{32}$$

and is equal to the largest singular value of Z (with the $\boldsymbol{x}$ that maximizes being the corresponding right singular vector). Accounting for the change of variables, we have:

$$\boldsymbol{\alpha}^{\text{rep}} = cG^{-1}\tilde{\boldsymbol{\alpha}}^{\text{rep}} \tag{33}$$

$$= cG^{-1}\boldsymbol{s}_1 \tag{34}$$

where $\boldsymbol{s}_1$ is the right singular vector corresponding to the spectral norm of VMG^{-1}. □

Experiments

While the framework we have described is broadly applicable to linear predictors, here we focus on the setting where Bob is forecasting future values of a time series. Specifically, we use a linear autoregressive model and recursive forecasting strategy.

We select this setting based on the following motivations:

i. Data manipulation attacks are a real-world concern in forecasting.
ii. Prior work (Alfeld, Zhu, and Barford 2016) has determined optimal (attractive) attacks against linear forecasters, specifically in the context of futures markets.
iii. Linear auto-regressive models for univariate time series yield an intuitive understanding of coefficients. θ_i is the weight of i steps ago when predicting the current value.

Bob uses the values for the last d time periods $x_{-d}, \ldots, x_{-1}$ to forecast the next h values into the future $(\hat{x}_0, \ldots, \hat{x}_{h-1})$. He does so with an order-d linear autoregressive model: $x_t = \sum_{i=1}^{d} \theta_i x_{t-i}$, and recursive forecasting strategy. Without loss of generality we assume $h > d$. We note that the forecasting function may be written as:

$$f(\boldsymbol{x}) = M\boldsymbol{x} \overset{\text{def}}{=} S^h Z\boldsymbol{x} \tag{35}$$

[3]For any norm $\|\cdot\|$, matrix Z, vector $\boldsymbol{x}$ and constant k, $||Z(k\boldsymbol{x})|| = ||kZ\boldsymbol{x}|| = k||Z\boldsymbol{x}||$

Where S is the $h \times h$ one-step matrix for model $\boldsymbol{\theta}$, and Z is the $h \times d$ zero-padding matrix.

$$S \overset{\text{def}}{=} \left[\begin{array}{c|cc} \mathbf{0}_h & \multicolumn{2}{c}{I_{(h-1)\times(h-1)}} \\ \hline & \mathbf{0}^\top_{(h-d-1)\times 1} & \overleftarrow{\boldsymbol{\theta}}^\top \end{array}\right], Z \overset{\text{def}}{=} \begin{bmatrix} \mathbf{0}_{(h-d)\times d} \\ I_{d\times d} \end{bmatrix}$$

Where we denote the reverse of $\boldsymbol{\theta}$ as $\overleftarrow{\boldsymbol{\theta}}$: $\overleftarrow{\theta}_i = \theta_{d-i+1}$.

We assume that Alice's original set $\mathcal{A}$ of feasible attacks is an d-ellipsoid defined by $\{\boldsymbol{\alpha} : \|\boldsymbol{\alpha}\|_C \leq c\}$. In defining Bob's set of possible actions $\mathcal{B}$, we consider performing an inspection on a single time period. In the general setting of prediction, this is akin to Bob independently verifying a single feature. For simplicity of demonstration, we let one time period be one day, and assume that on the day Bob performs an inspection, Alice is unable to lie — on all other days, Alice is bound by her original set of feasible attacks $\mathcal{A}$. Therefore Bob is restricting her to a $(d-1)$-ellipsoid. Recall that Bob's actions may be defined in terms of G_β^{-1} directly. For the action β corresponding to inspecting day i, we let $G_\beta^{-1}[i,j] = G_\beta^{-1}[j,i] = 0$ for all j. This encodes Alice's inability to affect day i; her value for $\boldsymbol{\alpha}_i$ is ignored. Note that other, non-elliptical constraints may also be used, which we discuss in a later section.

We use this construction of $\mathcal{B}$ to illustrate a subtle yet powerful flexibility of our framework. Namely, the different $\mathcal{A}_\beta$ need not have the same dimension. Note that the definition of feasible attacks does not directly yield a set defined by a Mahalanobis norm as in (19). To encode the restriction that Alice cannot poison day i into the condition $\|\cdot\|_C \leq c$, it must be that for all j, $C_{ij} = C_{ji} = \infty$. Therefore C is not in the space of real matrices, and the definitions of G and G^{-1} are unclear. However, in computing the optimal repulsive attack, C is never used explicitly (nor is G) — only G^{-1}. By letting $G_{ij}^{-1} = G_{ji}^{-1} = 0$ for all j, we are able to encode Alice's inability to affect day i; whatever value she selects for $\boldsymbol{\alpha}_i$ will be ignored.

We illustrate the process of computing the optimal defense action using our running example where Bob uses Wednesday and Thursday to predict Friday and Saturday. In this case, we find that inspecting Wednesday (meaning Bob ensures that $\alpha_{\text{W}} = 0$) allows an optimal attack of $\alpha_{\text{T}} = 1$, for which Bob suffers loss ≈ 3.8. In contrast, inspecting Thursday allows an optimal attack $\alpha_{\text{W}} = 1$, and Bob's loss is 3.25 — inspecting Thursday is the optimal defense.

For brevity, we consider only $d = 5, h = 7$ with spherical loss (total squared deviation) for Bob and (unit) spherical constraints for Alice for each experiment: $W = C = I, c = 1$. Because computing the optimal repulsive attack is accomplished via the SVD, runtime was not a concern — all experiments ran quickly on consumer hardware. All figures were made with Matplotlib (Hunter 2007) v 1.5.1.

Futures Markets Experiments

For ten different futures markets, we obtained[4] daily settle price data. For each, we estimated $\boldsymbol{\theta}$ using Yule-Walker esti-

[4]Data is freely available from www.quandl.com. Identification codes for individual datasets are provided in Figure 1.

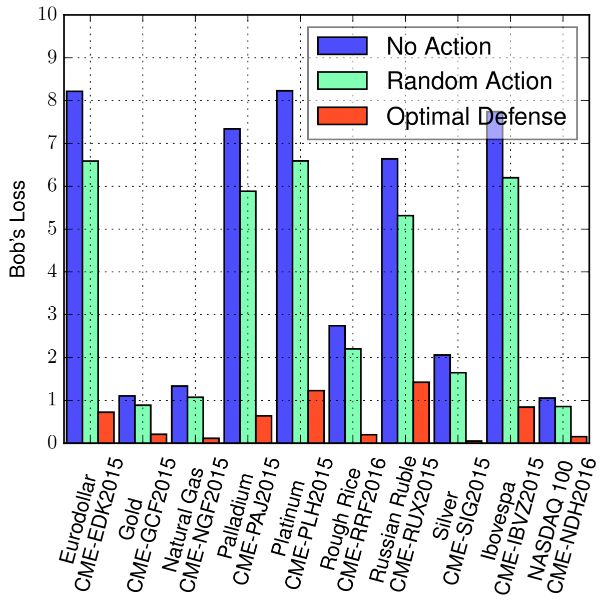

Figure 1: For each the 10 futures markets, we show Bob's loss against the worst case attack if he takes no defense action (blue), a random defense action (green), and his optimal defense action (red).

mation (Box, Jenkins, and Reinsel 2011) on approximately one month's worth of centered data — the exact dates used varied across markets based on values available. We compared Bob's loss under the optimal defense, selecting a day to inspect at random, and the null strategy where he takes no action on each future market. We report loss under all three strategies in Figure 1. Universally, taking the optimal defense action considerably reduces Bob's worst-case loss — defending resulted in a 78% (Russian Ruble) to $> 97\%$ (Silver) reduction in loss compared to the null strategy across the ten markets. Note that while we used $c = 1$ for these experiments, the percentages are independent of c. By Theorem 2 and the fact that for any matrix norm $||\cdot||$, constant k and matrix M we have $||kM|| = |k|||M||$, Bob's loss scales with c^2. We note that on each futures market, the optimal action was to lock down day -1 (the last day), and for each learned model we find $|\theta_1| > |\theta_i|, i = 2, \ldots, 5$.

Synthetic Experiments

From the futures markets data, as well as our running example, one may be tempted to form two natural hypotheses: (a) The optimal action is always to select the day $i_{\max}$ corresponding to the maximal (in magnitude) θ_i: $i_{\max} \overset{\text{def}}{=} -\arg\max_i |\theta_i|$. (b) The optimal action is always to select the last day. Hypothesis (a) is supported by the observation that Bob's first prediction will be most affected by the value $\boldsymbol{\alpha}_{i_{\max}}$, and subsequent predictions will in turn be affected by the first. Hypothesis (b), in contrast, is motivated by the observation that while x_{-d} directly affects only $\hat{x}_0$ (all later predictions are affected by x_{-d} only through $\hat{x}_0$), the value x_{-1} directly affects predictions $\hat{x}_0, \ldots, \hat{x}_{d-1}$.

To test these hypotheses, we run an additional experiment. To emulate models that may be encountered in practice, we construct 10,000 stationary models (Box, Jenkins, and Reinsel 2011) $\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(10000)}$ by drawing each $\boldsymbol{\theta}^{(i)}$ *iid* from a standard Gaussian, and then rejecting any non-stationary models. We then determine the percentage of models on which hypotheses (a) and (b) yield the optimal defense action. We find that selecting the day corresponding with the maximal $|\theta_i|$ (hypothesis (a)) is optimal only $\approx 55\%$ of the time. Selecting the last day (hypothesis (b)) is optimal only $\approx 49\%$ of the time. We further note that in $\approx 24\%$ of models, neither the last day nor the day $i_{\max}$ was optimal, whereas in $\approx 29\%$ both were optimal (and in $\approx 31\%$, $i_{\max}$ was the last day —- that is, the two hypotheses suggested the same day).

To further explore this behavior, we conduct a second experiment where we focus specifically on the case of $d = 2$. In following with our running example, we consider predicting Friday and Saturday using Wednesday and Thursday. For $(\theta_1, \theta_2) \in [-3, 3]^2$ with a resolution of 0.01, we determine the optimal action. In Figure 2 (top left) we show the space of models, and we mark black each model for which it is better to inspect Thursday (white indicates it is better to inspect Wednesday). We observe a "bat-wing" effect, where the boundary between sets of models with the same optimal defense action is non-linear, and each set is non-convex. Note that hypothesis (a) would result in the figure having an "X" shape with the left and right portions black, and hypothesis(b) would result in every point being black. While both are suboptimal, we note that pictorially, they are reasonable approximations to the true bat-wing structure (especially hypothesis (a)).

We further investigate this by rerunning our experiment for higher h (See Figure 2). Indeed, even in the relatively simple case of spherical loss and constraints, the optimal action is a complex function of the underlying model, and it is unclear that any simple heuristic would perform optimally. We note the symmetry in $\boldsymbol{\theta}_1$, and how the structure varies with the parity of h. Further investigation of these phenomena is left as future work.

Extensions

We have described above only a subset of examples of our framework. In this section we briefly describe extensions of two forms: other types of defense actions and loss functions.

In what we have described thus far, Bob's action defines the feasible set of attacks for Alice. Our methods, however, easily extend to other forms of $\mathcal{B}$. Consider for example the case where Bob's actions are to select one of a set of potential models $\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(k)}$. Here, Bob is not affecting $\mathcal{A}$, but we determine his best action similar to as before. For model $\boldsymbol{\theta}^{(i)}$, let $S^{(i)}$ be the induced one-step matrix. The optimal attack is then the solution to (20) with S replaced with $S^{(i)}$.

Common loss functions in time series forecasting are the sum of absolute deviations, and maximum deviation (c.f., (Box, Jenkins, and Reinsel 2011; Alfeld and Barford 2014)). In these cases, Bob's loss function is ℓ_1 or ℓ_∞, respectively.

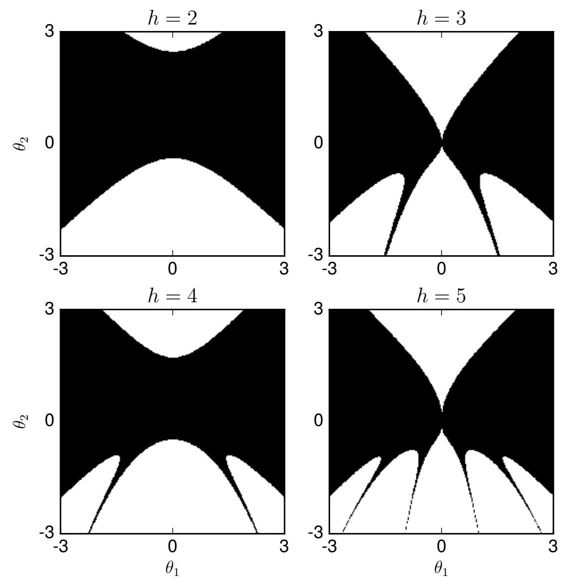

Figure 2: AR(2) models for various h. Using Wednesday and Thursday to forecast Friday and Saturday, the point (θ_1, θ_2) being black indicates that it is better (for Bob) to inspect Thursday's value than Wednesday's.

In such cases, we may still find the optimal repulsive attack by computing an induced norm, so long as Alice is constrained in a similar way ($\|\boldsymbol{\alpha}\|_1 \leq c$ or $\|\boldsymbol{\alpha}\|_\infty \leq c$). For these particular norms, the induced matrix norm is simple to find: $\|M\|_1$ is the maximum absolute row sum of M and $\|M\|_\infty$ is the maximum absolute column sum (see (Horn and Johnson 2012) for a deeper discussion of matrix norms). Other pairs of loss and constraint functions, including the cases where they differ (*e.g.,* Bob has a quadratic loss but Alice has linear constraints), often lead to intractable optimization problems, and their exploration is left as future work.

Related Work

The setting of so-called test-set attacks has been examined under a variety of titles. One such example is "evasion attacks", where the predictor performs binary classification (*e.g.,* spam detection (Nelson et al. 2009; Lowd and Meek 2005), intrusion detection (Tan, Killourhy, and Maxion 2002)) and the attacker aims to have their bad (*e.g.,* "spam" or "intrusion") sample classified as good (*e.g.,* "ham" or "normal traffic"). Neural Networks for computer vision have been investigated in the context of security (Goodfellow, Shlens, and Szegedy 2014; Papernot et al. 2016). Robust Learning (Globerson and Roweis 2006; El Ghaoui et al. 2003) considers the setting where a test set is drawn from a distribution distinct from the training set's. The setting presented herein is distribution free, and an example of covariate shift (Quionero-Candela et al. 2009). (Goodfellow, Shlens, and Szegedy 2015) argues that linearity in the models is a primary cause of attack vulnerability. This theory is supported by our work and warrants further investigation.

Separate from test-set attacks, training set attacks have also been heavily explored. Support vector machines have received special attention (Biggio, Nelson, and Laskov 2012; Biggio et al. 2014; Xiao et al. 2014; Xiao, Xiao, and Eckert 2012). Adversarially corrupting training sets has been put into the framework of machine teaching (Goldman, Rivest, and Schapire 1993; Goldman and Kearns 1995; Zhu 2015). A general framework, similar to the one presented here was recently proposed for training-set attacks against learners in (Mei and Zhu 2015).

A primary goal of this line of research is defense. We borrow from the framework and methodology used in (Alfeld, Zhu, and Barford 2016), which derived optimal (attractive) attacks against autoregressive forecasters. A separate line of research has posed the problem of learning in the presence of adversaries in game theoretic contexts (Liu and Chawla 2009; Brückner, Kanzow, and Scheffer 2012; Dalvi et al. 2004; Brückner and Scheffer 2009; 2011; Hardt et al. 2016; Brückner and Scheffer 2011). (Dalvi et al. 2004) and (Letchford and Vorobeychik 2013) phrase the interplay between Alice and Bob as game similar to ours, and specifically addresses Bob's defense strategy. They consider training-set attacks against classifiers and find the Nash equilibrium of the induced game.

Conclusions

The framework for our study is a predictor targeted by an attacker which seeks to influence its predictions. Our goal is to identify an optimal defense against such an attack. We allowed for a powerful, knowledgeable attacker, yielding a two player, non-zero sum Stackelberg game. By, in essence, constructing a phantom attacker based on Bob's loss function, we are able to phrase this interplay as a standard minimax formulation. We utilize our framework to identify the optimal defense action for worst-case attacks against linear predictors, which may be computed by reducing to the classicial problem of calculating an induced matrix norm.

We conducted empirical experiments on autoregressive forecasting models trained on real-world futures markets data, considering actions of inspecting (limiting the attacker's ability to lie) a single day. We found that the optimal defense action reduces Bob's loss by between 78 and 97%. Through further investigation on synthetic data, we discovered that the mapping from model to optimal defense action is highly non-linear, and simple heuristics for selecting a defense action fail in general. In future work, we plan to apply our methods to non-linear predictors, in hopes to derive tractable methods of identifying optimal defense actions.

Acknowledgments

This work is supported in part by NSF grants IIS-0953219, IIS-1623605, DGE-1545481, and by DHS grant BAA 11-01 and AFRL grant FA8750-12-2-0328. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the NSF, DHS or AFRL.

References

Alfeld, S., and Barford, P. 2014. Targeted residual analysis for improving electric load forecasting. In *Energy Conference (ENERGYCON), 2014 IEEE International*, 459–466. IEEE.

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

Biggio, B.; Corona, I.; Nelson, B.; Rubinstein, B. I.; Maiorca, D.; Fumera, G.; Giacinto, G.; and Roli, F. 2014. Security evaluation of support vector machines in adversarial environments. In *Support Vector Machines Applications*. Springer. 105–153.

Biggio, B.; Nelson, B.; and Laskov, P. 2012. Poisoning attacks against support vector machines. In *Twenty-Ninth International Conference on Machine Learning*.

Box, G.; Jenkins, G.; and Reinsel, G. 2011. *Time series analysis: forecasting and control*, volume 734. Wiley.

Brückner, M., and Scheffer, T. 2009. Nash equilibria of static prediction games. In *Advances in neural information processing systems*.

Brückner, M., and Scheffer, T. 2011. Stackelberg games for adversarial prediction problems. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM.

Brückner, M.; Kanzow, C.; and Scheffer, T. 2012. Static prediction games for adversarial learning problems. *The Journal of Machine Learning Research*.

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

El Ghaoui, L.; Lanckriet, G. R. G.; Natsoulis, G.; et al. 2003. *Robust classification with interval data*. Computer Science Division, University of California.

Globerson, A., and Roweis, S. 2006. Nightmare at test time: robust learning by feature deletion. In *Proceedings of the 23rd international conference on Machine learning*, 353–360. ACM.

Goldman, S. A., and Kearns, M. J. 1995. On the complexity of teaching. *Journal of Computer and System Sciences* 50(1):20–31.

Goldman, S. A.; Rivest, R. L.; and Schapire, R. E. 1993. Learning binary relations and total orders. *SIAM Journal on Computing* 22(5):1006–1034.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2015. Explaining and harnessing adversarial examples. *International Conference on Learning Representations*.

Hardt, M.; Megiddo, N.; Papadimitriou, C.; and Wootters, M. 2016. Strategic classification. In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science*, 111–122. ACM.

Horn, R. A., and Johnson, C. R. 2012. *Matrix analysis*. Cambridge university press.

Hunter, J. D. 2007. Matplotlib: A 2d graphics environment. *Computing In Science & Engineering* 9(3):90–95.

Letchford, J., and Vorobeychik, Y. 2013. Optimal interdiction of attack plans. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, 199–206. International Foundation for Autonomous Agents and Multiagent Systems.

Liu, W., and Chawla, S. 2009. A game theoretical model for adversarial learning. In *Data Mining Workshops, 2009. ICDMW'09. IEEE International Conference on*, 25–30. IEEE.

Lowd, D., and Meek, C. 2005. Good word attacks on statistical spam filters. In *Conference on Email and Anti-Spam*.

Mei, S., and Zhu, X. 2015. Using machine teaching to identify optimal training-set attacks on machine learners. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Nelson, B.; Barreno, M.; Chi, F. J.; Joseph, A. D.; Rubinstein, B. I.; Saini, U.; Sutton, C.; Tygar, J.; and Xia, K. 2009. Misleading learners: Co-opting your spam filter. In *Machine learning in cyber trust*. Springer. 17–51.

Nocedal, J., and Wright, S. 2006. *Numerical optimization*. Springer Science & Business Media.

Papernot, N.; McDaniel, P.; Goodfellow, I.; Jha, S.; Berkay Celik, Z.; and Swami, A. 2016. Practical black-box attacks against deep learning systems using adversarial examples. *arXiv preprint arXiv:1602.02697*.

Quionero-Candela, J.; Sugiyama, M.; Schwaighofer, A.; and Lawrence, N. D. 2009. *Dataset shift in machine learning*. The MIT Press.

Tan, K. M.; Killourhy, K. S.; and Maxion, R. A. 2002. Undermining an anomaly-based intrusion detection system using common exploits. In *Recent Advances in Intrusion Detection*, 54–73. Springer.

Tzu, S. Circa 500 B.C.E. *The Art of War*.

Xiao, H.; Biggiob, B.; Nelson, B.; Xiao, H.; Eckerta, C.; and Rolib, F. 2014. Support vector machines under adversarial label contamination. *Neurocomputing*.

Xiao, H.; Xiao, H.; and Eckert, C. 2012. Adversarial label flips attack on support vector machines. In *European Conference on Artificial Intelligence*.

Zhu, X. 2015. Machine teaching: An inverse problem to machine learning and an approach toward optimal education. In *AAAI Conference on Artificial Intelligence (Senior Member Track)*.