Using Machine Teaching to Identify Optimal Training-Set Attacks on Machine Learners

Shike Mei and Xiaojin Zhu
Department of Computer Sciences,
University of Wisconsin-Madison, Madison WI 53706, USA
{mei, jerryzhu}@cs.wisc.edu

Abstract

We investigate a problem at the intersection of machine learning and security: training-set attacks on machine learners. In such attacks an attacker contaminates the training data so that a specific learning algorithm would produce a model profitable to the attacker. Understanding training-set attacks is important as more intelligent agents (e.g. spam filters and robots) are equipped with learning capability and can potentially be hacked via data they receive from the environment. This paper identifies the optimal training-set attack on a broad family of machine learners. First we show that optimal training-set attack can be formulated as a bilevel optimization problem. Then we show that for machine learners with certain Karush-Kuhn-Tucker conditions we can solve the bilevel problem efficiently using gradient methods on an implicit function. As examples, we demonstrate optimal training-set attacks on Support Vector Machines, logistic regression, and linear regression with extensive experiments. Finally, we discuss potential defenses against such attacks.

Introduction

The study on security threats to intelligent agents has a long history (Nelson et al. 2009; Tan, Killourhy, and Maxion 2002; Rubinstein et al. 2008; Barreno et al. 2010; Laskov and Lippmann 2010; Barreno et al. 2006; Dalvi et al. 2004; Liu and Chawla 2009; Biggio, Fumera, and Roli 2013; Laskov and Kloft 2009). One important type of threat is training-set attack (a.k.a. causative or poisoning attack), where an attacker modifies the training data in order to mislead a machine learner toward a model profitable to the attacker. We foresee training-set attacks to increase in the future as more intelligent systems (e.g. wearable devices, cars, smart houses and robots) include a "life long learning" component. The attacker may not be able to directly hack the machine learning code inside these systems, but may readily poison the training data these systems receive.

In order to defend against training-set attacks, it is imperative to first formalize them mathematically. Prior work in training-set attacks tends to utilize heuristic computational methods and lacks a unifying framework (Xiao, Xiao, and Eckert 2012; Chung and Mok 2007; Tan, Killourhy, and Maxion 2002; Lowd and Meek 2005; Wittel and Wu 2004; Rubinstein et al. 2008). Our first contribution is a bilevel optimization framework that specifies training-set attacks. Intuitively, a malicious attacker seeking to cover their tracks wants to make the fewest manipulations to the training set as possible, and to make the learned model as close to the original solution as possible but still incorporates their target attack. We formalize this intuition as a trade-off between the attacker's effort E_A and risk R_A, to be defined below.

Our second contribution is an efficient solution to the bilevel optimization problem for a broad families of attack settings. The solution utilizes the Karush-Kuhn-Tucker (KKT) conditions to convert the bilevel problem into a single level optimization problem. Our third contribution is a demonstration of our training-set attack framework on support vector machines (SVMs), logistic regression, and linear regression. Defenses against training-set attacks are discussed at the end, and is left for future work.

Training-Set Attacks and Machine Teaching

In this paper we assume the attacker has full knowledge of the learning algorithm.[1] The attacker seeks the minimum training-set poisoning to attack the learned model. We consider machine learners that can be posed as an optimization problem:

$$\hat{\theta}_D \in \text{argmin}_{\theta \in \Theta} \quad O_L(D, \theta) \tag{1}$$

$$\text{s.t.} \quad g_i(\theta) \leq 0, \; i = 1 \ldots m \tag{2}$$

$$h_i(\theta) = 0, \; i = 1 \ldots p \tag{3}$$

where D is the training data. In classic machine learning, D is an iid sample from the underlying task distribution. $O_L(D, \theta)$ is the learner's objective: For example, in regularized risk minimization $O_L(D, \theta) = R_L(D, \theta) + \lambda\Omega(\theta)$ for some learner's empirical risk function R_L and regularizer Ω. The g and h functions are potentially nonlinear; together with the hypothesis space Θ they determine the feasible region. $\hat{\theta}_D$ is the learned model (recall argmin returns the set of minimizers).

Given the machine learner, the attacker carries out the attack by manipulating the original training data (henceforth

[1] This is a strong assumption but serves as a starting point toward understanding optimal attacks.

denoted as D_0) into D. For example, in classification the attacker may add some foreign items (x', y') to D_0, or change the value of some existing features x and labels y in D_0. The learner is unaware of the changes to the training data,[2] and learns a model $\hat{\theta}_D$ instead of $\hat{\theta}_{D_0}$. The attacker's goal is to make $\hat{\theta}_D$ beneficial to the attacker, e.g. mislead a spam filter to pass certain types of spam emails. We characterize the attacker's goal using an *attacker risk function* $R_A(\hat{\theta}_D)$. For example, the attacker may have a specific target model θ^* in mind and wants the learned model $\hat{\theta}_D$ to be close to θ^*. In this example, we may define $R_A(\hat{\theta}_D) = \|\hat{\theta}_D - \theta^*\|$ with an appropriate norm.

Meanwhile, the attacker may be constrained by certain feasible manipulations, or simply prefers "small" manipulations to evade detection. We encode the former as a search space $\mathbb{D}$ from which the attacker chooses D. For example, say the attacker can change at most B items in the original training set. We can encode it as $\mathbb{D} = \{D : |D_\Delta D_0| \le B\}$ where $_\Delta$ is set symmetric difference and $|\cdot|$ the cardinality. Separately, for the latter preference on small manipulations we encode it as an *attacker effort function* $E_A(D, D_0)$. For example, if the attack changes the design matrix X_0 in D_0 to X, we may define $E_A(D, D_0) = \|X - X_0\|_F$ the Frobenius norm of the change. We will give more concrete examples of R_A and E_A when we discuss attacks on SVMs, logistic regression, and linear regression later. Let

$$O_A(D, \hat{\theta}_D) = R_A(\hat{\theta}_D) + E_A(D, D_0) \tag{4}$$

be the overall attacker objective function. With these notations, we define the training-set attack problem as:

$$\begin{aligned}
\min_{D\in\mathbb{D},\hat{\theta}_D} \quad & O_A(D, \hat{\theta}_D) && (5)\\
\text{s.t.} \quad & \hat{\theta}_D \in \operatorname{argmin}_{\theta\in\Theta} O_L(D, \theta) && (6)\\
& \text{s.t. } \mathbf{g}(\theta) \le \mathbf{0},\ \mathbf{h}(\theta) = \mathbf{0}. && (7)
\end{aligned}$$

Note that the machine learning problem Eq (1) appears in the constraint of problem Eq (5). This is a bilevel optimization problem (Bard 1998). The optimization over data D in Eq (5) is called the upper-level problem, and the optimization over model θ given D is called the lower-level problem.

Our training-set attack formulation is closely related to *machine teaching* (Zhu 2013; Patil et al. 2014). Both aim to maximally influence a learner by carefully designing the training set. Machine teaching has focused on the education setting where the learner is a human student with an assumed cognitive model, and the teacher has an educational goal θ^* in mind. The teacher wants to design the best lesson so that the student will learn the model θ^*. There is a direct mapping from teacher to attacker and from student to intelligent agent. However, previously machine teaching formulation was only applicable to very specific learning models (e.g. conjugate exponential family models). One major contribution of the present paper is this bilevel optimization formulation of training-set attack and its efficient solutions, which significantly widens the applicability of machine teaching.

Identifying Attacks by the KKT Conditions

Bilevel optimization problems are NP hard in general. We present an efficient solution for a broad class of training-set attacks. Specifically, we require the attack space $\mathbb{D}$ to be differentiable (e.g. the attacker can change the continuous features in D for classification, or the real-valued target in D for regression). Attacks on a discrete $\mathbb{D}$, such as changing the labels in D for classification, are left as future work. We also require the learner to have a convex and regular objective O_L.

Under these conditions, the bilevel problem Eq (5) can be reduced to a single-level constrained optimization problem via the Karush-Kuhn-Tucker (KKT) conditions of the lower-level problem (Burges 1998). We first introduce KKT multipliers $\lambda_i, i = 1 \ldots m$ and $\mu_i, i = 1 \ldots p$ for the lower-level constraints $\mathbf{g}$ and $\mathbf{h}$, respectively. Since the lower-level problem is regular, we replace the lower-level problem with its KKT conditions (the constraints are stationarity, complementary slackness, primal and dual feasibility, respectively):

$$\begin{aligned}
\min_{D\in\mathbb{D},\theta,\lambda,\mu} \quad & O_A(D, \theta) && (8)\\
\text{s.t.} \quad & \partial_\theta\left(O_L(D,\theta) + \lambda^\top \mathbf{g}(\theta) + \mu^\top \mathbf{h}(\theta)\right) = \mathbf{0}\\
& \lambda_i g_i(\theta) = 0,\ i = 1 \ldots m\\
& \mathbf{g}(\theta) \le \mathbf{0},\ \mathbf{h}(\theta) = \mathbf{0},\ \lambda \ge \mathbf{0}.
\end{aligned}$$

This equivalent single-level optimization problem allows us to use the descent method on an implicit function (Colson, Marcotte, and Savard 2007). In iteration t, we update the data D by taking a projected gradient step (the upper level problem):

$$D^{(t+1)} = \mathrm{Prj}_{\mathbb{D}}\left(D^{(t)} + \alpha_t \nabla_D O_A(D, \theta^{(t)})\Big|_{D=D^{(t)}}\right), \tag{9}$$

where α_t is a step size. Then, we fix $D^{(t+1)}$ and solve for $\theta^{(t+1)}, \lambda^{(t+1)}, \mu^{(t+1)}$ which is a standard machine learning problem (the lower level problem). The gradient of the upper-level problem is computed by the chain rule

$$\nabla_D O_A(D, \theta) \quad = \quad \nabla_\theta O_A(D, \theta) \frac{\partial \theta}{\partial D}. \tag{10}$$

$\nabla_\theta O_A(D, \theta)$ is often easy to compute. Let the machine learning model θ have d parameters, and the training set D have n variables that can be manipulated. Then $\frac{\partial\theta}{\partial D}$ is the $d \times n$ Jacobian matrix, and is the main difficulty because we do not have an explicit representation of θ w.r.t. D. Instead, we note that the equalities in the KKT conditions defines a function $\mathbf{f} : \mathbb{R}^{n+d+m+p} \mapsto \mathbb{R}^{d+m+p}$:

$$\mathbf{f}(D, \theta, \lambda, \mu) = \begin{pmatrix} \partial_\theta\left(O_L(D,\theta) + \lambda^\top \mathbf{g}(\theta) + \mu^\top \mathbf{h}(\theta)\right) \\ \lambda_i g_i(\theta),\ i = 1 \ldots m \\ \mathbf{h}(\theta) \end{pmatrix}. \tag{11}$$

A data set D and its learned model θ, λ, μ will satisfy $\mathbf{f}(D, \theta, \lambda, \mu) = \mathbf{0}$. The implicit function theorem guarantees that, if $\mathbf{f}$ is continuously differentiable w.r.t. its parameters and the Jacobian matrix $\left[\frac{\partial \mathbf{f}}{\partial \theta}\middle|\frac{\partial \mathbf{f}}{\partial \boldsymbol{\lambda}}\middle|\frac{\partial \mathbf{f}}{\partial \boldsymbol{\mu}}\right]$ is of full rank, then $\mathbf{f}$ induces a unique function $(\theta, \boldsymbol{\lambda}, \boldsymbol{\mu}) = i(D)$ in an open

[2] D may not be *iid* samples anymore, but the learner still carries out its optimization in (1).

neighborhood. Furthermore,

$$\frac{\partial i}{\partial D} = -\left[\frac{\partial \mathbf{f}}{\partial \theta}\Big|\frac{\partial \mathbf{f}}{\partial \boldsymbol{\lambda}}\Big|\frac{\partial \mathbf{f}}{\partial \boldsymbol{\mu}}\right]^{-1}\left(\frac{\partial \mathbf{f}}{\partial D}\right). \tag{12}$$

where $\frac{\partial \mathbf{f}}{\partial \theta}$ is the Jacobian matrix of $\mathbf{f}(D, \theta, \boldsymbol{\lambda}, \boldsymbol{\mu})$ w.r.t. θ, and so on. Then $\frac{\partial \theta}{\partial D}$ is the first d rows of $\frac{\partial i}{\partial D}$.

We now have a generic procedure to optimize the attacking training-set D. In the rest of the paper we derive three concrete cases of training-set attacks against SVMs, logistic regression, and linear regression, respectively. These are examples only; attacks against many other learners can be readily derived.

Attacking SVM

There are many possible training-set attack settings against an SVM. Our work differs from an earlier work in attacking SVM which greedily changes one data point at a time (Biggio, Nelson, and Laskov 2012), in that we have a formal framework for optimal attacks, that we allow different attack settings, and that we have a convex attack solution for the specific attack setting below (proof of convexity in (Mei and Zhu 2014)). The attack setting we consider here is mathematically illuminating while also relevant in practice. Let $D_0 = (\mathbf{X}_0, \mathbf{y}_0)$ be the original data set. The attacker is only allowed to change the features. Formally, the attack search space is $\mathbb{D} = \{(\mathbf{X}, \mathbf{y}_0) \mid \mathbf{X} \in \mathbb{R}^{d \times n}\}$.

Recall that given training set $D \in \mathbb{D}$, an SVM learns weights $\hat{\mathbf{w}}_D$ and bias $\hat{b}_D$ by solving the lower level problem Eq (1):

$$\begin{aligned} O_L(D, \mathbf{w}, b, \xi) &= \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_i \xi_i && (13)\\ g_i &= 1 - \xi_i - y_i(\mathbf{x}_i^\top \mathbf{w} + b) && (14)\\ g_{i+n} &= -\xi_i && (15) \end{aligned}$$

for $i = 1 \ldots n$, where ξ_i is the hinge loss and C the regularization parameter.

On the original data D_0, the SVM would have learned the weight vector $\hat{\mathbf{w}}_{D_0}$. We assume that the attacker has a specific target weight vector $\mathbf{w}^* \neq \hat{\mathbf{w}}_{D_0}$ in mind, and the attacker risk function is

$$R_A(\hat{\mathbf{w}}_D) = \frac{1}{2}\|\hat{\mathbf{w}}_D - \mathbf{w}^*\|_2^2. \tag{16}$$

That is, the attacker wants to "nudge" the SVM toward $\mathbf{w}^*$. We also assume that the attacker effort function is the Frobenius norm

$$E_A(D, D_0) = \frac{\lambda}{2}\|\mathbf{X} - \mathbf{X}_0\|_F^2. \tag{17}$$

These fully specifies the attack problem in Eq (5). We reduce the SVM KKT conditions to:

$$w_j - \alpha_i \sum_i \mathbb{I}_1(1 - y_i(\mathbf{x}_i^\top \mathbf{w} + b) \geq 0) y_i x_{ij} = 0 \tag{18}$$

for $j = 1 \ldots d$, where $\mathbb{I}_1(z) = 1$ if z is true and 0 otherwise, α_i is a value between $[0, C]$ which is uniquely determined by D. The derivation is in the longer version (Mei and Zhu 2014). We plug in the KKT conditions to reduce the bilevel attack problem to a constrained single-level optimization problem:

$$\begin{aligned} \min_{D \in \mathbb{D}, \mathbf{w}} \quad & \frac{1}{2}\|\mathbf{w} - \mathbf{w}^*\|_2^2 + \frac{\lambda}{2}\|\mathbf{X} - \mathbf{X}_0\|_F^2 && (19)\\ \text{s.t.} \quad & w_j - \alpha_i \sum_i \mathbb{I}_1(1 - y_i(\mathbf{x}_i^\top \mathbf{w} + b) \geq 0) y_i x_{ij} = 0. \end{aligned}$$

As discussed in the previous section, we use gradient descent to solve Eq (19). The gradient is

$$\nabla_{\mathbf{X}} = \nabla_{\mathbf{w}} R_A(\mathbf{w})\Big|_{\hat{\mathbf{w}}(\mathbf{X})} \frac{\partial \hat{\mathbf{w}}(\mathbf{X})}{\partial \mathbf{X}} + \nabla_{\mathbf{X}} E_A(D, D_0) \tag{20}$$

with

$$\begin{aligned} \nabla_{\mathbf{w}} R_A(\mathbf{w}) &= \mathbf{w} - \mathbf{w}^* && (21)\\ \nabla_{x_{ij}} E_A(D, D_0) &= \lambda(X_{ij} - X_{0,ij}). && (22) \end{aligned}$$

To compute $\frac{\partial \hat{\mathbf{w}}(\mathbf{X})}{\partial \mathbf{X}}$, denote the left hand side of Eq (18) as $\mathbf{f}(\mathbf{w}, \mathbf{X})$. Under suitable conditions the implicit function theorem holds and the Jacobian matrix of $\frac{\partial \mathbf{f}}{\partial \mathbf{w}}$ is the identity matrix:

$$\frac{\partial \mathbf{f}}{\partial \mathbf{w}} = I. \tag{23}$$

The Jacobian $\frac{\partial \mathbf{f}}{\partial \mathbf{X}}$ at row j' and column ij as

$$\left[\frac{\partial \mathbf{f}}{\partial \mathbf{X}}\right]_{j',ij} = -\alpha_i y_i \mathbb{I}_1(j = j')\mathbb{I}_1(1 - y_i \mathbf{x}_i^T \mathbf{w} \geq 0). \tag{24}$$

The element at row j' and column ij in $\frac{\partial \hat{\mathbf{w}}(\mathbf{X})}{\partial \mathbf{X}}$ is

$$\left[\frac{\partial \hat{\mathbf{w}}(\mathbf{X})}{\partial \mathbf{X}}\right]_{j',ij} = \alpha_i y_i \mathbb{I}_1(j = j')\mathbb{I}_1(1 - y_i \mathbf{x}_i^\top \mathbf{w} \geq 0). \tag{25}$$

Intuitively, modifying a point $\mathbf{x}$ will only have an effect on $\mathbf{w}$ if $\mathbf{x}$ incurs hinge loss (i.e. $\mathbf{x}$ is a support vector), a well-known fact.

Attacking Logistic Regression

As another example, we show training-set attacks on logistic regression. As in attacking SVM, the attacker can only change the features in $D_0 = \{\mathbf{X}_0, \mathbf{y}_0\}$. The attack space is $\mathbb{D} = \{(\mathbf{X}, \mathbf{y}_0) \mid \mathbf{X} \in \mathbb{R}^{d \times n}\}$

Given training set D, logistic regression estimates the weight $\hat{\mathbf{w}}_D$ and bias $\hat{b}_D$ by solving the following lower level problem:

$$O_L(D, \mathbf{w}, b) = \sum_i \log\left(1 + \exp(-y_i \hat{h}_i)\right) + \frac{\mu}{2}\|\mathbf{w}\|_2^2, \tag{26}$$

where $\hat{h}_i \triangleq \mathbf{w}^\top \mathbf{x}_i + b$ is the predicted response on instance i, $\log\left(1 + \exp(-y_i \hat{h}_i)\right) = -\log\left(\sigma(y_i \hat{h}_i)\right)$ is the negative likelihood $P(y_i \mid \mathbf{x}_i, \mathbf{w}, b)$, $\sigma(a) \triangleq 1/(1 + \exp(-a))$ is the logistic function, and μ is the regularization parameter.

Let the attack goal be defined by a target weight vector $\mathbf{w}^* \neq \hat{\mathbf{w}}_{D_0}$. The attacker risk function is $R_A(\hat{\mathbf{w}}_D) = \frac{1}{2}\|\hat{\mathbf{w}}_D - \mathbf{w}^*\|_2^2$, and the attacker effort function is $E_A(D, D_0) = \frac{\lambda}{2}\|\mathbf{X} - \mathbf{X}_0\|_F^2$. Note that the logistic regression problem Eq (26) is convex, we reduce it to the equivalent KKT condition

$$\sum_i -\big(1-\sigma(y_i\hat{h}_i)\big)y_i x_{ij} + \mu w_j \quad = \quad 0. \tag{27}$$

By replacing the lower-level problem Eq (26) with the KKT condition Eq (27), the training-set attack bilevel problem reduces to the single-level constrained problem:

$$\begin{aligned} \min\nolimits_{D\in\mathbb{D},\mathbf{w}} \quad & \frac{1}{2}\|\mathbf{w}-\mathbf{w}^*\|_2^2 + \frac{\lambda}{2}\|\mathbf{X}-\mathbf{X}_0\|_F^2 \\ \text{s.t.} \quad & \sum_i -\big(1-\sigma(y_i\hat{h}_i)\big)y_i x_{ij} + \mu w_j = 0. \end{aligned} \tag{28}$$

To solve it, we still use the gradient descent method. The gradient is $\nabla_{\mathbf{X}} = \nabla_{\mathbf{w}} R_A(\mathbf{w})\Big|_{\hat{\mathbf{w}}(\mathbf{X})} \frac{\partial \hat{\mathbf{w}}(\mathbf{X})}{\partial \mathbf{X}} + \nabla_{\mathbf{X}} E_A(D,D_0)$ where $\nabla_{\mathbf{w}} R_A(\mathbf{w}) = \mathbf{w}-\mathbf{w}^*$ and $\nabla_{x_{ij}} E_A(D,D_0) = \lambda(X_{ij}-X_{0,ij})$. To calculate $\frac{\partial \hat{\mathbf{w}}(\mathbf{X})}{\partial \mathbf{X}}$, we denote the left hand side of Eq (27) as $\mathbf{f}(\mathbf{w},\mathbf{X})$. Under suitable conditions, the implicit function theorem holds and we compute $\frac{\partial \hat{\mathbf{w}}(\mathbf{X})}{\partial \mathbf{X}}$ using Eq (12). All we need is the two right hand side terms of Eq (12). The first term has the element at the $j'-$th row and the $j-$th column as

$$\begin{aligned} \left[\frac{\partial \mathbf{f}}{\partial \mathbf{w}}\right]_{j',j} \quad = \quad & \sum_i \sigma(y_i\hat{h}_i)(1-\sigma(y_i\hat{h}_i))x_{ij}x_{ij'} \\ & +\lambda\mathbb{I}_1(j=j'). \end{aligned} \tag{29}$$

We note that the inversion of matrix $\frac{\partial \mathbf{f}}{\partial \mathbf{w}}$ is well-defined because the matrix is positive definite. It is the Hessian matrix of $O_L(D,\mathbf{w},b)$. The second term on the right hand side of Eq (12) has the element at the $j'-$th row and the $ij-$th column

$$\begin{aligned} \left[\frac{\partial \mathbf{f}}{\partial \mathbf{X}}\right]_{j',ij} \quad = \quad & \sigma(y_i\hat{h}_i)(1-\sigma(y_i\hat{h}_i))w_j x_{ij'} \\ & -(1-\sigma(y_i\hat{h}_i))y_i\mathbb{I}_1(j=j'). \end{aligned} \tag{30}$$

Attacking Linear Regression

For demonstration, we consider the simplest linear regression: ordinary least squares (OLS). Denote the $n \times d$ input matrix X and the response vector $\mathbf{y}$. Therefore, given data $D=(X,\mathbf{y})$, OLS assumes $\mathbf{y} = X\beta + \epsilon$, where the noise $\epsilon \sim \mathcal{N}(0,\sigma^2\mathbf{I})$. The maximum likelihood estimate (MLE) corresponds to the lower level problem

$$O_L(D,\beta) = \|\mathbf{y} - X\beta\|^2. \tag{31}$$

This problem is convex and unconstrained. The equivalent KKT condition is simply the familiar OLS solution

$$\hat{\beta}_D - (X^\top X)^{-1}X^\top \mathbf{y} = 0. \tag{32}$$

Without loss of generality, for original training set $D_0 = \{\mathbf{X}_0, \mathbf{y}_0\}$, we assume $\hat{\beta}_{D_0,1} > 0$. For demonstration purpose let us suppose that the attack goal is to modify the data such that $\hat{\beta}_D$, the OLS estimate on the modified data, has $\hat{\beta}_{D,1} \le 0$. Let the attacker be allowed to only change $\mathbf{y}$ by adding an n-vector δ, that is $\mathbf{y} \leftarrow \mathbf{y}_0 + \delta$. We will give a concrete example in the experiment section.

The attacker may define the attacker risk function as a hard constraint on $\hat{\beta}_{D,1}$, that is $R_A(\hat{\beta}_D) = \mathcal{I}(\hat{\beta}_{D,1} \le 0)$, where the indicator function $\mathcal{I}(z)$ is zero when z is true and infinity otherwise. In other words, $\mathbb{D} = \{(\mathbf{X}_0, \mathbf{y}_0 + \delta) : \delta \in \mathbb{R}^n\}$.

There are different ways to define the attacker effort function, which measures some notion of the magnitude of δ. Interestingly, different choices lead to distinct attacking strategies. We discuss two attacker effort functions. The first one is the ℓ_2-norm effort, defined as $E_A(D,D_0) = \|\delta\|_2^2$. Let $A = (\mathbf{X}_0^\top \mathbf{X}_0)^{-1}\mathbf{X}_0^\top$. According to the KKT conditions Eq (32), $\hat{\beta}_D = A(\mathbf{y}_0 + \delta) = \hat{\beta}_{D_0} + A\delta$. The optimal attack problem (5) reduces to

$$\min_\delta \|\delta\|_2^2 \quad \text{s.t. } \hat{\beta}_{D,1} \le 0. \tag{33}$$

Instead of using the gradient descent method described in previous sections to solve it, one can obtain an analytic solution for δ. Let a_1 be the first column in $A^\top$: $a_1 = A^\top \mathbf{e}$, where $\mathbf{e} = (1,0,\ldots)^\top$ is the first unit vector. One can show that the attack solution is $\delta = -\frac{\hat{\beta}_{D_0,1}}{a_1^\top a_1}a_1$ (Mei and Zhu 2014).

The second attacker effort function is the ℓ_1-norm effort, defined as $E_A(D,D_0) = \|\delta\|_1$. The optimal attack problem becomes

$$\min_\delta \|\delta\|_1 \quad \text{s.t. } \hat{\beta}_{D,1} \le 0. \tag{34}$$

Without loss of generality assume $a_1 = (\alpha_1, \ldots, \alpha_d)^\top$ with $|\alpha_1| \ge \ldots \ge |\alpha_d|$, then the solution is $\delta = (-\frac{\hat{\beta}_{D_0,1}}{\alpha_1}, 0, \ldots, 0)^\top$ (Mei and Zhu 2014). The attack solution is sparse: δ changes only a single data point. This sparsity is distinct from standard ℓ_1-regularization such as LASSO, in that it is not on the model parameters but on the training data.

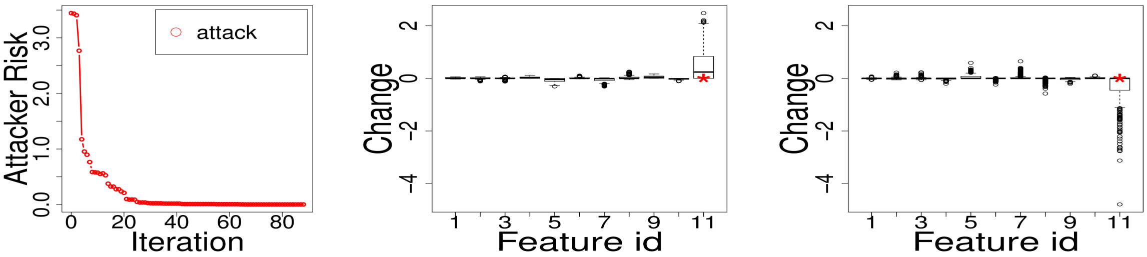
Experiments

Using the procedure developed in the previous section, we present empirical experiments on training-set attacks. Our attack goals are meant to be fictitious but illustrative.

Attack Experiments on SVM

We demonstrate attacks using the LIBLINEAR SVM implementation as the learner (Fan et al. 2008). The regularization parameter C in the learner is set to 1 by a separate cross validation process, and we assume that the attacker knows C. D_0 is the wine quality data set (Cortez et al. 2009). The data set has $n = 1600$ points, each with $d = 11$ numerical features $\mathbf{x}$ and a wine quality number ranging from 1 to 10. We normalize each feature dimension to zero mean and standard deviation one. We threshold the wine quality number at 5 to produce a binary label y.

The fictitious attacker's goal is to make it seem like only the feature "alcohol" correlates with wine quality. We simulate this goal by first generating the attack target weight



(a) attacker risk R_A (b) feature changes on positive data (c) feature changes on negative data

Figure 1: Training-set attack on SVM. The "alcohol" feature is marked by a red star in (b,c).

vector $\mathbf{w}^*$ as follows. We create another set of labels y_i' for each data point in D_0 by thresholding the (normalized) alcohol feature $x_{i,alcohol}$ at 0. We train a separate SVM on $D' = \{(\mathbf{x}_i, y_i')\}_{i=1}^n$, and let the learned weight be $\mathbf{w}^*$. As expected, $\mathbf{w}^*$ has a much larger positive weight on "alcohol" compared to other dimensions. We then let the attacker risk function be Eq (16) with this $\mathbf{w}^*$. We let the attacker effort function be Eq (17) with the weight $\lambda = 0.1$. We then run the optimization procedure to identify the optimal attacking training-set D. The step length α_t of gradient defined in Eq (9) is set to $\alpha_t = 0.5/t$.

We compare the optimal attack against a heuristic baseline called *naive attack*, which moves every data point $\mathbf{x}_i$ to its nearest point with zero hinge loss under the attack target weight: $\mathbf{x}_i \leftarrow \text{argmin}_{\mathbf{z}} \|\mathbf{z} - \mathbf{x}_i\|_2$ s.t. $(1 - y_i \mathbf{z}^\top \mathbf{w}^*)_+ = 0$.

The attack results are shown in Figure 1. The optimal training-set attack rapidly decreased the attacker risk R_A to zero, with an effort $E_A = 370$. In contrast, naive attack also achieved zero attacker risk but not until an effort of 515. Figures 1(b,c) are box plots showing how the training-set attacker manipulated the data points for positive and negative data, respectively. As expected, the attack basically changed the value of the "alcohol" feature (the right-most or the 11th) to made it appear to strongly correlate with the class label. This behavior increased the weight on "alcohol", achieving the attack.

Attack Experiments on Logistic Regression

We use logistic regression in the LIBLINEAR package as the learner, which has the regularization parameter $C = 0.01$ set separately by cross validation. D_0 is the Spambase data set (Bache and Lichman 2013). This data set consists of $n = 4601$ instances, each of which has $d = 57$ numerical features $\mathbf{x}$ and a binary label y_i (1 for spam and -1 for not spam). Each feature is the percentage of a specific word or character in an email. We denote the logistic regression model weights trained from D_0 as $\mathbf{w}_0$.

The attack is constructed as follows. We picked the word "credit" and assume that the attacker wanted the learned model to ignore this (informative) feature. To this end, we generate a new feature vector $\mathbf{x}_i'$ for each instance by copying $\mathbf{x}'$ from D_0 but set $x'_{i,freq\ credit} = 0$. We then produce the attack target weights $\mathbf{w}^*$ as the learned weights of logistic regression given data $D' = \{(\mathbf{x}_i', y_i)\}_{i=1}^n$. As expected, $\mathbf{w}^*$ has zero weight on the feature "credit" and is similar to $\mathbf{w}_0$ on other dimensions. We then let the attacker risk function be Eq (16) with this $\mathbf{w}^*$. We let the attacker effort function be Eq (17) with the weight $\lambda = 0.01$. We then run the optimization procedure to identify the optimal attacking training-set D. The step length α_t of gradient defined in Eq (9) is set to $\alpha_t = 1/t$.

The data set D' used to generate $\mathbf{w}^*$ is the baseline we compare against.

Figure 2 shows the results. Our training-set attack procedure rapidly decreased the attacker risk R_A to zero with an attacker effort of $E_A = 232$. The baseline D' achieved zero R_A (since the target weights $\mathbf{w}^*$ was learned from D'), but its attacker effort is a much higher 390. Therefore, our procedure efficiently attacked logistic regression. The optimal training-set attack essentially changed the feature "credit" (the 20th feature) in the following way to decrease the corresponding weight to zero: It decreased (increased) the feature value for positively (negatively) labeled instances.

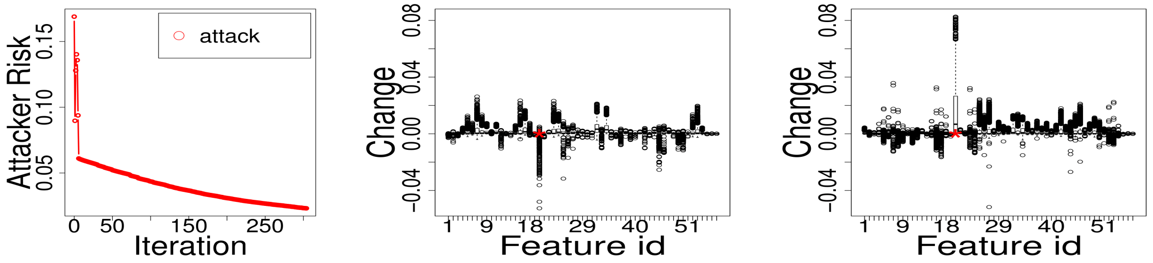
Attack Experiments on Linear Regression

We demonstrate an attack on OLS. D_0 comes from the Wisconsin State Climatology Office, and consists of annual number of frozen days for Lake Mendota in Midwest USA from 1900 to 2000 [3]. It is a regression problem with $n = 101$ points. Each $\mathbf{x}_i$ is a two dimensional vector of year and the constant 1 for the bias term. The response y_i is the number of frozen days in that year. On D_0 OLS learns a downward slope $\hat{\beta}_1 = -0.1$, indicating a warming trend for this lake, see Figure 3(a).

We now construct an attack whose goal is to hide the warming trend, i.e., $\beta_1^* \geq 0$. As stated earlier, this can be achieved by an attacker risk indicator function $R_A(\hat{\beta}_D) = \mathcal{I}(\hat{\beta}_{D,1} \leq 0)$. The attacker can manipulate $\mathbf{y}$ in D by adding a vector δ. Interestingly, we demonstrate how different attacker effort functions affect the optimal attack:

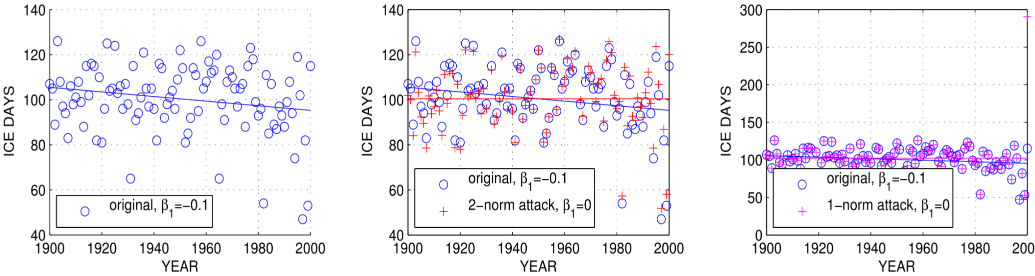
(Results with ℓ_2-norm attacker effort) With the attacker effort function $\|\delta\|_2^2$, the optimal attack is specified by Eq (33). We show the optimal attack in Figure 3(b). The optimal changes δ "pivoted" each and every data point, such that OLS learned a flat line with $\hat{\beta}_1 = 0$.

[3] data available at http://www.aos.wisc.edu/~sco/lakes/Mendota-ice.html



(a) attacker risk R_A (b) feature changes on positive data (c) feature changes on negative data

Figure 2: Training-set attack on logistic regression. The 20th feature on “frequency of word credit” is marked



OLS on D_0 ℓ_2 attacker effort ℓ_1 attacker effort

Figure 3: Training-set attack on OLS

(Results with ℓ_1-norm attacker effort) If the attacker effort function is $\|\delta\|_1$ instead as in Eq (34), the optimal attack δ is drastically different: Only the rightmost data point was changed (by a lot) while all other points remained the same, see Figure 3(c). The learned model after attack became a flat line with $\hat{\beta}_1 = 0$, too.

A Discussion on Defenses

Although we focused on formulating the optimal training-set attack in this paper, our ultimate goal is to design defenses in the future.

There is a related line of research on *robust learning* (Globerson and Roweis 2006; Torkamani and Lowd 2013; El Ghaoui et al. 2003; Xu, Caramanis, and Mannor 2009; Kim, Magnani, and Boyd 2005; Dekel, Shamir, and Xiao 2010). Promising as the name suggests, we point out that robust learning is *not* an appropriate defense against training-set attacks. In robust learning the learner receives *clean* training data D_0, and wishes to learn a model that performs well on future contaminated test data. The contaminated test data is unknown ahead of time, but is assumed to be within a given distance from D_0 under an appropriate metric. One may view robust learning as an extreme case of domain adaptation: the training data comes from the source domain and the test data from the target domain, but there are no examples from the target domain except for the distance constraint. The test data may not be *iid* either, but can be adversarially contaminated. Robust learning typically employs a minimax formulation to mitigate the worst possible test risk. Obviously, robust learning’s assumption that the learner receives clean training data D_0 is immediately broken by a training-set attack. If a robust learner in fact receives a poisoned training set D, it will unnecessarily try to guard against further contamination to D. This will likely make the learned model highly obtuse because the model needs to be “doubly robust.” Bridging robust learning and defenses against training-set attacks remains an open problem.

Our optimal training-set attack formulation opens the door for an alternative defense: flagging the parts of training data likely to be attacked and focus human analysts’ attention on those parts. For instance, in our attack on OLS with the ℓ_1-norm effort function we observed the attack behavior that only the extreme data item was changed, and by a large amount. This suggests that under this attack setting the extreme data items have a high risk of being manipulated, and the analysts should examine such items. As another example, the optimal attack on logistic regression drastically increases the count for the feature “credit” in non-spam emails, and the optimal attack on SVM drastically increases the feature value of “alcohol” in good-quality wines. Such attacks can potentially be noticed by analysts once they know where to look.

Acknowledgments We thank Daniel Lowd and the anonymous reviewers for their constructive suggestions. Support for this research was provided in part by NSF grant IIS 0953219 and by the University of Wisconsin–Madison Graduate School with funding from the Wisconsin Alumni Research Foundation.

References

Bache, K., and Lichman, M. 2013. UCI machine learning repository.

Bard, J. F. 1998. *Practical Bilevel Optimization: Algorithms And Applications*. Kluwer Academic Publishers.

Barreno, M.; Nelson, B.; Sears, R.; Joseph, A.; and Tygar, J. 2006. Can machine learning be secure? In *CCS*.

Barreno, M.; Nelson, B.; Joseph, A. D.; and Tygar, J. 2010. The security of machine learning. *Machine Learning Journal* 81(2):121–148.

Biggio, B.; Fumera, G.; and Roli, F. 2013. Security evaluation of pattern classifiers under attack. *IEEE TKDE*.

Biggio, B.; Nelson, B.; and Laskov, P. 2012. Poisoning attacks against support vector machines. In *ICML*.

Burges, C. 1998. A tutorial on support vector machines for pattern recognition. *Knowledge Discovery and Data Mining* 2(2).

Chung, S., and Mok, A. 2007. Advanced allergy attacks: Does a corpus really help. In *RAID*.

Colson, B.; Marcotte, P.; and Savard, G. 2007. An overview of bilevel optimization. *Annals of operations research* 153(1):235–256.

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

Dalvi, N.; Domingos, P.; Mausam; Sanghai, S.; and Verma, D. 2004. Adversarial classification. In *SIGKDD*.

Dekel, O.; Shamir, O.; and Xiao, L. 2010. Learning to classify with missing and corrupted features. *Machine learning* 81(2):149–178.

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

Fan, R.-E.; Chang, K.-W.; Hsieh, C.-J.; Wang, X.-R.; and Lin, C.-J. 2008. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research* 9:1871–1874.

Globerson, A., and Roweis, S. T. 2006. Nightmare at test time: robust learning by feature deletion. In *ICML*.

Kim, S.-J.; Magnani, A.; and Boyd, S. 2005. Robust Fisher discriminant analysis. In *Advances in Neural Information Processing Systems*, 659–666.

Laskov, P., and Kloft, M. 2009. A framework for quantitative security analysis of machine learning. In *The 2nd ACM Workshop on AISec*.

Laskov, P., and Lippmann, R. 2010. Machine learning in adversarial environments. *Machine Learning* 81(2):115–119.

Liu, W., and Chawla, S. 2009. A game theoretical model for adversarial learning. In *ICDM Workshops*.

Lowd, D., and Meek, C. 2005. Good word attacks on statistical spam filters. In *CEAS*.

Mei, S., and Zhu, X. 2014. Using machine teaching to identify optimal training-set attacks on machine learners. Technical Report Computer Science TR1813, University of Wisconsin-Madison.

Nelson, B.; Barreno, M.; Chi, F.; Joseph, A.; Rubinstein, B.; Saini, U.; Sutton, C.; Tygar, J.; and Xia, K. 2009. Misleading learners: Co-opting your spam filter. In *Machine Learning in Cyber Trust: Security, Privacy, Reliability*. Springer.

Patil, K.; Zhu, X.; Kopec, L.; and Love, B. 2014. Optimal teaching for limited-capacity human learners. In *Advances in Neural Information Processing Systems (NIPS)*.

Rubinstein, B.; Nelson, B.; Huang, L.; Joseph, A.; Lau, S.; Taft, N.; and Tygar, D. 2008. Compromising PCA-based anomaly detectors for network-wide traffic. Technical Report UCB/EECS-2008-73, EECS Department, University of California, Berkeley.

Tan, K.; Killourhy, K.; and Maxion, R. 2002. Undermining an anomaly-based intrusion detection system using common exploits. In *RAID*.

Torkamani, M., and Lowd, D. 2013. Convex adversarial collective classification. In *Proceedings of The 30th International Conference on Machine Learning*, 642–650.

Wittel, G., and Wu, S. 2004. On Attacking Statistical Spam Filters. In *Proc. of the Conference on Email and Anti-Spam (CEAS)*.

Xiao, H.; Xiao, H.; and Eckert, C. 2012. Adversarial label flips attack on support vector machines. In *ECAI*.

Xu, H.; Caramanis, C.; and Mannor, S. 2009. Robustness and regularization of support vector machines. *The Journal of Machine Learning Research* 10:1485–1510.

Zhu, X. 2013. Machine teaching for Bayesian learners in the exponential family. In *NIPS*.