E0:270 - Machine Learning - Adversarial Machine Learning

Prithiv Natarajan [1] **Suraj Jagtap** [1]

Abstract

Many machine learning algorithms can be attacked under adversarial settings. Adversaries can achieve this by adding a small perturbation in the input data which can significantly change the output of the classifier. We will discuss about reverse engineering attacks against Support Vector Machines (SVMs). Also, we implement the randomization strategy to defend against the reverse engineering attacks. In randomization, instead of learning a fixed classifier, a random classifier is selected from the pool of classifiers for predicting the outcome. Results show that we can incorporate this strategy with large variance without incurring any significant loss in prediction accuracy.

1. Introduction and motivation

Machine learning techniques are powerful because they can detect complex hidden patterns in huge datasets to give reasonably accurate predictions. Unfortunately, these techniques are prone to security related issues. In most of the machine learning techniques we assume that both the training and testing data are generated from a stationary environment. But this assumption is not valid in many of the real life scenarios.

Adversarial examples can be defined as inputs to theoretical models that has been intentionally designed to cause an error and make the system ineffective. These attacks are possible in multiple fields. Such as attack against email spam detection, where attackers can misspell bad words or add good words in spam emails to evade the system (Faucett, 2003). Similar attack is possible in computer intrusion detection; fraud detection (Faucett, 1997); counter-terrorism (Jensen et al., 2003); comparison shopping (Doorenbos et al., 1997) and web page rankings (Guernsey, 2003).

Attempts at defending against attacks have been shown previously. One of the many ways being training the model in the brute force method to get the desired outputs for known adversarial attacks. Another method is where the model is trained to output probabilities of classes rather than making strict or hard decisions about the same.

Designing a defense that can protect a model against powerful adversaries is certainly one of the most important research areas. This not only applies to fields like Cryptography, but also very much in deep learning. As the adversaries become more creative, it becomes imperative for defenders to design a defense for adaptive attackers, as blocking one loophole might often leave or generate another loophole for one to attack. Most importantly, adversarial examples generally fall into the category of complicated non-convex or non-linear or both kind of problems, and to have a theoretical framework for such optimization problems is often rare.

Under adversarial environments, adversaries can generate exploratory attacks against the defender such as evasion and reverse engineering. In this work, we show that reverse engineering attacks can be carried out quite efficiently against fixed classifiers, and analyze the use of randomization as a suitable strategy for mitigating their risk. We learn a distribution of classifiers subject to the constraint that any single classifier picked at random from such distribution provides reliable predictions with a high probability. We also analyze the tradeoff between variance of the distribution and its predictive accuracy.

In order to make system robust we need to be able to make reliable predictions without revealing much information about the decision algorithm. Also in worst case scenario we would want a system which can work even when the adversary knows the system.

We will show that randomization strategy can be used effectively to add robustness to SVMs against adversarial reverse engineering attacks without incurring notable loss in predictive accuracy.

2. Literature review

Adversarial attack strategies can be classified by the type of attack (causative or exploratory) and by the nature of proposed solutions (attack or defense strategy).

[1]Department of Chemical Engineering, Indian Institute of Science, Bengaluru. Correspondence to: Prithiv Natarajan <prithivn@iisc.ac.in>, Suraj Jagtap <jsuraj@iisc.ac.in>.

Project Report, E0:270, Machine Learning, 2018.

2.1. Causative attack strategy

Causative attacks are carried out during the training phase of the machine learning. These attacks can be formalized as a game between the classifier and the adversary, in which each player seeks an optimal strategy knowing that its opponent would also seek an optimal strategy (Alabdulmohsin et al., 2014). Biggio et al. discussed attack strategy against SVM classifier using gradient ascent method (Biggio et al., 2012).

2.2. Causative defense strategy

Several methods have been proposed to defend the causative attacks such as adversarial collective classification, ensemble methods, kernel matrix correction, robustness/regularization, multiple-instance learning, game-theoretic strategies, reject on negative impact (RONI)(Tygar, 2011). In RONI, the training examples that are found to be quite influential to the decision boundary are rejected.

2.3. Exploratory attack strategy

Exploratory attacks are carried out post-training while the system is operational assuming no learning takes place during the attack. Two categories of attacks under this are evasion attack and reverse engineering attack. Good words attack to evade the spam filter system is an example of evasion attack (Lowd & Meek, 2005). In reverse engineering adversary tries to model the classifier's decision boundary. For this an adversary can send probe queries to the classifier and observe the decision to estimate the decision boundary. We will show how reverse engineering attacks can be carried out against the fixed classifier in following sections.

2.4. Exploratory defense strategy

Exploratory attacks can be defended by keeping the entire learning processes as confidential as possible. Another strategy is to increase the complexity so that it will be harder for the attacker to learn the classifier. And last strategy is using randomized classifier. In this work we will implement the randomization strategy (Alabdulmohsin et al., 2014) against reverse engineering attacks.

3. Problem statement

A classifier is supplied with m training examples $\{(x_1, y_1), ..., (x_m, y_m)\}$, drawn i.i.d. from a fixed unknown distribution D, where $x_i \in R^n$ and $y_i \in \{+1, -1\}$. The objective of the classifier is to make a prediction y'_t given a new example x_t such that prediction is accurate with high probability.

For reverse engineering the classifier, adversary can send the probe query x_q and observe the response y_q. To get maximum information about the classifier's decision boundary, one positive and one negative query can be selected by random sampling. And by generating this dataset an adversary can learn it's own classifier (soft-SVM in our case), thus reverse engineering the decision boundary.

The defender can learn a distribution of classifiers subject to the constraint that any single classifier picked at random from such distribution provides reliable predictions. Once learning is concluded, the risk of reverse engineering attacks can be minimized by picking up a classifier at random for every query observed.

3.1. Randomization strategy

The convex optimization problem was derived to learn the distribution of the classifiers (Alabdulmohsin et al., 2014).

$$\begin{aligned} &\underset{\mu, s, \xi}{\text{minimize}} \quad \frac{1}{2}\frac{\mu^T.\mu}{1^T.s} + C\sum_{i=1}^{m}\xi_i \\ &\text{subject to} \\ &y_i(\mu^T x_i) \geq 1 + \Phi^{-1}(\nu)\sum_{j=1}^{n} {x_{i,j}}^2 s_j - \xi_i \end{aligned} \tag{1}$$

$s_j \geq 0, \xi_i \geq 0$ for j=1,..m and i=1,2,...n

Where μ and s are mean and standard deviation of the weights, ξ_i are slack variables, Φ is cumulative density function of the standard Gaussian distribution and ν is the probability that the inequalities are satisfied ($>$0.5).

4. Dataset description

Reverse engineering technique is explained by attacking against a synthetic data. This dataset has two classes distributed normally around the means (-5,-5) and (5,5) with standard deviation of 2. The data is represented in Figure 1 (top-left).

For validating the randomization strategy against the reverse engineering attacks, 3 datasets are selected. First is the synthetic data used for reverse engineering. Second is IRIS dataset (4 attributess, 100 data examples). And last one is VERTEBRAL dataset (6 attributes, 310 examples) obtained from UCI machine learning repository.

5. Implementation and results

5.1. Reverse engineering attack

First we show that the reverse engineering can be carried out effectively by using random sampling using the synthetic dataset (Figure 1). Top left figure represents the synthetic dataset. From the data one positive and one negative example (a probe pair) was selected randomly and then de-

cision boundary was obtained by learning the soft-SVM. The second plot on the top row shows how weights evolve as number of probes increase. It can be seen that after about 13 probes weights remain constant. Bottom row shows the decision boundary for different number of probe pairs, decision boundary for 15 probes resembles with the true decision boundary (top right plot).

5.2. Randomization strategy

For randomization strategy, we define its normalized variation $0 \leq \chi \leq 1$ using:

$$\chi = \sqrt{\frac{\sum_{j=0}^{n} {\sigma_j}^2}{{||\mu||_2}^2 + \sum_{j=0}^{n} {\sigma_j}^2}}$$

where μ and σ are mean and standard deviations in weights learnt by solving the equation (1).

We have used 3 datasets to implement randomization as described earlier. In each dataset, the first exercise is to plot the tradeoff curve (Figure 2 top row). This can be implemented by recording predictive accuracy and χ for various choices of C and ν. Once a model (C, ν) is selected and the optimization problem in equation (1) is solved, we obtain the desired distribution N(μ,σ).

Table 1 shows that without loss of accuracy we can incorporate the variation in the learning classifiers.

Reverse engineering attacks are harder to carry out when the decision boundary is picked at random from a distribution with large variance.

To show this, we define estimation accuracy by normalized difference between the weights obtained by adversary and the actual randomized weights, which is given by,

$$Error_{estimation} = {||\frac{w}{||w||} - \frac{\hat{w}}{||\hat{w}||}||^2}_2$$

where w is true classifier and $\hat{w}$ is adversary's estimate of w.

Figure 2 (bottom row) and the estimation error as a function of number of probes used for reverse engineering attack (refer to Table 2 for the values of estimation errors). It is clear from the plots that in case of random classifiers (blue plot) estimation error is high compared to the fixed classifiers (red plot) in all the datasets.

6. Discussion

We showed that the risk of reverse engineering can be mitigated by learning a distribution of classifiers, rather than following the conventional practice of learning a fixed classifier. Using accuracy-variance tradeoff curves we showed that one can learn the classifier without having to significantly lose on the accuracy.

Once the adversary obtains an estimate of the defender's decision boundary using reverse engineering, the adversary can generate additional attacks such as evasion. Alabdulmohsin et al. (2014) have shown that the randomization increases robustness even against the evasion attacks by showing the reduction in the attack success rate in randomization as compared to the fixed classifier.

$Datasets$	Acc_F	Acc_R	χ
Synthetic data	100.0	100.0	0.77
IRIS	100.0	100.0	0.26
VERTEBRAL	86.3	81.8	0.25

Table 1. Accuracy of SVM on Fixed and randomized classifier Acc_F and Acc_R are accuracies for fixed and randomized classifiers χ is optimum variation obtained from the accuracy-variance tradeoff curves (Top row in Figure 2)

$Datasets$	Er_F	Er_R
Synthetic data	0.0000	0.3940
IRIS	0.0610	0.5224
VERTEBRAL	0.0007	0.0303

Table 2. Reverse engineering estimation errors for fixed and randomized classifier Er_F and Er_R are estimations errors in fixed and randomized classifiers (Bottom row in Figure 2)

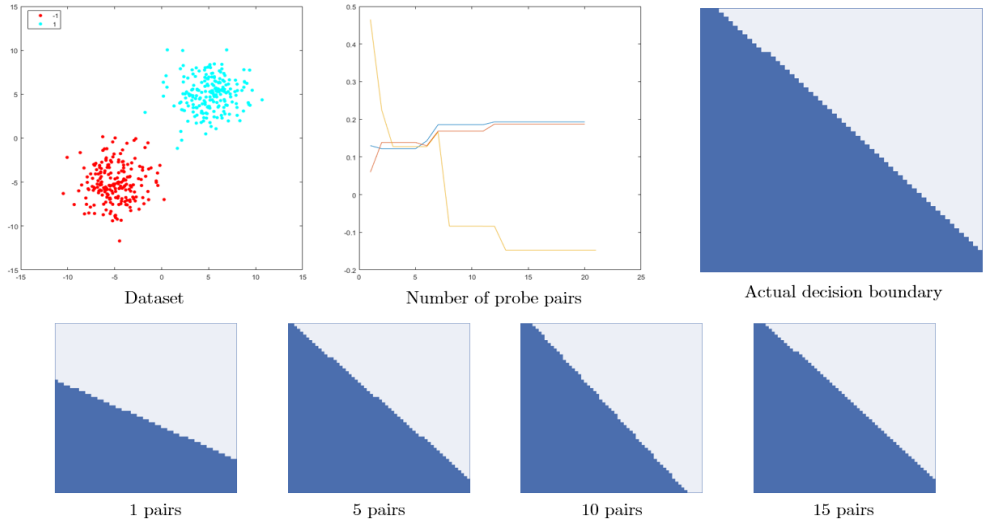

Figure 1: Reverse engineering attack against fixed classifier. Only about 12 probes are required to get the actual decision boundary

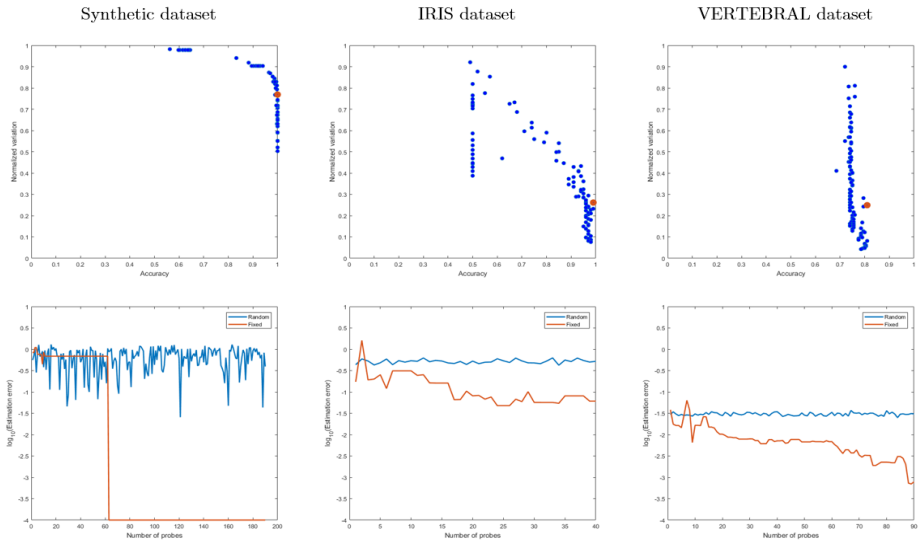

Figure 2: Randomization strategy against reverse engineering attacks. For three datasets (1. synthetic dataset 2. IRIS dataset and 3. VERTEBRAL dataset) Top row: Accuracy vs variation trade-off (red spot shows the optimum parameters), bottom row: Estimation error (Blue: random classifier, red: fixed classifier)

References

Alabdulmohsin, I., Gao, X., and Zhang, X. Adding robustness to support vector machines against adversarial reverse engineering. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pp. 231–240, Shanghai, China, 2014.

Biggio, B., Nelson, B., and Laskov, P. Poisoning attacks against support vector machines. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pp. 1467–1474, Edinburgh, Scotland, 2012.

Doorenbos, R. B., Etzioni, O., and Weld, D. S. A scalable comparison-shopping agent for the world-wide web. In *Proceedings of the First International Conference on Autonomous Agents*, pp. 39–48, Marina del Rey, CA, 1997.

Faucett, T. Adaptive fraud detection. *Data Mining and Knowledge Discovery*, 1(3):291–316, 1997.

Faucett, T. In-vivo spam filtering: A challenge problem for kdd. *SIGKDD Explorations*, 5(2):140–148, 2003.

Guernsey, L. Retailers rise in google rankings as rivals. Technical report, New York Times, 2003.

Jensen, D., Rattigan, M., and Blau, H. Information awareness: A prospective technical assessment. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 378–387, Washington, DC, 2003.

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

Tygar, J. Adversarial machine learning. *IEEE Internet Computing*, 15(5):4–6, 2011.