Adversarial Data Mining: A Game Theoretic Approach[1]

CLASSIFICATION NATO UNCLASSIFIED

Murat Kantarcioglu,
Department of Computer Science,
University of Texas at Dallas
800 W. Campbell Road; MS EC31
Richardson, TX 75080 U.S.A.
muratk@utdallas.edu

Bowei Xi,
Department of Statistics,
Purdue University
250 N. University Street
West Lafayette, IN 47907 U.S.A.
xbw@purdue.edu

[1] This work is funded by Army Research Office Grant W911NF-12-1-0558

Summary

Many real world applications, ranging from spam filtering to intrusion detection, are facing malicious adversaries who actively transform the objects under their control to avoid detection. Data mining techniques are highly useful tools for cyber defense, since they play an important role in distinguishing the legitimate from the destructive. Unfortunately, traditional data mining techniques are insufficient to handle such adversarial problems directly. The adversaries adapt to the data miner's reactions, and data mining algorithms constructed based on a training dataset degrades quickly. Our proposed adversarial data mining framework addresses the challenges posed by malicious adversaries. In this survey article, we discuss the theory, the techniques, and the applications of our proposed adversarial data mining framework. We model the adversarial data mining applications as a Stackelberg game, with an emphasis on the sequential actions of the adversary and the data miner, allowing both parties to maximize their own utilities. We analyze the equilibrium behavior of both parties under our proposed game theoretic framework, which offers insight into the long term effectiveness of a defensive algorithm. Furthermore we apply the equilibrium information to cost sensitive attribute selection. We then derive adversarial support vector machine models against an adversary whose attack strategy is defined under general and reasonable assumptions. We investigate how the performance of the resulting solutions changes under different attack models. The empirical results suggest that our adversarial support vector machine algorithms are robust against various degrees of attacks.

Key Words:

Adversarial data mining, game theory, support vector machine, classification

1 Introduction

Data mining techniques such as clustering and classification have wide applications in real-world problems. However traditional data mining techniques follow a major implicit assumption that the current data and the future data share the same properties. When this assumption is satisfied, a data mining algorithm developed using the current observed data performs well with the future data. Unfortunately this assumption no longer holds in an adversarial environment where data miner faces malicious active adversaries. Examples of such adversarial applications are spam filtering, intrusion detection, advance persistent threat detection etc. Unlike in a static environment, the attack instances are controlled by malicious adversaries who actively transform the objects under their control to avoid detection. For example, in spam filtering, spammers increase the length of spam emails and insert “good” words to make the spam emails resemble the legitimate emails and pass spam filters. Performance of traditional data mining techniques deteriorates quickly when facing active adversaries. To address these challenges, we need to develop resilient and robust data mining techniques in an adversarial environment to distinguish the legitimate objects from the attack objects.

The challenges we face from the active adversaries are different from the concept drift problem, because the adversaries launch targeted attacks and maliciously change the properties of the objects under their control to maximize their payoff. One approach facing malicious adversaries is to be protected against the worst case scenario, i.e. the minimax solution. The drawback of the minimax solution is that they are too pessimistic because they do not consider the adversary's utility function and their potential motives.

This survey article summarizes our previous work in adversarial data mining. We model the adversarial data mining applications as a two player Stackelberg game, where the data miner and the adversary make sequential moves, and each player aims to maximize its own utility. Our approach is not to stay ahead of the adversary by constantly adapting to the adversary’s actions. We focus on a data mining algorithm’s long term performance, i.e. its equilibrium performance. At an equilibrium, neither player has an incentive to change its action. Based on a data mining algorithm’s equilibrium performance, we are able to carry out cost sensitive attribute selection. Our game theoretic framework is general, where the zero-sum game and the corresponding minimax solution is a special case under our framework.

The article is organized as follows: In Section 2, we describe our game theoretic framework for the adversarial data mining applications, and the stochastic search algorithm we use to find the equilibrium information. The equilibrium information measures the long term effectiveness of a data mining algorithm. We then use the equilibrium information to perform cost sensitive attribute selection. In Section 3 we extend our basic model, and present our adversarial support vector machine algorithms. Section 4 concludes the survey article.

1.1 Related Work

Dalvi et al. [3] propose a game theoretic framework for adversarial problems where there is an optimal opponent. They define the problem as a game between two cost-sensitive opponents: a Naïve Bayes classifier and an adversary playing optimal strategies. They assume all parameters of both players are known to each other and the adversary knows the exact form of the classifier. Their adversary-aware Naïve Bayes classifier makes constant automatic adjustments according to the adversary’s expected actions.

Lowed and Meek [12] point out that assuming the adversary has perfect knowledge of the classifier is unrealistic. Instead they suggest the adversary can confirm the membership of an arbitrary instance by sending queries to the classifier. They also assume the adversary has available an adversarial cost function over the sample space that maps samples to cost values. This assumption essentially means the adversary needs to know the entire feature space to issue optimal attacks. They propose an adversarial classifier reverse engineering algorithm to learn the vulnerabilities of given learning algorithms.

Adversarial data mining problems are often modeled as games played between two opponents. We summarize our work in this area in this survey article (based on Kantarcioglu et al. [8]). We model the problem as a sequential Stackelberg game. As we discuss in details below, we assume the two players know each other's utility function. A stochastic search algorithm is used, such as simulated annealing and genetic algorithm, to search for an equilibrium. Later on such equilibrium is used to choose optimal set of attributes that give good equilibrium performance. After [8], Bruckner et al. [2] also model adversarial prediction

problems as Stackelberg games. However, their formulation assumes that the data miner is the leader and the adversary is the follower in the game. Compared with [2], in [8], we let the data miner be the follower. Furthermore our formulation can be generalized, and the data miner can have the option to take an action that does not optimize its utility but instead punishes the adversary at its own expense. Improved models in which Nash strategies are played have also been proposed [1, 10]. Other game theoretic models play zero-sum minimax strategies. Globerson and Roweis [7] consider a problem where some features may be missing at testing time. This is related to adversarial data mining in that the adversary may simply delete highly weighted features in malicious data to increase its chance to evade detection. They develop a game theoretic framework in which classifiers are constructed to be optimal in the worst case scenario. Their idea is to prevent assigning too much weight on any single feature. They use the support vector machine model which optimally minimizes the hinge loss when at most K features can be deleted. El Ghaoui et al [6] apply a minimax model to training data bounded by hyper-rectangles. Their model minimizes the worst-case loss over data in given intervals. Other robust learning algorithms for handling classification-time noise are also proposed [9, 10, 11, 13].

In a follow-up work [14], discussed in Section 3, we show how to build robust support vector machine classifiers without making strong assumptions on what is known to either side of the players. In our SVM models, both wide range attacks and targeted attacks are considered and incorporated into the SVM framework. We discuss the details of our adversarial SVM models in Section 3.

2 A Game Theoretic Framework

2.1 Adversarial Stackelberg Game

Assume the "good" class S_g consists of the legitimate objects and the "bad" class S_b consists of the attack objects. The "bad" class S_b is controlled by one or several adversaries. Our basic game theoretic model formulates the adversarial data mining applications as a two class problem. Data miner measures q attributes from an object, $X=(X_1, X_2, ..., X_d)$. Assume each class has a probability density function $f_i(X)$, i = g or b. Assume p_i is the proportion of each class in the overall population, i = g or b. We have $p_g + p_b = 1$. The overall population is a mixture of two classes, with the density function

$$f(X) = p_g f_g(X) + p_b f_b(X) .$$

The adversary transforms the objects under its control to avoid detection, applying transformation T to the attack objects. The transformed "bad" class has a new density function $f^T_b(X)$. We assume the proportions of two classes, p_g and p_b, stay the same under attack. This assumption can be easily relaxed and adopted into our basic model if an attack significantly increases the number of attack objects. When an attack object is misidentified as a legitimate object, it generates a profit for the adversary. On the other hand, transformation of an attack object suffers from a penalty. For example, by purchasing links a website can increase its search ranking. The cost of purchasing links is a form of penalty to its profit. The ultimate penalty for the website is to be removed from the search index if it is caught doing so. Following many real-world applications, the adversary and the data miner often take the following sequential actions:

1. Given the initial overall distribution with the density function f(X), the adversary chooses a transformation T from its strategy space *S*, which is the set of all feasible transformations, and applies the transformation T to the objects under its control.

2. After observing the transformation T, the data miner sets parameter values for a data mining algorithm and creates a defensive rule h(X).

Let L(h,g) and L(h,b) be the regions where the objects are identified as legitimate or not respectively. We allow each player to have their own utility function. Let $u_g(T,h)$ denote the data miner's utility function, such as –C(T,h) where C(T,h) is the misclassification cost. Let g(T,X) be the profit generated by a transformed attack object when it is misidentified as legitimate. The utility function for the adversary is defined as the expected value of the profit generated by a misidentified transformed attack object.

$$u_b(T,h) = \int_{L(h,g)} g(T,X) f^T{}_b(X)\, dX .$$

We use a Stackelberg game with two players to model the adversary and the data miner's sequential actions. We define the Adversarial Stackelberg Game as follows.

Adversarial Stackelberg Game $G=(N,H,P,u_b,u_g)$:
N = {adversary, data miner}. Set of sequences H = { ∅, (T), (T,h) } s.t. T ∈ *S*, and h ∈ *C*, where *S* and *C* are the strategy spaces for the adversary and the data miner respectively. Function P assigns a player to each sequence in H: P(∅)=adversary and P((T))=data miner. There is a corresponding function A that assigns a strategy space to each sequence in H: A(∅) = *S*, A((T)) = *C*, and A((T,h)) = ∅. Payoff functions u_b and u_g are defined as above.

We assume the adversary knows which data mining algorithm is used and which attributes are being measured by the data miner. Furthermore we assume each player knows the other's utility function. Hence the players have perfect information in the game. We assume both are rational players. Hence each player wants to maximize their own utility. Therefore a sub-game perfect equilibrium of the Adversarial Stackelberg Game can be expressed as follows.

Let $h_T(X)$ be the data miner's best defensive rule against transformation T. Let $L(h_T,g)$ be the region where the objects are identified as legitimate under the defensive rule $h_T(X)$. The adversary gain W(T) of applying transformation T is the expected value of the profit generated by a misidentified transformed attack object under the data miner's best defensive rule against T.

$$W(T) = u_b(T,h_T) = \int_{L(h_T,g)} g(T,X) f^T{}_b(X)\, dX .$$

A sub-game perfect equilibrium is $(T^e, h_T{}^e)$, where

$$T^e = \operatorname{argmax}_{T \in S} W(T) .$$

When an equilibrium is reached, obviously neither player has the incentive to change its action. When the strategy space is compact and the adversary gain is continuous, there exists a solution for the optimization problem. If the two players' utility functions have this relationship, $u_b(T,h) = -u_g(T,h)$, the Adversarial Stackelberg Game becomes a zero-sum game. Hence our game theoretic framework is more general and is able to handle the minimax solution as well.

Compared with previous work, in our model, the type of data mining algorithm and the set of attributes chosen by the data miner is an initial step that is not directly modeled in our game theoretic framework. Actually the data miner has an advantage by this initial choice. Being able to choose the type of data mining algorithm and the set of attributes means ultimately the data miner controls the rule of the game. When the equilibrium payoff for the data miner is unsatisfactory, the data miner is able to change the rule of the game by

1) increasing the penalties for certain attributes;
2) selecting a different set of attributes;
3) switching to another data mining algorithm.

2.2 Search for An Equilibrium

It is straight forward to see from the above expression that we cannot obtain an explicit expression of a sub-game perfect equilibrium even under a simple distribution such as Gaussian mixture. We therefore use computational algorithms to search for an equilibrium.

The first step is to be able to evaluate the adversary gain given a transformation. Given a transformation T, we can generate transformed attack objects based on the observed data, and use the generated objects to re-train the data mining algorithm with the chosen set of attributes to obtain the optimal defensive rule against the transformation, $h_T(X)$. We then use Monte Carlo integration to evaluate the adversary gain W(T) for the transformation T. The adversary gain can be written as

$$W(T) = \int I_{L(h_T,g)}(X)\, g(T,X) f^{T}{}_{b}(X)\, dX ,$$

where $I_{L(h_T,g)}(X)$ is an indicator function. It equals to 1 if an attack object is not identified by the defensive algorithm, and 0 otherwise. We generate *m* transformed attack objects under transformation T, examine which ones can successfully pass the optimal defensive rule $h_T(X)$. Each unidentified attack object generates a profit for the adversary according to g(T,X). The average of the profits from the unidentified attack objects is an estimate of the adversary gain W(T). Increasing *m* will increase the accuracy of the estimated adversary gain.

Next we use a stochastic search algorithm to find an equilibrium of the Adversarial Stackelberg game. Any stochastic search algorithm with good properties will serve the purpose. In [14], we used the simulated annealing algorithm to search for an equilibrium. Simulated annealing algorithm is able to converge to a global optimum. The algorithm goes through a slow cooling process and settles down at the lowest energy state when the computation budget is unlimited. The simulated annealing algorithm is as follows.

- Set parameters TempMin, TempMax, reduction rate R (0<R<1), sample size N
- Let T_c be the starting transformation with evalc = $W(T_c)$
- Let T_g be the best transformation in the search so far with evalg = $W(T_g)$
- $T_g = T_c$ and $W(T_g) = W(T_c)$
- TempCurrent = TempMax

- WHILE TempCurrent >= TempMin DO
  FOR i = 1 to N DO
  Randomly select T_n in the neighborhood of T_c
  Let evaln = $W(T_n)$
  IF evaln > evalc THEN
  $T_c = T_n$, evalc = evaln
  IF evalg < evaln THEN
  $T_g = T_n$, evalg = evaln
  END IF
  ELSE IF rand(0,1) <= exp{(evaln-evalc)/TempCurrent} THEN
  $T_c = T_n$, evalc = evaln
  END IF
  END FOR
  TempCurrent = TempCurrent × R
- END WHILE

An Adversarial Stackelberg game may have multiple equilibria. The adversary receives the same maximum adversary gain at each equilibrium, but the data miner may see different payoffs. Theoretically we should run a stochastic algorithm multiple times and examine the data miner's worst and best equilibrium payoffs.

2.3 Equilibrium Performance and Attribute Selection

A data mining algorithm's initial success cannot guarantee its good performance against active adversaries. The data miner needs a proper criterion to measure a defensive algorithm's long term performance. There are three quantities that can potentially serve as such a measure.

1) Adversary's equilibrium transformation;
2) Adversary's equilibrium gain;
3) Data miner's equilibrium payoff.

We believe a data mining algorithm's long term success should be measured from the data miner's perspective. When the adversary significantly alters the attack objects to make them similar to the legitimate objects, the unidentified attack objects then become harmless. As long as the data miner does not suffer from major damages, it might tolerate a few unidentified attack objects. For instance, we still receive a lot of spam emails today. A defensive algorithm is put in place mainly to prevent the data miner from suffering significant damages. Hence we choose the data miner's equilibrium payoff to measure a defensive algorithm's long term effectiveness.

Often there are a number of attributes that can be measured from an object. A data mining algorithm may monitor only a few of them. Data miner's equilibrium payoff can then be used to select the best subset of attributes. Notice many factors interact with each other to determine the attributes' equilibrium performance. Below is an example. Consider a data mining algorithm that monitors only one attribute, and there are three attributes available. Assume the profit function of a transformed attack object is the following.

$$g(T,X) = \max(1 - a|T^{-1}(X)-X|, 0) .$$

X is the transformed value, and $T^{-1}(X)$ is the original one. $|T^{-1}(X)-X|$ measures the extent of transformation. a is the penalty per unit transformation. The maximum profit generated by an unidentified attack object is 1, and it decreases linearly due to transformation. The minimum profit is 0.

Further assume the "good" and the "bad" class each follows a Gaussian distribution. $f_i(X)$, is the density function of $N(\mu_i, \sigma^2_i)$, i = g or b. Assume two classes are of the same size, $p_g = p_b = 0.5$. Let T be a real number and the value of a transformed attribute is simply $T(X) = T \times X$. Under transformation T, $f^T_b(X)$ is the density of $N(T \times \mu_b, T^2 \times \sigma^2_b)$.

Data miner uses a Bayesian classifier to detect the attack objects. Let c(i, j) be the cost of classifying an object into class S_i given it actually belongs to class S_j, i = g or b, and j = g or b. In this simple example we set c(g,g) = c(b,b) = 0 and c(g,b) = c(b,g) = 1. The optimal Bayesian classification rule against transformation T is the following.

$$h_T(X) = S_g \text{ if } p_b \times f^T_b(X) <= p_g \times f_g(X); \ S_b \text{ otherwise.}$$

Given the above set-up, the adversary gain can be re-written as follows.

$$W(T) = \int_{L(h_T,g)} \max(1 - a|T^{-1}(X)-X|, 0) \times f^T_b(X)\, dX .$$

A sub-game perfect equilibrium is (T^e, h_{T^e}), where

$$T^e = \text{argmax}_{T \in R} W(T) .$$

Under this set-up, Table 1 shows three attributes initial distributions, penalties, and their equilibrium classification errors respectively.

Attribute	S_g	S_b	Penalty	Equilibrium Error Rate
X_1	N(1,1)	N(3,1)	a=1	0.16
X_2	N(1,1)	N(3.5,1)	a=0.45	0.13
X_3	N(1,1)	N(4,1)	a=0	0.23

Table 1. X_1, X_2, X_3 Equilibrium Performance.

X_1 sees the heaviest penalty, and X_3 has the best initial performance. However X_2 returns the smallest equilibrium classification error. We cannot select attributes based on penalty alone or their initial performance. To select a subset of attributes, we either exhaustively evaluate the equilibrium performance of every subset, or we can implement a forward or backward selection algorithm for attribute selection.

3 Adversarial Support Vector Machine

Now we focus on how to build resilient classification models against active adversaries. In the presence of active adversaries, data used for training by a data mining algorithm is unlikely to represent the future data the system would observe. What typically flunk these data mining algorithms are targeted attacks by the adversary that aim to make the data mining system dysfunctional by disguising malicious data. Existing data mining algorithms cannot be easily tailored to counter this kind of attack because there is a great deal of uncertainty in terms of how much the attacks would affect the structure of the sample space. Unlike the model discussed in the previous section, here we do not model the attacker's utility function directly. Instead, we assume certain limits on the attacker's capabilities. This implies that attacker's utility increases proportional with the loss of the data miner as long as the attack does not exceed the capabilities of the attacker. Therefore, attack models that foretell how far an adversary would go in order to breach the system, need to be incorporated into data mining algorithms to build a robust decision surface. In [14], we discuss two attack models that cover a wide range of attacks tailored to match the adversary's motives. Each attack model makes a simple and realistic assumption on what is known to the adversary. Optimal SVM strategies are then developed against the attack models.

In our model, let X be a d-dimensional vector of attributes measured from an object. In classification scenario, we often obtain a class label from an object as well. Let $y \in \{-1, +1\}$ be the class label. $y = +1$ indicates an attack object. We consider an adversarial data mining problem where the adversary has the freedom to move only the malicious attack object ($y_i = 1$) in any direction by adding a non-zero displacement vector δ to attack object vector. For example, in spam-filtering scenario, the adversary may add good words to spam e-mail to defeat spam filters. On the other hand, adversary will not be able to modify legitimate e-mails. We make no specific assumptions on the adversary's knowledge of the defensive system. Instead, we simply assume there is a trade-off or cost of changing malicious objects. For example, a practical strategy often employed by an adversary is to move the malicious objects in the feature space as close as possible to where the innocuous objects are frequently observed. However, the adversary can only alter a malicious object in such a way that its malicious utility is not completely lost. If the adversary moves an object too far away from its own class in the feature space, the adversary may have to sacrifice much of the malicious utility of the original object. For example, in order to prevent detection, the attacker may reduce the download speed of the sensitive data gathered after a successful cyber attack, since some of the recent intrusion detection systems try to detect sensitive data exfiltration from the system. On the other hand, reducing the sensitive data download speed to zero that may prevent detection would kill entire purpose of the attack. Clearly, different type of modifications by the attacker has different costs and benefits for the attacker. In this adversarial SVM work, we model these costs and benefits as a limit on the attacker's capabilities. In [14], we consider different attack models. To further clarify this modeling choice, we explain the free range attack model from [14] below.

3.1 Free Range Attack

In this attack model, we assume every attribute is bounded. For the j-th attribute, $x_j \in [\, x_j^{min}, x_j^{max} \,]$. In the free range attack model, the attacker can modify the j-th attribute of the i-th attack object, x_{ij}, by adding δ_{ij} to x_{ij}, where δ_{ij} satisfies the following constraint

$$C_f(x_{.j}^{\min} - x_{ij}) \le \delta_{ij} \le C_f(x_{.j}^{\max} - x_{ij}),$$

where C_f is between 0 and 1. In this model, the parameter C_f controls the extent of the attacker's capabilities on modifying the j-th attribute. $C_f = 0$ means attacker cannot modify the data at all. And $C_f = 1$ corresponds to the most aggressive attacks involving the widest range of permitted data movement. The great advantage of this attack model is that it is sufficiently general to cover all possible attack scenarios as far as data modification is concerned. When paired with a defensive algorithm, the combination would produce good performance against the most severe attacks. However, when there are only mild attacks, the defensive algorithm can become paranoid and its performance suffers accordingly.

After defining the above attack model, we can incorporate it into the SVM model to make the resulting classifier more resilient against various attacks. In the context of SVM, by using mathematical

tricks such adding a slack variable and considering the dual problem, we can rewrite the classic SVM optimization problem as follows:

$$
\begin{aligned}
\underset{w,b,\xi_i,t_i,u_i,v_i}{\operatorname{argmin}} \quad & \tfrac{1}{2}||w||^2 + C\sum_i \xi_i \\
s.t. \quad & \xi_i \geq 0 \\
& \xi_i \geq 1 - y_i \cdot (w \cdot x_i + b) + t_i \\
& t_i \geq \sum_j C_f \left(v_{ij}(x_j^{max} - x_{ij}) - u_{ij}(x_j^{min} - x_{ij})\right) \\
& u_i - v_i = \tfrac{1}{2}(1 + y_i)w \\
& u_i \succeq 0 \\
& v_i \succeq 0
\end{aligned}
$$

In the above formulation, the adversarial SVM algorithm explicitly considers the fact that objects can be transformed by the attacker by integrating C_f in the SVM optimization formulation. In other words, unlike the traditional SVM, our adversarial SVM algorithm takes into consideration of the future potential attacks by the adversary at model building time to make the defensive algorithm more resistant to attacks.

3.2 Simulation

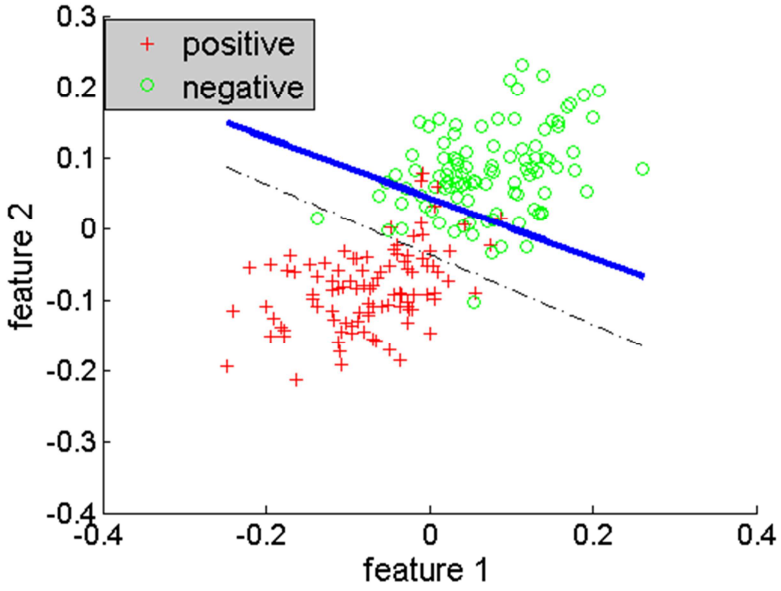
In order to illustrate the effect of the above discussed adversarial SVM modeling, we provide some results on synthetic datasets. For Figures 1(a) and 1(b), we use a synthetic data with two features. Figure 1(a) shows the classification boundaries on the training data, before the adversary lounged an attack. The standard SVM missed a few "bad" objects (e.g., spam emails) and correctly classified most of the "good" objects (e.g., legitimate emails). Adversarial SVM, in anticipation of a future attack, blocked most of the "bad" objects and misclassified a few "good" objects. Figure 1(b) shows the classification boundaries on the test data with attack objects, both the transformed ones and the ones in the original form. The standard SVM failed to block a large number of the attack objects, while Adversarial SVM succeeded in blocking most of the attack objects.

4 Conclusion

Our proposed game theoretic framework serves multiple purposes. First, it is used to evaluate the equilibrium performance of any adversarial data mining technique. The equilibrium performance of an adversarial data mining technique is an indicator of its long term effectiveness. When the equilibrium payoff for the data miner is unsatisfactory, the data miner can completely change the rule of the game. Secondly, our game theoretic model indicates that we need to build decision boundaries close to the good class but not too close to prevent too many false positives. Based on this intuition from our game theoretic model, we propose the Adversarial SVM algorithms, which exhibit robust performance against various attacks in the form of data modification. We further extend our basic model to an adversarial Bayesian relevance vector machine model [15] and the adversarial Hierarchical Mixtures of Experts [16]. Given the limited space we cannot cover the extensions in details in this survey article. More information can be found in [15] and [16].

One important conclusion of our work and similar adversarial data mining work is that when data analytics and data mining techniques are used to detect malicious behavior and events, the adaptation of the attacker strategies must be explicitly considered when building a defensive algorithm. Otherwise, the data mining techniques and decisions support systems used in cyber defense can quickly become useless due to the adversaries potential evasion techniques and adaptation.

(a)



(b)

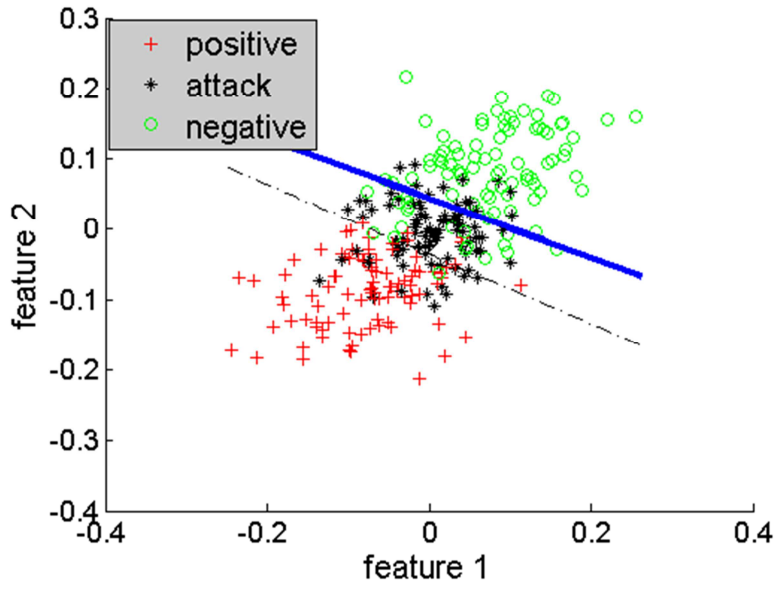


Figure 1. Classification boundaries. + is for the untransformed "bad" objects; o is for the "good" objects; ***** is for the transformed "bad" objects, i.e., the attack objects. The black dashed line is the standard SVM classification boundary, and the blue line is the Adversarial SVM classification boundary. Both the untransformed and the transformed "bad" objects are what we want to detect and block.

5 References

[1] M. Bruckner and T. Scheer. Nash equilibria of static prediction games. In Advances in Neural Information Processing Systems. MIT Press, 2009.

[2] M. Bruckner and T. Scheer. Stackelberg games for adversarial prediction problems. In Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2011.

[3] N. Dalvi, P. Domingos, Mausam, S. Sanghai, and D. Verma. Adversarial classification. In Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '04, pages 99-108, New York, NY, USA, 2004. ACM.

[4] O. Dekel and O. Shamir. Learning to classify with missing and corrupted features. In Proceedings of the International Conference on Machine Learning, pages 216-223. ACM, 2008.

[5] O. Dekel, O. Shamir, and L. Xiao. Learning to classify with missing and corrupted features. Machine Learning, 81(2):149-178, 2010.

[6] L. El Ghaoui, G. R. G. Lanckriet, and G. Natsoulis. Robust classification with interval data. Technical Report UCB/CSD-03-1279, EECS Department, University of California, Berkeley, Oct 2003.

[7] A. Globerson and S. Roweis. Nightmare at test time: robust learning by feature deletion. In Proceedings of the 23rd international conference on machine learning, ICML '06, pages 353-360. ACM, 2006.

[8] M. Kantarcioglu, B. Xi, and C. Clifton. Classifier evaluation and attribute selection against active adversaries. Data Min. Knowl. Discov., 22:291-335, January 2011.

[9] G. R. G. Lanckriet, L. E. Ghaoui, C. Bhattacharyya, and M. I. Jordan. A robust minimax approach to classification. Journal of Machine Learning Research, 3:555-582, 2002.

[10] W. Liu and S. Chawla. Mining adversarial patterns via regularized loss minimization. Mach. Learn., 81:69-83, October 2010.

[11] D. Lowd. Good word attacks on statistical spam fillters. In Proceedings of the Second Conference on Email and Anti-Spam (CEAS), 2005.

[12] D. Lowd and C. Meek. Adversarial learning. In Proceedings of the eleventh ACM SIGKDD international conference on knowledge discovery in data mining, KDD '05, pages 641-647, 2005.

[13] C. H. Teo, A. Globerson, S. T. Roweis, and A. J. Smola. Convex learning with invariances. In Advances in Neural Information Processing Systems, 2007.

[14] Y. Zhou, M. Kantarcioglu, B. M. Thuraisingham, B. Xi. Adversarial support vector machine learning. KDD 2012: 1059-1067

[15] Y. Zhou, M. Kantarcioglu, B. M. Thuraisingham. Sparse Bayesian adversarial learning using relevance vector machine ensembles. ICDM 2012: 1206-1211

[16] Y. Zhou and M. Kantarcioglu. Adversarial learning with Bayesian hierarchical mixtures of experts. SIAM Data Mining 2014, to appear.