# A Game Theoretical Framework for Adversarial Learning

Murat Kantarcioglu
University of Texas at Dallas
Richardson, TX 75083, USA
muratk@utdallas

Bowei Xi
Purdue University
West Lafayette, IN 47907, USA
xbw@stat.purdue.edu

Chris Clifton
Purdue University
West Lafayette, IN 47907, USA
clifton@cs.purdue.edu

## Abstract

*Many data mining applications, ranging from spam filtering to intrusion detection, are faced with active adversaries. In all these applications, initially successful classifiers will degrade easily. This becomes a game between the adversary and the data miner: The adversary modifies its strategy to avoid being detected by the current classifier; the data miner then updates its classifier based on the new threats. In this paper, we investigate the possibility of an equilibrium in this seemingly never ending game, where neither party has an incentive to change. Modifying the classifier causes too many false positives with too little increase in true positives; changes by the adversary decrease the utility of the false negative items that aren't detected. We develop a game theoretic framework where the equilibrium behavior of adversarial learning applications can be analyzed, and provide a solution for finding the equilibrium point. A classifier's equilibrium performance indicates its eventual success or failure. The data miner could then select attributes based on their equilibrium performance, and construct an effective classifier.*

## 1 Introduction

Many data mining applications, both current and proposed, are faced with an active adversary. Problems range from the annoyance of spam to the damage of computer hackers to the destruction of terrorists. In all of these cases, statistical classification techniques play an important role in distinguishing the legitimate from the destructive. There has been significant investment in the use of learned classifiers to address these issues, from commercial spam filters to research programs such as those on intrusion detection [8] These problems pose a significant new challenge not addressed in previous research: The behavior of a class (the adversary) may adapt to avoid detection. A classifier constructed by the data miner in a static environment won't maintain its optimal performance for long, when facing an active adversary.

An intuitive approach to fight the adversary is to let the classifier adapt to the adversary's actions, either manually or automatically. Such a classifier was proposed in [1], which left open the following issue. The problem is that this becomes a never-ending game between the classifier and the adversary. Or is it never-ending? Will we instead reach an equilibrium, where each party is doing the best it can and has no incentive to deviate from its current strategy? If so, does this equilibrium give a satisfactory result for those using the classifier? Or does the adversary win?

Our approach is *not* to develop a learning strategy for the classifier to stay ahead of the adversary. We instead predict the end state of the "game" – an equilibrium state. We model the problem as a two-player game, where the adversary tries to maximize its return and the data miner tries to minimize the amount of misclassification. We examine under which conditions an equilibrium would exist, and provide a method to estimate the classifier performance and the adversary's behavior at such an equilibrium point (e.g., the players' equilibrium strategies).

Spam filtering is one motivating application. There are many examples of spam e-mails where words are modified to avoid spam filters. We could see that those transformations the adversary makes to defeat the data miner come with a cost: lower response rates. Combining the fact that the reward to the adversary decreases as they try to defeat the data miner, with the data miner's interest in avoiding false positives as well as false negatives, can lead us to an equilibrium where both are best served by maintaining the status quo.

Spam filtering also validates our model by confirming an insight we obtain from the model we develop and experiment in Section 3.2. There is currently intensive debate about the effectiveness of content based filtering, where attributes are easy to modify. On the other hand, by checking the IP address of each email and blocking port 25 from home computers, two attributes expensive to modify, people successfully reduced the overall proportion of spam emails in 2005 [10]. The tools developed in this paper would help developers to arrive at such a decision without spending a long time on making small and constant adjustment to an (eventually) ineffective strategy. The model assumptions shown later in the paper are originally based on spam filtering.

Section 2.1 contains our game theoretic model and how we determine the Subgame Perfect Equilibrium ([9]) and the associated strategies. Section 2.2 discusses techniques used to calculate the equilibrium. Section 3 has the simulation results. We conclude with a discussion of future work. First, however, we will discuss related work, both in adversarial learning and game theory.

### 1.1 Related Work

Learning in the presence of an adaptive adversary is an issue in many different applications. Problems ranging from intrusion detection ([8]) to fraud detection ([3]) need to be able to cope with adaptive malicious adversaries. As discussed in [1], the challenges created by the malicious adversaries are quite different than those previous work such as concept drift ([5]), because the concept is maliciously changed based on the reactions of the classifier. There have been applications of game theory to spam filtering. In [6], the spam filter and spam emails are considered fixed, the game is if the spammer should send legitimate or spam emails, and the user decides if the spam filter should be trusted or not. In [7], the adversary tries to reverse engineer the classifier to learn the parameters. In [1], the authors applied game theory and aimed to produce a Naïve Bayes classifier that could automatically adapt to the adversary's expected actions. They concentrated on a single-shot version of the game. While recognizing the importance of an equilibrium state, they simplified the situation by assuming the adversary bases its strategy on the Naïve Bayes classifier rather than their proposed adaptive strategy.

We take a different approach, directly investigating the equilibrium state of the game, at which point all parties will stick to their current strategies. We aim at providing a guide for how to construct classifiers that could lead to the data miner's eventual success in the game.

## 2 Problem Formulation

In this section we present a game theoretic model for adversarial learning applications, and provide a solution for finding the equilibrium strategies using stochastic simulated annealing and Monte Carlo integration.

### 2.1 A Game Theoretic Model

The adversarial learning scenario can be formulated as a two class problem, where class one ($\pi_1$) is the "good" class and class two ($\pi_2$) is the "bad" class. $n$ attributes would be measured from a subject coming from either classes. Denote the vector of attributes by $x = (x_1, x_2, \ldots, x_n)'$. Assume the attributes of a subject $x$ would follow different distribution for different class. Let $f_i(x)$ be the probability density function of class $\pi_i$, $i = 1, 2$. The overall population is formed by combining the two classes. Let $p_i$ denote the proportion of class $\pi_i$ in the overall population. Note $p_1 + p_2 = 1$. The distribution of the attributes $x$ for the overall population could be considered as a mixture of the two distributions, with the density function written as $f(x) = p_1 f_1(x) + p_2 f_2(x)$.

Assume that the adversary can control the distribution of the "bad" class $\pi_2$. In other words, the adversary can modify the distribution by applying a transformation $T$ to the attributes of a subject $x$ that belong to $\pi_2$. Hence $f_2(x)$ would be changed into $f_2^T(x)$. Each such transformation would have a cost. At the same time, the adversary gains a profit when a "bad" instance ($\pi_2$) is classified as a "good" instance ($\pi_1$). We assume that the values of $p_1$ and $p_2$ will not be affected by the transformation, meaning that adversary would transform the distribution of $\pi_2$ but in a short time period would not significantly increase or decrease the amount of "bad" instances. Here we examine the case where a rational adversary and a rational data miner play the following game:

1. Given the initial distribution and density $f(x)$, the adversary will choose a transformation $T$ from the set of all feasible transformations $S$.

2. After observing the transformation $T$, the data miner will create a classifier $h$.

Consider the case where data miner wants to minimize its (mis)classification cost. Define $c_{ij}$ be the cost of classifying a subject $x \in \pi_i$ given that $x \in \pi_j$. Given transformation $T$ and the associated $f_2^T(x)$, the data miner uses a classifier $h(x)$, and let $L_i^h$ be the region where the instances are classified as $\pi_i$ based on $h(x)$, $i = 1, 2$. The expected cost of classification can be written as ([4]):

$$\begin{aligned} c(T,h) &= \int_{L_1^h} \left[c_{11}p_1 f_1(x) + c_{12}p_2 f_2^T(x)\right] dx \\ &\quad + \int_{L_2^h} \left[c_{21}p_1 f_1(x) + c_{22}p_2 f_2^T(x)\right] dx \end{aligned}$$

Define the payoff function of data miner as $u_2(T,h) = -c(T,h)$. Note that the value of $c(T,h)$ is always positive assuming positive $c_{ij}$ values. In order to maximize payoff $u_2$, data miner needs to minimize $c(T,h)$.

Note that adversary will only profit from the "bad" instances that are classified as "good". Also note that the transformation may change the adversary's profit of an instance that successfully passed the detection. Define $g^T(x)$ as the profit function for a "bad" instance $x$ being classified as a "good" one, after the transformation $T$ being applied. Define the adversary's payoff function of a transformation $T$ given $h$ as the following:

$$u_1(T,h) = \int_{L_1^h} \left(g^T(x) f_2^T(x) dx\right)$$

Within the vast literature of game theory, the *extensive game* provides a suitable framework for us to model the sequential structure of adversary and data miner's actions. Specifically, the *Stackelberg game* with two players suits our need. In a Stackelberg game, one of the two players chooses an action $a_1$ first and the second player, after observing the action of the first one, chooses an action $a_2$. The game ends with payoffs to each player based on their payoff functions $u_1$, $u_2$ and $a_1$, $a_2$. In our model, we assume all players act rationally throughout the game. For the Stackelberg game, this implies that the second player will respond with the action $a_2$ that maximizes $u_2$ given the action $a_1$ of the first player. The assumption of acting rationally at every stage of the game eliminates the Nash equilibrium with non-credible threats and creates an equilibrium called *subgame perfect equilibrium*. Further more, we assume that each player has perfect information about the other. Here in this context, "perfect information" means that each player knows the other player's utility function. Further more, player two observes the $a_1$ before choosing an action. In applications such as spam filtering, this is a reasonable assumption due to publicly available data.

Hence we define the **Adversarial Learning Stackelberg Game**: *A game $G = (N, H, P, u_i)$ is called an Adversarial Learning Stackelberg Game if $N = \{1, 2\}$, set of sequences $H = \{\emptyset, (T), (T, h)\}$ s.t. $T \in S$ and $h \in C$, where $S$ is the set of all admissible transformations for adversary, and $C$ is the set of all possible classification rules given a certain type of classifier. Function $P$ assigns player to each sequence in $H$ where $P(\emptyset) = 1, P((T)) = 2$ (i.e., there exists an corresponding function $A$ that assigns action space to each sequence in $H$ where $A(\emptyset) = S, A((T)) = C$, $A((T,h)) = \emptyset$). Payoff functions $u_1$ and $u_2$ are defined as above.*

We use the minimum cost Bayesian classifier as an example to illustrate how we would solve for the subgame perfect equilibrium. First we will find the best response function for data miner given a transformation $T$. Using the population proportion $p_i$ of each class as the prior probabilities, and after observing $T$ being applied to the "bad" class ($f_2^T(x)$), the optimal classification rule becomes:

$$h_T(x) = \begin{cases} \pi_1 & (c_{12} - c_{22})p_2 f_2^T(x) \le (c_{21} - c_{11})p_1 f_1(x) \\ \pi_2 & \text{otherwise} \end{cases}$$

$h_T(x)$ is the decision rule that minimizes the expected classification cost of the data miner. Given $T$, $h_T$ is the best response of data miner, i.e., $R_2(T) = h_T$. Then the adversary would find the transformation $T$ that belongs to $S$ which maximizes its profit, given the data miner would use $h_T = R_2(T)$ defined above as its classification rule. Let $L_1^{h_T} = \{x : (c_{12} - c_{22})p_2 f_2^T(x) \le (c_{21} - c_{11})p_1 f_1(x)\}$ be the region where the instances are classified as $\pi_1$ given $h_T$. The adversary gain of applying transformation $T$ is:

$$g_e(T) = u_1(T, R_2(T)) = E_{f_2^T}(I_{\{L_1^{h_T}\}}(x) \times g^T(x))$$

which is the expected value of the profit generated by the "bad" instances that would pass detection under transformation $T$. Therefore we can write the subgame perfect equilibrium as $(T^*, h_{T^*}(x))$, where

$$T^* = argmax_{T \in S} \; (g_e(T)) \,. \tag{1}$$

*Game theory ([9]) established that the solution of the above maximization problem is a subgame perfect equilibrium. Furthermore if the action space $S$ is compact and $g_e(T)$ is continuous, the maximization problem has a solution.*

Another important aspect of the Adversarial Learning Stackelberg game and its subgame perfect equilibrium is that once an equilibrium point is reached, even if the game is repeated, both parties will not have an incentive to change their actions.

**Theorem 1.** *Let us assume that the adversarial learning Stackelberg game is played $n$ times for finite $n$. Let us also assume that current $f(x) = p_1 f_1(x) + p_2 f_2(x)$ is reached after playing the game $k$ times and after adversary used $T^*$, the subgame perfect equilibrium strategy defined by Equation 1, in the $k^{th}$ game. Also assume that parties will change their actions if they increase their payoff. This implies that adversary will not change $f_2(x)$ in the $j^{th}$ round where $k < j \le n$. Similarly, the data miner will not change $h_{T^*}(x)$ in the $j^{th}$ round where $k < j \le n$.*

*Proof.* Omitted. □

The above formulation could accommodate any well defined set of transformations $S$, any appropriate distributions with densities $f_1(x)$ and $f_2(x)$, and any meaningful profit function $g^T(x)$. Next we present how above equations can be solved in practice.

### 2.2 Solving for the Equilibrium

Since the domain of the integration $L_1^{h_T}$ for the adversary gain $g_e(T)$ is a function of the transformation $T$, finding an analytical solution to the maximization problem is very challenging. In addition, even calculating the integration analytically for a specific transformation is not possible for high dimensional data. Instead, we use Monte Carlo integration technique that generally converts a given integration problem to computing an expected value. The adversary gain $g_e(T)$ can be written as:

$$g_e(T) = \int \left( I_{L_1^{h_T}}(x) \times g^T(x) \right) \; f_2^T(x)dx$$

In the above formula, $I_{L_1^{h_T}}(x)$ is the indicator function and returns 1 if $x$ is classified into $\pi_1$, else it returns 0. $f_2^T(x)$ is naturally a probability density function. Therefore $g_e(T)$ could be calculated by sampling $m$ points from $f_2^T(x)$, and taking the average of $g^T(x)$ for the sample points that satisfy $(c_{12} - c_{22})p_2 f_2^T(x) <= (c_{21} - c_{11})p_1 f_1(x)$.

We consider stochastic search algorithms for finding an approximate solution for Equation 1. Especially, in our case, a stochastic search algorithm with the ability to converge to the global optimal solution is desired. To satisfy this goal, a simulated annealing algorithm is implemented to solve for the subgame perfect equilibrium. [2]

## 3 Simulation Study

We have done simulations to examine various equilibrium strategies. Gaussian distributions and minimal cost Bayesian classifier are applied in the experiments. Gaussian distributions have a particularly helpful property: after a linear transformation of the attributes, we still have a Gaussian distribution and an explicit expression for the density. This combination as a simple example gives us important insight about how costs could affect the equilibrium strategies.

### 3.1 Profit Function and Gaussian Mixture

First define the profit function $g^T(x)$ as:

$$g^T(x) = g - a \left| T^{-1}(x) - x \right|_1 , \qquad (2)$$

where $x$ is the transformed “bad” instance, $T^{-1}(x)$ is the original one, and $g$ and $a$ are positive constant numbers. To quantify the difference of the “bad” instance $T^{-1}(x)$ before and after transformation $T$, we compute the $L_1$ norm of $T^{-1}(x) - x$. This is simply adding up the absolute differences of the individual attributes before and after transformation $T$. The constant value $g$ is the constant profit generated by original instances. In our preliminary simulation study, we assume the profit would decline linearly according to the extent of the transformation. Here $a$ is the reduction rate. This definition of the profit is based on the following intuition: The more the original distribution changes, the higher the cost for the adversary. Although more “bad” instances could avoid being detected, each instance would generate less profit for the adversary. Hence it is possible to reach a point that adversary stops modifying the instances, and the equilibrium is established. Further assume that each class $\pi_i$, $i = 1, 2$, has a Gaussian distribution. $f_i(x)$ is the density function for Gaussian distribution $N(\mu_i, \Sigma_i)$.

Consider the set of linear transformations $S$. Define $T$ as a $n \times n$ real matrix, the transformed instance $x$ has every element $x_j$ as a linear combination of the original attributes $(T_1^{-1}(x), T_2^{-1}(x), ..., T_n^{-1}(x))'$. In our preliminary study $S$ will be limited to a certain region, not the entire space of the real matrices. Under transformation $T$, $f_2^T(x)$ becomes the density of $N(T\mu_2, T\Sigma_2 T')$, which is the new distribution for the “bad” class $\pi_2$. Here $T'$ is the transpose of $T$.

Rewrite the subgame perfect equilibrium using the above specifics as follows:

$$T^* = argmax_T \left( \int_{L_1^{h_T}} \left( g - a \left| T^{-1}(x) - x \right|_1 \right) \; f_2^T(x)dx \right)$$

, where $f_2^T(x)$ is the density of $N(T\mu_2, T\Sigma_2 T')$.

### 3.2 Experimental Results

It is interesting to see what the equilibrium strategies would become in response to different classification costs and transformation costs. Due to space limitations, we show only one set of experiments. In our setting a classifier changes when the classification cost matrix changes, and the adversary's gain is affected by the profit function under a transformation $T$. In this section we search for approximate equilibrium results under various classification cost matrices and profit functions. Table 1 contains the parameter values (rounded to 4 digits after the decimal point) for the Gaussian distributions. Notice there is no linear transformation $T$ such that $f_2^T(x) = f_1(x)$.

In our cost matrices, the correct classification costs are fixed to be 0, i.e., $c_{11} = c_{22} = 0$. We would modify the misclassification costs of classifying a “bad” instance as “good” and a “good” instance as “bad”. (Please note that $c_{ij}$ is the cost of deciding $x \in \pi_i$ given that $x \in \pi_j$. In our case, $\pi_2$ is the “bad” class and $\pi_1$ is the “good” class). Different profit reduction rates for the adversary are also considered.

**Table 1. Mean and standard deviation for $\pi_1$ and $\pi_2$.**

| | $\pi_1$ | | $\pi_2$ | |
|---|---|---|---|---|
| Attribute | $\mu$ | $\sigma$ | $\mu$ | $\sigma$ |
| 1 | $-0.7564$ | $0.9595$ | $-0.6461$ | $0.7056$ |
| 2 | $-0.7324$ | $1.0411$ | $-0.5501$ | $0.8935$ |
| 3 | $-1.5979$ | $0.8483$ | $-2.1507$ | $0.7973$ |
| 4 | $-2.8988$ | $1.1486$ | $-1.7248$ | $0.9477$ |
| 5 | $2.4559$ | $0.9872$ | $3.7256$ | $1.2581$ |
| 6 | $3.9976$ | $1.4711$ | $5.3755$ | $1.2593$ |

**Table 2. Experiment Results**

| | $a = 0$ | $a = 0.2$ | $a = 0.7$ | Initial Gain |
|---|---|---|---|---|
| $c_{21}/c_{12} = 1$ | 0.4950 | 0.2036 | 0.1958 | 0.1926 |
| $c_{21}/c_{12} = 2$ | 0.8430 | 0.3156 | 0.3134 | 0.3098 |
| $c_{21}/c_{12} = 10$ | 0.9830 | 0.6250 | 0.6234 | 0.6102 |

The adversary's gain is the expectation of the profit generated by a certain transformation $T$. Note that in the profit function, there are two parameters: the profit without transformation $g$, and the profit reduction rate $a$. In the experiments, without loss of generality, we fix $g$ to be $1$ and change the value of $a$.

Combining the cost matrices and profit functions defined above, we performed nine experiments corresponding to combinations of the above. We restricted our search space to matrices with entries chosen from $[-1, 1]$.

For each cost matrix of the data miner, the initial gain of the adversary (i.e., choosing the identity matrix as the transformation) and our experimental results are reported in Table 2.

The experiments show that for increasing profit reduction rate $a > 0$, simulated annealing cannot find a transformation within the search space that improves the gain of the adversary significantly better than the identity transformation. For $a = 0$, the adversary can increase its gain significantly by using transformation to defeat the filter.

The experiments identified two rather extreme equilibrium strategies. 1) The cost for misclassified "good" instances is much higher than for misclassified "bad" instances (i.e., $c_{12}p_2 < c_{21}p_1$), and there is no penalty for the adversary to perform transformations. The equilibrium strategy for the classifier is to pass most of the instances, good and bad alike; the adversary would transform its class ($\pi_2$) to have the similar distribution as the "good" class ($\pi_1$). 2) Under equal misclassification costs, equal population size, and severe penalty for transformation, the classifier would minimize the total number of misclassified instances; the adversary would not attempt to perform a transformation (i.e., perform the identity transformation). We could see when under more severe penalty, an adversary has less incentive to change.

## 4 Conclusion

Many classification problems operate in a setting with active adversaries: while one party tries to identify the members of a particular class, the other tries to reduce the effectiveness of the classifier. Although this may seem like a never-ending cycle, it is possible to reach a steady-state where the actions of both parties stabilize. The game has an equilibrium because both parties facing costs: costs associated with misclassification on the one hand, and for defeating the classifier on the other. By incorporating such costs in modeling, we can determine where such an equilibrium could be reached, and whether it is acceptable to the data miner.

## References

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

[2] R. Duda, P. E. Hart, and D. Stork. *Pattern Classification*. John Wiley & Sons, 2001.

[3] T. Fawcett and F. J. Provost. Adaptive fraud detection. *Data Mining and Knowledge Discovery*, 1(3):291–316, 1997.

[4] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, San Diego, CA, 1990.

[5] G. Hulten, L. Spencer, and P. Domingos. Mining time-changing data streams. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 97–106, San Francisco, CA, 2001. ACM Press.

[6] D. K. V. Ion Androutsopoulos, Evangelos F. Magirou. A game theoretic model of spam e-mailing. In *Proceedings of the 2nd Conference on Email and Anti-Spam (CEAS 2005)*, 2004.

[7] D. Lowd and C. Meek. Adversarial learning. In *KDD '05: Proceeding of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 641–647, New York, NY, USA, 2005. ACM Press.

[8] M. V. Mahoney and P. K. Chan. Learning nonstationary models of normal network traffic for detecting novel attacks. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 376–385, New York, NY, USA, 2002. ACM Press.

[9] M. J. Osborne and A. Rubinstein. *A Course in Game Theory*. MIT Press, 1999.

[10] Spam accounts for 68% of year's email, 2005.