A Game Theoretic Approach to Adversarial Learing

by

Murat Kanatarcioglu
The University of Texas at Dallas

Bowei Xi
Purdue University

Chris Clifton
Purdue University

Technical Report #05-06

Department of Statistics
Purdue University

October 2005

A Game Theoretic Approach to Adversarial Learning*

Murat Kantarcıoğlu
University of Texas at Dallas
muratk@utdallas.edu

Bowei Xi
Purdue University
xbw@stat.purdue.edu

Chris Clifton
Purdue University
clifton@cs.purdue.edu

Abstract

Open environments such as the internet lead to conflict between those whose goals are at odds: email users vs. spam, legitimate users vs. malicious hackers, web search engines vs. pages desiring high rankings, etc. This paper uses a game theoretic approach to identify a steady-state: What happens when both parties are doing the best they can to achieve their (conflicting) goals? We demonstrate that in a spam email setting, filters should concentrate on attributes that are expensive for the spammer to modify.

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, data mining has been proposed as a solution: from training spam filter to using data mining to identify terrorists.

These problems pose a significant new challenge: The behavior of a class (the adversary) may adapt to avoid detection. Of course, the data miner can also adapt, in a never-ending information "arms race".

Or is it never-ending? Will we instead reach a Nash Equilibrium[7], where each party is doing the best it can? If so, does this equilibrium give a satisfactory result for the data miner? Or does the adversary win?

In the worst case, the adversary's data would be indistinguishable from "good" data. The result would be that data mining would be useless: no matter what we do, we could do no better than a pure guess. However, the more the adversary's data looks like real data, the less value it has to the adversary. Imagine spam that exactly matched real email, viruses that had no impact, or terrorists who never did anything wrong. For the adversary's strategy to have value, it must have some impact - and the more the adversary tries to look like real data, the less the value will be.

For example, look at the emails in Figures 1-3. Email 1 clearly identifies what it is trying to accomplish: convince the reader to buy a product. Email 2 is actually a link to

*This material is based upon work supported by the National Science Foundation under Grant No. 0428168.

Subject: Clean up your PC with Window Washer
From: Real Accessories <NEWS@REAL-EMAIL.NET>
Date: Tue, 27 Sep 2005 20:18:05 -0700
To: dreamxxx@yahoo.com

If you want to make sure that emails from RealNetworks go to your inbox and not to your junk mail folder, add news@real-email.net to your address book.

Real Accessories

You can't re-write history, but now you can erase it.

Window Washer is an award-winning privacy and clean-up utility that scours your browser's cache, cookies, history, recent document list and much more.

Get more Information now

Improve your computer's performance and protect your privacy with Window Washer.

- Erase any history of Internet and PC activities
- "Bleach" deleted files to make them unrecoverable
- Unclog disk space to increase system performance
- Clean files as you work
- Fully customisable

Get More Information Now

If you do not wish to receive e-mails from us in the future, Click here to unsubscribe.

Need Customer Support?

Contact us at: http://service.real.com/realone

Have questions regarding our email privacy policy?

Contact us at:
Email Privacy Policy Group
RealNetworks, Inc.
P.O. Box 91123
Seattle, WA 98111-9223

privacy policy

Figure 1: Straightforward advertising spam email.

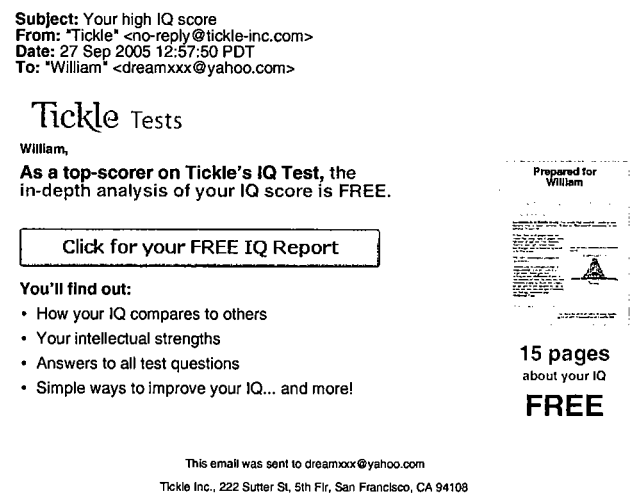

Figure 2: Misleading email designed to avoid filters.

```
From: "Ezra Martens" <ezrabngktbbem...
To: "Eleftheria Marconi" <clifton@pu...
Subject: shunless Phaxrrmaceutical
Date: Fri, 30 Sep 2005 04:49:10 -0500

Hello,
Easy Fast =
Best Home Total
OrdeShipPrricDelivConf
ringpingeseryidentiality
VIAAmbCIALevVALXan
GRAienLISitraIUMax
$  $ $
3.33 1.21 3.75
Get =
additional informmation attempted to
```

Figure 3: Spam email with text modified to avoid filters.

advertisements that have nothing to do with an IQ test – the "test results" is simply a way to convince filters (including the human) to let the email pass. Email 3 is heavily modified to avoid looking like what it actually is: An advertisement for mail-order pharmaceuticals (the HTML version uses strange font and style commands to try to be slightly more readable; the text version is shown to show just how far spammers go to avoid filters.) Which is more likely to get the reader to make a purchase? The response rate (actual sales) of the straightforward request will probably be higher than the same message placed through an ad linked from the IQ test, as many likely purchasers would ignore the bogus IQ test results.

Thus we see that the transformations the adversary makes to defeat the data miner come with a cost. 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.

In this paper, we model this issue with respect to a spam filtering problem. Representing the problem as a two-player game, where the spammer tries to maximize return and the filter tries to minimize the amount of spam while retaining good email, we test where a Nash equilibrium occurs. In many respects, this is similar to the work of [3]; the difference is that we carry the game to a steady-state conclusion.

In addition to modeling the problem, we present simulation results. The simulation is based on the Spam dataset in the UCI machine learning repository [1]. This repository does not contain sufficient data to actually know how spammers would transform their advertisements to avoid detection. Instead, we use the repository to set parameters distinguishing spam and non-spam emails, giving a starting point for the game. We then search for the equilibrium point for various misclassification costs (for the filter) and transformation costs (for the spammer.)

Before going into details, we would like to emphasize that our formulation applies to many real life scenarios, such as intrusion detection and profiling for homeland security, not only spam filtering. Wherever there are two parties involved, and adversary would try to avoid being detected by modifying their current strategy to mimic the behavior of the other party, the scenario would fit into our formulation.

For example, the proposed Computer Assisted Passenger Prescreening System II (CAPPS II)[2], designed by the Transportation Security Agency (TSA), will classify passengers into three groups. If the passenger is classified as red, he or she will not be able to fly. If the passenger is classified as yellow, he or she will be subject to an increased search. Finally, a passenger classified as green will go through the regular screening process. Clearly, real terrorists will ob-

serve the system and alter their behavior so that they will be classified as green. Similarly, consider an intrusion detection system where each TCP/IP connection is monitored for intrusion. The rules used to identify an attacker would be based on unusual patterns compared with a real user. An adversary would then change its attack pattern in order to avoid detection.

Because we have access to a real spam email data, spam filtering is our experimental application, where the parameter values are the ones observed from the real data and functions meaningful to the application. The results show that such an equilibrium can be reached. Perhaps more important, they show that if the cost to the adversary for transforming a message is high (e.g., the response rate of the transformed spam is very low), then there is little or no benefit to the spammer from transforming the messages to defeat spam filters. This has two impacts: First, spam filters need to be designed to look at features that are expensive to modify. Second, perhaps by publicizing this work, spammers will realize that messages like those in Figures 2 and 3 are worthless, and will instead concentrate on spam that is easily filtered but likely to be of interest to those who do receive it.

In Section 2.1 we show specifically how we model the problem, and derive an approach to determining the Nash Equilibrium. Section 2.2 discusses techniques used to calculate the equilibrium. We give a simple example in 2.3. Section 3 shows how we model the actual spam dataset using this approach, and gives results. We conclude with a discussion of future work. First, however, we will discuss related work, both in spam filtering 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[11] to fraud detection[5] need to be able to cope with adaptive malicious adversaries. The challenges brought by the malicious adversaries are quite different than the previous work such as concept drift[8]. In our problem, the concept is maliciously changed based on the reactions of the classifier.

There have been applications of game theory to spam filtering. In [9], 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. We instead look at this from the view of designing an optimal filter in the presence of spam emails designed to get past the filter. In [10], the adversary tries to reverse engineer the classifier to learn the parameters. In our work, we assume that the classifier and the adversaries actions are known to each other and try to find the Nash equilibrium in such a setting. To our knowledge, only the work of Dalvi et. al. [3] is directly related to our problem. In [3], authors developed a game theoretical framework where the adversary modifies its input to avoid being detected by a "Naive Bayes" classifier. They gave heuristic solutions to find the best action by the adversary given the parameters of the "Naive Bayes" classifier and vice versa. Compared to their work, we provide a general formulation that can be applied to many different settings and many different classifiers (i.e, not specific to "Naive Bayes" classifier). Also, using backward induction, we show how stochastic search techniques can be used to find Nash equilibria for adversarial learning games.

2 Dynamic Games with Complete Information

In this section, we rigorously formulate the problem using a game theory approach, and provide a solution based on stochastic simulated annealing and Monte Carlo integration.

Everyday, spammers change their e-mails to pass spam filters in order to reach the users under their protection. Based on the changes done by spammers, the classification rules are updated in spam filters to block the modified spam e-mails. This process can be considered as a game between a spammer and a spam filter.

2.1 Formulation of the Game Our goal is to give a reasonable framework where we can analyze such games and find the Nash equilibriums of those games. Nash equilibrium for our application is the point where given the spammer's strategy, the spam filter has no incentive to change its rules and given the spam filter rules, the spammer has no incentive to modify his e-mails ([7]).

The spam filtering scenario can be formulated as a two class problem, where class one (π_1) is the regular class and class two (π_2) is the "spam" 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 a different distribution for different class values. Let $f_i(x)$ be the probability density function of class i, $i = 1, 2$. The overall population is formed by combining the two classes. Let p_i denote the proportion of class 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 "spam" class π_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 π_2. Hence $f_2(x)$ would be changed into $f_2^T(x)$. Each such transformation would have a cost, for example due to the lost effectiveness of modified spam emails. On the other hand, the adversary gains a profit when a spam (π_2) is classified as a regular e-mail (π_1).

The spam filter attempts to classify the messages and block the spam emails. Here we examine the case where

a rational adversary and a rational spam filter play the following two stage 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 spam filter will create a classifier that minimizes its cost.

In reality misclassifying a regular email into a spam email would have more serious consequence than failing to block a spam email. Hence we consider a minimum cost Bayesian classifier. Define c_{ij} be the cost of classifying a subject $x \in \pi_i$ given that in fact $x \in \pi_j$. We stress that the framework and the solution we present in the paper could adopt any classifier.

Using the population proportion p_i of each class as the prior probabilities, and after observing the transformation being applied to the "spam" class ($f_2^T(x)$), the Bayesian classifier considering the cost of each action ([6]) is:

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

We assume that the values of p_1 and p_2 will not be affected by the transformation, meaning that spammer would modify the emails but in a short time period would not significantly increase or decrease the number of spam emails sent out.

The spammer's goal is to find the transformation T that belongs to S which maximizes his profit. By using transformation T, the spammer attempts to increase the number of spam instances that are classified as regular ones. Meanwhile the transformation may change the adversary's gain of an instance successfully passed the detection. Define $g^T(x)$ as the profit function for a spam instance x which is classified as a regular one, after the transformation T being implemented.

After observing the transformation T, spam filter would use $h_T(x)$ defined above as its classification rule. Let $L_1^T = \{x : (c_{12} - c_{22})p_2 f_2^T(x) \leq (c_{21} - c_{11})p_1 f_1(x)\}$ be the region where the instances are classified as π_1 based on $h_T(x)$. Define the adversary gain of applying transformation T as:

$$g_e(T) = \int_{L_1^T} \left(g^T(x) f_2^T(x) dx\right) = E_{f_2^T}(I_{\{L_1^T\}}(x) \times g^T(x)),$$

which is the expected value of the profit generated by the spam instances that pass the spam filter under transformation T. The above definition of the adversary gain would work for any classifier. Here L_1^T is the region that email instances would be classified into class 1 (regular emails) for that classifier, after adjusted to the transformation T performed by the adversary.

Using backwards induction ([7]), we can write the Nash equilibrium as $(T^*, h_{T^*}(x))$ where

$$T^* = argmax_{T \in S} \ (g_e(T)) \tag{2.1}$$

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 a solution under this general setting.

2.2 Stochastic Search Algorithm Since the domain of the integration L_1^T for the adversary gain $g_e(T)$ is a function of the transformation T, finding an analytical solution to Equation 2.1 is very challenging. In addition, even calculating the integration analytically for a specific transformation is not possible for high dimensional data. We have to numerically evaluate $g_e(T)$. Because of these limitations, we consider stochastic search algorithms for finding an approximate solution to Equation 2.1. A typical stochastic search algorithm for maximization problems works as follows: First, algorithm assigns a random initial point and tries to search the solution space by randomly moving to different points based on some selection criteria. Usually, the selection criteria involve calculating the function that needs to be maximized for the current and the new point in the solution space. Clearly, this implies a computationally efficient method for calculating the integration for $g_e(T)$ is required, since the process will be repeated for hundreds of thousands transformations in S. Furthermore a stochastic search algorithm with ability to converge to the global optimal solution is desired.

In the rest of this section, Monte Carlo integration method is introduced to compute $g_e(T)$ and simulated annealing algorithm is implemented to solve for the Nash equilibrium.

2.2.1 Monte Carlo Integration Monte Carlo integration technique generally converts a given integration problem to computing an expected value. Assume that we would like to calculate $\int_S g(x)dx$. If we can find a probability density function $f(x)$ ($\int_S f(x)dx = 1$) which is easy to sample from, then

$$\int_S g(x)dx = \int_S \frac{g(x)}{f(x)} \times f(x)dx = E_f[\frac{g(x)}{f(x)}].$$

The integration equals to the expected value of $g(x)/f(x)$ with respect to the density $f(x)$.

The expectation of $g(x)/f(x)$ is estimated by computing a sample mean. Generate m samples $\{x^{(i)}\}$ from $f(x)$ and calculate $\mu_m = 1/m \times \sum_1^m \left(g(x^{(i)})/f(x^{(i)})\right)$. When sample size m is large enough, μ_m provides an accurate estimate of $\int_S g(x)dx$.

The integration for computing $g_e(T)$ can be rewritten

as:

$$g_e(T) = \int \left(I_{L_1^T}(x) \times g^T(x) \right) f_2^T(x) dx \tag{2.2}$$

In the above formula, $I_{L_1^T}(x)$ is the indicator function and returns 1 if x is classified into π_1, i.e. $(c_{12} - c_{22})p_2 f_2^T(x) <= (c_{21} - c_{11})p_1 f_1(x)$, else 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 $(c_{12} - c_{22})p_2 f_2^T(x) <= (c_{21} - c_{11})p_1 f_1(x)$. The pseudo-code for Monte Carlo integration is given in Algorithm 2.2.1.

Algorithm 2.1 Monte Carlo Integration

{Evaluating $g_e(T)$ for a given transformation T}
Generate m samples $\{x^{(i)}\}$from $f_2^T(x)$
$sum = 0$
for $i = 1$ to m **do**
  if $(c_{12}-c_{22})p_2 f_2^T(x^{(i)}) <= (c_{21}-c_{11})p_1 f_1(x^{(i)})$ **then**
    $sum = sum + g^T(x^{(i)})$
  end if
end for
return sum/m

2.2.2 Simulated Annealing Simulated annealing is a stochastic search method that is based on an analogy taken from physics[4]. Physical systems that have many interacting components can be in any of the possible states based on some probability distribution. For high temperatures, a system can be in any one of the possible states with roughly equal probability. As the temperature decreases, the system will choose a low energy state with higher probability. Similarly, when the temperature is high, our simulated annealing algorithm will select a solution from the search space with roughly equal probability. As the temperature gets lower later in the search, the algorithm will converge to a globally optimal solution.

Our version of simulated annealing algorithm, first selects few random transformations and tries to get a good starting transformation. (Algorithm 2.2.2, Lines:1-3).

After the selection of the initial transformation, for each temperature, a few hundred transformations are selected from the neighborhood of the current transformation (Lines:7-9). New transformation replaces the current transformation if it gives a greater value of $g_e(T)$ (Lines:10-13). In case the new transformation is not better than the current one, simulated annealing algorithm may choose it with some probability. This probability is calculated using the value of the new transformation, the value of the current transformation and the current temperature (Lines:15-17). This probabilistic step enables the algorithm to escape local maxima[4]. In Algorithm 2.2.2 (Line:15), $rand(0, 1)$ generates a random number uniformly distributed between 0 and 1. Also current temperature is reduced by multiplying it with a reduction rate (Lines:19). The whole process is repeated until the algorithm freezes.

Later in our simulation study (Section 3.2), the algorithm appears to converge extremely slowly even when there are only 6 attributes. We force the algorithm to stop when the temperature drops below a prespecified minimum temperature. This may cause the algorithm to stop at a local optimal value. Nonetheless, we obtained good simulation results.

Algorithm 2.2 Simulated Annealing Algorithm for Solving for Nash Equilibrium

Require: $TempMin$,$TempMax$,$0 < ReductionRate < 1$,$SampleSize$
1: Select random T and evaluate $g_e(T)$
2: Let T_c be the starting transformation with value $evalc = g_e(T_c)$
3: Let T_g be the best transformation seen in the search with value $evalg = g_e(T_g)$
4: $T_g = T_c$, $evalg = evalc$
5: $TempCurrent = TempMax$
6: **while** $TempCurrent \geq Tempmin$ **do**
7: **for** $i = 1$ *to* $SampleSize$ **do**
8: Randomly select T_n in neighborhood of T_c
9: Let $evaln = g_e(T_n)$ for T_n
10: **if** $evaln > evalc$ **then**
11: $T_c = T_n$, $evalc = evaln$
12: **if** $evalg < evaln$ **then**
13: $T_g = T_n$, $evalg = evaln$
14: **end if**
15: **else if** $rand(0,1) \leq e^{\frac{evaln-evalc}{TempCurrent}}$ **then**
16: $T_c = T_n$, $evalc = evaln$
17: **end if**
18: **end for**
19: $TempCurrent \times = ReductionRate$
20: **end while**

2.3 Example with Mixture of Gaussian Distributions Although the simulated annealing algorithm terminates at a prespecified temperature instead of freezing naturally, if the temperature is low enough, the algorithm should return a solution in the correct global optimal region. In this section we will demonstrate the power of the search algorithm on a specific setting of the profit function $g^T(x)$, distribution densities $f_1(x)$ and $f_2(x)$, and a set of transformations S.

2.3.1 Profit Function and Gaussian Mixture First define the profit function $g^T(x)$ as:

$$g^T(x) = g - a \left| x^T - x \right|_1, \tag{2.3}$$

where x^T is the transformed spam instance, x is the original one, and g and a are positive constant numbers. To quantify the difference of the spam instance x before and after transformation T, we compute the L_1 norm of $x^T - x$. This is simply summarizing the absolute differences of the individual attributes before and after transformation. The constant value g is the constant profit generated by original instances. We assume the profit would decline linearly according to the extent of the transformation. Here a is the declining 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. Hence it is possible to reach a point that adversary stops modifying the instances, and the Nash equilibrium is established.

Further assume that each class π_i, $i = 1, 2$, has a Gaussian distribution. $f_i(x)$ is the density function for Gaussian distribution $N(\mu_i, \Sigma_i)$. This is based on our observation of a real dataset. Refer to Section 3.1. After log transformation, some variables reveal a Gaussian structure. The Gaussian mixture is used in our simulation study in Section 3.2 as well.

Consider the set of linear transformations S. Define T as a $n \times n$ real matrix, the transformed instance $x^T = Tx$ has every element x_j^T as a linear combination of the original attributes $(x_1, x_2, ..., x_n)'$. Both in the artificial examples in this section and in our simulation study in Section 3.2, 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 "spam" class π_2. Here T' is the transpose of T.

Rewrite Equation 2.1 using the above specifics as follows:

(2.4)

$$T^* = argmax_T \left(\int_{L_1^T} (g - a\,|Tx - x|_1) \times f_2^T(x) dx \right),$$

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

In the rest of this section the profit function $g^T(x)$ is simplified by setting $g = 1$ and $a = 0$, i.e. no cost for transformation. Let $c_{11} = c_{22} = 0$ and $c_{12} = c_{21} = 1$, i.e. no cost for correctly classifying the instances and equal cost for misclassification. Assume $p_2 <= p_1$, meaning the number of regular instances is no less than that of spam instances. Then the adversary gain $g_e(T)$ would reach a maximum value 1 if there exists a T^0 such that two classes are not distinguishable after the transformation T^0, i.e. $f_2^{T^0}(x) = f_1(x)$. Next the structure of the adversary gain function $g_e(T)$ will be examined.

2.3.2 Adversary's Gain If there exists a transformation T^0 that could map the "spam" class π_2 into the regular class π_1, the adversary gain would reach its maximum value 1. However by defining the adversary gain as the expected value of the profit, $g_e(T)$ is discontinuous at T^0.

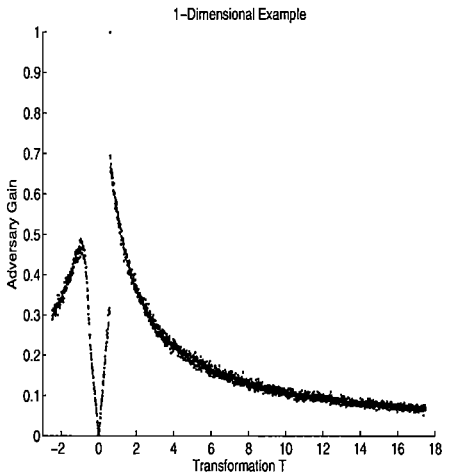

Figure 4: Adversary gain with true transformation equal to 0.6

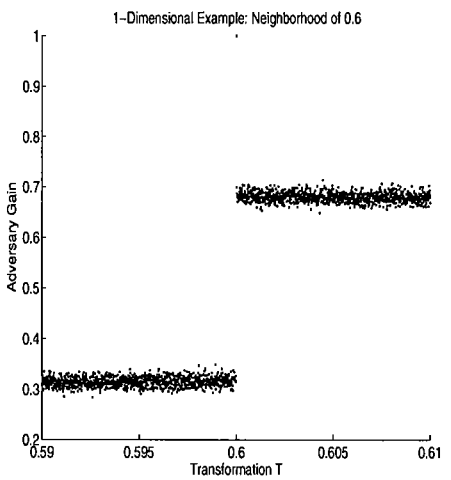

Figure 5: Adversary gain in the neighborhood of true transformation 0.6

This is demonstrated by a $1-$dimensional example. Set $f_1(x)$ be a density for $N(0.2160, 0.3168)$ and $f_2(x)$ be a density for $N(0.36, 0.88)$. Let $g^T(x) = 1$ for all x, $c_{11} = c_{22} = 0$ and $c_{12} = c_{21} = 1$, and $p_1 = p_2 = 0.5$. A transformation $T = 0.6$ would transform $f_2(x)$ into $f_1(x)$. In Figure 4, the value of the adversary gain $g_e(T)$ is plotted against the transformation T. $g_e(T)$ is continuous everywhere else except at the true transformation. It has local minimal and maximal regions.

In the neighborhood around the true transformation, $g_e(T)$ is increasing, but not approaching the maximum value 1. Refer to Figure 5. Fortunately since $g_e(T)$ has

a higher value in the neighborhood of true transformation than in other local regions, we could get close to the true value. In Section 2.3.3, another example also suggests the neighborhood is a good local region. With good properties of the surrounding neighborhood, we could find a solution close to the true transformation even when the adversary gain $g_e(T)$ is discontinuous at the optimal solution T^0.

The discontinuity is caused by the sudden change of the classification region, where "spam" class will be classified as regular. At the true transformation, the "spam" class coincides with the regular class and will all be classified as regular instances. Then we will integrate over the entire space to compute $g_e(T)$. For any other transformation T^a, the classification region L_1^T will experience a continuous change with respect to T^a and will never be the entire space.

Within the set of linear transformations S, there may not exist a transformation T that could map $f_2(x)$ into $f_1(x)$ even if both are Gaussian densities. For example, this happens when the two densities have the same mean vectors and the variance-covariance matrices are different. In this case, the adversary gain $g_e(T)$ is continuous over the entire S, and simulated annealing algorithm will converge to the optimal linear transformation and maximize the adversary gain.

To test the effectiveness of the simulated annealing algorithm we first ran it on a toy example.

2.3.3 An Artificial Example Use a two dimensional example. Again set $g^T(x) = 1$ for all x, $c_{11} = c_{22} = 0$, $c_{12} = c_{21} = 1$, and $p_2 = p_1 = 0.5$. First choose a bivariate normal distribution $N(\mu_2, \sigma_2)$ for $f_2(x)$, a transformation T, and let $f_1(x)$ as density for $N(T\mu_2, T\sigma_2 T')$. Since the cost of transformation is 0, we expect the simulated annealing algorithm to find a solution close to T to achieve near maximum gain when forced to stop. In our experiment, set the parameter values as:

$$\mu_2 = \begin{bmatrix} -0.6461 \\ -0.5501 \end{bmatrix}$$

$$\Sigma_2 = \begin{bmatrix} 0.4978 & 0 \\ 0 & 0.7984 \end{bmatrix}$$

Choose T as:

$$T = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}$$

The cooling schedule is set by fixing the temperature reduction rate to 0.9. Stop the algorithm when reaching minimum temperature 0.001. Set maximum temperature to 1.5 and sample 100 points at every temperature. Our simulated annealing algorithm managed to find the following result which gives a near optimal value for the adversary.

$$T^* = \begin{bmatrix} 0.4300 & 0.0320 \\ 0.0330 & 0.4770 \end{bmatrix}$$

This result suggests that the surrounding neighborhood is an optimal region and if we allow the algorithm to converge, we would reach an even better result. Simulated annealing is proven to converge to the globally optimal solution but in reality has a slow convergence rate. Other stochastic search algorithms are being considered to improve the speed. Please refer to Section 4 for details.

3 Experimental Analysis

3.1 Spam Data Analysis In order to obtain the information related to the variables used in spam filters and the associated distributions and parameter values, a spam dataset documented in the UCI (University of California, Irvine) machine learning repository [1] is examined.

The spam dataset documented in the UCI repository contains 4601 instances. Among those, 1813 belong to spam emails, which is 39.4% of the total. 58 attributes are reported on each instance. One is the class, indicating whether the instance belongs to a regular or a spam email. 48 attributes measure the occurrence frequency of certain words. 6 measure the occurrence frequency of certain characters, and 3 attributes are for the length of sequences of capital letters. The spam and regular emails are analyzed separately to examine the properties for each class. We worked with the 57 numerical attributes.

The values of the 54 attributes focusing on the word or character frequencies contain many 0s, meaning there are many instances that these words or characters did no appear in the emails. The overall distribution of one attribute for spam emails could be formulated as the following:

$$P_s(X < x) = P_s(X = 0) + p_s \times P_s(X < x | X > 0),$$

where $p_s = P_s(X > 0)$, the probability that a word or character does appear in a spam email. The distribution for the regular emails $P_r(X < x)$ could be expressed in a similar fashion. Given an email does include a certain word or character, we discovered that many such conditional distributions ($P_{s/r}(X < x | X > 0)$) could be approximated by lognormal distribution, for both spam and regular emails. The three attributes for the length of sequence of capital letters have minimum values greater or equal to 1 and large maximum values. Some of them could be directly approximated by lognormal distribution.

In the literature, Zipf distribution is used to characterize the distribution of word frequencies in a natural language. It describes a phenomenon that a few words occur frequently while others appear rarely. In our study we focus on the occurrence of a certain word or character in a large number of emails. A word (character) would appear more frequently in some emails (maybe a certain type of advertisement) than others. Hence the distribution given an email does include the word (character) is also a skewed distribution, but not with such a long tail such as the Zipf distribution. Some

could be modeled as lognormal, as shown in Figures 6 and 7.

A correlation study shows that the 57 attributes for spam emails are mostly uncorrelated. Only two pairs of the attributes have correlation coefficients either greater than 0.5 or smaller than -0.5, showing modest to large correlation: attributes 25 and 26; and attributes 50 and 56. On the other hand, the attributes for regular emails exhibit much stronger correlation structure. 30 pairs have correlation coefficients either greater than 0.5 or smaller than -0.5 (a second look shows these are all positive correlations), with the largest equal to 0.9988. We suspect this is due to the fact that regular emails have more coherent content. Therefore words or characters would appear or disappear in groups.

It is difficult to modify the values of p_s and p_r. Spam emails serve a purpose, such as advertising, and inevitably some words would appear more often while others almost never. Note regular emails can not avoid these words completely. Therefore it is possible to transform the conditional distributions with some cost, so given a word (character) occurs in spam emails, the occurrence follows a distribution similar to the one for regular emails. Then it would be difficult to separate the two classes. Further notice if a variable Y follows a lognormal distribution, $log(Y)$ is normally distributed, $\log(Y) \sim N(\mu, \sigma^2)$. To transform a lognormal distribution is essentially to transform the underlying normal distribution. In the simulation study, we would work directly with normal distributions.

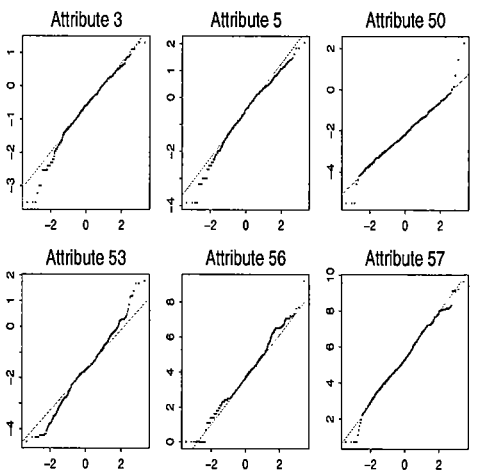

Figure 6: Lognormal probability plot of the 6 selected attributes for spam emails.

We picked 6 attributes, for which lognormal is a good fit for both spam and regular emails and the probability of occurring in each class is not too small. They are attributes 3, 5, 50, 53, 56 and 57. Refer to Fig. 6 and Fig. 7. On the probability plot, the log of the variable values are plotted against normal percentiles. A straight line is desired for a good fit.

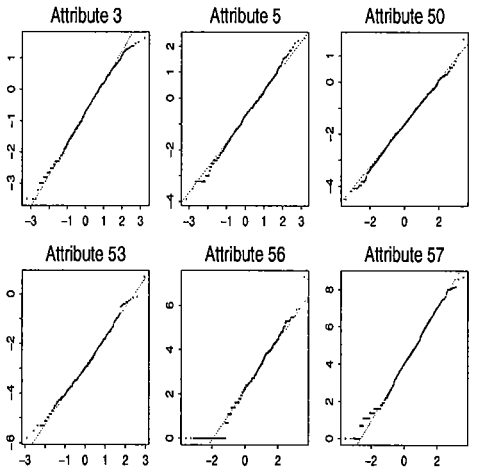

Figure 7: Lognormal probability plot of the 6 selected attributes for regular emails.

The correlation among these attributes for the spam emails are:

$$\begin{bmatrix} 1.00 & 0.03 & -0.02 & 0.05 & 0.11 & 0.06 \\ 0.03 & 1.00 & -0.03 & -0.06 & 0.00 & -0.09 \\ -0.02 & -0.03 & 1.00 & 0.12 & 0.61 & 0.21 \\ 0.05 & -0.06 & 0.12 & 1.00 & 0.12 & 0.15 \\ 0.11 & 0.00 & 0.61 & 0.12 & 1.00 & 0.48 \\ 0.06 & -0.09 & 0.21 & 0.15 & 0.48 & 1.00 \end{bmatrix}$$

And the correlation for the regular emails are:

$$\begin{bmatrix} 1.00 & 0.03 & 0.01 & -0.02 & -0.01 & -0.02 \\ 0.03 & 1.00 & -0.02 & 0.00 & -0.02 & -0.03 \\ 0.01 & -0.02 & 1.00 & 0.02 & 0.02 & 0.05 \\ -0.02 & 0.00 & 0.02 & 1.00 & 0.14 & 0.06 \\ -0.01 & -0.02 & 0.02 & 0.14 & 1.00 & 0.36 \\ -0.02 & -0.03 & 0.05 & 0.06 & 0.36 & 1.00 \end{bmatrix}$$

Since most of the correlation is weak and in both classes only one to two pairs show a modest correlation, later we will use 6 independent variables in our simulation study. Taking the log of non-zero values, we estimated the mean and standard deviation for the lognormal distributions. Tables 1 and 2) contain the estimated parameter values and they are the ones we will use in simulation:

3.2 Experimental Set-up It is interesting to see what the adversary's strategy would become in response to different classification rules and transformation costs. According to our setting a classification rule changes when the cost matrix changes, and the adversary's gain is affected by the

Table 1: Mean and standard deviation of the log of non-zero attribute values for spam emails.

Attribute	μ	σ
3	−0.6460696	0.7055715
5	−0.5501368	0.8935253
50	−2.1506839	0.7973027
53	−1.7247619	0.9477220
56	3.7256119	1.2580866
57	5.3754914	1.2592832

Table 2: Mean and standard deviation of the log of non-zero attribute values for regular emails.

Attribute	μ	σ
3	−0.7563790	0.9595315
5	−0.7323973	1.0411335
50	−1.5978695	0.8483226
53	−2.8987724	1.1485875
56	2.4558962	0.9872055
57	3.9976330	1.4711475

profit function under a transformation T. In this section we search for approximate Nash equilibrium results under various classification cost matrices and profit functions.

In constructing our cost matrices, the correct classification costs are fixed to be 0, i.e., $c_{11} = c_{22} = 0$. We would modify the misclassification cost of classifying a spam instance as non-spam and a non-spam instance as a spam. (Please note that c_{ij} is the cost of deciding $x \in \pi_i$ given that $x \in \pi_j$. In our case, π_2 is the class of spam e-mails and π_1 is the class of non-spam e-mails). Different transformation costs are also considered.

Increasing misclassification cost for spam filter: Usually having more spam emails pass the filter would cost users a little extra time to delete them. However blocking an important email which happens to be put in a wrong format would cause serious consequence. Here we gradually increase the cost of misclassifying a non-spam instance and examine the effect in the experiments.

Equal cost ($c_{21}/c_{12} = 1$) In this setting, the cost of misclassifying a spam e-mail as non-spam is equal to misclassifying a non-spam e-mail as spam. We use the following parameters for the cost matrix: $c_{11} = 0$, $c_{12} = 1, c_{21} = 1$, $c_{22} = 0$.

Low cost ($c_{21}/c_{12} = 2$) For this cost matrix, we assumed that the cost of misclassifying a non-spam e-mail as spam is twice the cost of misclassifying a spam e-mail as non-spam. We used the following parameters for the cost matrix: $c_{11} = 0, c_{12} = 1, c_{21} = 2$, $c_{22} = 0$.

High Cost ($c_{21}/c_{12} = 10$) For this cost matrix, we assumed that the cost of misclassifying a non-spam e-mail as spam costs ten times as much as misclassifying a spam e-mail as non-spam. We used the following parameters for the cost matrix: $c_{11} = 0$, $c_{12} = 1$, $c_{21} = 10$, $c_{22} = 0$.

Increasing transformation cost for adversary: The adversary's gain is the expectation of the profit generated by a certain transformation T. Note that in the profit function (Equation 2.3), there are two parameters: the profit of an instance 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. Please note that as the value of a increases the cost of transforming the spam class also increases. Based on this observation, we created three different gain functions.

No Transformation Cost: In this setting, we assume that using a linear transformation would not introduce any cost to the adversary, set $a = 0$, and use the following profit function: $g^T(x) = 1$.

Low Transformation Cost: In this setting, we assume that using a linear transformation creates a small cost to the adversary, set $a = 0.2$, and use the following profit function: $g^T(x) = 1 - 0.2\,|Tx - x|_1$.

High Transformation Cost: In this setting, we assume that applying a linear transformation imposes a high cost on the adversary, set $a = 0.7$, and use the following profit function: $g^T(x) = 1 - 0.7\,|Tx - x|_1$.

Using the cost matrices and profit functions defined above, we performed 9 experiments corresponding to each and every combination of cost matrix and profit function. In every experiment, we set the parameters of simulated annealing algorithm as the following: the initial temperature is 1.5; temperature reduction rate is 0.94; minimum temperature is 0.001. At every temperature, we search through 200 randomly generated transformations. To evaluate the integration, we used 10000 samples. We restricted our search space to the matrices with entries chosen from (0,1).

3.3 Experiment Results Our experiment results are reported in the Figure 8. Each line in the figure corresponds to a different cost matrix for the spam filter. The gain of the adversary is reported for every transformation cost level.

For each cost matrix of the spam filter, the initial gain of the adversary (i.e., choosing the identity matrix as the transformation) is given in Table 3.

Figure 8 shows that for transformation cost $a > 0$, simulated annealing cannot find a transformation within the search space that improves the gain of the adversary. For $a = 0$, the adversary can increase its gain significantly by using transformation to defeat the filter.

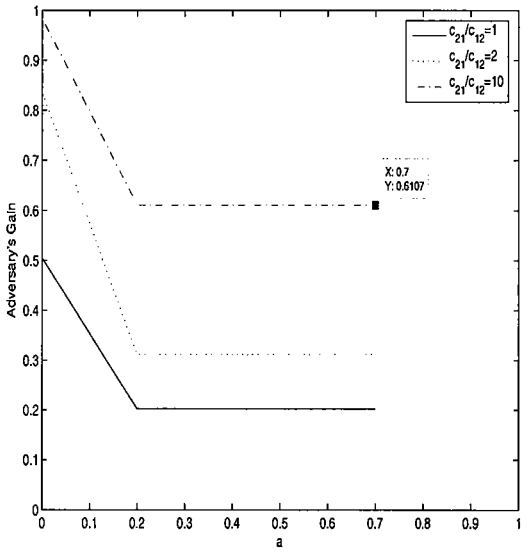

Figure 8: Benefit to the adversary under different cost matrices.

Table 3: Initial Gain of the Adversary

c_{21}/c_{12}	Initial Gain
1	0.2023
2	0.3114
10	0.6107

These initial results indicate that even under moderate transformation cost, an adversary has little incentive to modify its input. This suggests that in adversarial settings classifiers should be built using attributes that are costliest to change for adversaries. Another indication of this initial result is that spam filtering based on the easily modifiable attributes cannot be really effective, as the adversary will find ways to defeat them.

4 Conclusions

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. Achieving such a Nash Equilibrium requires that the adversaries and the classifier face 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 a steady state could be reached, and how best to build a classifier to make it as effective as possible at that steady state.

This paper has evaluated this game theoretic approach in the spam email domain. Costs to the classifier come both from subjecting the reader to (misclassified) spam, and from rejecting legitimate email. The spammer faces a cost from transforming email to defeat the classifier; while the transformed email may get past the filter, it is much less likely to obtain the desired response from the recipient.

While this is early work, the results are interesting. If the cost of rejecting legitimate email is low, then (as expected), the spammer gains by transforming the email. Perhaps surprisingly, however, if the cost of rejecting legitimate email is high then the spammer does as well by *not* transforming email. This is because a certain proportion of spam emails will get through, and the value of those spam emails is as high as getting a lot more of unreadable ones through. This also has implications for the filter designer: By searching for features that are expensive for the spammer to transform, the filter can remove incentives for the spammer to send hard to read, but hard to filter, "garbage".

The proposed approach is not limited to application to email. The same basic technique can be applied to anything from intrusion detection to homeland security. Our formulation of the problem could accommodate a wide range of distributions, classifiers and profit functions. With the application to spam filtering we performed experiments on a combination of Bayesian classifier with cost matrix, Gaussian mixture distribution and linear loss of profit for transformations. For the spam filtering application we would like to obtain information about classification rules used by real spam filters.

Further the ability of the simulated annealing algorithm to find the Nash equilibrium is examined in this paper. Though it is able to find the global optimal solution, in our experiments, the algorithm experienced a slow convergence problem. We are considering other stochastic search methods. Stochastic hill climbing technique is an alternative we have tested. With stochastic hill climbing, a new point is chosen only if it improves the current result. To prevent the algorithm sticking in a local optimal region, the algorithm is repeated with a few different random starting points. Our initial results with the example given in Section 2.3.3 indicates that stochastic hill-climbing technique could quickly find a good local optimal solution, returning a better solution given limited running time. For the example in Section 2.3.3, our stochastic hill-climbing algorithm found the following transformation T_g. Though clearly it is far from the true transformation, the gain of the results produced by both algorithms are very close, stochastic hill-climbing with slightly higher gain. In higher dimensional space, i.e., considering more attributes simultaneously, stochastic hill-climbing technique is an alternative, since it will be extremely time consuming for simulated annealing to converge. We would examine the structure of adversary gain $g_e(T)$ and explore other methods which could lead to a good solution with improved convergence rate targeted on $g_e(T)$.

$$T_g = \begin{bmatrix} 0.3502 & 0.3736 \\ -0.3577 & 0.3519 \end{bmatrix}$$

With minor adjustments, our formulation could fit many real life scenarios. We plan to extend our work in the following directions:

Other Applications: We would like to test our formulation on other scenarios, such as homeland security and intrusion detection. It would be interesting to investigate whether in other applications a Nash equilibrium is achievable.

Other Games: The two stage game with complete information discussed in this paper might be too limited. First we will consider what will happen when adversary could only have incomplete information. Next we would like to extend our formulation to games that involve many players and possibly with incomplete information. (For example, what if a spammer was unable to determine if email was classified as spam?)

Designing Effective Classifiers: Our current results indicate that parameter selection is very important in an adversarial setting. We would like to design better classifiers that do not give the adversary a cost effective transformation ability.

In summary, modeling adversarial classification problems using game theory is a valuable tool. First, it gives an idea of how effective we can expect a classifier to be in the long term. Second, it gives insights into the problems that enable us to build better classifiers/filters both short and long term.

References

[1] C.L. Blake and C.J. Merz. UCI repository of machine learning databases, 1998.

[2] Fact sheet: CAPPS II at a glance, February 12 2004. URL `http://www.dhs.gov/dhspublic/display?theme=20&content=3161`.

[3] Nilesh Dalvi, Pedro Domingos, Mausam, Sumit Sanghai, and Deepak 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.

[4] Richard Duda, Peter E. Hart, and David Stork. *Pattern Classification.* John Wiley & Sons, 2001.

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

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

[7] Robert Gibbons. *Game Theory for Applied Economists.* Princeton University Press, 1992.

[8] 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.

[9] Dimitrios K. Vassilakis 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.

[10] Daniel Lowd and Christopher 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.

[11] Matthew V. Mahoney and Philip 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.