Advanced Computing: An International Journal (ACIJ), Vol.2, No.2, March 2011

SPAM DETECTION BY STACKELBERG GAME

Alireza Naeimi Sadigh[1] , Sattar Hashemi[2] and Ali Hamzeh[3]

[1]Department of Computer Engineering, Shiraz University, Shiraz, Iran
anaeimi@gmail.com
[2]Department of Computer Engineering, Shiraz University, Shiraz, Iran
s_hashemi@shirazu.ac.ir
[3]Department of Computer Engineering, Shiraz University, Shiraz, Iran
ali@cse.shirazu.ac.ir

ABSTRACT

Many data mining applications, ranging from Spam filtering to intrusion detection, are forced with active adversaries. Adversary deliberately manipulate data in order to reduce the classifier's accuracy, in all these applications, initially successful classifiers will degrade easily.
In this paper we model the interaction between the adversary and the classifier as a two person sequential non cooperative Stackelberg game and analyze the payoff when there is a leader and a follower. We then proceed to model the interaction as an optimization problem and solve it with evolutionary strategy.
Our experimental results are promising; since they show that our approach improves accuracy spam detection on several real world data sets.

KEYWORDS

Spam detection, Adversarial learning, Stackelberg game, Evolutionary strategy

1. INTRODUCTION

As the Internet started to gain popularity in the early 1990s, it was quickly recognized as an excellent advertising tool because, at practically no cost, a person can use the Internet to send numerous email messages to thousands of people. When this message contains an unsolicited advertisement, the email boxes of many interest users get cluttered with all this so-called unsolicited bulk email also known as "Spam" or "junk mail". Regardless of whether Spam actually benefits the advertiser, to those who receive these Spams daily, the problem is serious.
So, being incredibly cheap to send, Spam causes a lot of trouble for the Internet community: large amount of Spam-traffic between servers cause delays in delivery of legitimate email, people with Internet access have to spend bandwidth downloading junk mail, sorting out the unwanted messages takes time and introduces a risk of deleting normal mail by mistake and other similar serious issues.
So, as a counter course, many methods to help users to automatically distinguish Spams have been proposed yet [2]. Some of those are simple preliminary approaches such as blocking Spammers IP-address [1], plain personal involvement and email filtering software [1]. But unfortunately, no perfect way for eliminating Spam proposed yet and finding a solution is a serious area of research nowadays [2].
However, automatic email filtering, which means the processing of email to organize it according to specified criteria, seems to be the most effective method for countering Spam at the moment [3].
Automatic email filtering divides into two general categories including: Static filtering and Adaptive filtering [1]. In the former approach, filter Spam based on incoming email attributes, such as: sender name, subject title, email content and etc. A set of rules is created according to

DOI : 10.5121/acij.2011.2203

the users choices concerning selecting messages as Spam or Ham (means not an Spam). A set of such rules should be created either by the user or by some other authority (e.g. the software company that provides a particular rule-based Spam filtering tool). The major drawback of this method is that this set of rules must be constantly updated, and maintaining it is not convenient for most users [2].

Unfortunately, email filtering as classifiers become more widely deployed, the incentive for defeating them increases. So nowadays, many classification tasks, such as Spam filtering, intrusion detection are complicated by an "adversary" who wishes to avoid detection. In Spam filtering for instance, they often disguise their messages by adding unrelated words, sentences, or even paragraphs more indicative of legitimate email than Spam. The adaptive filtering try to tackle this issue even if the adversary has perfect knowledge of the classifier [4]. Although in the adversary's side, gathering this knowledge is somehow impossible in practice and adversaries must learn about the competing classifier using some combination of prior knowledge, observation and experimentation.

But, as a solution for this problem, in [5], it was proposed that the adversary has the ability to issue some membership queries to the classifier and ask for the label of some unlabeled instances which is an acceptable assumption (suppose the adversary uses the same version of the filter for himself).

As a new solution, in this paper, we are going to propose a new algorithm which tries to model the adversarial classification paradigm as a sequential Stackelberg game [6] in which the adversary makes the first move to which the classifier responds.

We also use a heuristic algorithm to compute the Stackelberg equilibria in infinite case when the players do not know the payoff function of each other which is more acceptable in the real world.

The rest of this paper is organized as follows: Section 2 we present a general game and discuss its Stackelberg solution. In Section 3 we formulate Stackelberg games with infinite strategy space for spam detection. Then, in Section 4, we explain the evolutionary strategy we design to search for the needed equilibrium. After that, in Section 5, we describe the experimental setup and presents and discuss the experimental results. Finally, Section 6 presents conclusions and future works.

2. STACKELBERG GAME

We consider a two-person repeated sequential game with *L* as the natural leader (first player to move at each period) and *F* as the natural follower. In our case, the spammer is the leader and the classifier is the follower. The spammer always makes the first move.

Let *U* and *V* be the action spaces of the leader and the follower respectively, with their generic elements denoted by *u* and *v*. Let $J^i : U * V \to R$ be the payoff function for player *i*, where *i* = *L; F*.

Assume that *U* and *V* are some compact, nonempty and convex spaces. Then, if we also assume that $J^L(u * v)$ and $J^F(u * v)$ are functions strictly convex and differentiable in *u* and *v*, respectively, then we know from [7] that there exist some reaction functions T^L and T^F that are continuously differentiable mappings.

Hence, the best reaction function $T^L : V \to U$ of the leader is defined by Equation 1:

$$T^L = \arg\max_{v \in V} J^L(u, v) \qquad (1)$$

And likewise the best reaction function of the follower, $T^F : U \to V$ is given by Equation 2:

$$T^F = \arg\max_{u \in U} J^F(u, v) \qquad (2)$$

2.1. The Stackelberg Equilibrium Solution

In a game, equilibrium [7] is a strategy profile from which none of the players has any incentive to deviate. In particular, no player can achieve strictly greater payoffs by choosing any strategy other than the one prescribed by the profile, given that all other players choose their prescribed strategies.

A Stackelberg equilibrium is a situation where the leader of a group knows that it is the leader. It makes decisions, and its followers estimate the best response to apply, according to this decision. The leader can then estimate what reactions will the other agents have, and makes the decision which will bring him the best reward, relative to those reactions. If the objective of the leader is to maximize the group's reward, it will make the decision which will bring the best reward to this group.

A solution for the Stackelberg equilibrium is generated by the following well-known procedure [7]. Knowing the reaction function of the follower T^F , the leader maximizes his own payoff function. This involves an optimization problem as Equation 3:

$$u^* = \arg\max_{u \in U} J^L(u, T^F(u)) \qquad (3)$$

Given this action, the follower reacts by an optimal action found by Equation 4.

$$v^* = T^F(u^*) \qquad (4)$$

Regarding these definitions, a chain of tuple $< u^*, v^* >$ where $u^* \in U$ and $v^* \in V$, with u^* given by Equation (3) and $v^* = T^F(u^*)$, is a solution to achieve the Stackelberg equilibrium.

2.2. Stackelberg Equilibrium as a Bilevel Programming Problem

There is a mathematical method for solution the Stackelberg equilibrium applied to the two-user game, this method is called bi-level programming problem [7].

The general formulation of a bilevel programming problem (BPP) is

$$\max_x \quad F(x, y)$$
$$s.t. \quad G(x, y) \leq 0$$

$$\max_y \quad f(x, y)$$
$$s.t. \quad g(x, y) \leq 0$$

The variables of this problem are divided into two levels, namely the *upper-level* variables *x* and the *lower-level* variables *y*. Similarly, the functions *F* and *f* are the *upper-level* and *lower-level* objective functions respectively, while the vector-valued functions *G* and *g* are called the upper-level and lower-level constraints respectively. Upper-level constraints involve variables from both levels (in contrast with the constraints specified by the set X) and play a very specific role. Indeed, they must be enforced indirectly, as they do not bind the lower-level decision-maker .

Now, In the particular framework of Stackelberg games, we can express equation 3 as a BPP [7], while the leader maximizes its profit, the follower maximize its own profit by choosing among a set of competitors. The follower problem is hence a constraint of the leader problem:

$$\max_u J^L(u,v) \qquad (5)$$
$$s.t.\ \ g(u,v) \le 0$$

$$v \in \arg\max\{J^F(u,v) \mid h(u,v) \le 0\} \qquad (6)$$

Here *g* and *h* are general constraints which capture the space of the actions *U* and *V* above.

The BPP problem is known to be NP-Hard even when all the functions in the problem are linear [7]. This is because of the presence of the nonlinear argmax constraint. This motivates the use of heuristic approaches to calculate the equilibrium.

3. GAME MODEL FOR SPAM DETECTION

We model the game between the spammer (adversary) and linear classifier as a two-class classification problem. For simplicity, without lose of generality, we assume the data are from one dimensional feature space and varying on normal distribution.

Suppose the distribution of the spam is $P \sim N(\mu_1, \sigma_1)$ and of the legitimate email is $Q \sim N(\mu_2, \sigma_2)$, where $\mu_1 > \mu_2$ (see figure1). Adversary plays by moving $\mu_1 - u$ (towards μ_2) as shown in figure 2, while the classifier reacts by moving boundary from $\frac{\mu_1 + \mu_2}{2}$ to *w* (also towards μ_2) as shown figure 3.

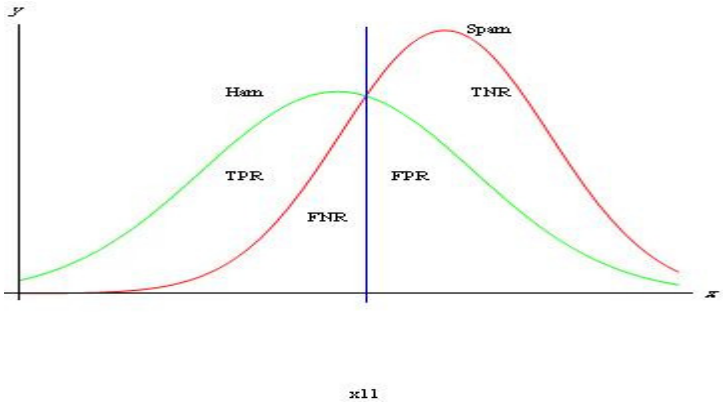

Figure 1. Initial state

To estimate the influence of transformation *u* on the original intrusion data, we use the Kullback-Leibler divergence [8] (*KLD*) to measure the effects of transformation from $N_1(\mu_1, \Sigma_1)$ to $N_2(\mu_2, \Sigma_2)$:

$$D_{KL}(N_1 \setminus N_2) = \frac{1}{2}(\log_e(\frac{\det\Sigma_2}{\det\Sigma_1})) + tr(\Sigma_2^{-1}\Sigma_1) + (\mu_2 - \mu_1)^T \Sigma_2^{-1}(\mu_2 - \mu_1) \qquad (7)$$

where *det* and *tr* stands for the determinant and trace of matrices, *T* in superscript means transpose.

From the probability density function a Normal distribution $N(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$, we can calculate cumulative a Normal distribution as follow:

$$F(t,\mu.\sigma) = \int_{-\infty}^{z} N(t,\mu,\sigma)dx$$

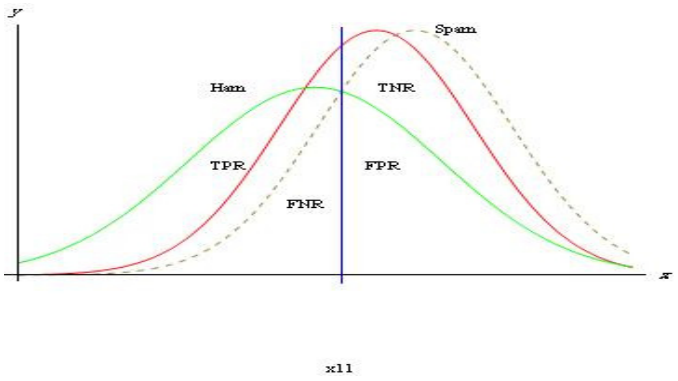

Figure 2. The adversary's movement

We define the payoff for the adversary as the increase in the false negative rate (FNR) minus KLD for moving the distribution:

$$\begin{aligned} J^{L}(u,w) &= FNR - \alpha KLD(\mu_1,\sigma_1,\mu_1-u,\sigma_1) \\ &= F(w,\mu_1-u,\sigma_1) - \alpha KLD(\mu_1,\sigma_1,\mu_1-u,\sigma_1) \end{aligned} \quad (8)$$

The parameter α in Equation 8 determines the strength of the KLD penalty.

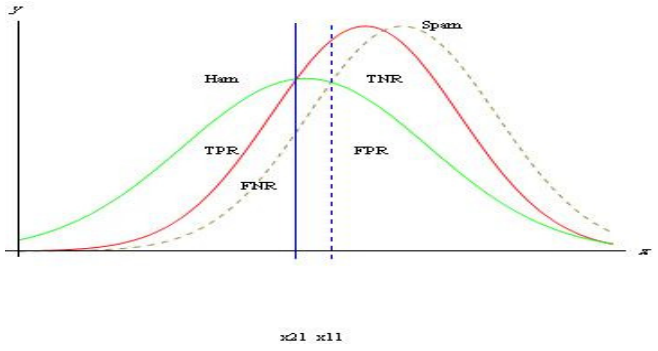

Figure 3. The classifier's movement

The payoff of the classifier is given by increasing both the true positive and the true negative rate (TPR and TNR):

$$\begin{aligned} &J^{F}(u,w) = TPR+TNR \\ &= F(w,\mu_2,\sigma_2) - F(w,\mu-u_1,\sigma_1) + (1-F(w,\mu_1-u,\sigma_{1_1})) - (1-F(w,\mu_2,\sigma_2)) \\ &= 2F(w,\mu_2,\sigma_2) - 2F(w,\mu_1-u,\sigma_1) \end{aligned} \quad (9)$$

For the case of datasets with multiple attributes, we assume all attributes are conditionally independent, final result payoff functions can be defined as follows:

$$J^L(U,W) = \frac{1}{q}\sum_{i=1}^{q} J^L(u_i, w_i) \qquad (10)$$

$$J^F(U,W) = \frac{1}{q}\sum_{i=1}^{q} J^F(u_i, w_i) \qquad (11)$$

where q is the number of attributes.

In next section, we explain how do we use evolutionary strategy to solve this optimization problem.

4. EVOLUTIONARY STRATEGY

We use evolutionary algorithms to solve the Stackelberg equilibrium game.
Evolutionary algorithms suggested by Rechenberg [9] in 1975 and initially applied for parameter optimization. The main difference between ES and GA are in the presentation of population and the types of evolution operators. In ES, instead of binary strings, we use real values to present parameters of optimization. Also ES
just use mutation. ES are easier to implement and might be faster than GA. The basic ES algorithm is shown below [9].

1. Randomly generate a parent population of solutions.
2. Evaluate all parents to determine their fitness.
3. Apply reproduction operators to create λ offspring.
4. Evaluate and keep the fittest individuals.
5. Go to step 3 unless an acceptable solution has been found or a fixed number of generations has been produced and evaluated.

Every point in the search space is an individual. The ES uses a population of μ individuals to conduct the search for possibly better solutions [9]. During each generation, each individual is mutated to produce offspring. This means the ES is simultaneously investigating several regions of the search space, which greatly decreases the amount of time required to locate good solutions The initial population of individuals is randomly generated but, ideally, should be uniformly distributed throughout the search space so that all regions may be explored. Each individual in each generation is evaluated to determine its fitness.
Individuals with high fitness represent approximations which produce low error estimates. The ES terminates after a fixed number of generations have been produced
and evaluated or earlier if the acceptance criterion is reached [9].

4.1. Initial Population

We randomly initiate k transformations $u_i, i \in$ (1; 2; …; k) from uniform distribution. The constraint on u is that $u \in [\mu_2, \mu_1]$, where $\mu_2 < \mu_1$.

4.2. Mutation

Mutation is implemented through adding some random noise drawn from Gaussian distribution, mutation parameters are changed during a run of the algorithm which is defined as follows:

$$(\sigma_i' = \sigma_i . \exp(\tau . N(0,1)) \qquad for\ \ i = 1,2,...,k \qquad (12)$$

$$x_i' = x_i + \sigma_i' . N_i(0,1) \qquad for\ i = 1,2,...,k \qquad (13)$$

$$\tau = 1 / k^{1/2} \qquad (14)$$

where k is the number of population size.

4.3. Selection

After creating the offspring through recombination, we need to select parents into the next generation. We select best $\frac{k}{2}+1$ offspring (the individual with the highest adversarial gain is selected as the best transformation of this generation u_i) and select randomly $\frac{k}{2}-1$ offspring survive.

4.3. Termination Condition

The algorithm terminates when the $10k$ number of iterations is reached. To define the number of iterations, we follow the practice of previously researchers [6].
The algorithm returns the best transformation of the last generation as u^* and w^* .

5. RESULT

In this section, we use synthetic data for all attributes to demonstrate the process of searching for an equilibrium by evolutionary strategy in training step. Also at test step we show results for accuracy filter Spam detection with genetic algorithm (GA) and evolutionary strategy (ES) .

5.1. Experimental Setup

The data set consists of spam emails obtained from [10]. It is collected from an anonymous individual's mailbox of six months' time. The datasets have 20 attributes, for simplicity, we assume all attributes are independent and follow normal distributions.
10-fold cross-validation was used in all experiments: datasets were partitioned randomly into ten parts, and the experiment was repeated ten times, each time reserving a different part for testing, and using the remaining nine parts for training, also ES runs for 100 iterations. The figures below show performance of each method in each experiment.

5.2. Experiment for Training

For example, we proposed first attribute for January dataset. The distributions of Spam and legitimate instances are $P \to N(0.15,0.31)$ and $Q \to N(0.07,0.29)$ respectively. We used linear classifier and runs 100 times by GA [6] and ES. $(\alpha = 0.01)$
Figure 4 show adversary gain for this example. As we displayed, ES find the better than GA adversary gain (for algorithms comparison, an algorithm is better if adversary has the less payoff and it could be less fooling the classifier).

5.2. Experiment for Testing

At this step, the filter final position in the equilibrium point which was determined with train data for each attribute helps to compute the filter accuracy according to below processes:

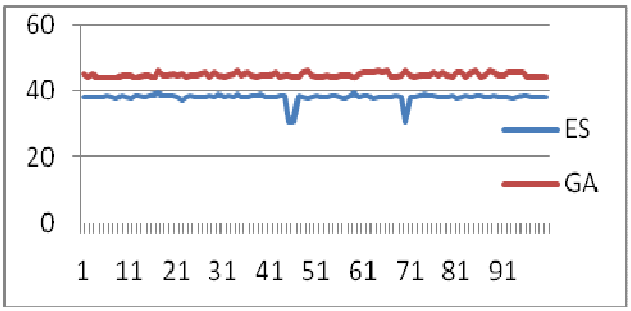

Figure 4. Adversary gain for ES vs. GA

- From each test data choose one sample and specify every attribute by using the filter is distinguish Spam or Ham.
- Class label this sample obtains by voting method.

Table 1 show the filter accuracy for Spam and Ham detection with GA [6] and ES algorithms.

Table 1. Filter Accuracy

Data Set	GA		ES	
	Spam	Ham	Spam	Ham
JAN	92.6%	94.6%	94%	97.5%
FEB	66%	91.3%	67.3%	93%
MAR	64.5%	87.3%	64.9%	88%
APR	50.8%	95.6%	51.7%	96.3%
MAY	66.6%	94.8%	68.2%	95.6%
JUN	67.18%	95.7%	69.3%	96.2%

As in Table 1 is determined to reach the equilibrium point by ES algorithm in compared to the genetic algorithms is more accurate.

6. CONCLUSIONS

The game between adversary and the classifier can be describe in a game theoretical framework. These two players will reach Stackelberg equilibrium when they are playing their best strategy at the same time. Our experiments illustrate that evolutionary strategy have the ability of solving this equilibrium, since they produce effective optimizations on adversarial gains.
Given those insights, in future our research will focus on designing feature selection algorithms that can extract inherently expensive to manipulate features, and also designing classifiers with nonlinear functions and movement cut- point can obtain the more accuracy filtering.

REFERENCES

[1] G, Schryen, (2007) *Anti-spam measures analysis and designe*, Springer Press.

[2] S, Heron**,** (2009) **"**Technologies for spam detection", *Network Security*, Vol. 2009, No. 1, pp. 11-15.

[3] T, Guzella & W, Caminhas (2009) "A review of machine learning approaches to Spam filtering", *Journal of Expert Systems with Applications Elsevier* , NO. 36, pp. 10206-10222

[4] Nilesh, D. & Pedro, D. (2004) "Adversarial classification.", *In KDD*, New York, NY, USA, pp. 99-108

[5] D, Lowd & C, Meek (2005) "Adversarial learning.", *In KDD*, New York, NY, USA, pp. 641–647.

[6] Liu, W. & Chawla S. (2009) "A Game Theoretical Model for Adversarial Learning*"*, *IEEE on Data Mining,* pp. 25-30.

[7] K. Binmore (2007) *Playing for real: a text on game theory*. Oxford University Press, USA.

[8] S., Kullback & RA, Leibler (1951) "On information and sufficiency*"*. *The Annals of Mathematical Statistics,* pp. 79–86.

[9] H., Beyer & H., Schwefel (2002) "Evolution strategies", *Natural Computing*, Vol. 3, pp. 150-200.

[10] http://www.comp.dit.ie/sjdelany/datasets/staticdatasets.zip.

Authors

Alireza Naeimi Sadigh: He received his B.Sc. degree in Engineering from Ferdowsi university of Mashhad, Iran in 2006. He is currently M.Sc. student in artificial intelligence in Shiraz University. His research interests include adversarial learning, game theory and optimization algorithms.

Sattar Hashemi received the PhD degree in Computer Engineering from the Iran University of Science and Technology, in conjunction with Monash University, Australia, in 2008. He is currently a lecturer in the Electrical and Computer Engineering School, Shiraz University, Shiraz, Iran. His research interests include data stream mining, database intrusion detection, dimension reduction and adversarial learning.

Ali Hamzeh received his Ph.D. in artificial intelligence from Iran University of Science and Technology (IUST) in 2007. Since then, he has been working as assistant professor in CSE and IT Department of Shiraz University. There, he is one of the founders of local CERT center which serves as the security and protection service provider in its local area. As one of his research interests, he recently focuses on cryptography and steganography area and works as a team leader in CERT center to develop and break steganography method, especially in image spatial domain. Also, he works as one of team leaders of Soft Computing group of shiraz university working on bio-inspired optimization algorithms. He is co-author of several articles in security and optimization.