

# Adversarial Learning: A Game Theoretic Approach

## Thesis Proposal

Orhan Sönmez

Max Planck Institute for Computer Science

July 2007

### Abstract

This paper is a master thesis proposal about a game theoretic approach for the adversarial learning problem. The main aspects of the approach would be opponent modeling by the classifier, repeated games between the classifier and the adversary, feature addition by the adversary in addition to feature removal, and the optimality in the average case.

## 1 Introduction

Most of the machine learning algorithms assume that the classifier receives unmanipulated data directly from the data source. But in some domains e.g. spam filtering, this assumption wouldn't hold as there exists an adversary which actively manipulates the data in order to prevent classifier to generate an appropriate model for the data.

Adversarial Learning Problem is a general machine learning problem where the classifier generates a model for the data without disregarding the existing of an adversary and even adapts itself to the evolving actions of the adversary.

## 2 State-of-Art

In [2], after assuming the complete information for both the classifier and the adversary, the optimal strategies for them are introduced by using a cost-sensitive learning method. Even though, the problem is modeled as a game between the classifier and the adversary, the strategies are calculated only for the very first step of the game. Also, The performance of the algorithm could be improved against the subopti-

mal adversaries, by using an opponent modeling method.

In contrast to [2], the algorithm in [1] doesn't assume that the classifier information is complete for the adversary. Hence, a new problem, namely adversarial classifier reverse engineering problem (ACRE) is introduced. In our problem, the information would be assumed as complete both for the classifier and the adversary.

[3] introduces a feature deletion method which calculates the  $K$  features that would be most probably manipulated by adversary and trains the classifier with ignoring those  $K$  features. However, it is a pessimistic method that works optimal only in the worst case scenario. When the adversary is not optimal, the method lacks of the usage of whole data as it generally ignores the  $K$  features that have the most informative power. Hence, an opponent modeling method would increase the performance of the algorithm.

## 3 Problem Proposal

### 3.1 Aim

After assuming complete information, creating a framework and an appropriate algorithm to generate a classifier  $C$  that works optimally in the average case with the help of an opponent modeling technique to model the activities of the adversary  $A$ . Hence, in order to model the opponent repeated games should be held between the classifier  $C$  and the adversary  $A$ .

### 3.2 Definition

Let a data instance be  $X = (x_1, \dots, x_n)$  where  $x_j$  is the  $j^{th}$  feature. The instance  $X$  and its corresponding classification  $c^*(X)$  is generated

by a data distribution  $D$ . Each instance  $X$  has a cost matrix  $M$  that consists of a true positive benefit  $TP_X$ , a true negative benefit  $TN_X$ , a false negative cost  $FN_X$  and a false positive cost  $FX$  with respect to the classification of  $C$ . Also, each feature  $x_j$  has a changing cost of  $a_j$  for  $A$ . Without losing generality, we assume that all features of  $X$  are boolean and  $c^*(X)$  is either positive(+) or negative(-).

After  $C$  is initially trained by the data obtained from  $D$ , we define a zero-sum game between  $A$  and  $C$ . At each step  $i$ , a further data instance  $X_i$  is generated by  $D$ . Then,  $A$  chooses a boolean toggle vector  $T_i = (t_{i,1}, \dots, t_{i,n})$  in order to change the instance  $X_i$  to prevent  $C$  to classify it correctly. It has a total cost of

$$a_T = \sum a_j t_{i,j}$$

for  $A$ . Then, the manipulated sample vector  $S_i = (s_{i,1}, \dots, s_{i,n})$  is calculated as

$$s_{i,j} = x_{i,j} \oplus t_{i,j}$$

and given to  $C$ . With only knowing  $S_i$ ,  $C$  makes a classification  $c(X_i, T_i)$  for  $X_i$ . According to the values of  $c(X_i, T_i)$  and  $c^*(X_i)$ , the game step ends with the corresponding value  $m_i$  in  $M$  for  $C$  and  $-m_i - a_T$  for  $A$ .

Finally, at the end of each game step,  $X_i$  is revealed to  $C$ . Hence,  $C$  now can calculate  $T_i$  in order to adapt itself to the moves of  $A$ . And then the game between  $C$  and  $A$  is repeated for the next instance  $X_{i+1}$ .

### 3.3 Problems

#### 3.3.1 Optimal Strategies

With assuming complete information, optimal strategies for both the classifier  $C$  and the adversary  $A$  would be calculated for a fixed cost matrix  $M$ .

$C$  would try to minimize the classification error with respect to the cost matrix  $M$ . On the other hand,  $A$  would try to maximize its benefit while minimizing the total cost of change of the instances. With assuming the game as zero-sum, maximizing the classification error will maximize the benefits for  $A$ .

#### 3.3.2 Opponent Modeling

As more games played,  $C$  would gather more information about the strategy of  $A$  by calculating  $T_i$  values for each step. Instead of assuming

an optimal adversary, this information would be used to adapt  $C$  to the actions of  $A$ .

#### 3.3.3 Cost Matrix Learning

Without assuming a fix cost matrix  $M$ ,  $C$  would approximate  $M$ , according to the results of the previous games. Then also, each instance  $X_i$  could have a different cost matrix  $M_i$  which depends on its own feature values  $x_{i,j}$ .

#### 3.3.4 Toggle Vector Learning

Revealing  $T_i$  to classifier  $C$  after each step is indeed not a realistic assumption for the real world applications. Hence,  $C$  would model the actions of  $A$  and generate an approximation for  $T_i$  at each step.

## References

- [1] Lowd D. Meek C. Adversarial learning. *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2005.
- [2] Dalvi et. al. Adversarial classification. *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2004.
- [3] Globerson A. Rowies S. Nightmare at test time: Robust learning by feature deletion. *Proceedings of the 23rd International Conference on Machine Learning*, 2006.