Adversarial Learning

"If you know the enemy and know yourself, you need not fear the result of a hundred battles.”
-- Sun Tzu, The Art of War

Abstract

Many classification tasks, such as spam filtering, intrusion detection, and terrorism detection, are complicated by an adversary who wishes to avoid detection. Previous work on adversarial classification has made unrealistic assumptions that the attacker has perfect knowledge of the classifier [Dalvi et al., 2004]. In this paper, we introduce the adversarial classifier reverse engineering (ACRE) learning problem, the task of learning sufficient information about a classifier to construct adversarial attacks. We present efficient algorithms for reverse engineering linear classifiers with either continuous or Boolean features and demonstrate their effectiveness using real data from the domain of spam filtering.

ACRE Learning

A classifier, \( c(x) \), maps \( n \)-dimensional feature vectors (instances) to classes + and -. An adversarial cost function, \( a(x) \), represents the adversary’s preference for some instances over others. We visualize this as constant-cost contours in the instance space:

In an adversarial classifier reverse engineering (ACRE) learning problem, the adversary tries to minimize \( a(x) \) subject to the constraint \( c(x) = + \). The adversary is given one positive instance, one negative instance, and may issue membership queries to the classifier. In each query, the adversary learns the class of a single chosen instance.

A set of classifiers and cost functions is \( k \)-ACRE learnable if the adversary can always find a negative instance within a factor \( k \) of the minimum cost using a polynomial number of queries.

Spammer goal: minimally modify a spam message to achieve a spam that gets past a spam filter.

Corresponding ACRE problem: spam filter \( \rightarrow \) linear classifier with Boolean features “minimally modify” \( \rightarrow \) uniform linear cost function

Filter configuration:
- Naïve Bayes (NB) and maxent (ME) filters
- 500,000 Hotmail messages for training
- > 250,000 features

Adversary feature sets:
- 23,000 English words (Dict)
- 1,000 most frequent English words (Freq)
- 1,000 random English words (Rand)

Results (see table):
- Reduced feature set almost as good
- Cost ratio is excellent
- Number of queries is reasonable (parallelize)

Conclusion: ACRE algorithms are practical!

Empirical Evaluation: Spam

Table 1: Empirical Results in Spam Domain

<table>
<thead>
<tr>
<th></th>
<th>med. cost</th>
<th>max cost</th>
<th>med. ratio</th>
<th>max ratio</th>
<th>med. queries</th>
<th>max queries</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dict NB</td>
<td>23</td>
<td>723</td>
<td>1.136</td>
<td>1.5</td>
<td>261k</td>
<td>6,472k</td>
</tr>
<tr>
<td>Dict ME</td>
<td>10</td>
<td>49</td>
<td>1.167</td>
<td>1.5</td>
<td>119k</td>
<td>640k</td>
</tr>
<tr>
<td>Freq NB</td>
<td>34</td>
<td>761</td>
<td>1.105</td>
<td>1.5</td>
<td>25k</td>
<td>656k</td>
</tr>
<tr>
<td>Freq ME</td>
<td>12</td>
<td>72</td>
<td>1.108</td>
<td>1.5</td>
<td>10k</td>
<td>95k</td>
</tr>
<tr>
<td>Rand NB</td>
<td>31</td>
<td>769</td>
<td>1.120</td>
<td>1.5</td>
<td>23k</td>
<td>755k</td>
</tr>
<tr>
<td>Rand ME</td>
<td>12</td>
<td>64</td>
<td>1.158</td>
<td>1.5</td>
<td>9k</td>
<td>78k</td>
</tr>
</tbody>
</table>

For each scenario, we list the median and maximum number of changes required, ratio (relative to optimal), and number of queries required.