We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS BOOK CITATION INDEX INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

27

Countering Good Word Attacks on Statistical Spam Filters with Instance Differentiation and Multiple Instance Learning

Yan Zhou, Zach Jorgensen and Meador Inge
University of South Alabama
USA

1. Introduction

Although the impact of e-mail spam has been greatly reduced by existing spam filters, spam remains a great challenge to Internet Service Providers and the average user. Given that there are millions of e-mail users, profit-driven spammers have great incentives to spam. With as little as 0.001% response rate, a spammer could potentially profit $25,000 on a $50 product (Carpinter and Hunt, 2006). Over the years, spammers have grown in sophistication with cutting-edge technologies and have become more evasive. The best evidence of their growing effectiveness is a recent estimate of over US $10 billion worldwide spam-related cost in terms of wasted resources and lost productivity (Jennings, 2005). The spam problem has been generally accepted as a long lasting problem, unlikely to end anytime soon.

Spam filtering has become an extensively studied subject due to the severity of the spam problem. However, relatively little research has been done on countering adversarial attacks on existing spam filtering systems. In recent years, adversarial attacks have become an increasing challenge to the anti-spam community. Common adversarial attacks on spam filters are exploratory attacks. These attacks attempt to discover ways to disguise spam messages so that spam filters are unable to correctly detect them. However, the adversary who initiates this type of attacks has no intention to influence or alter the training process of spam filters that are the targets of attack. The good word attack (Lowd and Meek, 2005b) is one of the most popular exploratory attacks employed by spammers. This technique involves appending to spam messages sets of "good" words that are common to legitimate e-mails (ham) but rare in spam. Spam messages injected with good words are more likely to bypass spam filters. So far, relatively little research has been done on how spam filters can be trained to account for such attacks. This chapter presents an effective defence strategy, using instance differentiation and multiple instance (MI) learning, against spam disguised with good words.

Multiple instance (MI) learning (Dietterich et al., 1997) differs from single instance supervised learning in that an example is represented by a set, or bag, of instances rather than as just a single instance. The bag is assigned a class label (either positive or negative) based on the instances it contains; however, the instances within the bag are not necessarily labelled. Classic MI learning assumes that a bag is positive if at least one instance in the bag is positive and negative if all instances are negative. Therefore, the goal of multiple instance learning is to learn a classification function that accurately maps a given bag to a class.

Our spam filtering strategy adopts the classical MI assumption. Each e-mail is transformed into a bag of instances. An e-mail is classified as spam if at least one instance in the corresponding bag is spam, and as legitimate if all the instances in it are legitimate. The other integral component of our defence strategy is the model for instance differentiation, i.e., dividing an e-mail into a bag of multiple instances. The challenge of defining such a model includes finding a clean separation between the spam content and the good words that are common to legitimate e-mail, and ensuring that the model itself does not introduce a loophole for adversarial attacks. With an effective instance differentiation model, we can split each e-mail into multiple instances so that even when spam is injected with good words, a multiple instance learner is still able to recognize the spam part of the message.
In this chapter, an overview of recent research in the area of adversarial learning and multiple instance learning is provided. The models for performing instance differentiation and multiple instance learning are presented and investigated. Experimental results are given to show that a multiple instance learner stands up better to good word attacks than its single instance counterpart and the commonly practiced Bayesian filters. Then further development and future directions for this research are discussed.

2. Related work

Our work is primarily motivated by recent research on adversarial learning (Dalvi et al., 2004; Lowd & Meek, 2005a; Kolter & Maloof, 2005). Dalvi et al. (2004) consider classification to be a game between classifiers and adversaries in problem domains where adversarial attacks are expected. They model the computation of the adversary's optimal strategy as a constrained optimization problem and approximate its solution based on dynamic programming. Subsequently, an optimal classifier is produced against the optimal adversarial strategy. Their experimental results demonstrate that their game-theoretic approach outperforms traditional classifiers in the spam filtering domain. However, in their adversarial classification framework, they assume both the classifier and the adversary have perfect knowledge of each other, which is unrealistic in practice.
Instead of assuming the adversary has perfect knowledge of the classifier, Lowd & Meek (2005a) formalized the task of adversarial learning as the process of reverse engineering the classifier. In their adversarial classifier reverse engineer (ACRE) framework, the adversary aims to identify difficult spam instances (the ones that are hard to detect by the classifier) through membership queries. The goal is to find a set of negative instances with minimum adversarial cost within a polynomial number of membership queries. Newsome et al. (2006) emphasize the point that the training data used to build classifiers for spam filtering and the similar problem of internet worm detection is, to a large extent, controlled by an adversary. They describe and demonstrate several attacks on the generators of such classifiers in which the adversary is able to significantly impair the learning of accurate classifiers by manipulating the training data, even while still providing correct labels for the training instances. The attacks involve inserting features, in a specific manner, into one or both classes of the training data and are specifically designed to cause a significant increase in false positives or false negatives for the resulting classifier. They conclude that the generation of classifiers for adversarial environments should take into account the fact that training data is controlled by an adversarial source in order to ensure the production of accurate classifiers.

Barreno et al. (2006) explore possible adversarial attacks on machine learning algorithms from multiple perspectives. They present a taxonomy of different types of attacks on machine learning systems. An attack is *causative* if it targets the training data, and is *exploratory* if it aims to discover information through, for example, offline analysis. An attack is *targeted* if it focuses on a small set of points, and is *indiscriminate* if it targets a general class of points. An *integrity* attack leads to false negatives, and an *availability* attack aims to cause (machine learning) system dysfunction by generating many false negatives and false positives. They also discuss several potential defences against those attacks, and give a lower bound on the adversary's effort in attacking a naïve learning algorithm.

A practical example of adversarial learning is learning in the presence of the good word attack. Lowd & Meek (2005b) present and evaluate several variations of this type of attack on spam filters. They demonstrate two different ways to carry out the attack: passively and actively. Active good word attacks use feedback obtained by sending test messages to a spam filter in order to determine which words are "good". The active attacks were found to be more effective than the passive attacks; however, active attacks are generally more difficult than passive attacks because they require user-level access to the spam filter, which is not always possible. Passive good word attacks, on the other hand, do not involve any feedback from the spam filter, but rather, guesses are made as to which words are considered good. Three common ways for passively choosing good words are identified. First, *dictionary attacks* involve selecting random words from a large collection of words, such as a dictionary. In testing, this method did not prove to be effective; in fact, it actually increased the chances that the e-mail would be classified as spam. Next, *frequent word attacks* involve the selection of words that occur most often in legitimate messages, such as news articles. This method was more effective than the previous one, but it still required as many as 1,000 good words to be added to the original message. Finally, *frequency ratio attacks* involve the selection of words that occur very often in legitimate messages but not in spam messages. The authors' tests showed that this technique was quite effective, resulting in the average spam message being passed off as legitimate by adding as few as 150 good words to it. Preliminary results were also presented that suggested that frequent retraining on attacked messages may help reduce the effect of good word attacks on spam filters.

Webb et al. (2005) also examined the effectiveness of good word attacks on statistical spam filters. They present a "large-scale evaluation" of the effectiveness of the attack on four spam filters: naïve Bayes, support vector machine (SVM), LogitBoost, and Spam-Probe. Their experiments were performed on a large e-mail corpus consisting of around a million spam and ham messages, which they formed by combining several public and private corpora. Such a large and diverse corpus more closely simulates the environment of a server-level spam filter than a client-level filter. The experimental results show that, on normal e-mail, i.e., e-mail that has not been modified with good words, each of the filters is able to attain an accuracy as high as 98%. When testing on "camouflaged messages", however, the accuracies of the filters drop to between 50% and 75%. In their experiments, spam e-mails were camouflaged by combining them with portions of legitimate messages. They experimented with camouflaged messages containing twice as much spam content as legitimate content, and vice versa. They also proposed and demonstrated a possible solution to the attack. By training on a collection of e-mails consisting of half normal and half camouflaged messages, and treating all camouflaged messages as spam, they were able to improve the accuracy of the filters when classifying camouflaged messages.

Our counterattack strategy against good word attacks is inspired by work in the field of multiple instance (MI) learning. The concept of MI learning was initially proposed by Dietterich et al. (1997) for predicting drug activities. The challenge of identifying a drug molecule that binds strongly to a target protein is that a drug molecule can have multiple conformations, or shapes. A molecule is positive if at least one of its conformations binds tightly to the target, and negative if none of its conformations bind well to the target. The problem was tackled with an MI model that aims to learn axis-parallel rectangles (APR). Later, learning APR in the multiple instance setting was further studied and proved to be NP-complete by several other researchers in the PAC-learning framework (Auer, 1997; Long and Tan, 1998; Blum and Kalai, 1998).

Several probabilistic models: diverse density (DD) (Maron & Lozano-Pérez, 1998) and its variation EM-DD (Zhang & Goldman, 2002), and multiple instance logistic regression (MILR) (Ray & Craven, 2005), employ a maximum likelihood estimation to solve problems in the MI domain. The original DD algorithm searches for the target concept by finding an area in the feature space with maximum diverse density, i.e., an area with a high density of positive points and a low density of negative points. The diverse density at a point in the feature space is defined to measure probabilistically how many different positive bags have instances near that point, and how far the negative instances are from that point. EM-DD combines EM with the DD algorithm to reduce the multiple instance learning problem to a single-instance setting. The algorithm uses EM to estimate the instance in each bag which is most likely to be the one responsible for the label of the bag. The MILR algorithm presented by Ray & Craven (2005) is designed to learn linear models in a multiple instance setting. Logistic regression is used to model the posterior probability of the label of each instance in a bag, and the bag level posterior probability is estimated by using softmax to combine the posterior probabilities over the instances of the bag. Similar approaches with different combining functions are presented by Xu & Frank (2004).

Many single-instance learning algorithms have been adapted to solve the multiple instance learning problem. For example, Wang & Zucker (2000) propose the lazy MI learning algorithms, namely Baysian-kNN and citation-kNN, which solve the multiple instance learning problem by using the Hausdorff distance to measure the distance between two bags of points in the feature space. Chevaleyre & Zucker (2001) propose the multi-instance decision tree ID3-MI and decision rule learner RIPPER-MI by defining a new multiple instance entropy function and a multiple instance coverage function. Other algorithms that have been adapted to multiple instance learning include the neural network MI-NN (Ramon & Raedt, 2000), DD-SVM (Chen & Wang, 2004), MI-SVM and mi-SVM (Andrews et al., 2003), multi-instance kernels (Gärtner et al., 2002), MI-Ensemble (Zhou & Zhang, 2003), and MI-Boosting (Xu & Frank, 2004).

In this chapter, we demonstrate that a counterattack strategy against good word attacks, developed in the framework of multiple instance learning, can be very effective, provided that a single instance can be properly transformed into a bag of instances. We also explore several possible ways to transform e-mails into bags of instances. Our experiments also verify earlier observations, discussed in other works (Lowd & Meek, 2005b; Webb et al., 2005), that retraining on e-mails modified during adversarial attacks may improve the performance of the filters against the attack.

3. Problem definition

Consider a standard supervised learning problem with a set of training data $D = \{< X_1, Y_1 >, ..., < X_m, Y_m >\}$, where X_i is an instance represented as a single feature vector, $Y_i = C(X_i)$ is the target value of X_i, where C is the target function. Normally, the task is to learn C given D. The learning task becomes more difficult when there are adversaries who could alter some instance so that $X_i \rightarrow X_i^{'}$ and cause $Y_i \rightarrow Y_i^{'}$, where $Y_i \neq Y_i^{'}$. Let ΔX_i be the difference between X_i and $X_i^{'}$, i.e., $X_i^{'} = X_i + \Delta X_i$. In the case of spam filtering, an adversary can modify spam e-mails by injecting them with good words. So, ΔX_i represents a set of good words added to a spam message by the spammer. There are two cases that need to be studied separately:

1. the filter is trained on normal e-mails, i.e., e-mails that have not been injected with good words, and tested on e-mails which have been injected with good words;
2. both the training and testing sets contain e-mails injected with good words.

In the first case, the classifier is trained on a clean training set. Predictions made for the altered test instances are highly unreliable. In the second case, the classifier may capture some adversarial patterns as long as the adversaries consistently follow a particular pattern.
In both cases, the problem becomes trivial if we know exactly how the instances are altered; we could recover the original data and solve the problem as if no instances were altered by the adversary. In reality, knowing exactly how the instances are altered is impossible. Instead, we seek to approximately separate X_i and ΔX_i and treat them as separate instances in a bag. We then apply multiple instance learning to learn a hypothesis defined over a set of bags.

4. Multiple instance bag creation

We now formulate the spam filtering problem as a multiple instance binary classification problem in the context of adversarial attacks. Note that the adversary is only interested in altering positive instances, i.e., spam, by injecting sets of good words that are commonly encountered in negative instances, i.e., legitimate e-mails, or ham. We propose four different approaches to creating multiple instance bags from e-mails. We call them split-half (split-H), split-term (split-T), split-projection (split-P), and split-subtraction (split-S). We will now discuss each of these splitting methods, in turn.

4.1 Split-H

In our first splitting method, *split-half*, we split every e-mail right down the middle into approximately equal halves. Formally, let $B = \{B_1, ..., B_i, ..., B_m\}$ be a set of bags (e-mails), where $B_i = \{X_{i1}, X_{i2}\}$ is the i^{th} bag, X_{i1} and X_{i2} are the two instances in the i^{th} bag, created from the upper half and the lower half of the e-mail respectively. This splitting approach is reasonable in practice because spammers usually append a section of good words to either the beginning or the end of an e-mail to ensure the legibility of the spam message.
This splitting method, because it relies on the physical positioning of words in an e-mail, could potentially be circumvented by the spammer. There are two obvious ways to do this.

One is to create a visual pattern with good words so that the original spam message is still legible after the attack, but the spam is fragmented in such a way that "spammy" words are well separated and become less indicative of spam in the presence of a number of good words. The following example demonstrates how this idea can work effectively to circumvent this splitting method.

From: foo@internet.org
To: foo-foo@email.org
Subject: meeting agenda

good words	…	**low**	…	good words
good words	…	**mortgage**	…	good words
good words	…	**rate**	…	good words

The second way to defeat the split-H method is to append a very large block of good words to an e-mail so that after the split, good words outweigh spam-indicative words in all instances. The next three splitting methods do not rely on the positions of the words and thus do not suffer from this vulnerability.

4.2 Split-T

The second splitting method, *split-term* (split-T), partitions a message into three groups of words (terms) depending on whether the word is an indicator of spam, an indicator of ham, or neutral, i.e., $B_i = \{X_{is}, X_{in}, X_{ih}\}$, where X_{is} is the spam-likely instance, X_{in} is the neutral instance, and X_{ih} is the ham-likely instance in bag B_i. The instance to which each word is assigned is based on a weight generated for it during preprocessing. These weights are calculated using word frequencies obtained from the spam and legitimate messages in the training corpus. More specifically, the weight of a term W is given as follows:

$$weight(W) = \frac{p(W \mid D_s)}{p(W \mid D_s) + p(W \mid D_h)},$$

where D_s and D_h are the spam and ham e-mails in the training set respectively. When splitting an e-mail into instances we used two threshold values, $thresh_s$ and $thresh_\ell$, to determine which instance (spam-likely, ham-likely, or neutral) each word in the e-mail should be assigned to. We considered any word with a weight greater than $thresh_s$ to be spammy, any word with a weight less than $thresh_\ell$ to be legitimate, and any word with a weight in between to be neutral. Given each training set, $thresh_s$ was selected such that some fraction, for example 20%, of the terms chosen during attribute selection (discussed in Section 6.1) would have a weight greater than or equal to it. $thresh_\ell$ was selected so that some other fraction, for example 50%, of the terms would have a weight less than or equal to it. Reasonable fractions of terms and threshold values can be determined by using cross validation on training e-mails.

4.3 Split-P

The third splitting method, *split-projection* (split-P), transforms each message into a bag of two instances by projecting the message vector onto the spam and ham prototype vectors. The prototype vectors are computed using all the spam and ham messages in the training set. If we view the spam and ham messages in the training set as two clusters, then the prototypes are essentially the centroid of the two clusters. More specifically, let C_s be the set of e-mails that are spam and C_l be the set of e-mails that are legitimate. The prototypes are computed using Rocchio's algorithm (Rocchio Jr., 1971) as follows:

$$P_s = \beta \cdot 1/|C_s| \cdot \sum_{i=1}^{|C_s|} C_{s_i} - \gamma \cdot 1/|C_\ell| \cdot \sum_{i=1}^{|C_\ell|} C_{\ell_i}$$

$$P_\ell = \beta \cdot 1/|C_\ell| \cdot \sum_{i=1}^{|C_\ell|} C_{\ell_i} - \gamma \cdot 1/|C_s| \cdot \sum_{i=1}^{|C_s|} C_{s_i}$$

where C_{s_i} is the i^{th} spam message in C_s and $C_{\ell i}$ is the i^{th} ham message in C_ℓ, β is a fixed constant suggested to be 16 and γ is a fixed constant suggested to be 4. Given a message *M*, two new instances, M_S and M_ℓ, are formed by projecting *M* onto P_s and P_ℓ:

$$M_s = \frac{M \cdot P_s}{|P_s|^2} P_s$$

$$M_\ell = \frac{M \cdot P_\ell}{|P_\ell|^2} P_\ell$$

The rationale of this splitting approach rests on the assumption that a message is close to the spam prototype in terms of cosine similarity if it is indeed spam, and a ham message is close to the ham prototype.

4.3 Split-S

The last splitting method, *split-subtraction* (split-S), like the former, uses prototype (centroid) vectors. In this method, however, the ham and spam prototypes are calculated by averaging the corresponding attribute values of all of the ham and spam e-mails, respectively.

$$P_s = 1/|C_s| \cdot \sum_{i=1}^{|C_s|} C_{s_i}$$

$$P_\ell = 1/|C_\ell| \cdot \sum_{i=1}^{|C_\ell|} C_{\ell_i}$$

where C_s is a set of spam and C_{s_i} is the i^{th} spam message in C_s; C_ℓ is a set of ham, and C_{ℓ_i} is the i^{th} ham message in C_ℓ. A message can then be transformed from a single instance attribute vector *M* into a bag of two instances by subtracting corresponding attribute values in the single instance vector from the ham prototype and the spam prototype, yielding a legitimate instance $M_\ell = M - P_\ell$ and a spam instance $M_s = M - P_s$, respectively (Zhang & Zhou, 2007).

Now that we have devised several techniques for creating multiple instance bags from e-mail messages, we can transform the standard supervised learning problem of spam filtering into a multiple instance learning problem under the standard MI assumption. For this research, we adopt the multiple instance logistic regression (MILR) model to train a spam filter that is more robust to adversarial good word attacks than traditional spam filters based on single instance models. We chose to use the MILR classifier over other MI classifiers mainly because its single instance counter-part, logistic regression (LR), which has been shown to be very effective in the spam filtering domain (Yih et al., 2006), appeared to be the best among the single instance learners considered in our experiments. The next section outlines the multiple instance logistic regression learning model.

5. Multiple instance logistic regression

Given a set of training bags $B = \{< B_1, Y_1 >, ..., < B_i, Y_i >, ..., < B_m, Y_m >\}$, let $Pr(Y_i = 1 \mid B_i)$ be the probability that the i^{th} bag is positive, and $Pr(Y_i = 0 \mid B_i)$ be the probability that it is negative. Here Y_i is a dichotomous outcome of the i^{th} bag (e.g., spam or legitimate). The bag-level binomial log-likelihood function is:

$$L = \sum_{i=1}^{m} [Y_i \log Pr(Y_i = 1 \mid B_i) + (1 - Y_i) \log Pr(Y_i = 0 \mid B_i)].$$

In a single instance setting where logistic regression is used, given an example X_i, we model the expected value of the dichotomous outcome of X_i with a sigmoidal response function, i.e., $Pr(Y_i = 1 \mid X_i) = \exp(p \cdot X_i + b)/(1 + \exp(p \cdot X_i + b))$, then estimate the parameters $\boldsymbol{p}$ and b that maximize the log-likelihood function. In a multiple instance setting, we do not have direct measure of bag-level probabilities in the log-likelihood function. Instead, we estimate the instance-level class probabilities $Pr(Y_{ij} = 1 \mid X_{ij})$ with a sigmoidal response function as follows:

$$Pr(Y_{ij} = 1 \mid X_{ij}) = \frac{\exp(p \cdot X_{ij} + b)}{1 + \exp(p \cdot X_{ij} + b)},$$

where X_{ij} is the j^{th} instance in the i^{th} bag, and $\boldsymbol{p}$ and b are the parameters that need to be estimated. Thus, $Pr(Y_i = 0 \mid B_i)$ with instance-level class probabilities can be computed as follows:

$$Pr(Y_{ij} = 0 \mid X_{ij}) = 1 - Pr(Y_{ij} = 1 \mid X_{ij}) = \frac{1}{1 + \exp(p \cdot X_{ij} + b)}.$$

Now we can compute the probability that a bag is negative based on the MI assumption that a bag is negative if and only if every instance in the bag is negative:

$$Pr(Y_i = 0 \mid B_i) = \prod_{j=1}^{n} Pr(Y_{ij} = 0 \mid X_{ij})$$

$$= \exp(-\sum_{j=1}^{n} (\log(1 + \exp(p \cdot X_{ij} + b)))),$$

where n is the number of instances in the i^{th} bag. Thus the probability

$$Pr(Y_i = 1 \mid B_i) = 1 - Pr(Y_i = 0 \mid B_i)$$

estimates how likely a bag is positive if at least one instance in the bag is positive. In our case, given a set of e-mails for training, X_{ij} is a vector of the frequency count (or other variations such as a *tf-idf* weight) of unique terms in each e-mail. We can apply maximum likelihood estimation (MLE) to maximize the bag-level log-likelihood function, and estimate the parameters $\boldsymbol{p}$ and b that maximize the probability of observing the bags in B.

6. Experimental setup

We evaluated our multiple instance learning counterattack strategy on e-mails from the 2006 TREC Public Spam Corpus (Cormack & Lynam, 2006). Good word attacks were simulated by generating a list of good words from the corpus and injecting them into spam messages in the training and/or test data sets. We compared our counterattack strategy, using the multiple instance logistic regression model and the four splitting methods introduced above, to its single instance learning counterpart—logistic regression (LR)—and to the support vector machine (SVM) and the multinomial naïve Bayes (MNB) classifiers.

6.1 Experimental data

Our experimental data consists of 36,674 spam and legitimate e-mail messages from the 2006 TREC spam corpus. We preprocessed the entire corpus by stripping HTML and non-textual parts and applying stemming and stop-list to all terms. The **to**, **from**, **cc**, **subject**, and **received** headers were retained, while the rest of the headers were stripped. Messages that had an empty body after preprocessing were discarded. Tokenization was done by splitting on nonalphanumeric characters. We did not take any measures to counter obfuscated words in the spam messages. Given that there are a large number of possibilities to disguise a word, most content-based spam filters will not be able to deobfuscate the text of a message efficiently (Carpinter & Hunt, 2006). Recently, an efficient complementary filter (Lee & Ng, 2005) has been demonstrated to be able to effectively deobfuscate text with high accuracy. In practice, this type of technique could be used during preprocessing.

For our experiments we sorted the e-mails in the corpus chronologically by receiving date and evenly divided them into 11 subsets $\{D_1,...,D_{11}\}$. In other words, the messages in subset n come chronologically before the messages in subset n+1. Experiments were run in an on-line fashion, i.e., training on subset n and testing on subset n+1. Each subset contains approximately 3300 messages. The percentage of spam messages in each subset varies as in

the operational setting (see Figure 1). We used the Multiple Instance Learning Tool Kit (MILK) (Xu, 2003) implementation of MILR and the Weka 3.4.7 (Witten and Frank, 2000) implementations of LR, SVM and multinomial naïve Bayes, in our experiments. We reduced the feature space to the top 500 words ranked using information gain. Retaining 500 features appeared to be the best compromise among the classifiers in terms of improved efficiency and impaired performance. Attribute values for each message are calculated using the *tf-idf* (term frequency inverse document frequency) weighting scheme.

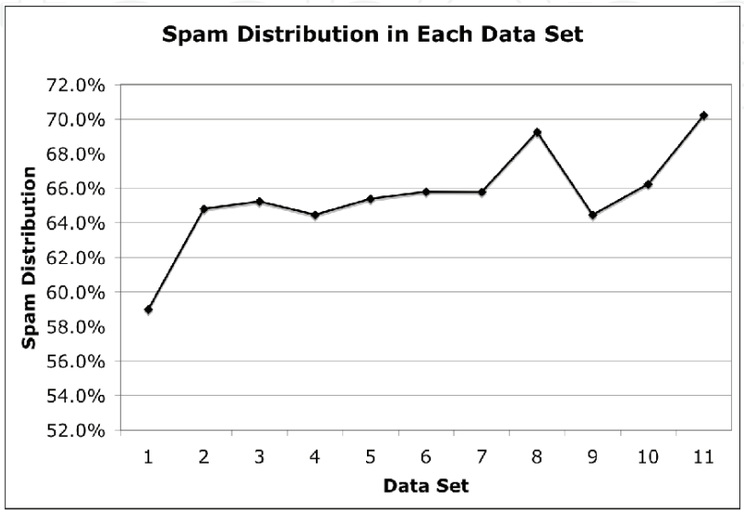


Fig. 1. Percentage of e-mails in each data set that are spam.

6.2 Good word list

To simulate the good word attacks, we generated a list of good words using all the messages in the TREC spam corpus. We ranked each word according to the ratio of its frequency in the legitimate messages over its frequency in the spam messages. We then selected the top 1,000 words from the ranking to use as our good word list. Generating the good word list in this manner has an important implication. Since the list was generated from the entire corpus rather than from the subset of messages used to train the classifiers, and since we represent e-mails using a feature vector of 500 features, some of the words in the list will not have an effect on the classification of messages that they are injected into. Such a list is more representative of the kind of list a spammer would be able to produce in practice, since the spammer would have no way of knowing the exact features used by the target filter. We noticed that in our experiments, only about 10% of the injected good words were actually retained in the feature vector, yet they had a significant impact on the classification.

7. Experimental results

We now present the results of two experiments in which we evaluate the effectiveness of our proposed multiple instance counterattack strategy. In the first experiment, we train all of the classifiers on normal e-mail (i.e., e-mail that has not been injected with good words) and then test them on e-mail that has been injected with good words. In the second experiment we train on both normal and attacked e-mails to observe how doing so affects classification of both normal and attacked e-mails.

7.1 Experiment 1: attacking the test set

In this experiment, we tested the ability of the MILR algorithm, using the four splitting methods introduced above, to classify e-mail injected with good words. We also tested the single instance logistic regression (LR), support vector machine (SVM) and multinomial naïve Bayes (MNB) classifiers for comparison. The classifiers were each trained and tested on the eleven chronologically sorted data sets in an on-line fashion. That is, all of the classifiers were trained on the same unaltered data set D_n, and then tested on the data set D_{n+1}, for n=1...10. Fifteen variations of each test set were created to test the susceptibility of the classifiers to good word attacks of varying strength. The first version of each test set was left unmodified, i.e., no good words were injected. Half of the spam messages (selected at random) in each of the remaining 14 variations of each test set were injected with some quantity of random good words from our good word list, beginning with 10 words. With each successive version of the test set, the quantity of good words injected into half of the spam messages was increased: first in increments of 10 words, up to 50, and then in increments of 50 words up to 500. The injected words were randomly selected, without replacement, from our good word list on a message by message basis. We chose to inject good words into only half of the messages in each test set because, in practice, spam messages injected with good words account for only a subset of the spam e-mails encountered by a given filter. The precision and recall values on each version of the test set for all 10 test sets were averaged and recorded for each classifier. In our results, we use "MILRH", "MILRT", "MILRP" and "MILRS" where split-H, split-T, split-P and split-S were used with MILR, respectively.

Figures 2 and 3 show how the average precision and average recall, respectively, of each classifier is affected as the good word attack increases in strength (that is, the quantity of good words injected into the spam e-mails in the test set increases).

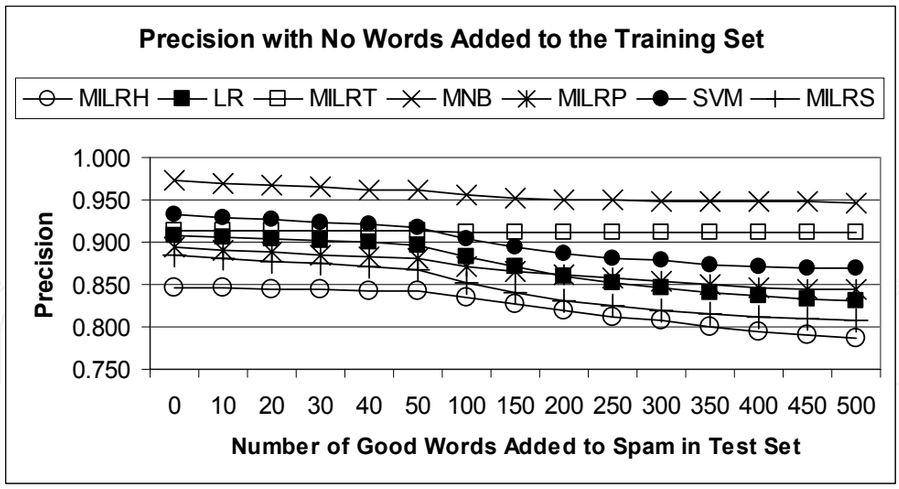


Fig 2. Change in average precision as good words are injected into the test Set.

From the results we can see that, with the exception of MILRT, each classifiers' ability to correctly identify spam e-mail was significantly reduced as a result of the simulated good word attack. Of all the classifiers MILRT was most resilient to the attack, dropping by only 0.3% (from 0.914 to 0.911) in average precision and by only 2% (from 0.901 to 0.883) in average recall after 500 good words had been injected into the test set. MILRH and MILRP stood up better to the attack than the single instance classifiers and the MILRS classifier, but

the attack still had a very noticeable effect on their ability to classify spam, reducing the average recall of MILRH by 27.1% (from 0.980 to 0.714) and the average recall of MILRP by 29.2% (from 0.957 to 0.678). The average precision values of MILRH and MILRP dropped by 7% (from 0.846 to 0.787) and by 5.7% (from 0.895 to 0.844), respectively. Of the single instance classifiers, LR was the most resilient; however, the attack still had a very significant effect on its ability to classify spam, reducing its average recall by 42.8% (from 0.966 to 0.553) and its average precision by 8.6% (from 0.909 to 0.831). The average recall of MNB and SVM dropped by 47.4% (from 0.914 to 0.481) and 46.2% (from 0.932 to 0.501), respectively. Their average precision values dropped by 2.7% (from 0.973 to 0.947) and by 6.9% (from 0.932 to 0.868), respectively. MILRS turned out to be nearly as vulnerable to the attack as the single instance classifiers, dropping by 37.8% (from 0.961 to 0.598) in average recall and by 8.7% (from 0.885 to 0.808) in average precision.

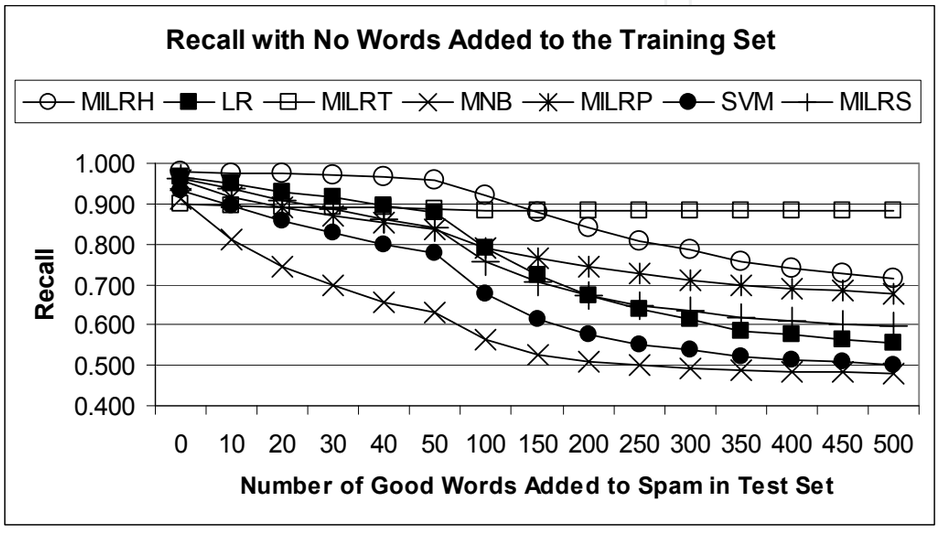


Fig 3. Change in average recall as good words are injected into the test set.

One thing that is clear from these results is that the effectiveness of our multiple instance counterattack strategy is very much dependent on the specific technique used to split e-mails into multiple instance bags. The success of the split-term method (MILRT) is due to the fact that the classifier is able to consider both spammy and legitimate terms independently, since they are placed into separate instances in the bag created from an e-mail. Under the multiple instance assumption, if at least one instance in a bag is spammy, the entire bag is labeled as spammy. When good words are injected into a spam message they end up in the legitimate instance of the bag and have no effect on the spammy instance; thus the bag still contains a spammy instance and is classified correctly as spam.

7.2 Experiment 2: training on attacked spam messages

In the second experiment, our goal was to observe the effect that training on messages injected with good words has on the susceptibility of the classifiers to attacks on the test set. As in the previous experiment, we tested each of the classifiers on the eleven chronologically sorted data sets in an on-line fashion. This time, however, in addition to creating 15 versions of the test set injected with increasing quantities of good words, we also created 5 versions of the training set. We injected 10 good words into half of the spam messages (selected at random) in the first version of the training set and then increased the number of injected

good words by 10 for each subsequent version, up to 50 good words for the fifth version. We also tried injecting larger numbers of good words, but after exceeding 50 words, the additional effect was minimal; therefore, those results are not shown here. For each version of the training set we tested the classifiers on the 15 versions of the corresponding test set. As before, good words were selected from our good word list randomly and without replacement on a message by message basis. For all ten tests, the precision and recall values on each version of the test set were averaged and recorded, separately for each of the 5 versions of the training set. In the interest of space, the results of this experiment are presented in terms of average F-measure that is the harmonic mean of the precision and recall values, and only the results of the experiments when 10, 30 and 50 good words are added to some spam in the training set are presented. Figures 4-6 show the average F-measure of each of the classifiers when 10, 30 and 50 good words are injected into half of the spam messages in the training set, respectively.

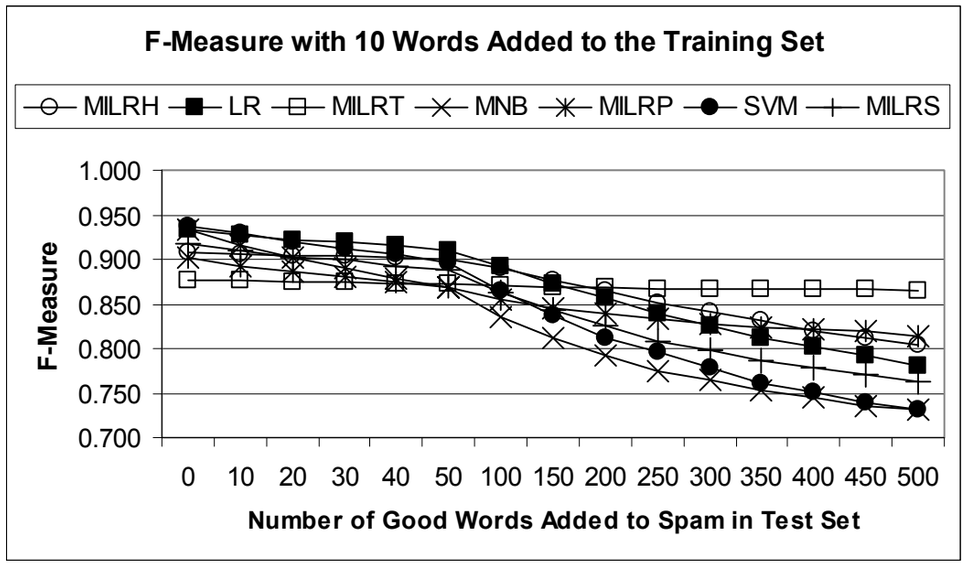


Fig 4. Change in average F-measure as good words are injected into the test set. 10 good words were also injected into the training set.

From these results we can see that injecting just 10 good words into half of the spam messages in the training set appeared to lessen the effect of the good word attack for almost all of the classifiers (see Fig. 4). Further increasing the number of good words injected into the training set continued to lessen the effect of the attack for all of the classifiers (see Fig. 5). After 30 good words had been injected into the training set, the presence of good words in the test messages actually began to increase the likelihood that such messages would be correctly classified as spam (see Fig. 6). These results confirm the observations of several other researchers (Lowd and Meek, 2005b; Webb et al., 2005) that retraining on normal and attacked e-mails may help to counter the effects of the good word attack. However, it is important to realize that this would only work in cases where the attacked messages being classified contained the same good words as the attacked messages that the spam filter was trained on. One of the major advantages of our proposed multiple instance strategy is that the spam filter need not be trained on attacked messages in order to be effective against attacks and further, that frequent retraining on attacked messages is not necessary for the strategy to maintain its effectiveness.

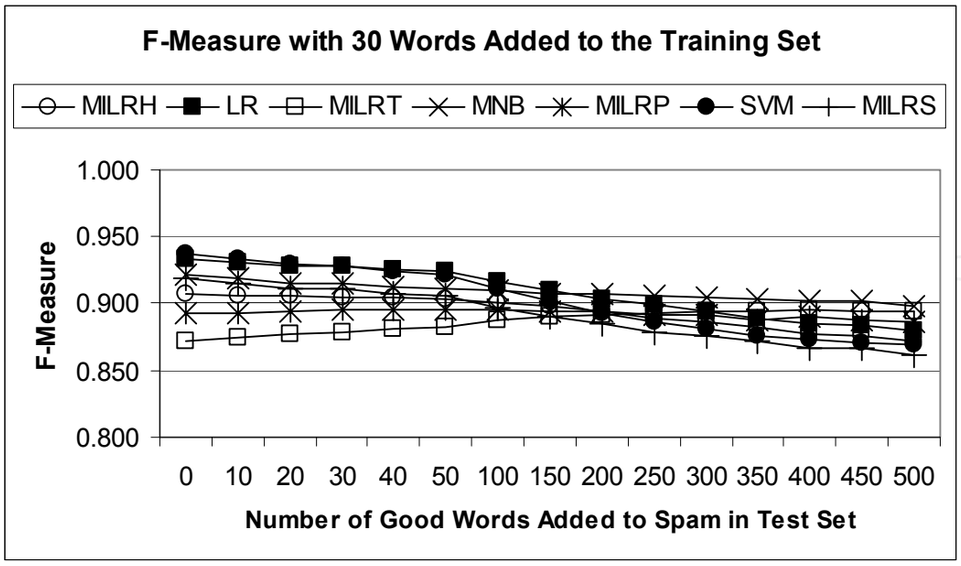


Fig 5. Change in average F-measure as good words are injected into the test set. 30 good words were also injected into the training set.

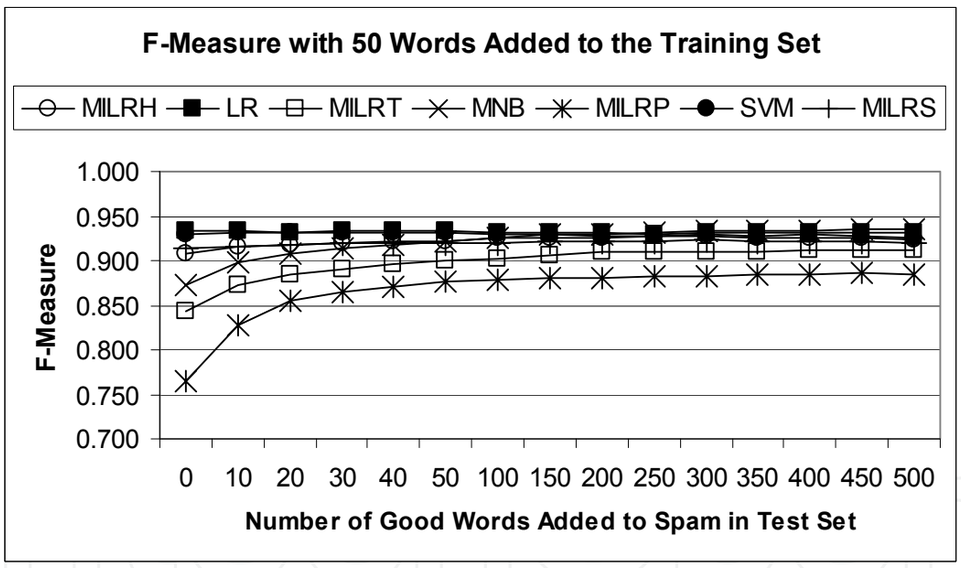


Fig 6. Change in average F-measure as good words are injected into the test set. 50 good words were also injected into the training set.

8. Conclusions and future work

A multiple instance learning counterattack strategy for combating adversarial good word attacks on statistical spam filters has been presented. In the proposed strategy, e-mails are treated as multiple instance bags and a logistic model at the instance level is learned indirectly by maximizing the bag-level binomial log-likelihood function. The proposed counterattack strategy has been demonstrated on good word attacks of varying strength and has been shown to be effective. Additionally, we have confirmed earlier reports that

retraining on attacked as well as normal e-mails may strengthen a spam filter against good word attacks. One of the advantages of our proposed strategy, as demonstrated by our experiments, is that it is effective even when trained on normal e-mail and that frequent retraining on attacked messages is not necessary to maintain that effectiveness. We presented several possible methods for creating multiple instance bags from e-mails. As was observed from our experimental results, the splitting method used ultimately determines how well the strategy performs. The splitting methods we presented here work fairly well, especially the split-term method, but there are possibly other, perhaps better, methods that could be used. We plan to investigate other possible splitting methods in the future, and investigate the vulnerability of each splitting method in adversarial environments. In our experiments, we simulated the real operational environment where the adversary does not have a complete knowledge of the training data. We plan to investigate the effectiveness of our proposed counterattack strategy in the extreme cases where we assume the adversary knows the make-up of the training examples.

Since it is an arms race between spammers and filter designers, we also plan to make our MI strategy adaptive as new spam techniques are devised, and on-line as the concept of spam drifts over time. In addition, we plan to investigate the possibility of extending the proposed multiple instance learning strategy to handle similar adversarial attacks in other domains.

9. References

S. Andrews, I. Tsochantaridis, and T. Hofmann. Support vector machines for multiple instance learning. In *NIPS 15*, pages 561–568. MIT Press, 2003.

P. Auer. On learning from multi-instance examples: Empirical evaluation of a theoretical approach. In *Proceedings of the 14th International Conference on Machine Learning*, pages 21–29, San Francisco, CA, 1997. Morgan Kaufmann.

M. Barreno, B. Nelson, R. Sears, A. D. Joseph, and J. D. Tygar. Can machine learning be secure? In ASIACCS '06: *Proceedings of the 2006 ACM Symposium on Information, computer and communications security*, pages 16–25, New York, NY, USA, 2006. ACM Press. ISBN 1-59593-272-0. doi: http://doi.acm.org/10.1145/1128817.1128824.

A. Blum and A. Kalai. A note on learning from multiple-instance examples. *Machine Learning*, 30(1):23–30, 1998.

J. Carpinter and R. Hunt. Tightening the net: A review of current and next generation spam filtering tools. *Computers and Security*, 25(8):566–578, 2006.

Y. Chen and J.Z. Wang. Image categorization by learning and reasoning with regions. *Journal of Machine Learning Research*, 5:913–939, 2004.

Y. Chevaleyre and J.D. Zucker. Solving multiple-instance and multiple-part learning problems with decision trees and rule sets. application to the mutagenesis problem. In *Proceedings of the 14th Biennial Conference of the Canadian Society for Computational Studies of Intelligence*, pages 204–214, 2001.

G. V. Cormack and T. R. Lynam. Spam track guidelines — TREC 2005-2007. http://plg.uwaterloo.ca / gvcormac/treccorpus06/, 2006.

N. Dalvi, P. Domingos, Mausam, S. Sanghai, and D. Verma. Adversarial classification. In *Proceedings of the 2004 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 99–108. ACM Press, 2004.

T.G. Dietterich, R.H. Lathrop, and T. Lozano-Pérez. Solving the multiple-instance problem with axis-parallel rectangles. *Artificial Intelligence Journal*, 89(1-2):31–71, 1997.

T. Gärtner, P. Flach, A. Kowalczyk, and A. Smola. Multi-instance kernels. In *Proceedings Of the 19th International Conference on Machine Learning*, pages 179–186, San Francisco, CA, 2002. Morgan Kaufmann.

R. Jennings. The global economic impact of spam. Technical report, Ferris Research, 2005.

J.Z. Kolter and M.A. Maloof. Using additive expert ensembles to cope with concept drift. In *Proceedings of the Twenty-second International Conference on Machine Learning*, pages 449–456, New York, NY, 2005. ACM Press.

H. Lee and A. Ng. Spam deobfuscation using a hidden markov model. In *Proceedings of the Second Conference on Email and Anti-Spam*, 2005.

P. Long and L. Tan. Pac learning axis-aligned rectangles with respect to product distribution from multiple-instance examples. *Machine Learning*, 30(1):7–21, 1998.

D. Lowd and C. Meek. Adversarial learning. In *Proceedings of the 2005 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 641 647. ACM Press, 2005a.

D. Lowd and C. Meek. Good word attacks on statistical spam filters. In *Proceedings of the 2nd Conference on Email and Anti-Spam*, 2005b.

O. Maron and T. Lozano-P´erez. A framework for multiple-instance learning. *Advances in Neural Information Processing Systems*, 10:570–576, 1998.

J. Newsome, B. Karp, and D. Song. Paragraph: Thwarting signature learning by training maliciously. In *Recent Advances in Intrusion Detection: 9th International Symposium (RAID)*, pages 81–105, 2006.

J. Ramon and L.D. Raedt. Multi instance neural networks. In *Proceedings of ICML-2000 workshop on Attribute-Value and Relational Learning*, 2000.

S. Ray and M. Craven. Supervised versus multiple instance learning: An empirical comparison. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 697–704, New York, NY, 2005. ACM Press.

J. Rocchio Jr. Relevance feedback in information retrieval. In *The SMART Retrieval System: Experiments in Automatic Document Processing*, pages 68–73. Prentice Hall, 1971.

J. Wang and J.D. Zucker. Solving the multiple-instance learning problem: A lazy learning approach. In *Proceedings of the 17th International Conference on Machine Learning*, pages 1119–1125, San Francisco, CA, 2000. Morgan Kaufmann.

S. Webb, S. Chitti, and C. Pu. An experimental evaluation of spam filter performance and robustness against attack. In *The 1st International Conference on Collaborative Computing: Networking, Applications and Worksharing*, pages 19–21, 2005.

I.H. Witten and E. Frank. *Data Mining: Practical machine learning tools with Java implementations*. Morgan Kaufmann, San Francisco, CA, USA, 2000.

X. Xu. Statistical learning in multiple instance problems. Master's thesis, University of Waikato, 2003.

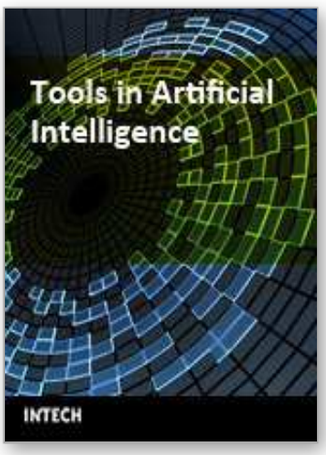
X. Xu and E. Frank. Logistic regression and boosting for labeled bags of instances. In *Proceedings of the Pacific-Asian Conference on Knowledge discovery and data mining*. Springer-Verlag, 2004.

W. Yih, J. Goodman, and G. Hulten. Learning at low false positive rates. In *Proceedings of the Third Conference on Email and Anti-Spam*, 2006.

Q. Zhang and S. Goldman. Em-dd: An improved multiple-instance learning technique. In *Proceedings of the 2001 Neural Information Processing Systems (NIPS) Conference*, pages 1073–1080, Cambridge, MA, 2002. MIT Press.

M. L. Zhang and Z. H. Zhou. 2007. Multi-label learning by instance differentiation. In The 22nd AAAI Conference on Artificial Intelligence (AAAI'07), pages 669–674, Vancouver, Canada.

Z.H. Zhou and M.L. Zhang. Ensembles of multi-instance learners. In *ECML-03, 15th European Conference on Machine Learning*, pages 492–502, 2003.



Tools in Artificial Intelligence

Edited by Paula Fritzsche

ISBN 978-953-7619-03-9
Hard cover, 488 pages
Publisher InTech
Published online 01, August, 2008
Published in print edition August, 2008

This book offers in 27 chapters a collection of all the technical aspects of specifying, developing, and evaluating the theoretical underpinnings and applied mechanisms of AI tools. Topics covered include neural networks, fuzzy controls, decision trees, rule-based systems, data mining, genetic algorithm and agent systems, among many others. The goal of this book is to show some potential applications and give a partial picture of the current state-of-the-art of AI. Also, it is useful to inspire some future research ideas by identifying potential research directions. It is dedicated to students, researchers and practitioners in this area or in related fields.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen