Highly Scalable SVM Modeling with Random Granulation for Spam Sender Detection

Yuchun Tang, Yuanchen He, Sven Krasser
Secure Computing Corporation
4800 North Point Parkway, Suite 300
Alpharetta, GA 30022, USA
{ytang, yhe, skrasser}@securecomputing.com

Abstract

Spam sender detection based on email subject data is a complex large-scale text mining task. The dataset consists of email subject lines and the corresponding IP address of the email sender. A fast and accurate classifier is desirable in such an application. In this research, a highly scalable SVM modeling method, named Granular SVM with Random granulation (GSVM-RAND), is designed. GSVM-RAND applies bootstrapping to extract a number of subsets of samples from the original training dataset. Each training subset is then projected into a feature subspace randomly selected from the original feature space. Here we call a granule such a subset of samples in such a feature subspace. A local SVM is then modeled in each granule. For a new sample, it is firstly projected into each granule in which the local SVM is fired to make a prediction. After that, all SVM predictions are aggregated by Bayesian Sum Rule for a final decision. GSVM-RAND is easy to be parallelized and hence efficient and highly scalable. GSVM-RAND is also effective by integrating a large number of weak, low-correlated local SVMs.

1 Introduction

In this research we present a novel algorithm based on Support Vector Machine (SVM) classification [17]. The proposed algorithm is able to classify high dimensional sparse data as encountered in text mining efficiently and effectively. We apply this new algorithm to an email dataset consisting of records of email subject text lines and the corresponding sender IP address. In this data, we mine malicious IP addresses (i.e. IPs sending spam, virus, and phishing messages).

Unsolicited spam, virus, and phishing messages pose a great threat to email communications and company networks can easily get overwhelmed with the high volume of these malicious messages. A recent development to stem this flood of messages is email reputation systems, which can assign a reputation to certain identifiers observed in a message [2]. This allows to quickly filter out a large amount of unwanted messaging traffic by doing a simple lookup in the reputation system's database.

One such reputation system is TrustedSource [1]. TrustedSource allows looking up reputation information and statistical data on identifiers like IP addresses, message fingerprints, and URLs. In previous work, we presented how even simple query information purely based on queries for the sending IP of a message can be used to detect malicious senders by aggregating global messaging data [14]. Ramachandran et al. propose a related approach that is able to detect malicious IPs if information on the destination domain is available [12].

In this research, we focus on the classification of IP addresses based on the email subjects they sent. Data is gathered from selected email servers that transmit pairs of a sending IP address and email subjects to the TrustedSource analysis center. Since this dataset is extensive and spam senders tend to have short periods of activity, an important design goal for a classifier is computational efficiency.

2 Machine Learning for Classification

2.1 Support Vector Machine

It is a binary classification problem to discriminate malicious sending IP addresses from legal ones. Given a training dataset Tr with n samples $(\mathbf{x_1}, y_1), (\mathbf{x_2}, y_2), ..., (\mathbf{x_n}, y_n)$, where $\mathbf{x_i}$ is a feature vector in a d-dimensional feature space R^d and $y_i \in \{-1, +1\}$ is the corresponding class label, $1 \leq i \leq n$, the task is to find a classifier with a decision function $f(\mathbf{x}, \theta)$ such that $y = f(\mathbf{x}, \theta)$, where y is the class label for $\mathbf{x}$, θ is a vector of unknown parameters.

SVM is a classification technique based on statistical learning theory [17]. Geometrically, the SVM modeling algorithm finds an optimal hyperplane with the maximal margin to separate two classes, which requires to solve the following optimization problem.

$$\begin{aligned} &\text{maximize} \\ &\quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K(\mathbf{x_i}, \mathbf{x_j}) \\ &\text{subject to} \\ &\quad \sum_{i=1}^{n} \alpha_i y_i = 0 \\ &\quad 0 \leq \alpha_i \leq C, i = 1, 2, ...n \end{aligned} \tag{1}$$

where α_i is the weight assigned to the training sample $\mathbf{x_i}$. If $\alpha_i > 0$, $\mathbf{x_i}$ is called a support vector. C is a "regulation parameter" used to trade-off the training accuracy and the model complexity so that a superior generalization capability can be obtained. K is a kernel function, which is used to measure the similarity between two samples. A popular RBF kernel function, as shown in (2), is used in this research.

$$K(\mathbf{x_i}, \mathbf{x_j}) = \exp(-\gamma \|\mathbf{x_i} - \mathbf{x_j}\|^2), \gamma > 0 \tag{2}$$

After the weights are determined, a test sample $\mathbf{x}$ is classified by

$$y = \text{sign}\left(\sum_{i=1}^{n} \alpha_i y_i K(\mathbf{x_i}, \mathbf{x}) \right), \tag{3}$$

$$\text{sign}(a) = \begin{cases} +1, & \text{if } a > 0 \\ -1, & \text{otherwise} \end{cases}$$

To determine the values of $< \gamma, C >$, a Cross Validation (CV) process is usually conducted on the training dataset. CV is also used to estimate the generalization capability on new samples that are not in the training dataset. A k-fold CV randomly splits the training dataset into k approximately equal-sized subsets, leaves out one subset, builds a classifier on the remaining samples, and then evaluates classification performance on the unused subset. This process is repeated k times for each subset to obtain the CV performance over the whole training dataset. CV is computationally expensive and hence is not feasible for time-critical classification tasks.

2.2 Granular Computing

Granular computing represents information in the form of some aggregates (called *information granules*) such as subsets, subspaces, classes, or clusters of a universe. It then solves the targeted problem in each information granule [10]. There are two principles in granular computing. The first principle is divide-and-conquer to split a huge problem into a sequence of granules (*granule split*); The second principle is data cleaning to define the suitable size for one granule to comprehend the problem at hand without getting buried in unnecessary details (*granule shrink*). As opposed to traditional data-oriented numeric computing, granular computing is knowledge-oriented [3]. By embedding prior knowledge into the granulation process for data modeling, better classification can be obtained.

A granular computing-based learning framework called Granular Support Vector Machines (GSVM) was proposed in [13]. GSVM combines the principles from statistical learning theory and granular computing theory in a systematic and formal way. GSVM works by extracting a sequence of information granules with granule split and/or granule shrink, and then building an SVM on some of these granules when necessary. The main potential advantages of GSVM are:

1. GSVM is more sensitive to the inherent data distribution by trading off between local significance of a subset of data and global correlation among different subsets of data, or trading off between information loss and data cleaning. Hence, GSVM may improve the classification performance.

2. GSVM may speed up the modeling process and the classification process by eliminating redundant data locally. As a result, it is more efficient and scalable on huge datasets.

2.3 Related Work

Some works have been reported on SVM ensembling with bagging [7, 6, 16, 18, 15].

Kim et al. [7] proposed to use the SVM ensembles with bagging to improve the classification accuracy. The proposed method demonstrated 1-2% accuracy improvement on SVM modeling in their experiments. However, the experiments were only conducted on two small datasets and no efficiency analysis was given.

Collobert et al. [6] designed a parallel mixture of SVMs for very large scale problems. Compared to one SVM, both effectiveness and efficiency improvements were observed with the gated SVM mixture. However, they did not use bagging but randomly divide the training dataset into approximately equal sized subsets. The subsets need to be adjusted in the following recursive process to ensure a balance among them. Without bagging, their method had to estimate the generalization capability with expensive cross-validation process. And the datasets in their experiments are low dimensional with only 54 or 135 input features.

In [7, 6], both of them observed the best performance by learning a meta classifier for aggregation. However, meta learning is expensive and hence maybe not desired on a large dataset.

Valentini et al. [16] proposed bagging of low-bias SVMs. The target is to reduce bias instead of classification error for each individual SVM. The experiments indicated that it often improves classification accuracy, compared to one well-tuned SVM and to bags of individually well-tuned SVMs. However, the idea was only tested on small datasets and no efficiency analysis was given. It is also expensive to tune individual SVMs separately in their method.

Yan et al. [18] presented an SVMs ensemble method based on bagging and fuzzy integral. The simulating results demonstrated their method outperformed a single SVM and traditional SVMs aggregation technique via major voting in terms of accuracy. They also proposed to tune each individual SVM separately. However, the idea was only tested on small datasets and no efficiency analysis was given. Only 3 or 8 times bagging were conducted in their experiments. With a small bagging number, the ensemble classifier may be not converged to the optimum. With a large bagging number, it is expensive to tune individual SVMs separately (with CV).

Tao et al. [15] used both bagging and random subspace for constructing an SVM ensemble to improve the relevance feedback performance in content-based image retrieval. Their ABRS-SVM algorithm focused on building SVM ensembles on highly imbalanced data in the specific application domain. They applied a two-layer process that generated several bootstrapping subsets first and then projected each subset into several subspaces. This method may limit diversity among different SVMs because some SVMs were built from the same subset. They tested ABRS-SVM on a small size but high dimensional dataset.

All of these works were not tested on a both "high" and "wide" dataset, i.e., with both a large number of samples and a large number of features, which is not uncommon in a large scale text mining application. And hence it is worth to investigate how an SVM ensemble method with bagging and random subspace projection behaves on such a high and wide dataset, in terms of both effectiveness and efficiency.

These works did not also evaluate classification effectiveness with Receiver Operating Characteristic (ROC) analysis [4], which is typically desirable in a real world classification system.

3 GSVM-RAND

In this research, we design a novel GSVM modeling algorithm by utilizing bootstrapping and random subspace projection for granulation. We also investigate three different SVM modeling aggregation methods.

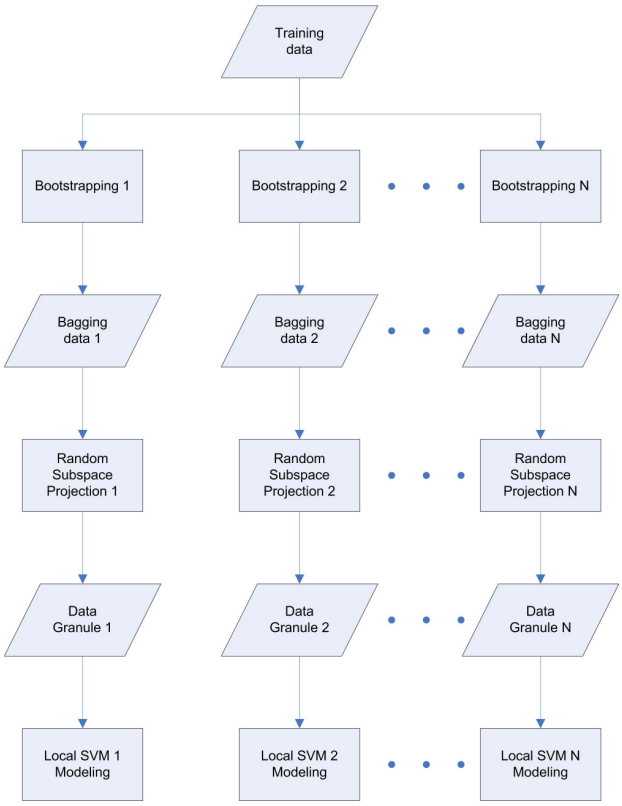

Figure 1. GSVM-RAND training

3.1 Bootstrapping

Given a training dataset Tr of n samples, a bootstrapping process generates a new in-bag subset Tr_IB of size $n'(n' <= n)$ by sampling uniformly from Tr with replacement. By sampling with replacement it is likely that some samples are repeated in Tr_IB. If $n' = n$ (100% bootstrapping), with large n the set Tr_IB is expected to have 63.2% samples of Tr, the rest being duplicates. And the remaining 36.8% samples form another out-of-bag subset Tr_OB. If $n' < n$, for example $n' \cong \frac{n}{10}$ (10% bootstrapping), less samples are expected to be in Tr_IB and more in Tr_OB.

We can do bootstrapping multiple times on Tr to generate many different in-bag subsets. A smaller n' would further encourage diversity among different in-bag subsets. A smaller n' is also beneficial to improve efficiency because the in-bag subsets are smaller.

3.2 Random Subspace Projection

After each bootstrapping is done, the in-bag subset and the out-of-bag subset are projected into a subspace. The subspace is constructed by randomly selecting a small part of features from the original feature space without replacement. Different bootstrapping datasets are (very likely) pro-

jected into different subspaces. And hence we can further enlarge diversity among different data subsets. Here we call a *granule* is created after a bootstrapping and a random subspace projection.

On the email subject data (as well as other text mining vector-space-modeled data), there are typically a lot of tokens as features. Each token feature is low-informative and there is large redundancy among them. The redundancy suggests us to select only a few of features in each granule. By selecting a smaller number of features each local SVM is less accurate but we can further enlarge diversity (low-correlation) among different granules. A smaller number of features also induces faster modeling.

3.3 Local SVM Modeling

On each granule a local SVM is modeled on the in-bag data subset in the feature subspace. Many previous research works with bagging suggest that a local classifier should be weak and low-correlated to each other [7, 6, 16, 18, 15]. Guided by this suggestion, the random granulation process is prone to select small granules with both a small number of samples and a small number of features. As such, it encourages diversity among different granules both data-wise and feature-wise. So we can expect low correlation among local SVMs, instead of classification strength of each local SVM.

3.4 Aggregation

After local SVMs are modeled, the next step is to aggregate the predictions from local SVMs to make a final decision. Here we try three different aggregation operators.

The simplest and most common aggregation method is Major Voting (MV), which is simply to sum up all prediction labels from local SVMs as the final prediction.

Because an SVM outputs a decision value as well as a label, we also try to sum up the decision values as the final prediction. Here we name it Decision Aggregation (DA).

We also try Bayesian Sum Rule (BSR) [9, 11, 8]. The basic idea is to output probability estimate from each local SVM. And then all probability estimates are summed up as the final prediction [18, 15].

Some previous works also proposed to learn a meta classifier for aggregation [7, 6]. GSVM-RAND does not adopt meta learning as it increases modeling complexity.

3.5 Out-Of-Bag Effectiveness Evaluation

Bootstrapping splits the training dataset into an in-bag subset and an out-of-bag subset. If we do 100 times 100% bootstrapping, averagely a sample is not used for 36.8 local SVMs training. These 36.8 local SVM predictions can be aggregated and used as the estimation of classification on unseen new samples. If the bootstrapping ratio is 10%, averagely 93.7 local SVM predictions can be aggregated for out-of-bag estimation.

Figures. 1-2 sketch the training phase and the testing phase of GSVM-RAND, respectively.

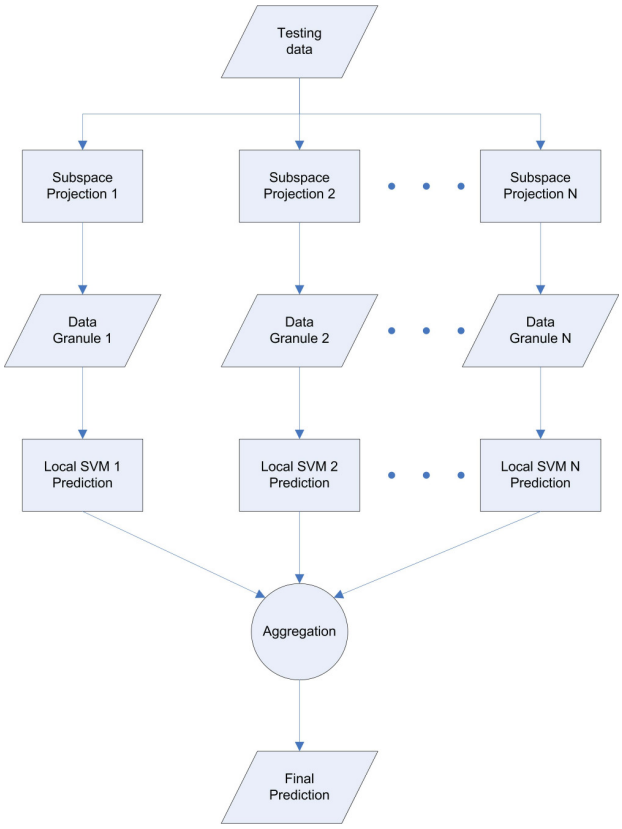

Figure 2. GSVM-RAND testing

Table 1. Subject data on NOV/06/2007

#tokens	569,823
#IPs	259,953
#non-spam IPs	12,131
#spam IPs	247,822

4 Experiments

Classification modeling is carried out on a workstation with a Intel Xeon CPU at 1.86GHz and 16 GB of memory.

4.1 Experiment Design

Table 1 lists the characteristic of the dataset for modeling. We retrieve 259,953 IPs sending emails in NOV/06/2007. For each IP, we concatenate the subject lines

Table 2. Effectiveness/Efficiency Result

Method	AUC-Testing	Modeling time
SVM	0.99385	933 mins
GSVM-BSR	0.98762	437 mins
GSVM-MV	0.98602	744 mins
GSVM-DA	0.95667	759 mins

of all emails sent from it as one single string, separated by spaces. And then these long subject strings are tokenized and 569,823 tokens are extracted. Finally Term Frequency-Inverse Document Frequency (TF-IDF) based feature vectors are built for these IPs on the high dimensional token space. The task is to build a classifier to discriminate spam IPs (labeled 1) from non-spam IPs (labeled -1) on such a dataset. The dataset is randomly divided into 3 equal-sized subsets, in which the ratio of non-spam over spam is also equal. Two subsets are combined as the training dataset and another subset is leaved as the testing dataset.

We choose LIBSVM [5] for SVM modeling with the RBF kernel. Because spam IPs are more than non-spam ones, the False Negative (FN) cost is 1 and the False Positive (FP) cost is 20.4272 for balance.

For SVM modeling, 5-fold CV is conducted on the training dataset for generalization capability estimation.

Notice that GSVM-RAND modeling does not need CV because out-of-bag accuracy is used as the estimation of generalization capability. We do 10% bootstrapping 100 times. In each bootstrapping 10% samples are randomly selected with replacement into the in-bag dataset and the remaining samples are used for out-of-bag estimation. In each granule we randomly select 6% of features for modeling. Averagely, each feature is selected into 6 granules. 10% of samples and 6% of features form a very small granule on which a local SVM can be quickly generated. The small size of each granule also encourages diversity among local SVMs instead of strength/accuracy of each local SVM. We expect bootstrapping aggregation can combine the 100 weak local SVMs into 1 highly accurate classifier. We try Major Voting, Decision Aggregation and Bayesian Sum Rule for aggregation of the 100 local SVMs.

4.2 Result Analysis

The modeling effectiveness/efficiency results are reported in Table 2. Area Under the Receiver Operating Characteristic Curve (AUC) is used for effectiveness evaluation. SVM denotes building a single SVM with 5-fold CV. GSVM-BSR denotes GSVM-RAND modeling with Bayesian Sum Rule for aggregation. GSVM-MV denotes GSVM-RAND modeling with Major Voting for aggregation. GSVM-DA denotes GSVM-RAND modeling with Decision Value Sum for aggregation.

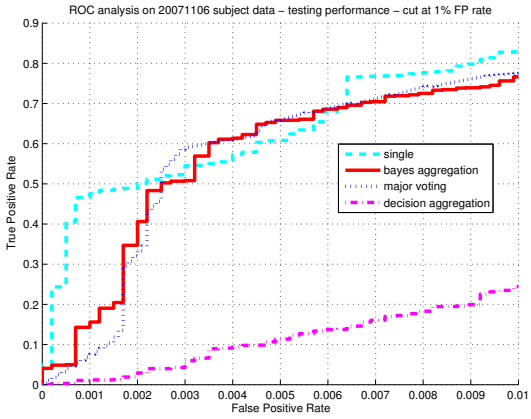

Figure 3. ROC analysis - testing

Table 3. Local SVM Effectiveness

AUC	Min	Max	Mean
Out-of-Bag	0.79912	0.89499	0.83926
Testing	0.80303	0.89612	0.84106

We observe that Bayesian Sum Rule is the most effective aggregation method. Major Voting is a little worse while Decision Aggregation significantly decreases classification effectiveness. Compared to one single SVM, GSVM-RAND with Bayesian Sum Rule is almost the same accurate but it is much faster. Notice that the modeling time including training time plus testing time.

GSVM-RAND is by nature very easy to be parallelized. With a well-implemented parallel GSVM-RAND on a computing cluster with 100 CPUs, we can build such an ensemble classifier in several minutes. This is proved to be critical to meet the business requirement for spam filtering.

AUC values alone cannot justify the effectiveness of GSVM-RAND. In our real spam sender detection production system, a classifier with $FP\_rate > 1\%$ is not acceptable. Fig. 3 depicts ROC curves at the cut of $FP\_rate \leq 1\%$. Once again, these curves show that GSVM-RAND is comparable to traditional SVM modeling in terms of effectiveness. Fig. 3 also compare different aggregation methods. Obviously Decision Aggregation is not suitable. Bayesian Sum Rule is slightly better than Major Voting.

Table 3 gives the effectiveness of local SVMs under GSVM-RAND with Bayes Sum Rule for aggregation. The AUC value is increased from averagely 0.84106 for one local SVM to 0.98762 after aggregation. Fig. 4 reports the effect of the number of granules on AUC. The classification effectiveness converges to the optimum with more local SVMs joining on ensembling.

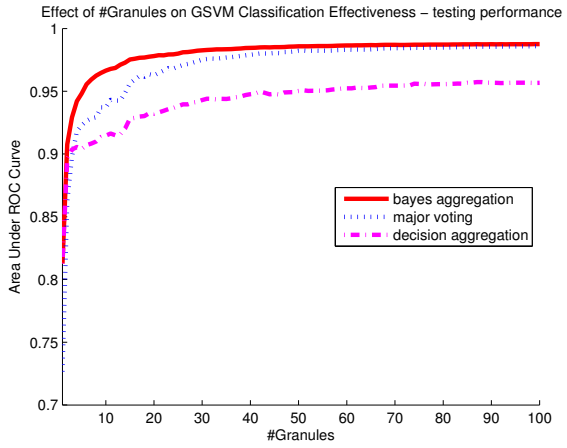

Figure 4. Effect of #granules - testing

5 Conclusion

Spam sender detection based on email subject data is a complex large-scale text mining task. A fast and accurate classifier is usually desirable in such an application. Support Vector Machine (SVM) is well-known as the state-of-the-art classifier for text mining in terms of accuracy. However, SVM modeling is computationally expensive, typically super-quadratic to the number of samples and linear to the number of features. And hence it cannot be comfortably applied onto large email subject data. In this work, a highly scalable SVM modeling method, named *Granular SVM with Random granulation (GSVM-RAND)*, is designed. GSVM-RAND applies bootstrapping to extract a number of subsets of samples from the original training dataset. Each training subset is then projected into a feature subspace randomly selected from the original feature space. Here we call a *granule* such a subset of samples in such a feature subspace. This random granulation process is conducted in such a way that encourages diversity among different granules. One local SVM is then modeled in each granule. For a new sample, it is firstly projected into each granule in which the local SVM is fired to make a prediction. After that, all SVM predictions are aggregated by Bayesian Sum Rule for a final decision. GSVM-RAND is easy to be parallelized and hence efficient and highly scalable. GSVM-RAND is also effective by integrating a large number of weak, low-correlated local SVMs. The experiment shows that GSVM-RAND can significantly speed up classification modeling with similar classification accuracy, compared to one single SVM on the whole dataset.

References

[1] Secure Computing Corporation, TrustedSource website, http://www.trustedsource.org.

[2] D. Alperovitch, P. Judge, and S. Krasser. Taxonomy of email reputation systems. In *Proc. of the First International Workshop on Trust and Reputation Management in Massively Distributed Computing Systems (TRAM)*, 2007.

[3] A. Bargiela and W. Pedrycz. *Granular Computing: An Introduction*. Kluwer Academic Pub, Kluwer, 2002.

[4] A. P. Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, 1997.

[5] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[6] R. Collobert, S. Bengio, and Y. Bengio. A parallel mixture of svms for very large scale problems. *Neural Computation*, 14(5):1105–1114, 2002.

[7] H. C. Kim, S. Pang, H.-M. Je, D. Kim, and S. Y. Bang. Support vector machine ensemble with bagging. In *Proc. of the First International Workshop on Pattern Recognition with Support Vector Machines*, pages 397–407, 2002.

[8] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239, 1998.

[9] H.-T. Lin, C.-J. Lin, and R. Weng. A note on platts probabilistic outputs for support vector machines. *Machine Learning*, 68(3):267–276, 2007.

[10] T. Y. Lin. Data mining and machine oriented modeling: A granular computing approach. *Applied Intelligence*, 13(2):113–124, 2000.

[11] J. Platt. Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In *Proc. of Advances in Large Margin Classifiers*, pages 61–74, 2000.

[12] A. Ramachandran, N. Feamster, and S. Vempala. Filtering spam with behavioral blacklisting. In *Proc. of the 14th ACM Conference on Computer and Communications Security (CCS)*, 2007.

[13] Y. C. Tang, B. Jin, and Y.-Q. Zhang. Granular support vector machines with association rules mining for protein homology prediction. *Artificial Intelligence in Medicine*, 35(1-2):121–134, 2005.

[14] Y. C. Tang, S. Krasser, P. Judge, and Y.-Q. Zhang. Fast and effective spam IP detection with granular SVM for spam filtering on highly imbalanced spectral mail server behavior data. In *Proc. of The 2nd International Conference on Collaborative Computing*, 2006.

[15] D. Tao, X. Tang, X. Li, and X. Wu. Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(7):1088–1099, 2006.

[16] G. Valentini and T. G. Dietterich. Low bias bagged support vector machines. In *ICML 2003*, pages 752–759, 2003.

[17] V. N. Vapnik. *Statistical Learning Theory*. John Wiley and Sons, New York, 1998.

[18] G. Yan, G. Ma, and L. Zhu. Support vector machines ensemble based on fuzzy integral for classification. In *ISNN 2006*, pages 974–980, 2006.