Proceedings of International Joint Conference on Neural Networks, Atlanta, Georgia, USA, June 14-19, 2009

Concentration Based Feature Construction Approach for Spam Detection

Ying Tan, Chao Deng and Guangchen Ruan

Abstract— **Inspired by human immune system, a concentration based feature construction (CFC) approach which utilizes a two-element concentration vector as the feature vector is proposed for spam detection in this paper. In the CFC approach, 'self' and 'non-self' concentrations are constructed by using 'self' and 'non-self' gene libraries, respectively, and subsequently are used to form a vector with two elements of concentrations for characterizing the e-mail efficiently. As a result, the design of classifier actually amounts to establishing a mapping between two real-value inputs and one binary output. The classification of the e-mail is considered as an optimization problem aiming at minimizing a formulated cost function. A clonal particle swarm optimization (CPSO) algorithm proposed by the leading author is also employed for this purpose. Several classifiers including linear discriminant, multi-layer neural networks and support vector machine are used to verify the effectiveness and robustness of the CFC approach. Experimental results demonstrate that the proposed CFC approach not only has a very much fast speed but also gives 97% and 99% of accuracy just using a two-element concentration feature vector on corpus PU1 and Ling, respectively.**

I. Introduction

SPAM has been considered as an increasingly serious problem to the infrastructure of Internet. According to the statistics from ITU (International Telecommunication Union), about 70% to 80% of the present emails in Internet are spam. Numerous spam not only occupies valuable communications bandwidth and storage space, but also threatens the network security when it is used as a carrier of viruses and malicious codes. Meanwhile, spam wastes much user's time to tackle with them, so decreases the productivity considerably.

Many classification algorithms have been put into practice for solving spam problems so far, which include Naïve Bayes [1], [2], Support Vector Machine (SVM) [3], [4], Artificial Neural Network (ANN) [5], [6], Artificial Immune System (AIS) [7], [8], [9], DNA Computing [10], and hybrid approaches [11], [12]. Approaches for feature extraction of email include simple approaches [8], term frequency analysis approaches [1], [3], [5] and heuristic approaches [7], [10], [13].

In this paper, inspired from human immune system (HIS), a Concentration-based Feature Construction (CFC) approach which utilizes a two-element concentration vector as the feature vector is proposed for spam detection. Both 'self' gene library and 'non-self' gene library, which contain the words with utmost representative of non-spam mail and spam mail, respectively, are generated for feature construction. 'Self' concentration and 'non-self' concentration are constructed by using 'self' gene library and 'non-self' gene library, respectively. Then they are used to form a two-element concentration vector to characterize the e-mail. Unlike traditional anti-spam methods, classification of e-mail is here considered as an optimization problem aiming at optimization of the formulated cost function. Optimal self concentration and non-self concentration is obtained as the one whose cost function associated to the classification is minimum. A clonal particle swarm optimization (CPSO) algorithm is used for the optimization process.

Comprehensive experiments are conducted on two public benchmark corpora PU1 and Ling. Comparisons on performance among different classifiers including linear discriminant, multi-layer or back-propagation neural networks (BPNN) and support vector machine (SVM), are made in accuracy, precision, recall and miss rate. Experimental results show that the proposed CFC approach achieves 97% and 99% of accuracy on corpus PU1 and Ling, respectively, by just using a two-element concentration feature vector, so outperforms current approaches.

The remainder of this paper is organized as follows. Algorithmic implementations of our proposed approach are elaborated in details in Section II. CPSO-based optimization for 'self' and 'non-self' gene library pair is presented in Section III. Several experimental results on two benchmark corpora are reported in Section IV. Finally, concluding remarks are given in Section V.

II. Concentration Based Feature Construction Approach

A. Overview of Our Proposed Approach

Our proposed approach can be mainly divided into three parts. (1) Generate 'self' gene library and 'non-self' gene library from training samples. (2) Construct the concentration vector of each training sample through the two gene libraries, then these concentration vectors are used as the input of a successive classification algorithm for training. (3) The trained classifier is used to predict the label of testing samples characterized by concentration vectors. Here we focus on the feature construction and do not emphasize the classification algorithms which would be SVM, neural networks, decision tree, Adaboost, just to name a few.

Authors are with the Key Laboratory of Machine Perception and Intelligence (MOE), Peking University, and Department of Machine Intelligence, School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, Haidian District, P. R. China.

This work was supported by the National High Technology Research and Development Program of China (863 Program), with grant number 2007AA01Z453, and partially supported by National Natural Science Foundation of China (NSFC), under grant number 60673020 and 60875080.

978-1-4244-3553-1/09/$25.00 ©2009 IEEE

B. Generation of gene libraries

Two gene libraries—*'self' gene library* and *'non-self' gene library* are generated from training samples in our proposed approach. The gene fragment in gene library is simply a word. *'Self' gene library* are composed of words with utmost representative of non-spam e-mails. On the contrary, *'non-self' gene library* contains those words with utmost representative of spam e-mails. Intuitively, a word that appears most in non-spam e-mails while seldom in spam e-mails is a good representative of non-spam e-mails. Consequently, the difference of its frequency in non-spam e-mails minus that in spam e-mails can be used to reflect a word's "proclivity". After calculating the difference of the frequency of each word, words are sorted in order of their differences. Consider the queue of words is sorted in descendent order, for instance, then two portions of words derived from the front and the rear of the queue with certain proportion can be used to construct *'self gene' library* and *'non-self gene' library*, respectively. Pre-processing is used to select candidate words. According to [14], the features that appear in most of the documents are not relevant to separate these documents because all the classes have instances that contain those features. For simplification, we discard the words that appear more than 95% in all messages of the corpus. In addition, stop word list is also used to remove those trivial words. The generation of gene libraries is described in Algorithm 1, in which P_S and P_N are parameters to be adjusted.

Algorithm 1 Generation of Gene Libraries

Use stop word list to exclude those trivial words in training set.
Drop the word whose frequency is more than 95% in training set.

for each remaining word **do**
  Calculate its frequency appearing in non-spam (denoted by f_s) and spam (denoted by f_n), respectively.
end for

for each remaining word **do**
  Calculate its "proclivity" via following formula: $f_d = f_s - f_n$.
end for

Sort the words in terms of f_d, in descendent order.

Extract $P_S\%$ of words in front of the queue to form *'self' gene library* and $P_N\%$ of words in rear of the queue to form *'non-self' gene library*, respectively.

C. Feature Construction

The *concentration* of an e-mail is defined as the proportion of the number of words of the e-mail which appear in *gene library* to the number of different words in the e-mail, which can be formulated as

$$c = \frac{N}{W} \tag{1}$$

where c denotes the concentration, N is the number of words appearing in both e-mail and *gene library*, W is the number of different words in the e-mail.

Note that the *gene library* can be either the *'self' gene library* or the *'non-self' gene library*. Therefore for an e-mail to be classified, a *'self' concentration* which describes its similarity to non-spam and a *'non-self' concentration* which describes its similarity to spam can be constructed, respectively. When parameters P_S and P_N take different values, different *'self' gene libraries* and *'non-self' gene libraries* are obtained. Consequently, different *concentrations* can be constructed. Theses *concentrations* are used to form a feature vector which is served as the input of a successive classification algorithm.

D. Complexity Analysis

In algorithm 1, the time complexity of sorting m words is

$$O(m \log m) \tag{2}$$

where m is the number of candidate words after preprocessing and this process only carries out once during training stage. During running stage, according to Eq. 1, the time complexity of constructing self concentration and non-self concentration is

$$O(n_s * n_m + n_n * n_m) \tag{3}$$

where n_s and n_n are the number of words in 'self' gene library and 'non-self' gene library, respectively. n_m is the number of words in the e-mail to be classified. As n_m is usually at the scale of constant, Eq. 3 can be further expressed as

$$O(n_s + n_n) \tag{4}$$

As $n_s + n_n < m$, the time complexity for constructing a two-element feature vector for an e-mail is at most $O(m)$.

III. CPSO-based Concentration Design

The generation of 'self' gene library and 'non-self' gene library, which in turn uniquely determine the self concentration and non-self concentration that form the two-element concentration vector, is here considered as an optimization problem. The optimal vector $P^* = \{P_S^*, P_N^*, P_1^*, P_2^*, \cdots, P_m^*\}$, composed of **gene library determinants** P_S^* and P_N^* as well as parameters $P_1^*, P_2^*, \cdots, P_m^*$ associated with a certain classifier, is the one whose cost function $CF(P)$ associated with classification is minimum, with

$$CF(P) = Err(P) \tag{5}$$

where $Err(P)$ is the classification error measured by 10-fold cross validation on the training set. Input vector P consists of two parts—gene library determinants P_S^* and P_N^*, and parameters $P_1^*, P_2^*, \cdots, P_m^*$ associated with a certain

classifier. Gene library determinants uniquely determine the construction of gene libraries, which in turn determine the two-element concentration vector used to represent the email. Therefore the part of gene library determinants correspond to the performance of feature construction. The other part of P—$P_1^*, P_2^*, \cdots, P_m^*$, is classifier-related parameters which influence the performance of a certain classifier. Different classifiers hold different parameters and lead to different performance. Parameters associated with neural network, which determine the structure of the network, include number of layers, number of nodes within a layer and each connection weight between two nodes. SVM-related parameters that determine the position of optimal hyperplane in feature space, include cost parameter C and kernel parameters, just to name a few. The vector P is the optimization objective whose performance is measured by $CF(P)$. Therefore, the optimization of concentrations can be formulated as follows.

Finding $P^* = \{P_S^*, P_N^*, P_1^*, P_2^*, \cdots, P_m^*\}$ so that

$$CF(P^*) = \min_{\{P_S, P_N, P_1, P_2, \cdots, P_m\}} CF(P). \quad (6)$$

Several optimization approaches not demanding an analytical expression of the objective function such as particle swarm optimization (PSO), genetic algorithms (GA) and so forth can be employed for the optimization process.

Figure 1 shows the optimization process when using CPSO to design concentrations. For detailed processes of clone, mutation and selection, please refer to literature [15].

IV. Experiments

A. Experimental Setup

Two corpora used to test our proposed CFC approaches are the PU1 corpus [1] and Ling corpus[1] [16]. PU1 corpus consists of 1,099 messages, with spam rate 43.77%, Ling corpus consists of 2,893 messages, with spam rate 16.63%. Each corpus is divided into ten partitions with approximately equal amount of messages and spam rate. The version with stop-word removal in used in our experiments.

All experiments are conducted on a PC with CPU of AMD Athlon 3200+ and 448M RAM. Accuracy, precision, recall and miss rate are used as performance indices. Linear discriminant, BPNN and SVM are employed to verify the effectiveness and robustness of the proposed CFC approach. LIBSVM software package is used for the SVM [17]. BP network and linear discriminant are implemented by the toolbox of MATLAB of version R2007a.

B. Experimental Results

1) Experiments on Different Concentrations: In this part, different *'self' concentrations* and *'non-self' concentrations*, which correspond to *'self' gene libraries* and *'non-self' gene libraries* with different P_S and P_N are tested, aiming to find the *concentrations* with best performance. The tested P_S and P_N range from 5% to 50% at a step size 5%. 10-fold cross validation is used to measure the performance. That is, in

[1]The PU1 corpus and Ling corpus may be downloaded from http://www.iit.demokritos.gr/skel/iconfig/

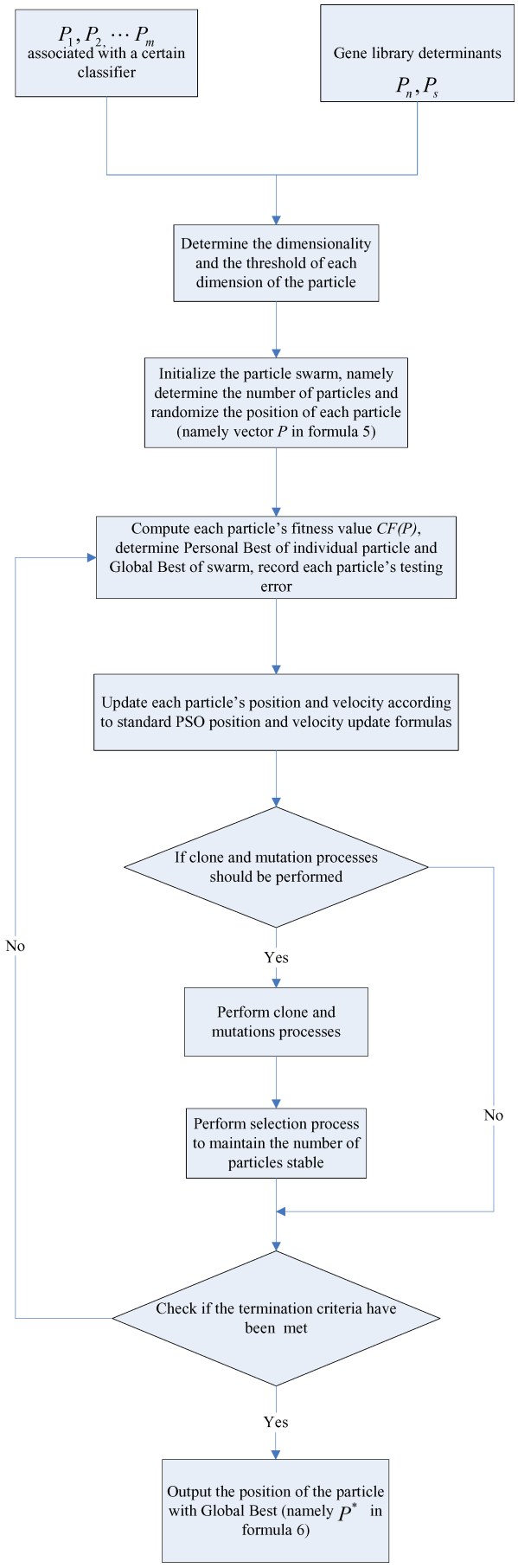


Fig. 1. Optimization process of concentration design.

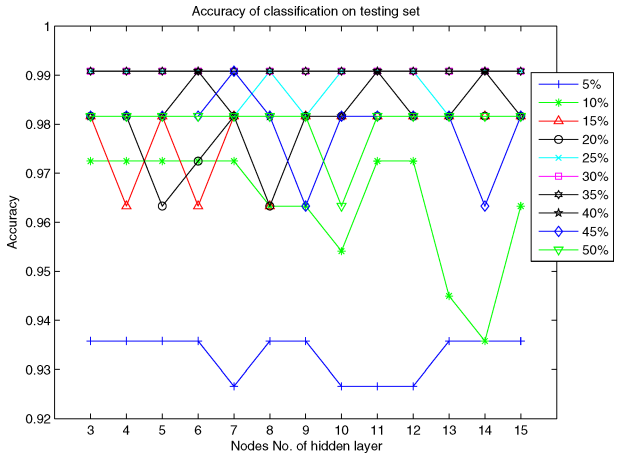
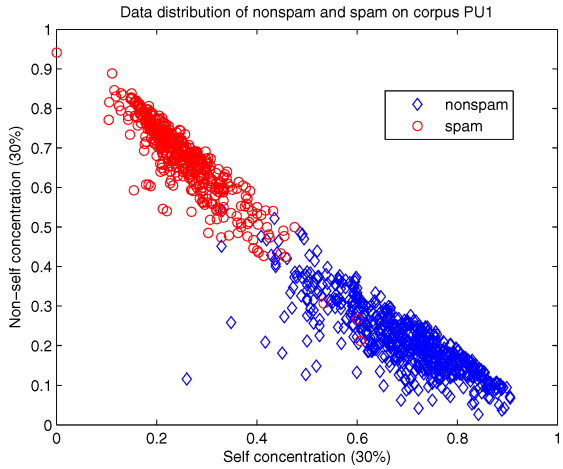


Fig. 2. Accuracy with different self concentrations on corpus PU1, leaving partition 1 as testing set.

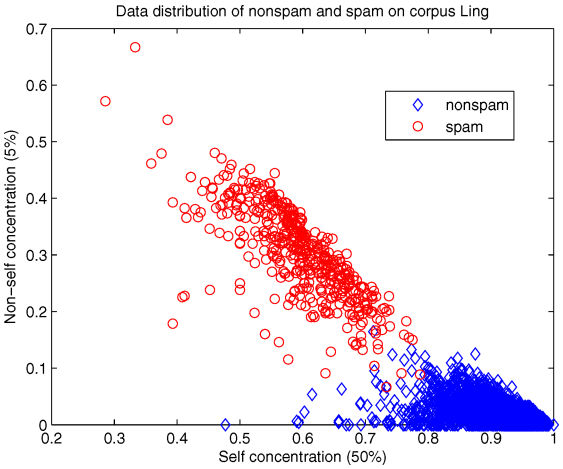
each iteration, 90% data are used for training while the rest 10% are used for its test.

A three layer BP network is used as the classifier. The number of nodes of input layer equals to the size of *concentration vector*. In this scenario, the *concentration vector* is unary. The number of nodes of hidden layer ranging from 3 to 15 is tested. There is only one node in output layer, output 1 indicates non-spam e-mail and 0 is for spam e-mail. The transfer functions of hidden layer and output layer are 'tansig' and 'purelin', respectively. The training function is 'trainlm'. Performance function is MSE. The network is trained for a maximum of 50 epochs to 0.01 of error goal. Figure 2 shows when leaving partition 1, the accuracy on corpus PU1 with different *'self' concentrations*. The performance measured by 10-fold cross validation shows that *'self' concentration* with $P_S = 30\%$ and *'non-self' concentrations* with $P_N = 30\%$ perform best on corpus PU1 ,respectively. For corpus Ling, the best performance is achieved with *'self' concentration* with $P_S = 50\%$ and *'non-self' concentration* with $P_N = 5\%$, respectively. Figure 3 shows the data distribution of non-spam and spam in feature space on corpora PU1 and Ling after feature construction with two-element *concentration vector* (That is, *'Self' concentration* with $P_S = 30\%$ and *'non-self' concentration* with $P_N = 30\%$ are used to form a two-element *concentration vector* to characterize each e-mail of corpus PU1. For corpus Ling, two-element *concentration vector* is composed of *'self' concentration* with $P_S = 50\%$ and *'non-self' concentrations* with $P_N = 5\%$.).

2) Experiments wih CPSO optimization: In this part, the selection of P_S and P_N as well as parameters of a certain classifier is considered as a dynamic optimization process carried out by CPSO. The cost function formulated by Eq. 5 is used as the objective function for optimization, therefore the fitness value of each particle is the classification error measured by 10-fold cross validation on the training set. The lower the classification error is, the better the fitness



(a) For corpus PU1



(b) For corpus Ling

Fig. 3. Data distributions of non-spam and spam in feature space for corpus PU1 (a) and Ling (b)

evaluation is, and vice versa. As the optimization process is somewhat time consuming, the scale of the data set in experiments is 10% of the original corpus. That is, in each independent test one partition of the corpus is used as data set, 90% of which is used for training and the rest 10% is used for testing. Linear discriminant, SVM (with linear kernel and RBF kernel) and BP neural network are used as classification algorithm.

Experiments of empirical two-element concentration vector (namely, $P_S = 30\%$ and $P_N = 30\%$ for corpus PU1, $P_S = 50\%$ and $P_N = 5\%$ for corpus Ling) are conducted on the same data set for comparison and following parameters are adopted. According to [3], the performance of SVM is remarkably independent of the choice of C as long as C is large (over 50). So the parameter C of SVM is set to be 100 for both SVM with linear kernel and SVM with RBF kernel in our experiments. In the initial tentative experiments, a range of parameter γ for RBF kernel are tested, and the performance is not sensitive to the variation of parameter γ.

TABLE I

PERFORMANCES OF LINEAR DISCRIMINANT, SVM, BP NEURAL NETWORK ON CORPUS PU1. ACCURACY, PRECISION, RECALL AND MISS RATE ARE THE AVERAGE PERFORMANCE OF 10 INDEPENDENT TESTS WITH EACH TEST USING A DIFFERENT PARTITION. USING $P_S = 30\%$ AND $P_N = 30\%$, RESPECTIVELY.

Methods	Acc (%)	Pre (%)	Rec (%)	MR (%)
LD	95.45	95.74	93.75	3.23
SVM (Linear)	95.41	95.74	93.75	3.28
SVM (RBF)	96.36	**97.83**	93.75	1.64
BPNN	**96.53**	97.76	**93.95**	**1.37**

TABLE II

PERFORMANCES OF LINEAR DISCRIMINANT, SVM, BP NEURAL NETWORK ON CORPUS LINGSPAM. ACCURACY, PRECISION, RECALL AND MISS RATE ARE THE AVERAGE PERFORMANCE OF 10 INDEPENDENT TESTS WITH EACH TEST USING A DIFFERENT PARTITION. USING $P_S = 50\%$ AND $P_N = 5\%$, RESPECTIVELY.

Methods	Acc (%)	Pre (%)	Rec (%)	MR (%)
LD	97.58	97.76	87.5	**0.41**
SVM (Linear)	**98.96**	95.92	**97.92**	0.83
SVM (RBF)	98.62	95.83	95.83	0.81
BPNN	**98.96**	**97.87**	95.83	**0.41**

So let γ be 10 for SVM with RBF kernel here. For BP neural network, the performance difference with different number of nodes of hidden layer is inapparent. So let the number of nodes of hidden layer be 3.

In Eq. 6, P_S and P_N are optimized in the real number interval $[0, 0.5]$. $P_1, P_2, \cdots, P_m$ are classifier related. For linear discriminant, there are no parameters. For BPNN, the number of nodes of hidden layer is optimized in the integer number interval $[3, 15]$. For SVM with linear kernel, the cost parameter C is optimized in the real number interval $[1, 200]$. For SVM with RBF kernel, the cost parameter C and γ are optimized in the real number interval $[1, 200]$ and $[1, 20]$. respectively. The stop criterion, i.e. a maximum number of generations, is set to be 200 in this study. In addition, the number of particles in a swarm is set to be 20. 10 independent tests are conducted with each test using a different partition. During each optimization process, as the randomness of CPSO, the performance and obtained P_S and P_N vary slightly, therefore the average performance of 10 independent runs are used to evaluate each test. For comparison test of empirical concentration, each test is only conducted once. The average performances of empirical and optimized concentration designs on corpora PU1 and LingSpam are reported in Table I to Table IV. Acc, Pre, Rec, MR and LD are abbreviations for accuracy, precision, recall, miss rate and linear discriminant, respectively.

Table V and Table VI show the performances of Naïve Bayesian, Linger-V and SVM-IG on corpora PU1 and Ling reported in [1], [5], [16], [18]. Linger-V is a NN-based system for automatic e-mail classification. For Naïve Bayesian, the version of the corpus adopted in the experiments is the original version, for Linger-V and SVM-IG, it is the stemming version. All these results are obtained by using 10-fold validation.

For Naïve Bayesian, 50 words with the highest mutual information scores are selected. LINGER-V and SVM-IG uses variance (V) and information gain (IG) as feature selection criteria respectively and the best scoring 256 features are chosen. It turns out that our CFC approaches are superiors to current approaches even if only a two-element concentration feature vector is employed.

TABLE V

PERFORMANCES OF NAÏVE BAYESIAN (NB) (50 FEATURES), LINGER-V (256 FEATURES) AND SVM-IG (256 FEATURES) ON CORPUS PU1, USING 10-FOLD CROSS VALIDATION

Methods	Acc (%)	Pre (%)	Rec (%)	MR (%)
NB	91.076	95.11	83.98	3.4
Linger-V	93.45	96.46	88.36	2.588
SVM-IG	93.18	95.7	88.4	3.1

V. CONCLUDING REMARKS

Instead of obtaining the approximately optimal concentrations in terms of empirical tentativeness, we establish an uniform framework for a general and systematical approach of feature construction. A cost function which measures the performance of classification of emails is formulated. Consequently, by minimizing this cost function by the CPSO, the optimal concentrations are obtained. Several classifier including linear discriminant, back propagation neural networks (BPNN), support vector machine (SVM) are employed to verify the effectiveness as well as robustness of the proposed feature construction approach. Comparisons of performances between concentration construction by empirical tentativeness and by optimization are conducted on different classifiers. Experimental results demonstrate that the performance of optimization based CFC approach outperforms that of the CFC approach by empirical tentativeness.

VI. ACKNOWLEDGEMENT

This work was supported by the National High Technology Research and Development Program of China (863 Program), with grant number 2007AA01Z453, and partially supported by National Natural Science Foundation of China (NSFC), under grant number 60673020 and 60875080. This work was

TABLE VI

PERFORMANCES OF NAÏVE BAYESIAN (NB) (50 FEATURES), LINGER-V (256 FEATURES) AND SVM-IG (256 FEATURES) ON CORPUS LING, USING 10-FOLD CROSS VALIDATION

Methods	Acc (%)	Pre (%)	Rec (%)	MR (%)
NB	96.408	96.85	81.10	0.539
Linger-V	98.2	95.62	93.56	0.875
SVM-IG	96.85	99	81.9	0.17

TABLE III

PERFORMANCES OF LINEAR DISCRIMINANT, SVM, BP NEURAL NETWORK ON CORPUS PU1. ACCURACY, PRECISION, RECALL AND MISS RATE ARE THE AVERAGE PERFORMANCE OF 10 INDEPENDENT TESTS WITH EACH TEST USING A DIFFERENT PARTITION. EACH TEST IS EVALUATED BY 10 INDEPENDENT RUNS. SELF CONCENTRATION AND NON-SELF CONCENTRATION ARE THE AVERAGE OPTIMAL CONCENTRATION DERIVED BY CPSO OPTIMIZATION.

Methods	**Acc** (%)	**Prec** (%)	**Rec** (%)	**MR** (%)	P_N (%)	P_S (%)
Linear Discriminant	97.27	97.87	95.83	1.64	28.37	27.83
SVM (Linear)	98.16	97.92	97.92	1.64	28.62	27.47
SVM (RBF)	98.18	97.92	97.92	**1.61**	28.68	27. 53
BPNN	**98.69**	**98.39**	**98.61**	**1.61**	28.56	27.59

TABLE IV

PERFORMANCES OF LINEAR DISCRIMINANT, SVM, BP NEURAL NETWORK ON CORPUS LING. ACCURACY, PRECISION, RECALL AND MISS RATE ARE THE AVERAGE PERFORMANCE OF 10 INDEPENDENT TESTS WITH EACH TEST USING A DIFFERENT PARTITION. EACH TEST IS EVALUATED BY 10 INDEPENDENT RUNS. SELF CONCENTRATION AND NON-SELF CONCENTRATION ARE THE AVERAGE OPTIMAL CONCENTRATION DERIVED BY CPSO OPTIMIZATION.

Methods	**Acc** (%)	**Prec** (%)	**Rec** (%)	**MR** (%)	P_N (%)	P_S (%)
Linear Discriminant	98.96	97.87	95.83	0.41	47.58	4.73
SVM (Linear)	99.65	**98.96**	99.02	0.45	48.31	4.58
SVM (RBF)	99.53	98.74	**99.32**	0.52	48.14	4.65
BPNN	**99.75**	98.89	98.81	**0.21**	47.83	4.51

also in part financially supported by the Research Fund for the Doctoral Program of Higher Education (RFDP) in China.

REFERENCES

[1] I. Androutsopoulos, J. Koutsias, K. V. Chandrinos, and C. D. Spyropoulos, "An experimental comparison of naive bayesian and keyword-based anti-spam filtering with personal e-mail messages," in *Proc. of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2000, pp. 160–167.

[2] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz, "A bayesian approach to filtering junk e-mail," in *AAAI Workshop on Learning for Text Categorization*, 1998, pp. 55–62.

[3] H. Drucker, D. Wu, and V. N. Vapnik, "Support vector machines for spam categorization," *IEEE Trans. Neural Netw.*, vol. 10, pp. 1048–1054, September 1999.

[4] Y. Tan and J. Wang, "A support vector network with hybrid kernel and minimal vapnik-chervonenkis dimension," *IEEE Trans. Knowl. Data Eng.*, vol. 26, pp. 385–395, April 2004.

[5] J. Clark, I. Koprinska, and J. Poon, "A neural network based approach to automated e-mail classification," in *Proc. IEEE International Conference on Web Intelligence (WI'03)*, Halifax, Canada, 2003, pp. 702–705.

[6] I. Stuart, S.-H. Cha, and C. Tappert, "A neural network classifier for junk e-mail," *Lecture Notes in Computer Science*, pp. 442–450, 2004.

[7] T. Oda and T. White, "Increasing the accuracy of a spam-detecting artificial immune system," in *Proc. IEEE Congress on Evolutionary Computation (CEC'03)*, vol. 1, Canberra, Australia, December 2003, pp. 390–396.

[8] A. Secker, A. A. Freitas, and J. Timmis, "AISEC: An artificial immune system for email classification," in *Proc. IEEE Congress on Evolutionary Computation (CEC'03)*, vol. 1, Canberra, Australia, December 2003, pp. 131–139.

[9] Y. Tan, "Multiple-point bit mutation method of detector generation for snsd model," *Lecture Notes in Computer Science 3973*, pp. 340–345, 2006.

[10] I. Rigoutsos and T. Huynh, "Chung-kwei: a pattern-discovery-based system for the automatic identification of unsolicited e-mail messages(spam)," in *Proc. of the first Conference on Email and AntiSpam (CEAS'04)*, Mountain View, CA, July 2004.

[11] B. Leiba and N. Borenstein, "A multifaceted approach to spam reduction," in *Proc. of the first Conference on Email and AntiSpam (CEAS'04)*, Mountain View, CA, July 2004.

[12] M.-W. Wu, Y. Huang, S.-K. Lu, I.-Y. Chen, and S.-Y. Kuo, "A multifaceted approach towards spam-resistible mail," in *Proc. IEEE Pacific Rim International Symposium on Dependable Computing*, December 2005, pp. 208–218.

[13] C.-Y. Yeh, C.-H. Wu, and S.-H. Doong, "Effective spam classification based on meta-heuristics," in *Proc. IEEE International Conference on Systems, Man and Cybernetics*, vol. 4, October 2005, pp. 3872–3877.

[14] M. H. Zuchini, "Aplições de mapas auto-organizáveis emmineração de dados e recuperação de informação," Master's thesis, UNICAMP, 2003.

[15] Y. Tan and Z. Xiao, "Clonal particle swarm optimization and its applications," in *Proc. IEEE Congress on Evolutionary Computation (CEC'07)*, Singapore, 2007, pp. 2303–2309.

[16] I. Androutsopoulos, J. Koutsias, K. V. Chandrinos, G. Paliouras, and C. D. Spyropoulos, "An evaluation of naive bayesian anti-spam filtering," in *Proc. European Conference on Machine Learning (ECML'00)*, 2000.

[17] C.-C. Chang and C.-J. Lin, *LIBSVM: a Library for Support Vector Machines*, 2001. [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/libsvm

[18] I. Koprinska, J. Poon, J. Clark, and J. Chan, "Learning to classify e-mail," *Information Science*, May 2007.