Winter 12-1-2010

An Effective Ensemble Approach for Spam Classification

Jin Tian

Minqiang Li

Fuzan Chen

AN EFFECTIVE ENSEMBLE APPROACH FOR SPAM CLASSIFICATION

Jin Tian, Minqiang Li, Fuzan Chen
School of Management, Tianjin University, Tianjin 300072, P.R. China.
E-mail: jtian_tju@yahoo.com.cn

Abstract

The annoyance of spam increasingly plagues both individuals and organizations. Spam classification is an important issue to distinguish the spam with the legitimate email or address. This paper presents a neural network ensemble approach based on a specially designed cooperative coevolution paradigm. Each component network corresponds to a separate subpopulation and all subpopulations are evolved simultaneously. The ensemble performance and the *Q*-statistic diversity measure are adopted as the objectives, and the component networks are evaluated by using the multi-objective Pareto optimality measure. Experimental results illustrate that the proposed algorithm outperforms the traditional ensemble methods on the spam classification problems.

Key word: neural network ensemble, cooperative coevolution, spam classification, web service

1. INTRODUCTION

Over the last few years, spam has become a serious problem ubiquitously throughout the world. Spam is the use of electronic messaging systems to send unsolicited commercial messages indiscriminately. The most widely recognized form of spam is e-mail spam and further the term is applied to similar abuses in other media, such as web search engine spam, online classified ads spam, mobile phone messaging spam, *et al*. Spam is rapidly eroding the value of legitimate online marketing and is causing problems for both users and the Internet generally. Many web services for anti-spam have been developed by Internet organizations and e-commerce companies. For example, Amazon has afforded a web service application on the market for all types of users to filter the unwanted emails before they reach the users' computers or mobiles. Thus spam filtering becomes one of the essential issues to most companies, governments, or even individual users.

Generally, spam filtering can be regarded as a binary classification problem. The classifier must distinguish between the legitimate and the spam. Various spam classification approaches have been proposed. The commonly used machine learning-based techniques include decision trees[1], support vector machine (SVM) [2], Naive Bayes[3], and neural network (NN) [4]. Previous researches have shown that NN can achieve high classification accuracies, sometimes more accurate than those of the symbolic classifiers [4]. Xu and Yu have designed a spam filtering system using revised back propagation network (BPN) and automatic thesaurus construction [5]. Wu and Tsai have applied a BPN model for spam classification using spamming behaviors as features [6]. Gavrilis *et al*. have presented a hybrid method that combines NN and genetic algorithms (GA) for robust detection of spam [7]. Cárpinteiro *et al*. have proposed a multilayer perceptron (MLP) model to classify spam emails and non-spam emails [8]. Cui *et al*. have produced a model based on the NN to classify personal emails, and the use of principal component analysis (PCA) as a preprocessor of NN to reduce the data in terms of both dimensionality and size [9]. These studies show that NN can be successfully used for spam classification. However, practical applications are difficult to be satisfied because of the problems of slow learning and the likelihood of being trapped into a local minimum especially when the size of the network is large.

Recently, ensemble learning receives increasing attention in the machine learning community [10]. Ensemble is composed of a finite number of component networks. Early work on ensembles suggested that the consensus of an ensemble may outperform component networks, especially when the components are quite different [11]. Ying *et al*. have presented an ensemble approach applied to classify spam e-mails, which consisted of SVM, BPN and decision tree [12]. Wei *et al*. have proposed an ensemble approach that combines the predictions made by Positive Naïve Bayes and the rough classifier of Positive Example-Based Learning for the unlabeled examples [13]. A prominent current in ensemble study is the combination ensemble learning with the evolutionary computation. Yao has demonstrated the success of combination of neural network ensemble (NNE) and the evolutionary computation in improving a classifier's generalization [14]. Liu *et al*. have introduced mutual information to measure similarity between component networks, and a diverse population of component networks could be evolved by adjusting the fitness sharing with mutual information [15]. Nicolás G-P *et al*. have developed an approach to ensemble design by means of coevolutionary algorithm [16].

In this paper, we focus on the parallel executable ability of the coevolutionary algorithm, and propose a multi-population coevolutionary NNE classification method (referred as MCNNE) for spam classification. MCNNE attempts to obtain the NNE model by a specially designed cooperative coevolutionary algorithm (Co-CEA) based on the multi-population paradigm. Generally, the Co-CEA utilizes a divide-and-cooperative mechanism to evolve subpopulations with evolutionary algorithms in parallel [17], which can boost up the search process. Radial Basis Function Neural Network (RBFNN) is utilized as the component network of the NNE. RBFNN is the most popular among all the NN applications for complex classification tasks, due to a number of advantages compared with other types of NNs, such as better classification capabilities, simpler network structures, and faster learning algorithms. The subpopulations of the component network adopt matrix-form chromosomes by encoding parameters of RBFNN's topology (the network centers, the radius widths, control variables). Each component network corresponds to a subpopulation in the coevolution mechanism and all the subpopulations are evolved simultaneously. In this mechanism, the fitness of an individual from a particular subpopulation is assessed by associating it with representatives from other subpopulations. The ensemble performance and the *Q*-statistic diversity measure are adopted as the objectives and the component networks are evaluated by using the multi-objective Pareto optimality measure. The performance of the proposed algorithm is verified on two real-world spam classification datasets.

The rest of the paper is organized as follows: in Section 2, the proposed algorithm is presented in detail. Section 3 illustrates the new algorithm's performance on two spam classification datasets in comparison with other learning algorithms. Finally, Section 4 summarizes the key points of the paper and concludes with remarks for the future research.

2. Configuration of NNE with Multi-populations

The Co-CEA is particularly well-suited for the configuration of the NNE on complicated classification problems. The idea is that the ensemble learning may benefit from encoding the different component networks into separate subpopulations, which are evolved concurrently to find the NNE model. Bootstrap resampling is a good solution to reduce the computing time and is applied in the proposed algorithm to generate several training subsets from the original training data. The component networks in the ensemble are then trained with these data subsets. For each component network, a matrix-form mixed encoding method is designed to generate the subpopulation of the RBFNN structure. The output weights between the hidden layer and the output layer of the RBFNN are calculated directly by the pseudo-inverse method. The subpopulations are then evolved respectively and fitness of individuals are assigned cooperatively. Specific genetic operators are developed to produce offspring for all subpopulations. The MCNNE outputs the complete ensemble solution by integrating the best individuals from the subpopulations. The majority vote method is utilized to calculate the ensemble output combination.

2.1 Encoding

The RBFNN can be viewed as a three-layer feed-forward NN with multi-inputs, multi-outputs, and linear output mapping. The RBFNN topological structure illustrates exactly the relationship between the *m*-dimension input vector $\boldsymbol{x} \in \boldsymbol{R}^m$ and the *n*-dimension output vector $\boldsymbol{y} \in \boldsymbol{R}^n$: $f : \boldsymbol{x} \to \boldsymbol{y}$. The response of a hidden node is produced by the node activity through a radial basis function:

$$\varphi_j(\boldsymbol{x}) = \exp\left\{-\frac{\left\|\boldsymbol{x}-\boldsymbol{\mu}_j\right\|^2}{\sigma_j^2}\right\} \tag{1}$$

where $\boldsymbol{x} = (x_1, x_2, \ldots x_m)^T$ is the input vector, $\boldsymbol{\mu}_j$ is the center, and σ_j is the radius width of the j^{th} hidden node. The output layer is linear and the i^{th} output may be expressed as a linear combination of the *k* radial basis functions:

$$y_i = \sum_{j=1}^{Nc} w_{ji}\varphi_j(\boldsymbol{x}) \tag{2}$$

In MCNNE, the component networks are encoded identically in subpopulations. According to the characteristics of the RBFNN, the real-encoded genotype representation can make the searching of the solution space more precise and efficient. Thus, a matrix-form mixed encoding genotype representation is designed for subpopulations, where the RBFNN structure, i.e., the hidden node centers and the radius widths, is encoded as real-valued encoding matrices, and a control vector, a binary string, is attached to the matrix. In this article, an individual in one subpopulation represents one component network structure. Thus each subpopulation contains L individuals, and each individual $\boldsymbol{P}_t^l = [\boldsymbol{c}_t^l \quad \boldsymbol{\sigma}_t^l \quad \boldsymbol{b}_t^l]$ ($l = 1, 2, \ldots, L$, $t = 1, 2, \ldots, M$) is a matrix of size $Nc_t \times (m+2)$. M is the ensemble size, Nc_t is the initial number of the hidden nodes, and m is the dimension of the input samples. $\boldsymbol{c}_t^l = [\boldsymbol{c}_t^{li}]_{Nc_t \times m}$ and $\boldsymbol{\sigma}_t^l = [\sigma_t^{li}]_{Nc_t \times 1}$ are the centers and widths of the hidden nodes of the l^{th} individual respectively in the t^{th}

subpopulation; $\boldsymbol{b}_t^l = [b_t^{li}]_{Nc_t \times 1}$ is the control vector, where $b_t^{li} = 0$ means that the i^{th} hidden node of the l^{th} individual in the t^{th} subpopulation is invalid and is excluded in the design of the network structure; otherwise $b_t^{li} = 1$ denotes that it is a valid hidden node in the network structure, $i = 1, 2, \ldots, Nc_t$.

2.2 Initialization

Breiman showed that Bagging is effective on "unstable" learning algorithms where small changes in the training set result in large changes in predictions and claimed that NN and decision tree are examples of unstable learning algorithms [18]. Thus we apply the bootstrap resampling method to obtain different training subsets, which generate different component networks.

The initialization of the proposed algorithm is done in three steps. Firstly, the total data are divided into three sets: the training set, the validation set, and the testing set. *M* training subsets are obtained by the bootstrap resampling method in original training set. Secondly, the *M* initial component networks are generated by the decaying radius selection clustering (DRSC) method [19], which makes the coevolution process work more effectively than beginning with randomly generated hidden nodes. The initial values of radius widths are calculated with the clustering information of the sample distribution [20]. Finally, *L* individuals are generated based on one initial component network to form one subpopulation, and the control vectors in an individual are initialized as 0 or 1 randomly, which indicates that the corresponding hidden nodes are inactive or active.

2.3 Multiobjective Evaluation of Individuals

With the coevolution paradigm, each subpopulation must contribute an individual to construct the complete NNE structure $\boldsymbol{\Theta}$. The best individual in each subpopulation is chosen as the representative to compose the elite pool $\boldsymbol{\Theta}^* = \{\boldsymbol{P}_1^*, \boldsymbol{P}_2^*, \ldots, \boldsymbol{P}_M^*\}$, *M* is the ensemble size. The elites in the initial elite pool, $\boldsymbol{P}_1^{*0}, \boldsymbol{P}_2^{*0}, \ldots, \boldsymbol{P}_M^{*0}$, are the RBFNNs obtained by DRSC with the different training subsets.

Individuals in one subpopulation are assigned fitness values in conjunction with individuals from other subpopulations, or an individual is evaluated in the context of the ensemble. The fitness of individuals is evaluated as a multi-objective optimization task in this algorithm because it is difficult to weigh different objectives as using the aggregating approach. Two objectives are used in the proposed algorithm:

(1) Classification accuracy

The classification accuracy is usually used as the fitness evaluation objective. In the proposed algorithm, this objective measures the contribution of individuals in subpopulations, and is calculated by the ensemble combination output. The majority vote method is adopted here. The label of a certain sample is determined by the majority voting in the ensemble components.

The l^{th} individual in the t^{th} subpopulation, $\boldsymbol{P}_t^l$, gets its fitness by calculating the combination output of the estimated ensemble structure $\boldsymbol{\Theta}_t^l = \{\boldsymbol{P}_1^*, \ldots, \boldsymbol{P}_{t-1}^*, \boldsymbol{P}_t^l, \boldsymbol{P}_{t+1}^* \ldots, \boldsymbol{P}_M^*\}$. This objective is represented as the ratio of correctly classified samples in the total validation set by the $\boldsymbol{\Theta}_t^l$:

$$A_{re}(\boldsymbol{P}_t^l) = \frac{N_{rv}(\boldsymbol{\Theta}_t^l)}{N_v} \tag{3}$$

where $N_{rv}(\boldsymbol{\Theta}_t^l)$ is the number of samples classified correctly on the validation set by the estimated network with hidden node structure $\boldsymbol{\Theta}_t^l$ and N_v is the size of the validation set.

In experiments, the similar accuracy rates of different individual structures usually give rise to smaller selection pressure in the population. So the objective is modified to increase the selection pressure as:

$$f_1(\boldsymbol{P}_t^l) = \alpha(1-\alpha)^{I(\boldsymbol{P}_t^l)} \tag{4}$$

where $I(\boldsymbol{P}_t^l)$ is the rank order of $\boldsymbol{P}_t^l$ with inversely sorting based on prediction accuracy of all $\boldsymbol{P}_t^l (t = 1, \ldots, M)$. $\alpha \in (0,1)$ is a pre-designed real number ($\alpha = 0.4$ for default).

(2) Diversity measure

The performance of the NNE algorithm for classification problems mainly depends on both the classification accuracy and the diversity between the component networks. In MCNNE, we want to find a tradeoff between the two evaluation measures. The second objective aims to evaluate the diversity of the components and the Yule's Q statistic is adopted to assess the similarity of two component networks' outputs [21].

$$Q_{ij} = \frac{N^{11}N^{00} - N^{01}N^{10} + 1}{N^{11}N^{00} + N^{01}N^{10} + 1} \tag{5}$$

where N^{ab} is the number of instances in the data set, classified correctly ($a = 1$) or incorrectly ($a = 0$) by the network *i*, and correctly ($b = 1$) or incorrectly ($b = 0$) by the network *j*. *Q* varies between -1 and 1.

In MCNNE, computing the diversity of individual $\boldsymbol{P}_t^l$ is to measure the difference between $\boldsymbol{P}_t^l$ and the representatives in other subpopulations,

$\boldsymbol{P}_1^*,\ldots,\boldsymbol{P}_{t-1}^*,\boldsymbol{P}_{t+1}^*\ldots,\boldsymbol{P}_M^*$. Q_{tj}^l is denoted as the Q values that assess the diversity between $\boldsymbol{P}_t^l$ and the representative $\boldsymbol{P}_j^*$, $j=1,\ldots,M$. The average of these Qs is a explicit index that illuminate the diversity of $\boldsymbol{P}_t^l$:

$$\bar{Q}_t^l = \frac{\sum_{j=1,\ldots,M;\, j\neq t} Q_{tj}^l}{M-1} \tag{6}$$

In order to normalize this measure to vary from 0 to 1, the objective is modified as:

$$f_2(\boldsymbol{P}_t^l) = \frac{1-\bar{Q}_t^l}{2} \tag{7}$$

The multiobjective algorithm is adopted to evaluate the fitness of individuals [22]. Since the objectives are in conflicts with each other, there is usually not a solution which maximizes all objectives simultaneously. Multiobjective optimization with conflicting objectives aims to find a set of optimal solutions instead of one optimal solution.

2.4 Selection

The selection operation adopted in this article is based on the Pareto ranking, similar to that in the NSGAII [23]. The successive Pareto fronts are obtained and nondominated individuals are assigned an equal rank. The individuals in the first nondominated front get the rank 1. The individuals in other fronts carried on their ranks successively. The individuals of a nondominated front are assigned identical fitness. Then the crowding distances [23] of individuals in each front are computed and the tournament selection is utilized to select individuals for the next generation.

In addition, the elitist selection [24] is adopted so that the best solutions in all subpopulations survive definitely to the next generation, which will keep the optimal solutions once they are found during the whole coevolution process.

2.5 Crossover

The crossover operation explores the whole search space and aims to find the global optima. The uniform crossover is used to exchange information between two individuals that are picked randomly from the mating pool to produce offspring. The individuals that undergo the crossover operation are grouped into pairs, and for every pair a binary mask string with the same length as the individual is generated randomly. The genes at the positions in first parent are selected when the corresponding bits in the mask string are 1, and the genes at the positions in second one are selected when the corresponding bits are 0. Thus, one offspring is produced. The second offspring is produced similarly by repeating the process again but with the 0 and 1 being exchanged in the mask string. It should be noted that all bits of the control vectors should not be zeros for both offspring.

2.6 Mutation

A ratio, p_{ad}, to decide whether the mutation occurs in the control bit or the real number part, has been introduced to accommodate the special chromosome structure of individuals. Suppose that $\boldsymbol{P}_t^l = \left[\boldsymbol{p}_t^{l1}, \boldsymbol{p}_t^{l2}, \ldots, \boldsymbol{p}_t^{lNc_t}\right]^T$ and $\boldsymbol{p}_t^{li} = \left[\boldsymbol{c}_t^{li} \quad \sigma_t^{li} \quad b_t^{li}\right]$ $(i=1,2,\ldots,Nc_t)$. For a hidden node $\boldsymbol{c}_t^{li}$ in $\boldsymbol{P}_t^l$, a random number r_{ad} is generated. If $p_{ad} > r_{ad}$, the operation only inverts the control bit (if the original bit is 0, it is mutated to 1, and vice versa). If $p_{ad} \le r_{ad}$ and $b_{ti}^l = 1$, the mutation introduces variances to the real-valued genes:

$$\boldsymbol{c}_t^{li\prime} = \boldsymbol{c}_t^{li} + N(0,1)\times(\boldsymbol{c}_t^{*,i} - \boldsymbol{c}_t^{li}) \tag{8}$$

$$\sigma_t^{li\prime} = \sigma_t^{li} + N(0,1)\times(\sigma_t^{*,i} - \sigma_t^{li}) \tag{9}$$

where $\boldsymbol{c}_t^{li\prime}$ and $\sigma_t^{li\prime}$ are the new values, $\boldsymbol{c}_t^{li}$ and σ_t^{li} are the current values, $\boldsymbol{c}_t^{*,i}$ and $\sigma_t^{*,i}$ are the corresponding values in the elite pool. $N(0,1)$ is a random number which obeys the standard normal distribution.

Finally, the representatives in the elite pool are output as the final estimation of the ensemble model. Incompact-training and collaborate-evaluation are good characters of the proposed multi-population frame. They make the algorithm more suitable for the multi-agent scheme to deal with the massive data.

3. Experimental Studies

Experiments were conducted on two real-world datasets, the Spam E-mail dataset from the UCI Repository and the Webspam dataset sponsored by Yahoo Research, to evaluate the performance of the proposed method. The first dataset contains 4601 instances, in which 1813 instances are spam. The collection of spam e-mails came from postmaster and individuals who had filed spam while the collection of non-spam e-mails came from filed work and personal e-mails. Each instance has 57 attributes, most of which are percentages of particular words or characters frequently occuring in the e-mail and the sequence length of consecutive capital letters. The second dataset Webspam is a dataset about the Web search engine spam, which is a large collection of annotated spam/ nonspam hosts labeled by a group of volunteers. Web search engine spam is one form of spam and refers to a practice on the World Wide Web of modifying HTML pages to increase the chances of them being placed high on search engine relevancy lists. The dataset contains 6479

instances, in which 344 instances are spam, 5709 instances are legitimate and the remainders are undecided.

The experiment parameters used in the MCNNE algorithm were set as follows. The population size L was 50, the maximum generations G was 200, and the ensemble size M=15. The probability of crossover p_c was 0.8. The non-structure mutation rate p_m was 0.2, and the structure mutation rate p_{ad} was 0.6. For each dataset, 30 runs of the algorithms were performed.

3.1 Experiment 1

The experiments were carried out to compare the performance of the MCNNE against conventional classification algorithms, such as Naïve Bayes, C4.5, MLP, K-nearest neighbor (KNN), RBFNN and SVM.

Table 1 reports the average testing accuracies (*Acc*), the standard deviation (*Std*), the maximum (*Max*) and minimum (*Min*) values of the MCNNE and the compared algorithms on 30 runs. The *t*-test statistics were computed to compare the difference of the testing accuracies of the MCNNE with the other algorithms.

Table 1. Testing accuracies of MCNNE and the traditional classification methods

	MC-NNE	Bayes	C4.5	MLP	KNN	RBF	SVM
				Spambase			
Acc	0.9165	0.7983	0.9136	0.9131	0.7149	0.8198	0.9067
Std	0.0067	0.0093	0.0075	0.0105	0.0126	0.0241	0.0085
Max	0.9296	0.8158	0.9304	0.9313	0.7365	0.8870	0.9217
Min	0.9043	0.7819	0.8974	0.8904	0.6835	0.7748	0.8930
t-test	-	56.52	1.579	1.498	77.48	21.21	4.950
				Webspam			
Acc	0.9807	0.8938	0.9330	0.9633	0.9229	0.8858	0.8811
Std	0.0035	0.0026	0.0068	0.0112	0.0175	0.0079	0.0003
Max	0.9868	0.8999	0.9500	0.9837	0.9560	0.9115	0.8815
Min	0.9709	0.8901	0.9197	0.9446	0.8800	0.8806	0.8809
t-test	-	109.6	34.05	8.154	17.74	60.47	156.9

Table 1 illustrates that the proposed algorithm is able to produce spam classification models with both higher accuracies and lower standard deviations compared with other classification algorithms. The *t*-test values show that the MCNNE outperforms the other methods significantly with a statistical confidence level of 95% on Webspam dataset and outperforms most methods on Spambase dataset except C4.5 and MLP by the *t*-test.

In terms of spam classification, false positives (marking good mail as spam) are very undesirable. Thus the experimental results are evaluated by spam precision (*SP*), spam recall (*SR*) and accuracy, which are defined as [3]:

$$SP = \frac{N_{SS}}{N_{SS} + N_{LS}} \tag{10}$$

$$SR = \frac{N_{SS}}{N_{SS} + N_{SL}} \tag{11}$$

$$Acc = \frac{N_{SS} + N_{LL}}{N_S + N_L} \tag{12}$$

where N_{SS} and N_{LL} are the number of instances that have been correctly classified to the spam and legitimateness, respectively; N_{SL} and N_{LS} are the number of spam and legitimate instances that have been misclassified; N_S and N_L are the total number of spam and legitimate instances in the testing set.

Table 2 gives a comparison of the average spam precisions and the average spam recalls between the proposed algorithm and the other classification methods.

Table 2. Comparison of the average spam precision and the average spam recall

	MC-NNE	Bayes	C4.5	MLP	KNN	RBF	SVM
				Spambase			
Acc	0.9165	0.7983	0.9136	0.9131	0.7149	0.8198	0.9067
SP	0.9727	0.9606	0.9301	0.9093	0.9467	0.7398	0.8871
SR	0.9107	0.6721	0.938	0.9073	0.7269	0.8564	0.9523
				Webspam			
Acc	0.9807	0.8938	0.9330	0.9633	0.9229	0.8858	0.8811
SP	0.9997	0.9231	0.8372	0.9825	0.8780	0.9016	0.8852
SR	0.7214	0.1750	0.9401	0.7892	0.7882	0.9819	0.8975

Table 2 indicates that the NNE models trained by MCNNE obtain higher spam precisions compared with other methods. MLP achieves a similar classification performance with MCNNE but its spam precisions are much lower than MCNNE. Particularly, the spam precision of MCNNE is 0.9997 on the Webspam dataset, which illuminates that MCNNE can prevent the legitimate instances being misclassified effectively.

3.2 Experiment 2

Experiments were conducted to verify the performance of the proposed method and some conventional ensemble algorithms, such as AdaBoost (AB) [25], Bagging (BA) [26], Dagging (DA) [27], Ensemble-selection (ES) [28], LogitBoost (LB) [29]and MultiBoost (MB) [30]. Table 3 reports the average testing accuracies of the MCNNE and other ensemble algorithms. The *t*-test statistics were computed to compare the difference of the testing accuracies of MCNNE with the other ensemble algorithms.

Table 3. Testing accuracies of MCNNE and the compared ensemble algorithms

	MC-NNE	AB	BA	DA	ES	LB	MB
				Spambase			
Acc	0.9165	0.8897	0.8281	0.8385	0.9113	0.7835	0.8692
Std	0.0067	0.0115	0.0148	0.0079	0.0071	0.0245	0.0222
Max	0.9296	0.9093	0.8549	0.8542	0.9235	0.8620	0.9054
Min	0.9043	0.8665	0.8012	0.8229	0.8983	0.7399	0.8306
t-test	-	11.01	29.87	41.32	2.928	28.63	11.18
				Webspam			
Acc	0.9807	0.8861	0.8948	0.9416	0.9746	0.8843	0.9022
Std	0.0035	0.0078	0.0233	0.0019	0.0040	0.0113	0.0236
Max	0.9868	0.9060	0.9423	0.9437	0.9800	0.9414	0.9424
Min	0.9709	0.8765	0.8811	0.9351	0.9659	0.8774	0.8811
t-test	-	60.71	19.95	53.89	6.361	44.55	18.06

As shown in Table 3, the MCNNE outperforms the other ensemble methods and achieves the best or near to the best classification accuracies. The NNE models trained by MCNNE achieved statistically significant increases in the testing accuracy in two spam classification problems. Furthermore, the MCNNE has lower standard deviation (Std.) than other ensemble algorithms, which illuminates that the proposed algorithm is more robust regarding classification accuracy.

Table 4 gives the average spam precisions and the average spam recalls obtained by MCNNE and the compared ensemble methods.

Table 4. Comparison of the average spam precision and he average spam recall

	MC-NNE	AB	BA	DA	ES	LB	MB
				Spambase			
Acc	0.9165	0.8897	0.8281	0.8385	0.9113	0.7835	0.8692
SP	0.9727	0.9032	0.9324	0.8762	0.9380	0.9243	0.7940
SR	0.9107	0.9151	0.9441	0.9581	0.9412	0.9309	0.9808
				Webspam			
Acc	0.9807	0.8861	0.8948	0.9416	0.9746	0.8843	0.9022
SP	0.9997	0.8750	1	1	0.9923	1	0.8746
SR	0.7214	0.0513	0.0992	0.1173	0.3315	0.0221	0.0517

The compared ensemble algorithms have high spam precisions but low spam recalls on the Webspam dataset partly due to the small number of the spam instances compared with the bulk of legitimate instances in this dataset. It indicates that these ensemble algorithms have high abilities to classify the legitimate instances correctly but low abilities to distinguish the spam ones. Although the spam precisions are high in testing set, most of the spam passed through the filter. Nevertheless, MCNNE achieves both high spam precisions and considerable spam recalls on the two datasets.

Totally, the MCNNE performs competitively compared with other conventional algorithms. Figure 1 gives the statistical performance of all algorithms over 30 runs in the box plots on every dataset.

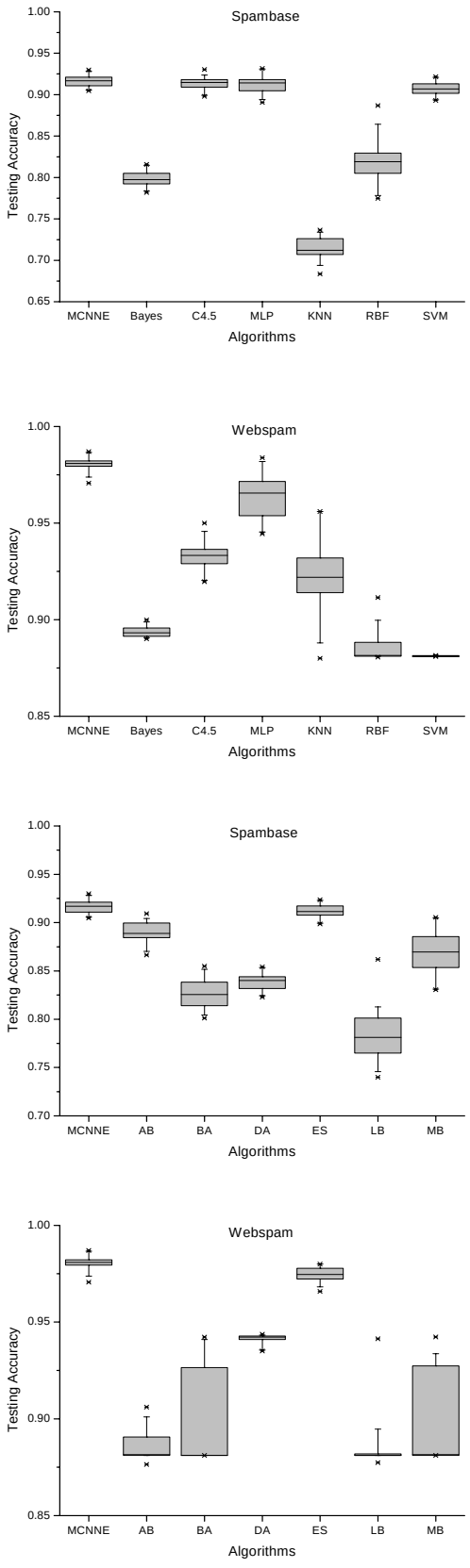


Figure 1 Statistic box plots for the testing accuracies of the algorithms over 30 run

3.3 Experiment 3

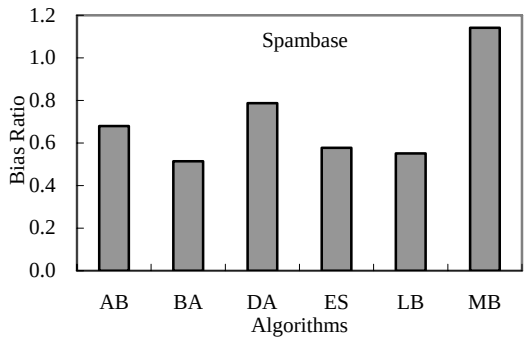
The bias-variance decomposition is often used in studying the performance of ensemble methods [31]. Originally, it was proposed for regression, but there are several variants for classification. In this section, we will study how the proposed algorithm behaves in a bias/variance decomposition test. Here we adopt the one proposed by Kohavi and Wolpert [32]. The bias and variance of the MCNNE and the compared ensemble approaches are shown in Table 5.

Table 5. The bias and variance of MCNNE and other ensemble algorithms

	MC	AB	BA	DA	ES	LB	MB
Spambase							
Bias	0.1161	0.0789	0.0597	0.0914	0.0670	0.0640	0.1325
Var.	0.0293	0.1148	0.0223	0.1204	0.0225	0.0960	0.1163
Ave	0.0727	0.0969	0.0410	0.1059	0.0448	0.0800	0.1244
Webspam							
Bias	0.0268	0.0319	0.0216	0.0153	0.0182	0.0229	0.0325
Var.	0.0128	0.0039	0.0215	0.0008	0.0237	0.0089	0.0031
Ave	0.0198	0.0179	0.0216	0.0081	0.0210	0.0159	0.0178

Note that since we care relative performance instead of absolute performance, the bias/variance of the ensemble algorithms has been normalized according to that of MCNNE and the average results of the relative bias/variance on the two datasets are shown in Figure 2. In other words, the bias/variance of MCNNE is regarded as 1.0, and the reported bias/variance of the ensemble algorithms is in fact the ratio against the bias/variance of the MCNNE.

Although MCNNE's ability of reducing the variance is not as good as those of some compared algorithms, such as ES and BA on Spambase dataset, it still can reduce the variance effectively. Moreover, MCNNE's ability of reducing the bias is better than AB and MB. This partially owes to its ability of significantly reducing both the bias and the variance.



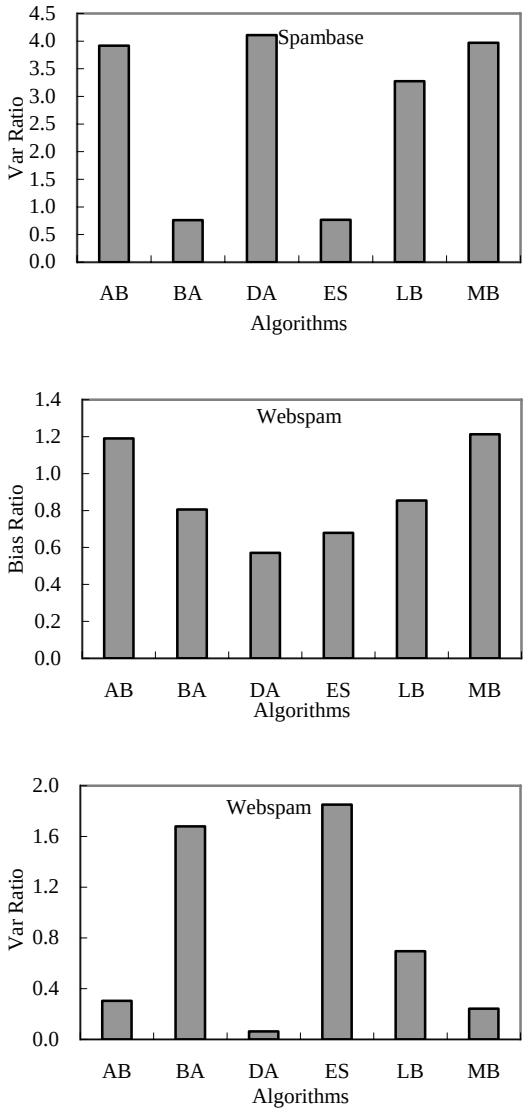


Figure 2 Comparison of the average relative bias and variance of the ensemble algorithms

4. Conclusion

This paper has presented an effective ensemble approach based on multi-population coevolution for spam classification. A component network of the ensemble in the proposed model corresponds to a separate subpopulation. The Co-CEA was introduced to realize the coevolution of the subpopulations in parallel. The RBFNN is employed as the component network. Experimental results illustrated that the spam classification performance of the proposed algorithm is superior to some traditional classification methods and ensemble algorithms on real-world spam datasets. And the proposed algorithm can also achieve high spam precisions and spam recalls. To sum up, the MCNNE is a quite competitive and powerful classification approach for the anti-spam problems in the web service.

The cooperative coevolution of multiple subpopulations provides a good paradigm to optimize the NNE model for complex classification problems. There are two issues to be addressed in the future research. One is the combination with the feature selection methodology to recognize the important features. The other is the introduction of new fitness measures to evaluate individuals in the paradigm of coevolutionary algorithms, which is good for keeping a more diversified population and making more explorative searching.

ACKNOWLEDGMENTS

This work was supported by grants from the Research Fund for the Doctoral Program of Higher Education of China (No. 20090032110065, No. 20090032120073) and the Self-innovation Research Fund of Tianjin University. The work was also supported by a grant from the General Program of the National Science Foundation of China (Grant No.70771074).

References

[1] Crawford, E., Kay, J., and McCreath, E. Automatic induction of rules for email classification, In: *Proceedings of the 6th Australasian document computing symposium*, Coffs Harbour, Australia, 2001, pp. 13–20.

[2] Amayri, O. and Bouguila, N. A study of spam filtering using support vector machines, *Artificial Intelligence Review*, 34(1), June 2010, pp. 73-108.

[3] Androutsopoulos, I., Koutsias, J., Chandrinos, K. V., *et al*. An evaluation of naive bayesian anti-spam filtering, In: *Proceedings of workshop on machine learning in the new information age*, Barcelona, 2000, pp. 9–17.

[4] Clark, J., Koprinska, I., and Poon, J. A neural network based approach to automated email classification, In: *Proceedings of IEEE/WIC international conference on web intelligence*, Halifax, Canada, 2003, pp. 702–705.

[5] Xu, H. and Yu, B. Automatic thesaurus construction for spam filtering using revised back propagation neural network, *Expert Systems with Applications*, 37(1), January 2010, pp. 18-23.

[6] Wu, C.-H. and Tsai, C.-H. Robust classification for spam filtering by back-propagation neural networks using behavior-based features, *Applied Intelligence*, 31(2), October 2009, pp. 107-121.

[7] Gavrilis, D., Tsoulos, I.G., and Dermatas, E. Neural recognition and genetic features selection for robust detection of e-mail spam, In: *Proceedings of the 4th Helenic conference on AI, Lecture notes in computer science*, 3955, 2006, pp. 498–501. Berlin: Springer.

[8] Carpinteiro, O.A.S., Lima, I., Assis, J.M.C., *et al*. A neural model in anti-spam systems, In: *ICANN, Lecture notes in computer science*, 4132, 2006, pp. 847–855. Berlin: Springer.

[9] Cui, B., Mondal, A., Shen, J., *et al*. On effective e-mail classification via neural networks, In: *Proceedings of the 16th international conference on database and expert systems applications (DEXA05)*, 2005, pp. 85–94.

[10] Sollich, P. and Kroph, A. Leaning with ensembles: How over-fitting can be useful, *Advances in Neural Information Processing Systems*, 8, 1996, pp. 190-196.

[11] Krogh, A. and Vedelsby, J. Neural network ensembles, cross validation, and active learning, *Advances in Neural Information Processing Systems*, 7, 1995, pp. 231-238.

[12] Ying, K.-C., Lin, S.-W., Lee, Z.-J. and Lin, Y.-T. An ensemble approach applied to classify spam e-mails, *Expert Systems with Applications*, 37(3), March 2010, pp. 2197-2201.

[13] Wei, C.-P., Chen, H.-C., and Cheng, T.-H. Effective spam filtering: A single-class learning and ensemble approach, *Decision Support Systems*, 45(3), June 2008, pp. 491-503.

[14] Liu,Y. and Yao, X. Ensemble learning via negative correlation, *Neural Networks*, 12(10), October 1999, pp. 1399-1404.

[15] Liu,Y., Yao, X., Zhao, Q. and Higuchi, T. Evolving a cooperative population of neural networks by minimizing mutual information, In: *Proceedings of the Congress on Evolutionary Computation*, Seoul, Korea, 2001, pp.384-389.

[16] García-Pedrajas, N., Hervás-Martínez, C. and Ortiz-Boyer, D. Cooperative Coevolution of Artificial Neural Network Ensembles for Pattern Classification, *IEEE Transactions on Evolutionary Computation*, 9, September 2005, pp. 271-302.

[17] Zhao, Q.F. and Higuchi, T. Evolutionary learning of nearest neighbor MLP, *IEEE Transactions on Neural Networks*, 7, July 1996, pp. 762-767.

[18] Breiman, L. Stacked regressions, *Machine Learning*, 24(1), January 1996, pp. 49-64.

[19] Berthold, M.R. and Diamond, J. Boosting the performance of RBF networks with dynamic decay adjustment, *Advances in Neural Information Processing Systems*, 7, 1995, pp. 512-528.

[20] Zhao, W.X. and Wu, L.D. RBFN structure determination strategy based on PLS and Gas,

Journal of Software, 13(8), August 2002, pp. 1450-1455.

[21] Kuncheva, L.I. and Whitaker, C.J. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy, *Machine Learning*, 51(2), February 2003, pp. 181–207.

[22] Ficici, S.G. and Pollack, J.B. Pareto optimality in coevolutionary learning, In: *European Conference on Artificial Life*, 2001, pp. 316-325.

[23] Deb, K., Pratap, A., Agarwal, S., *et al*. A Fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Transaction on Evolutionary Computation*, 6(2), February 2002, pp. 182-197.

[24] Li, M.Q., Kou, J.S., *et al*. The basic theories and applications in GA. *Science Pres*s, Beijing. 2002.

[25] Freund, Y. and Schapire, R. Experiments with a new boosting algorithm, In: *Thirteenth International Conference on Machine Learning*, 1996, pp. 148-156.

[26] Breiman, L. Bagging predictors, *Machine Learning*, 24(2), February 1996, pp. 123-140.

[27] Ting, K.M. and Witten, I.H. Stacking bagged and dagged models, In: *Fourteenth international Conference on Machine Learning*, 1997, 367-375.

[28] Caruana, R., Niculescu, A., Crew, G., *et al*. Ensemble selection from libraries of models, In: *Proceedings of the 21st International Conference on Machine Learning*, 2004, pp. 18-25.

[29] Friedman, J., Hastie, T. and Tibshirani, R. Additive logistic regression: a statistical view of boosting, *Annals of Statistics*, 28, 2000, pp. 337-407.

[30] Webb, G.I. Multiboosting: a technique for combining boosting and wagging, *Machine learning*, 40(2), August 2000, pp. 159-196.

[31] Zhou, Z.H., Wu, J. and Tang, W. Ensembling neural networks: Many could be better than all, *Artificial Intelligence*, 137(1-2), May 2002, pp. 239-263.

[32] Kohavi, R. and Wolpert, D.H. Bias plus variance decomposition for zero-one loss functions, In: *International Conference on Machine Learning*, 1996, pp. 275-283.