Application of the probability-based covering algorithm model in text classification①

ZHOU Ying[1,2]

[1] Department of Information Management, Nanjing University, Nanjing 210093, China;
[2] School of Management, Anhui University, Hefei 230039, China

Received Aug. 11, 2009
Revised Nov. 15, 2009
Accepted Nov. 23, 2009

Abstract The probability-based covering algorithm (PBCA) is a new algorithm based on probability distribution. It decides, by voting, the class of the tested samples on the border of the coverage area, based on the probability of training samples. When using the original covering algorithm (CA), many tested samples that are located on the border of the coverage cannot be classified by the spherical neighborhood gained. The network structure of PBCA is a mixed structure composed of both a feed-forward network and a feedback network. By using this method of adding some heterogeneous samples and enlarging the coverage radius, it is possible to decrease the number of rejected samples and improve the rate of recognition accuracy. Relevant computer experiments indicate that the algorithm improves the study precision and achieves reasonably good results in text classification.

Keywords Probability-based covering algorithm, Structural training algorithm, Probability, Text classification

With the recent development of networks and the vast increase in information output, more and more researchers are studying the processing of text information. Text classification and text clustering are the two most important aspects in the processing of text information and there are numerous algorithms used in this area. On a theoretical level, methods that process text classification can be divided into two categories: those based on statistical methods and those based on linguistic extraction and representation of documents. Researchers will utilize these two methods depending on their distinctive scientific perspectives. With the statistical-based methods, researchers construct a statistical or probability model without taking the meaning of each and every word or phrase into account. They then use the optimized parameters taken from the training set to classify the text for which

① This work is supported by the Fund for Philosophy and Social Science of Anhui Province and the Fund for Human and Art Social Science of the Education Department of Anhui Province (Grant Nos. AHSKF07-08D13 and 2009sk038).
Correspondence should be addressed to Zhou Ying (Email: zhouying97@yahoo.com.cn).

CJLIS
Vol. 2 No. 4, December 2009
pp 1–17
National Science Library, Chinese Academy of Sciences

the class label is unknown. While this method primarily relies on the development of a mathematical representation and the development of artificial intelligence, the other method relies on the natural language processing. This paper is focused on the statistical method for text classification. There are many methods used to obtain the classification model from a training set. Classifiers used include but are not limited to that of Naïve Bayes[1–2], methods based on neural networks[3–5], logistic regression[6], decision trees[7], expectation-maximization (EM) algorithm[8], and approximate inference[9]. Many algorithms can be applied when neural networks are used, such as the K-Nearest Neighbor (KNN) algorithm[3], Support Vector Machine (SVM)[4–5], Bagging algorithm[10], and Boosting algorithm[11] for the enhancement of learning. Although each of these algorithms has its own peculiar characteristics, they all can be optimized. The covering algorithm presented in this paper is a novel neural networks algorithm.

The covering algorithm(CA)[12–13], first proposed by Professor Zhang Bo and Professor Zhang Ling in 1998, has a few advantages over the current neural network calculation in such areas as network structure and accelerated speed of training. There are two methods to deal with the classification problem with the algorithms of neural networks. One is the searching-based method, such as the BP (Back Propagation) algorithm, and the other is the programming-based method, such as the SVM algorithm. The BP algorithm applies the gradient method, which is a local optimization method that makes use of the local information of samples. It obtains the neural network through repeated iteration, so the amount of calculation is large and the network structure is poor. The SVM algorithm proposes Structural Risk Minimization (SRM) and optimizes the network through inputting the kernel method. Some defects still exist, however, including that the SVM method does not work well for multiple classes, and that it is difficult to determine how to select and confirm the kernel function. The cover algorithm applies the combined method of modularizing and programming. It uses modularizing to decrease the amount of calculation and programming to optimize the network. The network structure and parameters of the cover algorithm are not defined before training but are defined in the course of data processing. Many improved methods and applications of the CA have been introduced in the past years. For example, the CA has been applied in finance forecasting[14], car plates recognition[15], and classification of signals[16], which have invariably shown good results.

One of the characteristics of the CA is that a number of the testing samples do not belong to any learned spherical neighborhood, that is, they are “rejected”. This has decreased the rate of classification accuracy. The present paper adopts the novel probability-based covering algorithm (PBCA) in the treatment of the “rejected” samples. Experiments show that this method decreases the rate of error and improves the rate of recognition and accuracy effectively.

National Science Library, Chinese Academy of Sciences

1 Basic contents of the covering algorithm

The covering algorithm is proposed on the basis of the geometrical meaning of the model of M-P neuron. The primary concept of the CA is as follows.

1.1 Primary concept of the covering algorithm

The main idea of the covering algorithm is to construct a network, which can classify samples according to certain conditions. This classification is equal to finding a set of spherical neighborhood that can classify the samples according to certain conditions. With this purpose in mind, we firstly propose that the samples be projected onto the high-dimension space. After the projection, every sample is located on the hyper sphere. Then, the neural network is constructed according to the location of the projected samples. As the CA does not need to scan and iterate the samples repeatedly during its implementation, it has the advantage of reducing the learning time. Instead of the traditional BP algorithm, which needs repeated iterative training and cannot guarantee favorable results, the CA can construct a neural network rapidly, which is highly accurate in the classification of the training samples.

1.2 Basic contents of the CA

There are N classes in the learning samples and the classes can be represented as $X=\{X_1, X_2, \ldots, X_N\}$. The CA uses the spherical neighborhood as the neural cell and separates the tested samples by constructing a three-layered network.

Firstly, the samples are projected from n-dimension space into high-dimension (n+1 dimension) space. This transfer is expressed by T, that is:

$$T: D \to S^{n+1}, T(x) = \left(x, \sqrt{r^2 - |x|^2}\right) \tag{1}$$

In formula (1), $r \geq \max\{|x|, x \in D\}$.

After projection, the samples are located on the hyper sphere S^{n+1}, with the origin point as the center and r as the radius. In that case, the construction of the three-layered neural network classification is equal to obtaining a set of the neighborhood, which can separate different samples from different classes. We then find the first sphere C^1, which only covers the first class samples and not the samples of other classes. After that, we delete the samples that are covered by C^1 and then find the second sphere C^2, which only covers the second class samples and not the other class samples. We then delete the samples that are covered by C^2 …… This repetitious process continues until all samples are deleted (or covered). When designing the structure of the neural network, we use every sphere as a neuron and adopt $\sigma(wx-\theta)$ as the activation function,

$$\sigma(x) = \begin{cases} 1, & x > 0 \\ 0, & x \leq 0 \end{cases} \tag{2}$$

In the course of learning, the method of constructing the sphere of learning sample X_k of the k class is as follows: Randomly select a sample a_i in X_k that has not been covered, and then calculate $d(a)$ according to the following formula:

$$d^1(a_i, x) = \max_{x \notin X_k} \{< a_i, x >\} \tag{3}$$

$$d^2(a_i, x) = \min_{x \in X_k} \{< a_i, x > | < a_i, x >> d^1(a_i, x)\} \tag{4}$$

$$d(a) = \frac{1}{2}\left(d^1(a_i, x) + d^2(a_i, x)\right) \tag{5}$$

Construct the coverage of $C(a_i)$:<ω, x$> - \theta > 0$ with a_i as the center and threshold $\theta = d(a)$, in which <x, y> is the scalar product of x and y. This method will enable all coverages of the samples to be obtained.

In the test, if a tested sample belongs to a sphere neighborhood of a certain class, the sample will be classified into that class. If a tested sample belongs to neither sphere neighborhood, the sample will be rejected.

2 Network model of the probability-based covering algorithm

Because the CA can only cover samples in the same class and not those in different classes, the size of the coverage projected in the space is restrictive. Although the training samples can be recognized completely during the training session, many samples are rejected.

In order to improve the rate of recognition accuracy and decrease the number of errors, the threshold value is increased and the covering sphere radius is enlarged so as to decrease the number of "rejected" samples. In addition, with the enlargement of the coverage radius, it is inevitable that some coverages will partially overlap. In other words, one tested sample may be covered by two or more sphere neighborhoods. The PBCA is used to solve this problem, by firstly processing the training samples via prior probability and by then deciding the classes of the previous results by voting.

2.1 Network model of the probability-based covering algorithm

In regards to the network structure, the CA is considered to be a feed-forward neural network, while the PBCA is a mixed neural network (see Fig. 1) composed of the following three-layered feed-forward network and the upper feedback network. The feed-forward network is used to accomplish the first classification of the samples, while the feedback network conducts the second classification of the samples that either belongs to two or more coverages or are rejected.

- Definition 1. In one particular coverage, the part that only includes the samples of the same class is called the inner core and the radius of the inner core is called the inner radius of the coverage. The part that includes the

samples of different classes is called the border, and the largest radius of the coverage is called the outer radius.

- Definition 2. In one particular coverage, the ratio between the number of samples in the same class and the total number of samples is called the convergence ratio, and the ratio between the number of samples in different classes and the total number of samples is called the divergence ratio.
- Definition 3. In one particular coverage, suppose that the number of samples included in the inner core is k and the inner radius is r, then k/r is called the linear density.

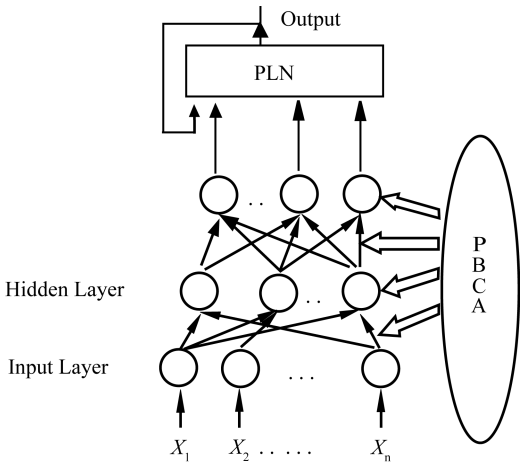

Fig. 1 The structure of the PBCA

In Fig. 1, the self-feedback network—the Probability Logical Network (PLN) is designed as follows: Suppose the training samples set $K=\{r^1(x^1, y^1),..., r=(x^p, y^p), x \in \{0,1\}^n\}$ has m classes $K_1, K_2, ..., K_m$, and the corresponding samples' input is $X=\{X_1, X_2, ..., X_m\}$. Further suppose the center of X_i is a_i, then choose the nearest sample x_i^0 to a_i in X_i and note $y_i=x_i^0$.

The PLN is then constructed as follows: neuron A_i, $i=1, ..., m$, is selected. Each A_i has m input. That is, the domain of each neuron is $\{(0,1)\}^m$. The feedback connection is $H=\{(i, i), i=1, ..., m\}$. That means the feedback y_i is connected to the component i (x_i). The structure of the PLN is shown in Fig. 2.

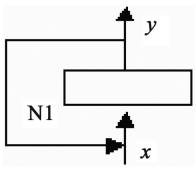

Fig. 2 The network structure of PLN

The activation function of the network is defined as formula (6)–(12)[17]:

$$K^0(i)=\left\{x^\alpha \middle| r^\alpha(y_i)=0, r^\alpha \in K\right\} \tag{6}$$

$$K^1(i)=\left\{x^\alpha \middle| r^\alpha(y_i)=1, r^\alpha \in K\right\} \tag{7}$$

In the above two formulas, suffix i represents two classes according to the value of the i output (y_i), which is 0 or 1. Define the distance function between the vector x and the vector set B as $\rho(x, B)$, and let:

$$D_i^0(x)=\rho\left(x, K^0(i)\right) \tag{8}$$

$$D_i^1(x)=\rho\left(x, K^1(i)\right) \tag{9}$$

If $h(x, y)$ represents the Hamming distance between x and y, then the distance between vector x and the set B is:

$$\rho(x,B)=\inf_{r\in B}\left\{h(x,r)\right\} \tag{10}$$

And let

$$t=g(t_1,t_2)=\frac{t_2}{t_1+t_2} \tag{11}$$

In formula (11), $t_1=D_i^1(x)$ $t_2=D_i^0(x)$ and then,

$$f(t)=\begin{cases}0, & t<0\\ (2t)^k/2, & 0\le t\le 1/2\\ 1-2(1-t)^k/2, & 1/2\le t\le 1\\ 1, & t>1\end{cases} \tag{12}$$

In formula (12), k is a parameter that is, in practice, an integer less than 5. In all experiments included in this paper, $k\ge 3$. When $k=\infty$, we get the following formula:

$$f(t)=\begin{cases}0 & t<1/2\\ 1/2 & t=1/2\\ 1 & t>1/2\end{cases} \tag{13}$$

Then,

$$y_i(t+1)=\begin{cases}0 & <D_i^1(x(t))\\ u & D_i^1(x(t))=D_i^1(x(t))\\ 1 & D_i^0(x(t))>D_i^1(x(t))\end{cases} \tag{14}$$

The physical meaning of formula (14) is as follows: when the Hamming distance between input $x(t)$ and the set with the value y_i equal to 0 is less than the Hamming distance between input $x(t)$ and the set with the value y_i equal to 1, the output of y_i

National Science Library, Chinese Academy of Sciences

at t+1, $y_i(t+1)$ is 0, otherwise, $y_i(t+1)$ is 1. When the two Hamming distances are equal, $y_i(t+1)$ is u, and that is therefore called rejected.

2.2 The realization of the PBCA

According to the network structure of the PBCA (see Fig. 1), the neural network is composed of two sections. That is, the three-layered feed-forward networks and the upper feedback network (PLN). On the whole, the PBCA is realized through mixed neural networks. The feed-forward network conducts the first classification of the tested sample. If the tested sample x is located in the inner core of one coverage, the network determines and generates (outputs) the class information result of the tested sample. Otherwise, when the tested sample x is located in the border of one or several coverages or is rejected, it is necessary to process those samples repeatedly by making use of the convergence ratio of the feedback network, until the tested sample is converged to y^i, and the sample x is classified to class i. The tested result is generated (output). This amounts to a second classification of the tested sample.

After comparison with the previous classification methods, we have determined this to be a different way in which to deal with tested samples located on the border of coverages. Generally, the existing methods use the "adjacency principle" or the "maximum probability principle", while the PBCA uses the probability distribution principle. For example, if a tested sample belongs to five classes with the probability of 0.25, 0.22, 0.18, 0.20 and 0.15, the probability of the first class is a little higher than the other four classes. According to the maximum probability principle, the tested sample should be classified into the first class (0.25). If such a judgment was made every time, samples could only be classified into the first class based on a probability of 0.25, with no possibility of being classified into any of the other four classes. That is, the rate of accuracy is only 25% and the error rate is 75%. This is because the probability distributions information obtained from the learning sessions are not fully utilized during classification, which is obviously not optimal. The PBCA is proposed precisely to overcome this shortcoming by imitating the random distribution of five classes by picking numbers at random. When there is little difference in the probability of these five classes, each and every class is provided with an equal chance in order to improve the classification accuracy of PBCA. Conversely, exercising favoritism each time for only a certain class with very low probability (0.25), consequently results in considerable errors (0.75).

3 The training algorithm and testing algorithm of the PBCA

3.1 The training algorithm of the PBCA

3.1.1 Algorithm 1: The training algorithm of the PBCA

- S1 Choose the maximum modulus r in the training samples set X, then project the samples in X into the hyper sphere with origin as the center and 2r

as the radius. In order to avoid the samples with larger modulus in the testing samples set, $2r$ is more appropriate than r as the radius of hyper sphere;

- S2 Initialize the number of coverage $j=1$ and the number of classes $i=1$;
- S3 Choose the sample of the i class and construct the j coverage C(j);
- S4 If no samples in X_i are uncovered, then go to S8 or else choose any uncovered sample x_i in X_i randomly;
- S5 Construct a coverage $C(j)$ with x_i as the center and θ as the threshold, according to formula (10)–(12);
- S6 If the number of samples of the same class covered by $C(j)$ is more than 2, some heterogeneous samples will be added to enlarge the radius of coverage. At the same time, the numbers of the samples of the same class and different classes will be saved;
- S7. $j=j+1$, go to S3;
- S8. End.

In step S6, the number of added heterogeneous samples is to be decided according to the specific conditions. Experiments show that it is better to add 2–4 heterogeneous samples so as to keep the rate of the same class samples and the different class samples less than 1/3. Heterogeneous samples should not be added when there is only one sample in the coverage.

3.2 The testing algorithm of the PBCA

There are three possible classifications for the tested samples. Firstly, the samples only belong to one or several coverages of one neighborhood. Secondly, the samples belong to several coverages of different neighborhoods. Thirdly, the samples do not belong to any coverage. For the first case, the class of the tested samples will be decided directly by the feed-forward network according to the coverage center. For the third case, the tested samples can be rejected. The class of the rejected samples is judged based on their corresponding coverages with the greatest linear density. As the threshold θ is larger in the training algorithm of PBCA than in the original covering algorithm, the second case emerges with a higher probability than in the original CA. As to the second case, there are again two possibilities, which are as follows.

When the tested sample is at the cross of the inner core of coverage C_1 and the border of coverage C_2, it will be considered to belong to the class of C_1; when the tested sample is at the cross of the border of coverage C_1 and the border of coverage C_2, the probability p of it belonging to coverage C_1 will be considered. In order to display more clearly the ideas of probability distribution of the random number, we use, in actual treatment, the ratio between the number of samples of the same class in the coverage and the total number of samples as the probability of the tested

samples belonging to the coverage. Supposing p_i is the convergence ratio of C_i ($i=1, 2$), that is, when p_i equals the ratio between the number of samples of the same class and the total number of samples of the coverage, then $p=p_1/(p_1+p_2)$, that is the probability of the tested samples belonging to C_1, $0<p<1$. During operation, the program may produce 100 random numbers between 0 and 1. If the numbers are equal to or are less than p, the tested sample belongs to C_1 otherwise it belongs to C_2. The results are then voted on to decide the classification of the sample. If there are more votes for C_1 than those for C_2, the sample belongs to the C_1 class otherwise it belongs to the C_2 class. The actual calculation goes as follows.

3.2.1 Algorithm 2: The testing algorithm of the PBCA

- S1 Input n, $i=1$; // n is the number of testing samples, i is the testing sample
- S2 Project the testing samples into the hyper sphere with the origin as the center and $2r$ as the radius;
- S3 While $i<=n$
- S4 For the testing sample i, calculate $d(x, C_i)=<x, x_i>$, $i=1,2...,j$, x_i is the center of the coverage C_i, j is the total number of the coverage;
- S5 If x only belongs to one coverage, then x belongs to the class in which x_i is found (go to S8);
- S6 If x does not belong to any coverage, then x is a "rejected" sample. Choose coverages C_1 and C_2 with the top two values of linear density — k/r (k is the number of samples in the same class, r is the radius of the coverage). If their classes are the same, then x belongs to that class (go to S8), otherwise the self-feedback algorithm of the PLN is used to deal with x;
- S7 If x belongs to more than one coverage and is in the inner core of one coverage, then x is judged to belong to that class of the coverage with the inner core covering x. If x is in the cross of more than one coverage border, then the coverages C_1 and, C_2 with the top two values of linear density are chosen. If their classes are the same, then x belongs to that class, otherwise the self-feedback algorithm of the PLN is used to deal with x;
- S8 $i=i+1$;
- S9 End
- S10 Calculate and output the error rate.

3.2.2 Algorithm 3 The self-feedback algorithm of the PLN

- S1 Calculate the convergence ratio p_1, p_2 of coverage C_1, C_2, $p_i=s_i / a_i$, $i=1,2$. s_i represents the number of the samples in the same class and a_i represents the total number of the samples in the coverage;

- S2 Calculate the value of $p = p_1/(p_1 + p_2)$;
- S3 Produce 100 random numbers between (0, 1), calculate n_1—the number which are bigger than p and n_2—the number which are smaller than p, then vote;
- S4 If $n_1 > n_2$, then $x \in$ the class of $C_1 - y^1$, otherwise $x \in$ the class of $C_2 - y^2$;
- S5 Output y^i ($i = 1$ or 2).

3.3 Limitations of the PBCA

The PBCA is an optimized algorithm of the original cover algorithm. Its optimized effect depends largely on the precision of the original cover algorithm, and its optimized effect can therefore be limited. That is, if the result of the original cover algorithm is ideal, a more optimized result can be obtained after processing via the PBCA. However, while results can be improved by using the PBCA if the original cover algorithm outcome is poor, the improvement may not be significant. Therefore, the limitation of the PBCA is the same as the limitation of the cover algorithm. That is, the most optimized coverages cannot be determined, such that the least amount of coverage while covering the most amounts of samples cannot be established. This has significance in relation to the sequence of the center of coverage and highlights the lack in current algorithms to ensure that coverage is the most optimized. The most optimized coverage corresponds to the most optimized neural network structure.

4 Experiment results and the analysis

The experiments conducted in this study included two sections. The results of the first section (here after mentioned as Experiment 1) were from University of California, Irvine (UCI), Machine Learning Repository and the results of the second section (here after mentioned as Experiment 2) were from the text classification corpus downloaded from the Chinese Nature Language Processing Platform (http://www.nlp.org.cn). All experiment results were obtained under MATLAB 7.0.

4.1 Data sources and results of Experiment 1

The six databases used in this section were obtained from the UCI Machine Learning Repository (www.ics.uci.edu/mlearn/mlrepository.html) (See Table 1). The training and testing algorithms of the PBCA were used. Between 2–4 heterogeneous samples were added into the coverage in which the number of samples was larger than one. The results of the experiments are shown in Table 2 (add $k = 2$ heterogeneous samples) and the comparative results with other classifiers are shown in Table 3 below.

Table 1 The selected databases for the experiment

Databases	Number of samples	Number of dimensions	Number of classes
Letter	20,000	16	26
Waveform	5,000	21	3
Glass	214	9	6
Iris	150	4	3
Wine	178	13	3
Pima Indians diabetes	768	8	2

Table 2 The testing results of the PBCA

Databases	Highest correct rate (%)	Average correct rate (%)	Number of training samples	Number of testing samples	Number of rejected samples
Letter	90.3	87.7	16,000	4,000	0
Waveform	81.5	80.9	300	5,000	0
Glass	67.23	63.86	169	45	0
Iris	99.2	98.4	120	30	0
Wine	100	94.2	148	30	0
Pima Indians diabetes	73.64	71.44	576	192	0

Table 3 The comparative results with other classifiers

Databases	Correct rate (%)				
	C4.5	NB/SVM	TAN	CBA	AVE
Letter	77.7	74.9	85.7	51.8	71.8
Waveform	70.4	78.5	79.1	75.3	75.8
Glass	69.6	69.7	58.4	–	65.9
Iris	94.4	93.3	–	–	93.9
Wine	71.5	94.8*	–	–	83.2
Pima Indians diabetes	71.1	73.1	72.8	68.3	71.3

Note: C4.5 is an algorithm used to generate a decision tree developed by Ross Quinlanis[18]; NB is Naive Bayesian Classifiers[19]; SVM is Support Vector Machine[20] and the result marked * is the result of SVM; TAN is Bayesian Network Classifiers[19] and "–" shows the missed classification results; CBA is the Classifier Based on Association Rule[21], and AVE represents the average correct rate of the above four classifiers.

4.2 Analysis of the Experiment 1 results

Compared to the average correct rate in Table 3, both the highest and the average correct rate of the six databases shown in Table 2 were higher than those in Table 3. It can be seen that the PBCA is effective in the training of neural networks. It adds a number of samples of different classes and enlarges the radius of the spherical coverages so as to decrease the number of coverages and the number of neurons in hidden layers and increase the accuracy of recognition. From the results obtained, the PBCA clearly showed high efficiency and improved the learning quality with only a small increase in learning time.

It is well known that the present classification methods usually adopt either the "adjacency principle" or the "maximum probability principle" for the re-classification of the bordered examination samples. When the maximum probability α is at a very low level, the error rate of the reclassification produced by the "maximum probability principle" is $1-\alpha$, which means a very high value. It is clearly not a reasonable approach. The reason for the high error rate is that it does not make full use of the learned probability distributing information. Conversely, however, the PBCA, displaying the idea of probability distribution, avoids this shortcoming and enhances classification precision.

4.3 Data sources of Experiment 2

The databases of Experiment 2 were obtained from the text classification corpus downloaded from the Chinese Nature Language Processing Platform (http://www.nlp.org.cn). There are 2,799 documents in the corpus, involving ten classes of environment, computer, transportation, education, economy, military affairs, physical education, medicine, arts, and politics. The corpus was divided into two equal sections with the proportion of 1 to 1, that is, the training corpus and the testing corpus. There were 1,370 documents in the former and 1,429 documents in the latter. The training corpus and the testing corpus are shown in Table 4 (the unit of the data is the number of the document).

Table 4 The training corpus and the testing corpus

Class	Environment	Computer	Transportation	Education	Economy	Military affairs	Physical education	Medicine	Arts	Politics	Total
Training corpus	100	90	100	110	160	120	220	100	120	250	1,370
Testing corpus	99	102	114	110	165	123	230	103	128	255	1,429
Total	199	192	214	220	325	243	450	203	248	505	2,799

4.4 Data process of the Experiment 2

The Chinese Lexical Analysis System provided by the Institute of Computing Technology, Chinese Academy of Sciences, was used to segment the document into words. The words were then processed with a statistics program written in JAVA. After deleting the stopped words and sparse words and counting the frequency of nouns and verbs, the document was expressed in a vector form (Term frequency), as shown in Table 5.

We used the data in Table 5 and obtained a $m \times n$ matrix D (comparative frequency of words) in which m represented the total number of different words in

Table 5 TF (Term frequency) form of document

	Document 1	Document 2	Document 3	…
Word 1	15	12	20	…
Word 2	12	2	0	…
Word 3	3	5	16	…
…	…	…	…	…

the document set and n represented the document number in the document set. That was, each different word corresponded to a row in matrix D and each document corresponded to a column in matrix D. The value of d_{ij} in matrix D was calculated according to formula (15).

$$W(t,d)=\frac{tf(t,d)\times\log(N/n_t+0.01)}{\sqrt{\sum_{t\in d}\left[tf(t,d)\times\log(N/n_t+0.01)\right]^2}} \tag{15}$$

Formula (15) is a generally used formula applied in calculating TF-IDF (Term Frequency–Inverse Document Frequency). In it, *W(t,d)* is the weight of word *t* in document *d*, *tf(t,d)* is the absolute frequency of the word *t* in the document *d* (TF factor), *N* is the total number of the training documents, n_t is the number of the documents that include word *t,* and the denominator is the unitary factor.

Because the number of words and documents are very large, and the words are widely distributed, the number of words, which appear in one document, is very limited. As such, matrix D is always called as a sparse matrix[22]. In matrix D, d_{ij} is the frequency of the *i* word appearing in the *j* document. Its value is between 0 and 1 after the unified process. The format of the matrix D is shown in Table 6.

Table 6 TF-IDF form of text

	Document 1	Document 2	Document 3	…
Word 1	0.005	0.12	0.05	…
Word 2	0.004	0.02	0	…
Word 3	0.0001	0.05	0.04	…
…	…	…	…	…

In the training algorithm of this experiment, there were 1,370 documents and 21,017 words prior to the deletion of the stopped and the sparse words. The number of the stopped words was 2,000 and the number of the sparse words was 10,259, which accounted for 9.5 percent and 48.8 percent of the total number respectively. After the stopped words and sparse words were deleted, 8,758 words remained, which accounted for 41.7 percent of the total number. We constructed a matrix D that was 8,758×1,370. In the testing algorithm of this experiment, there were 1,429 documents and 26,572 words prior to the deletion of the stopped and the sparse words. The number of the stopped words was 1,893 and number of the sparse words was 12,054,

which accounted for 7.1 percent and 45.4 percent of the total number respectively. After the stopped words and sparse words were deleted, 12,625 words remained, which accounted for 47.5 percent of the total number. We constructed a matrix D that was 12,625×1,429. The distance of the different documents was calculated via the cosine distance formula.

4.5 Results of the Experiment 2

To the processed 1,370 documents in the training set, the original coverage algorithm (CA)[13] and the Algorithm 1 of the training algorithm of PBCA were applied. In the case of the latter, a number of heterogeneous samples (decided as four) were added to the coverage. Then we got the classifier 1and classifier 2. The original coverage algorithm was applied to the processed 1,429 documents in the testing set. The experiment was conducted five times, after which the average was designated as the result of classifier 1. The actual training time of classifier 1 was 16.82 seconds, which did not include the time spent pre-processing the documents, such as segmenting the texts into words. To the same samples in the testing set, Algorithm 2 of the testing algorithm of the PBCA was applied. The experiment was also conducted five times, after which the average was designated as the result of classifier 2. The actual training time of classifier 2 was 25.66 seconds, which did not include the time spent pre-processing the documents, such as segmenting the texts into words. The classification results of the 10 classes are shown in Table 7.

Table 7 Comparison of the classification accuracy of two classifiers

Class	Accuracy rate %	
	Classifier 1	Classifier 2
Environment	90.5	92.6
Computer	91.4	93.9
Transportation	95.3	95.8
Education	95.2	95.9
Economy	93.1	93.8
Military affairs	91.2	90.8
Physical education	92.3	93.1
Medicine	93.6	95.3
Arts	96.8	97.1
Politics	90.6	91.3

4.6 Analysis of the Experiment 2 results

From the results shown in Table 7, there are nine classes in classifier 2 with a higher accuracy rate than in classifier 1. This is due to the difference in classification methods between the common algorithms and the PBCA. For common algorithms, when the information is determined, a border is defined by rounding the numbers up or down before calculation. In the PBCA, however, the rejected and crossed

samples are first described with the probability from the pre-experiment information of the samples in the coverage, and the results are calculated before rounding them up or down. Finally, the class of the testing samples is determined by voting on the processed results. Therefore, the PBCA is a different and novel classification method that calculates first and then rounds the results up or down. From the results obtained in this study, the PBCA can help achieve higher classification precision and is, therefore, obviously advantageous both in efficiency and accuracy. However, experiments also show that the No Free Lunch theorem[23] is perfectly correct, and while the PBCA does achieve higher classification precision, it adds to the testing time of the training. Consequently, we must treat different problems with respectively appropriate parameters in the coverage core, maximum coverage radius, and relevant linear density to achieve desired classification results.

The above analysis shows that the PBCA is highly effective in the training of neural networks. Compared with the original algorithms, it increases limited learning time and improves the learning precision on the smallest possible network scale.

5 Conclusions

The cover algorithm is a novel neural networks algorithm, which applies the method of combined modularizing and programming. It uses modularizing to decrease the amount of calculation and programming to optimize the network. The network structure and parameters of the cover algorithm are not defined before training but are defined in the course of data processing. It has many advantages in the structure of network and the training speed in comparison with the algorithms of current neural networks.

The PBCA introduced in this paper is an optimized algorithm to the original cover algorithm. It attempts to improve recognition precision by adding some heterogeneous samples in the coverage to enlarge the coverage radius and decrease the number of rejected samples. However, the increase of coverage radius inevitably results in the crossing of different coverages, that is, one tested sample belongs to more than one coverage at the same time. To classify such samples, the PBCA first divides the coverage in the learning process into core and border. It then treats the tested samples on the border with the pre-experiment probability of the learning samples in the coverage, and decides the final classification of the samples by voting on the processed results (calculation before determining the classification of the samples by voting). During the processing procedure, the probability treatment is determined with the use of the neural network of the probability logic.

Seen from the perspective of network structure, the PBCA is a mixed neural network composed of a three-layered feed-forward network and an upper one-layered feedback network. The feed-forward network determines the preliminary

classification of the samples. If the tested samples are rejected or are located on the border of one or several coverages, the feedback network will implement the secondary classification of the samples with the density information of the samples to decide the class of the samples. Computer modeling experiments indicate that the PBCA is highly effective in the training of neural networks. It increases limited learning time and improves the learning precision on the smallest possible network scale.

References

1 Chakrabarti, S., Dom, B., & Indyk, P. Enhanced hypertext categorization using hyperlinks. In Tiwary, A. & Franklin, M.(Eds.), Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data. Seattle, Washington, US: ACM Press, 1998: 307–318.

2 Neville, J & Jensen, D. Iterative classification in relational data. In Proceedings of the AAAI 2000 Workshop on Learning Statistical Models from Relational Data. Technical Report WS-00-06. Austin, Texas: AAAI Press, 2000: 42–49.

3 Zhou, Y., Li, Y. W., & Xia, S. X. An improved KNN text classification algorithm based on clustering. Journal of Computers, 2009, 4(3):230–237.

4 Shawe-Taylor, J. & Cristianini, N. Kernel methods for pattern analysis. Cambridge: Cambridge University Press, 2004.

5 Joachims, T. Text categorization with support vector machines: Learning with many relevant features. In Nedellec, C. & Rouveirol, C. (Eds.), Proceeding of the 10th European Conference on Machine Learning. Chemnitz, Germany: Springer, 1998:137–142.

6 Lu, Q & Getoor, L. Link based classification. In Fawcett, T & Mishra, N. (Eds.), Proceedings of the Twentieth International Conference on Machine Learning. Washington DC: AAAI Press, 2003: 496–503.

7 Jensen, D., Neville, J., & Gallagher, B. Why collective inference improves relational classification. In Kim, W., Kohavi, R., & Gehrke, J., et al. (Eds.), Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Seattle, WA: ACM Press, 2004: 593–598.

8 Nigam, K., McCallum, A., & Thrun, S., et al. Text classification from labeled and unlabeled documents using EM. Machine Learning, 2000, 39(2/3):103–134.

9 Macskassy, S. & Provost, F. Classification in networked data: A toolkit and a univariate case study. Journal of Machine Learning Research, 2007, 8(May):935–983.

10 Zhao, S., Li, X., & Liu, W. H. Bagging text classification algorithm based on classifier performance evaluation. Computer Engineering, 2008, 34(1):61–63.

11 Bloehdorn, S. & Hotho, A. Boosting for text classification with semantic features. Advances in Web Mining and Web Usage Analysis. In Mobasher, B., Liu, B., & Masand, B. et al. (Eds.), Proceedings of the 6th International Workshop and Knowledge Discovery from the Web, WebKDD 2004. Seattle, WA: Springer, 2004:149–166.

12 Zhang, L. & Zhang, B. A geometrical representation of McCulloch-Pitts neural model and its application. IEEE Transactions on Neural Networks, 1999, 10(4): 925–929.

13 Zhang, L. & Zhang, B. Learning Methods of Neural Networks. In Zhou, Z. Y. & Cao, C. G. (Eds.), Neural Network and Its Application (in Chinese). Beijing: Tsinghua University Press, 2004:1–35.

14 Wu, M.R., Zhang, B., & Zhang, L. Neural network based classifier for handwritten Chinese character recognition. In Proceedings of the 15th International Conference on Pattern Recognition. Barcelona: IEEE Computer Society, 2000, 2:561–568.

15 Zhang, L., Zhang, B., & Yin, H. F. The cross coverage algorithm of multi-layer forward propagation network. Journal of Software (in Chinese), 1999, 10(7):737–742.

16 Wu, M. R. Study of the classifier design in the large scale pattern recognition (in Chinese) (thesis). Beijing: Tsinghua University (2000).

17 Zhou, Y. A study of algorithms of neural networks as classifiers and their application in text classification (in Chinese) (thesis). Hefei: Anhui University (2006).
18 Quinlan, J. R. C4.5: Programs for Machine Learning. San Mateo: Morgan Kaufmann. 1993.
19 Friedman, N., Geiger, D., & Goldszmidt, M. Bayesian network classifiers. Machine Learning, 1997, 29(1): 131–163.
20 Zhang, L. Study of SVM and kernel method (in Chinese) (thesis). Xi'an: Xi'an University of Electronic Sciences (2002).
21 Liu, B., Hsu,W., & Ma,Y. Integrating classification and association rule mining. In Agrawal, R. & Stolorz, P. (Eds.), Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining. NewYork: AAAI Press, 1998:80–86.
22 Qian, T.Y., Wang Y. Z., & Feng, X. N. Chinese text classification relating class frequency. Journal of Chinese Information Processing (in Chinese), 2004, 18(6): 30–36.
23 Duda, R. O., Hard, P. E., & Stock, D. G. Pattern Classification (Second edition) (in Chinese). Translated by Li, H. D. & Yao, T. X. Beijing: China Machine Press, 2003.

(*Copy editor*: *Ning LI*; *Language revision*: *Christine WATTS*)