An Extended Corner Classification Neural Network Based Document Classification Approach*

CHEN En-hong[1], ZHANG Zhen-ya[1], Aihara Kazuyuki[2], WANG Xu-fa[1]

[1](*Department of Computer Science, University of Science and Technology of China, Hefei* 230027, *China*);

[2](*Department of Mathematical Engineering and Information Physics Faculty of Engineering, University of Tokyo, Tokyo* 113-8656, *Japan*)

E-mail: cheneh@ustc.edu.cn

http://mail.ustc.edu.cn/~zzychm

Received May 28, 2001; accepted November 14, 2001

Abstract: CC4 (the 4th version of corner classification) neural network is a new type of corner classification training algorithm for three-layered feedforward neural networks. It has been provided as a document classification approach for metasearch engine Anvish. On the condition that documents are almost of the same size, CC4 neural network is an effective document classification algorithm. However, when there is great difference in document sizes, CC4 neural network does not perform well. This paper aims to extend the original CC4 neural network for effectively classifying documents having much difference in sizes. To achieve this goal, the authors propose a MDS-NN based data indexing method thus making all documents be mapped to k-dimensional points while their distance information is kept well. The authors also extend CC4 neural network so that it can accept *k*-dimensional indexes of documents as its input, then transform these indexes to binary sequences required by CC4 neural network. The experimental results show that the performance of ExtendedCC4 is much better than that of InitialCC4 when there is a great difference in document sizes. At the same time, the high classification precision of ExtendedCC4 has much relationship with the effectiveness of indexing methods.

Key words: document classification; CC4 neural network; data indexing; distance information

With the explosive growth of information sources available on the World Wide Web, it has become increasingly necessary for users to utilize web search engine to find the desired information sources. Now a new generation of search engine has emerged (with Google as the representative), to provide convenience and efficiency for the user[1]. Information classification is playing an important role among all the information retrieval technologies. Once a search engine accepts a query of keywords from the user, it starts to retrieve and return web pages that match the query according to certain criteria. But the user may be interested in only a small portion of the search results. Providing accurate search results to users with information classification method can help users mine

* Supported by the National Natural Science Foundation of China under Grant No.60005004 (国家自然科学基金); the National Grand Fundamental Research 973 Program of China under Grant No.G1998030509 (国家重点基础研究发展规划 973 项目)

CHEN En-hong was born in 1968. He is an associate professor of Department of Computer Science, USTC. He received his Ph.D. degree in Computer Science from USTC in 1996. His current research areas include machine learning, information retrieval, data mining and XML. **ZHANG Zhen-ya** was born in 1972. He is a Ph.D. candidate at Department of Computer Science, USTC. He received his B.S. degree in Computer Science from Anhui Normal University in 1995. His research interests include intelligent information retrieval and data mining. **Aihara Kazuyuk**i was born in 1954. He is a professor of Department of Mathematical Engineering and Information Physics, the University of Tokyo. He received his Ph.D. degree in electronic engineering from the University of Tokyo in 1982. His current research areas include intelligent information processing, Chaos engineering and neural networks. **WANG Xu-fa** was born in 1948. He is a professor and doctoral supervisor of Department of Computer Science, USTC. His current research areas include machine learning, information retrieval, data mining and XML.

the Web more efficiently. The metasearch engine Anvish[2] has provided an efficient neural network-based classification algorithm, CC4 neural network. Because Anvish gets search results through different search engines[2,3], like Yahoo, WebCrawler, Excite, Infoseek, all pieces of information are almost of the same size. This is an important precondition that CC4 can work effectively. But it would be impractical to assume that all pieces of information on WWW are of the same size in reality.

This paper aims to provide efficient and effective solutions for document classification. To solve the problem incurred by the great difference in document sizes, we propose an MDS-NN based data indexing method thus making all documents be mapped to *k*-dimensional points while their distance information is kept well. We also extend CC4 neural network so that it can accept *k*-dimensional indexes of documents as its input.

In the following section, we will first briefly introduce CC4 neural networks. Then MDS-NN based textual data indexing method will be presented in Section 2. Section 3 describes our document index based classification method with CC4 neural network. Our experimental results and analysis are given in Section 4. Section 5 is our concluding remarks and future research directions.

1 CC4 Neural Networks

The CC4 algorithm[2] is a new type of corner classification training algorithm for three-layered feedforward neural networks. It has three layers: Input Layer, Hidden Layer, and Output Layer. The neurons between Input Layer and Hidden Layer, Hidden Layer and Output Layer are fully connected. The architecture of the network can be seen in Fig.1.

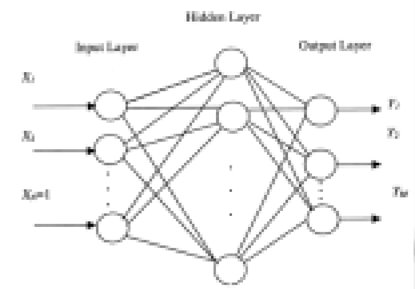

Fig.1 CC4 network architecture

The CC4 network maps an input binary vector *X* to an output vector *Y*. The neurons are all binary neurons with binary step activation function as follows:

$$y = f(\sum x_i) = \begin{cases} 1 & \sum x_i > 1 \\ 0 & \sum x_i \le 0 \end{cases} \tag{1}$$

where x_i=1 or 0. The number of input neurons is equal to the length of the input vector plus one, the additional neuron being the bias neuron which has a constant input of 1. The number of hidden neurons is equal to the number of training samples with each hidden neuron representing one training sample.

The training of the CC4 neural network is very simple. Let w_{ij} (i=1,2,….,N and j=1,2,…,H) be the weight of the connection from input neuron i to hidden neuron j and let X_{ij} be the input for the ith input neuron when the jth training sample is presented to the network. Then the weights are assigned as follows:

$$w_{ij} = \begin{cases} 1 & \text{if } x_{ij} = 1; \\ -1 & \text{if } x_{ij} = 0; \\ r - s + 1 & \text{if } i = n; \end{cases} \tag{2}$$

Here *r* is the user defined radius of generalization and s is the number of 1’s in the training vector.

Let u_{jk} (j=1,2,…,H and k=1,2,…,M) be the weight of connection from the *i*th hidden neuron to the *k*th output neuron and let Y_{jk} be the output of the *k*th output neuron for the *j*th training sample. The value of u_{jk} are determined by the following equation:

$$u_{jk}=\begin{cases}1 & \text{if } Y_{jk}=1;\\ -1 & \text{if } Y_{jk}=0.\end{cases} \tag{3}$$

In Anvish[2] metasearch engine, the CC4 neural network gets the TF vectors of documents as its input. In this way, the Anvish search engine can classify web pages in a precision around 80%. In Anvish however, the CC4 gets its input from the returned results of standard search engine like Yahoo, WebCrawler, Excite, Infoseek. All pieces of textual information are approximately of the same size. Thus, the classification precision is not so representative. In fact, there may be quite large differences between the sizes of several textual documents that belong to the same class. In this way, to accurately classify different size of documents that are in the same class by CC4, we must increase the radius of generalization *r*. But the increment of *r* may incur the inaccuracy of classification and result in misclassifying different classes of documents into the same class.

2 MDS-NN Based Textual Data Indexing

When performing indexing, mining and visualization operations on relational or text and multimedia data, a typical useful method is first to map n-dimensional data objects into low dimensional space while preserving distances between original data, then perform the corresponding operations[4]. When knowing the distances of pairs of data objects, Multi-Dimensional Scaling (MDS) is readily used to index original data in low dimensional space. If new data objects are added into data set, MDS is inefficient especially when the data size is large. To overcome the problems, we have proposed a BP neural network based incremental data indexing approach, called MDS-NN method. In this method, a small data set called sample data set is first indexed with MDS approach. For the size of sample data set is very small, the time spent on this step is very low. Then the indexing results are provided as training samples and supervisor signals to train neural network. The trained neural network is used to index newly added data. The quality of indexing is measured by *Stress*[4~6] function:

$$Stress=\sqrt{\frac{\sum_{i,j}(d_{ij}'-d_{ij})^2}{\sum_{i,j}d_{ij}^{\;2}}}, \tag{4}$$

where d'_{ij} is the distance between P_i and P_j, d_{ij} is the distance between O_i and O_j, and $P_i=(x_{i1},x_{i2},\ldots,x_{ik})$, a *k*-dimensional point called *k*-index of O_i, is the image of an original object O_i.

Definition 1. *K*-index: Suppose that there exists a mapping *T* that maps any original data *d* into a point *p* in *k*-dimensional space, then point *p* is called *k*-index of *d*.

To index all objects in *k*-d space means that all distances among objects should be kept as much as possible in *k*-dimensional space. For this purpose, *Stress* is used to find a good index in *k*-d space for each original object. When finishing indexing operation, *Stress* reaches a minimum value.

The steps of MDS method is as following:

1) Randomly map each item to a *k*-d point, i.e. the vector representation of a *k*-dimensional point;
2) Examine every point, compute its distance from the other points and move the point to minimize the *stress* function.
3) Iteratively perform step 2 until the *stress* value becomes stable.

Our proposed MDS-NN method is as following:

1) Build the k-d indexes of training sample data using MDS method.

2) Construct the sample data set and supervisor signal set for BP Neural Networks with the results obtained in step 1.

3) Train the Neural Networks with the data obtained in step 2.

4) Build the index of newly coming data with the trained Neural Networks.

3 Document Index Based Classification with CC4 Network

Through MDS-NN based textual data indexing method, all documents are mapped to points in k-dimensional space while their distance information is kept as much as possible. In this paper, we will extend the CC4 neural network so that it can accept k-dimensional indexes of documents as its input.

The CC4 neural networks consider only the Hamming Distances of the textual documents. To any document represented as TF vectors $tf=(tf[0], tf[1],..., tf[L])$ whose size is L. If $tf[i] > 0$, the i-th input of CC4 is 1, else 0. This encoding method may reduce the classification precision.

In one case, suppose that we have two documents whose TF vectors tf_1 and tf_2, and there exists a certain i_0 such that $tf_1[i_0]=tf_2[i_0]=0.98$, and for other j, $1\leq j\leq L$, $j\neq i_0$, $tf_1[j]\times tf_2[j]=0$ and $tf_1[j]+tf_2[j]\neq 0$. Thus the Hamming Distance between tf_1 and tf_2 is $L-1$ and the CC4 will classify these two documents into different classes. When measured with cosine similarity however, their distance is $1-0.98^2=0.0396$ These two documents can thus be regarded as in the same class. In order that the CC4 can classify these two vectors into the same class, we have to increase the radius of generalization to a high value, which will inevitably incur the misclassification of other TF vectors.

In another case, given two documents $tf_1$and tf_2, suppose that $tf_1[i_0]=0.98$, $tf_1[i_1]=0.02$, $tf_1[i_0]=0.02$, $tf_1[i_1]=0.98$, and $tf_1[j]=tf_1[j]=0$ for other i, $1\leq j\leq L$, $j\neq i_0,i_1$. The two documents are surely classified into the same class with CC4 networks. However, they should be in different classes when measured with cosine similarity.

Considering that the CC4 can only accept binary values as its input, each k-index of documents should be transformed to a 0/1 sequence. To avoid two cases mentioned above, this sequence should reflect the distance information of k-indexes of documents as Hamming distance as much as possible. In the following, we will notion of L-discretization sequence of real numbers first, and then L-discretization sequence of k-index.

Definition 2. Let x be a real number such that $x\in [a,b]$, S is a L-discretization sequence of x given that the frontmost k elements of S are all ones and the rest $L-k$ elements are all zeroes, where L is the length of S, $m=\frac{b-a}{L}$, $k=\left[\frac{x-a}{m}\right]$.

For example, let x=0.72, $x\in$[0,1], L=10, then m=0.1, k=7, and thus we get a L-discretization sequence 1111111000 for x=0.72 at interval [0,1].

Definition 3. L-discretization sequence of k-index: Suppose that k-index of an original data d is $(x_1,x_2,\ldots,x_k)$, $x_i\in[a_i,b_i]$, $i=1,2,...,k$, L is a given positive integer and S_i is L-discretization sequence of x_i at interval $[a_i,b_i]$, then $S=\langle S_{ij}\rangle=\langle S_{11}, S_{12},\ldots,S_{1L},S_{21},S_{22},\ldots,S_{2L},\ldots,S_{k1},S_{k2},\ldots,S_{kL}\rangle$ is the L-discretization sequence of k-index of data d, where $i=1,2,\ldots,k$, $j=1,2,\ldots,L$, $S_{ij}=S_i[j]$ is the jth element of the L-discretization sequence of k-index of x_i.

Considering the merit and shortcoming of the CC4 neural networks, we modify the input of the CC4 neural networks considerably by integrating our research work in textual data indexing. We do not directly use the TF or TF-IDF vectors of textual documents as the input of the CC4 any more, but the L-discretization sequence of k-index of textual documents. Thus the input vectors of the CC4 are of the same size. We call our method ***ExtendedCC4*** for short, in contrast to the CC4 (we call it ***InitialCC4***) using the TF or TF-IDF representation of textual documents as its input. The discretization method we propose is as following:

```
1)  Decide a value $L$ for $L$-discretization sequence of $k$-index;
2)  Initialize S to be an empty sequence;
3)  For each element $e_i$ of $k$-index of document $d$ do
    3.1) Determine the interval for each element of $k$-index. Suppose that the
            length of the interval is $I$, set $Step= [I/L]$ ；
    3.2) $Length = [e_i/Step]$;
    3.3) Assign $\langle S_{i1},S_{i2},\ldots,S_{iL}\rangle$ to be $L$-discretization sequence of $k$-index of
            element $e_i$, where $S_{ij}=1$ for $j=1,2,..., Length$, and $S_{ij}=0$ for $j=Length+1, ..., L$ ；
    3.4) Append $\langle S_{i1},S_{i2}, \ldots,S_{iL}\rangle$ to the tail of sequence $S$.
```

When training ExtendedCC4 to classify documents, each training document is indexed with MDS-NN method and then its *L*-discretization sequence of *k*-index is calculated as the input of ExtendedCC4. The topic of the corresponding document is served as the supervisor signal of ExtendedCC4 Neural Networks. For newly coming textual documents, MDS-NN method is applied to get their *L*-discretization sequence of *k*-indexes and then the trained ExtendedCC4 neural network is subsequently used to get them to be classified.

4 Theoretical Analysis of ExtendedCC4

To theoretically analyze the classification behavior of ExtendedCC4, we first introduce the notion of δ-neighborhood of *k* dimensional point *X*, and then give the relationship between radius of generalization and classification ability of ExtendedCC4.

Definition 4. Suppose that *X* is the center of a hyper-cube whose length of each edge is 2δ, then the continuos area covered by the hyper-cube is called δ-neighborhood of *X* and denoted as $N_\delta(X)$, *X* is the representative of the area.

Definition 5. Suppose that $X=(x_1,x_2,\ldots,x_k)\in[0,1]^k$, $Y=(y_1,y_2,\ldots,y_k)$, $x_i,y_i\in[0,1]$, $i=1,2,\ldots,k$. If $|x_i - y_i|\le\delta$, where $\delta>0$, then *Y* belongs to the δ-neighborhood of *X* and is denoted as $Y\in N_\delta(X)$.

Theorem 1. Suppose that *k*-index $X=(x_1,x_2,\ldots,x_k)$ is the center of training set for class *C*, $x_i\in[0,1]$, $i=1,2,\ldots,k$. Let $L=s$ for *L*-discretization sequence of *k*-index *X* and $r=[\delta/s]$. To any $Y=(y_1,y_2,\ldots,y_k)$, $y_i\in[0,1]$ for $i=1,2,\ldots,k$, if the Hamming distance of *L*-discretization sequences of x_i, y_i is at most *n*, $n\le r$ iff $Y\in N_\delta(X)$.

Proof. First, we know that $r=[\delta/s]$, hence $rs\le\delta\le(r+1)s$. For $n\le r$ and $s>0$, thus $n\le rs\le\delta$. Hence the Hamming distance of *L*-discretization sequences of *k*-index of *X* and *Y* is at most δ. Thus we can conclude that $Y\in N_\delta(X)$.

Conversely, given that $Y\in N_\delta(X)$, thus $|x_i-y_i|\le\delta$, $i=1,2,\ldots,k$. For $ns\le|x_i-y_i|<(n+1)s$, hence $n\le\delta$. For $rs\le|x_i-y_i|<(r+1)s$, hence $ns\le rs$, hence $n\le r$, and the theorem is proved.

Obviously, the distribution of these points is completely determined once all documents are mapped to the points in *k*-dimensional space. Because the original objects come from several different classes, these points are thus expected to be partitioned into several different continuous areas and each area correspond to a different class. ExtendedCC4 is aimed to determine which area each document belongs to based on its *L*-discretization sequence of *k*-index. When training ExtendedCC4, training samples for all classes are chosen first, then the center of training examples for each class is calculated as the real input of ExtendedCC4, i.e. the center, not training examples themselves, are used for training purpose. For the ease of clarity, we denote the class that the first center used to train ExtendedCC4 belongs to as the first class, and so on. By Theorem 1, we know that more and more points will be covered by the δ-neighborhood of each training center with the increase of the radius of generalization when training ExtendedCC4 and thus improve the classification precision of trained ExtendedCC4. The precision will reach to its highest value at a certain radius of generalization. Afterwards, with the increase of the radius of generalization, more and more points are covered by the δ-neighborhoods of the centers that belong to other classes, thus leading to the decrease of classification precision. However, when the radius of generalization is larger than a value r_0, called threshold value, the δ-neighborhood of the center used as the first training sample for ExtendedCC4 will cover all points. It thus makes the classification precision stay at a stable level, i.e. around the percentage of test samples belonging to the first class. This classification behavior of ExtendedCC4 will be demonstrated in the following experiments.

5 Experimental Results and Analysis

Our experiments are performed on real data downloaded from UCI KDD Archive site http://kdd.ics.uci.edu.

We randomly select 10 groups of news data downloaded and pick out the frontmost 50 news in each group as our experimental data. We set k=3. Thus all documents are mapped into points in 3-dimensional space. To test the performance of our MDS-NN indexing model, we select different numbers of documents to train neural networks. Figure 2 shows *Stress* values with the different numbers of documents as training samples. The detailed experimental steps are given below:

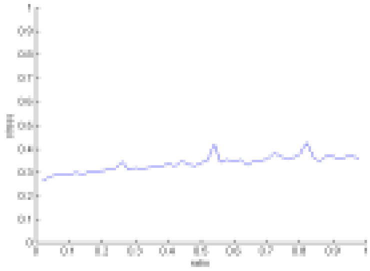

Fig.2 The Stress value for different ratio of documents used as training samples of MDS-NN

1) In each news group, determine a value for *ratio* such that *ratio* = size of training set / size of entire data set, then calculate the size of training set *SamplesNumber*.

2) Build a dictionary based on the terms obtained from the frontmost *SamplesNumber* articles in each news group. It can be used for the construction of the TF vector of each document.

3) Build TF vectors of the frontmost *SamplesNumber* news in each group, then calculate the center vector of training set in each group. Each component of the center vector is the median of all the corresponding components of training vectors.

4) Take the center vectors as the training data set of MDS-NN, and the news in each group as the test data set, we can build the k-d index of the test data using MDS-NN.

5) Take the k-d index of the test data as input, ExtendedCC4 is used to classify each test news.

Given different ratios of training documents, Figs.3~6 show the influences of the radius of generalization on the classification precision of ExtendedCC4 and InitialCC4. It can be observed that when ratio value is fixed, the highest classification precision of ExtendedCC4 will be much better than that of InitialCC4.

From Figs.3~6, we can also observe that when the radius of generalization is larger than a threshold value r_0 the classification precision of ExtendedCC4 and InitialCC4 stays at a stable level, i.e. around at the percentage, i.e. 10%, of test samples belonging to the first class. To confirm our observation, we also perform further experiments by picking up 10, 20, 30, 40 and 50 documents respectively from the first news group, and 50 documents from all other nine groups. The documents used to generate the center used as the training sample for each group are 10 percent of all documents that each group contains. Therefore, there are respectively 9, 18, 27, 36, 45 documents in the first news group are used as test documents. Table 1 shows the results obtained when the classification precision of ExtendedCC4 converges.

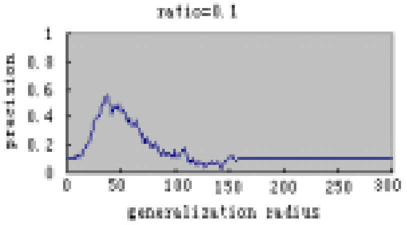

Fig.3 The influence of radius of generalization on classification precision of ExtendedCC4 (*ratio*=0.1)

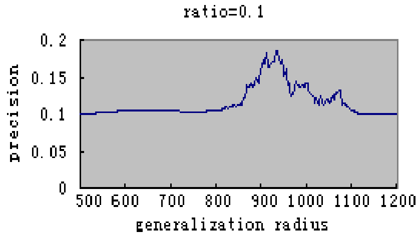

Fig.4 The influence of radius of generalization on classification precision of InitialCC4 (*ratio* = 0.1)

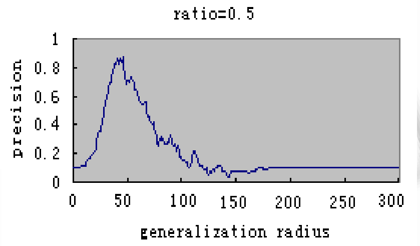

Fig.5 The influence of radius of generalization on classification precision of ExtendedCC4 (*ratio*=0.5)

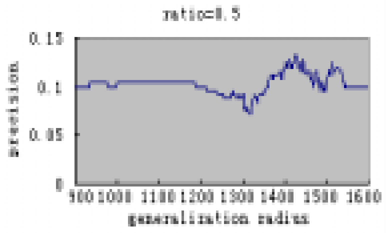

Fig.6 The influence of radius of generalization on classification precision of InitialCC4 (*ratio*=0.5)

Table 1 The expected and practical classification precision of ExtendedCC4 network when the radius of generalization reaches threshold value

No. of documents from the first news group	No. of documents from all other nine news group	Threshold value of radius of generalization	Expected precision	Practical precision
9	414	157	0.021739	0.021739
18	423	159	0.042553	0.042553
27	432	159	0.0625	0.0625
36	441	159	0.081633	0.081633
45	450	159	0.1	0.1

6 Conclusion and Further Work

When the input vectors of documents are almost the same size, InitialCC4 neural networks exhibit good performance. It would be impractical however to assume that all pieces of information on WWW is nearly of the same size. Our ExtendedCC4 aims to solve this problem by indexing documents in low dimensional space. The experiments show that the performance of ExtendedCC4 is much better than that of InitialCC4. Actually, the high classification precision of ExtendedCC4 relies on the effectiveness of indexing methods.

In the future, we will further study the integration of CC4 and incremental indexing method. Because in our ExtendedCC4, continuously valued data index can be discretilized as CC4 neural network input, we also plan to extend the method to other non-text data classification applications.

References:

[1] Brin, S., Page, L. Anatomy of a large scale hypertextual web search engine. In: Ashman, H., Thistlewaite, P., eds. Proceedings of the 7th International World Wide Web Conference. Amsterdam: Elsevier Science Publishers B.V., 1998. 107~117.

[2] Shu, B., Kak, S. A neural network-based intelligent meta search engine. Information Sciences, 1999,120(1):1~11.

[3] Gudivada, V.N. Raghavan, V.V., Grosky, W.I., *et al*. Information retrieval on the world wide web. IEEE Internet Computing, 1997, 1(5):59~68.

[4] Faloutsos, C. FastMap: a fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. In: Proceedings of the ACM SIGMOD Conference. New York: ACM Press, 1995. 163~174.

[5] Jagadish, H.V. A retrieval technique for similar shapes. In: Garcia-Molina, H., Jagadish, H.V., eds. Proceedings of the ACM SIGMOD Conference. New York: ACM Press, 1990. 208~217.

[6] Kruskal, J.B., Wish, M. Multidimensional Scaling. SAGE Publications, Beverly Hills, CA., 1978. 1~27.

基于扩展角分类神经网络的文档分类方法

陈恩红[1], 张振亚[1], 合源一幸[2], 王煦法[1]

[1](中国科学技术大学 计算机系,安徽 合肥 230027);

[2](东京大学 数学工程与信息物理系,东京 113-86-56,日本)

摘要: CC4 神经网络是一种三层前馈网络的新型角分类(corner classification)训练算法,原用于元搜索引擎 Anvish 的文档分类.当各文档之间的规模接近时,CC4 神经网络有较好的分类效果.然而当文档之间规模差别较大时,其分类性能较差.针对这一问题,本文意图扩展原始 CC4 神经网络,达到对文档有效分类的效果.为此,提出了一种基于 MDS-NN 的数据索引方法,将每一文档映射至 k 维空间数据点,并尽可能多地保持原始文档之间的距离信息.其次,通过将索引信息变换为 CC4 神经网络接受的 0,1 序列,实现对 CC4 神经网络的扩展,使其能够接受索引信息作为输入.实验结果表明对相互之间规模差别较大的文档,扩展 CC4 神经网络的性能优于原始 CC4 神经网络的性能.同时,扩展 CC4 神经网络的分类精度与文档索引方法有密切关系.

关键词: 文档分类;CC4 神经网络;数据索引;距离信息

中图法分类号: TP311 文献标识码: A