Available online at www.ijpe-online.com

International Journal of Performability Engineering

vol. 15, no. 4, April 2019, pp. 1141-1150
DOI: 10.23940/ijpe.19.04.p9.11411150

Four-Layer Feature Selection Method for Scientific Literature based on Optimized K-Medoids and Apriori Algorithms

Hongchan Li* and Ni Yao

School of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou, 450002, China

Abstract

With the increase in scientific literature, classifying scientific literature has become an important focus. Effectively selecting representative features from scientific literature has become a key step in scientific literature classification and influences the performance of scientific literature classification. According to the structural characteristics of scientific literature, we combine an optimized K-medoids algorithm, which firstly adopts information entropy to empower clustering objects to correct the distance function and then employs the corrected distance function to select the optimal initial clustering centres, with the Apriori algorithm to propose a four-layer feature selection method. The proposed feature selection method firstly divides scientific literature into four layers according to their structural characteristics, selects features layer by layer from the former three layers by means of the optimized K-medoids algorithm, subsequently mines the maximum frequent item sets from the fourth layer by the Apriori algorithm to act as the features of the fourth layer, and finally merges selected features of every layer and eliminates duplicate features to obtain the final feature set. Experimental results show that the proposed four-layer feature selection method achieves higher performance in scientific literature classification.

Keywords: feature selection; k-medoids algorithm; information entropy; apriori algorithm

(Submitted on November 11, 2018; Revised on December 14, 2018; Accepted on January 13, 2019)

1. Introduction

With the improvement of scientific literature retrieval techniques, more and more users are accustomed to retrieve scientific literature by means of various academic literature retrieval platforms and digital library systems. However, with the rapid increase in knowledge update speed, new themes, new concepts, and new disciplines are arising constantly. The amount and the types of information are also sharply increasing, causing the amount of scientific literature to grow exponentially every year [1]. Such a large amount of scientific literature often makes it time-consuming for users to retrieve their required literature. In this case, how to organize scientific literature effectively to meet the needs of users has become a research hotspot in the data mining field [2]. At present, many scientific literature retrieval platforms have classified scientific literature; for example, we can input the keyword "green network" in the China National Knowledge Internet (CNKI) and obtain 15,902 scientific literature records, of which 4,613 records are in the macroeconomic management and sustainable development field, 3,391 records are in the industrial economy field, 2,539 records are in the building science and engineering field, 1,461records are in the agricultural economy field, 2,419 are records in the computer software and computer application field, and so on. However, the current discipline domain classification of scientific literature is not precise enough; it may miss some literature according to users' retrieval requirements, and it is a great challenge to the current scientific literature retrieval technique.

Scientific literature mainly exists in the form of text, so scientific literature classification is text classification. Feature selection is the prerequisite for efficient text classification and is an important link of automatic text classification [3], and the performance of feature selection algorithm will directly affect the final effect of text classification.

At present, the Vector Space Model (VSM) is often used to describe texts [4], but if we adopt the obtained features by means of word segmentation algorithm and word frequency method to represent text vectors, then the dimensions of text

* Corresponding author.
E-mail address: zhuhaodong80@163.com

vectors will be much larger. The untreated text vectors will not only bring greater computational overhead for subsequent text classification and cause lower efficiency for the whole processing, but also damage the precision of text classification [5], so that it is difficult to obtain satisfactory classification results. Therefore, it is necessary to further purify text vectors to find the more representative text features of texts on the basis of guaranteeing the original meaning of texts. At present, the mainstream feature selection methods mainly contain the following kinds: mutual information [6], information gain [7], x^2 statistic method [8], expected cross entropy [9], document frequency, etc. When they are applied to ordinary text classification, the classification performance is better. However, because ordinary text and scientific literature have bigger differences in structural characteristics, when they are used for scientific literature classification, the classification performance is unsatisfactory.

Scientific literature mainly exists in the form of text and is composed of the title, abstract, key words, introduction, conclusion, and rest of the main body. The best representative parts of scientific literature are the title and keywords, followed by the abstract, introduction, conclusion, and rest of the main body. In this case, if we adopt the current mainstream feature selection methods and the relevant improved methods to select features from scientific literature, then the selected features may be unrepresentative. In order to more accurately select features from scientific literature, according to the structural characteristics of scientific literature, we put forward a four-layer feature selection method to select features layer by layer.

2. Relevant Basic Knowledge

2.1. K-Medoids Algorithm

K-medoids is a commonly used clustering algorithm based on a partitioning method [10]. Its core thought is that it selects cluster centres from the cluster data regions, divides the data objects into different clusters by means of iterative operation, and minimizes the target function, so that the generated clusters are as compact and independent as possible. The *K*-medoids algorithm is described as follows:

Input

K: The number of clusters.
D: The sample data set that contains *N* samples.

Output *K* clusters.

Step 1 Randomly select *K* samples as the initial clustering centres from *D*.

Step 2 Assign the rest samples to respective nearest clustering centre according to the distances between the rest of the samples and the clustering centres.

Step 3 Randomly select *K* samples from unrepresentative samples as the current interim clustering centres O_i.

Step 4 Calculate the current interim absolute error after O_i replaces O_j (O_j represents the precursor cluster centres).

Step 5 If the current interim absolute error is less than the precursor absolute error, then O_i formally replaces O_j as the current clustering centres, and *K* new clustering centres are formed.

Step 6 Go to step 2 until the termination condition is met.

Step 7 The *K*-medoids algorithm ends and outputs *K* clusters.

2.2. Apriori Algorithm

The Apriori algorithm is a frequent item set mining algorithm. Its core thought is that it generates the association relationship between data items by the candidate set generation and the downward sealing detection [11]. It firstly generates frequent 1-item sets, then generates frequent 2-item sets by the frequent 1-item sets, and continues to generate frequent *k*-item sets by the frequent (*k*-1)-item sets until the maximum frequent *k*-item sets are generated.

The Apriori algorithm adopts the following property to seek the maximum frequent *k*-item sets [12]:

Property 1 L_{k-1} is the frequent (*k*-1)-item sets, and X_k is a frequent *k*-item set. If the number of (*k*-1)-item subsets that contain X_k of L_{k-1} is less than *k*, then X_k cannot be the maximum frequent *k*-item set.

In order to improve the search efficiency of frequent item sets, the Apriori algorithm employs the following property to compress the search space [12]:

Property 2 X_k is a *k*-item set. If any (*k*-1)-item subset of X_k is not a frequent item set, then X_k is not a frequent item set.

3. Improved Initial Clustering Centres of K-Medoids

The traditional *K*-medoids algorithm randomly selects *K* data samples as the initial clustering centres and ignores the different clustering effect of attributes of data samples in the process of clustering, potentially leading to spotty initial clustering centres. Therefore, in order to obtain more reasonable initial clustering centres, we need to reprocess the initial clustering centres. The basic idea is as follows: we firstly equally divide the data sample set into $k_1(k_1 > K)$ data sample subsets and randomly select a sample from every data sample subset, and then we consider the k_1 selected samples as seed clustering centres to cluster and calculate the empowerment value σ_i of each category. Finally, we arrange the empowerment values in descending order and select the corresponding centres of the top *K* clusters as the initial clustering centres.

Information entropy refers to the emergence probability of specific information [13]. Given a data sample set $D = \{d_1, d_2, \cdots, d_q\}$ and the probability measure $p_i = P[X = d_i]$, the information amount of a data sample can be expressed as: $I(d_i) = \log\frac{1}{p_i}$. The average information amount (information entropy) of the data sample set *D* can be expressed as:

$$H(X) = \sum_i p_i \log\frac{1}{p_i}, \quad i = 1, 2, \cdots, n \tag{1}$$

In Equation (1), it is assumed that the data sample set $X = \{X_r \mid X_r \in R^m, r = 1, 2, \cdots, n\}$, $X_1 = (x_{11}, x_{12}, \cdots, x_{1j})$, $\cdots$, $X_i = (x_{i1}, x_{i2}, \cdots, x_{ij})$. We determine the attribute weight by information entropy as follows:

Step 1 Calculate the proportions of the attribute *j* of data sample *i*. We need to standardize the data and compress the data sample into the interval [0, 1]:

$$M_{ij} = x_{ij} / \sum_{i=1}^{n} x_{ij} \tag{2}$$

In Equation (2), x_{ij} is the attribute *j* of the data sample *i*, M_{ij} is the proportion of x_{ij}, $i = 1, 2, \cdots, n$, and $j = 1, 2, \cdots, m$.

Step 2 Calculate the information entropy of the j^{th} dimension:

$$H_j = -\frac{1}{\ln n} \sum_{i=1}^{n} M_{ij} \ln M_{ij} \tag{3}$$

In Equation (3), if $M_{ij} = 0$, then $M_{ij} \ln M_{ij} = 0$. For a given *j*, if x_{ij} are all equal, then $M_{ij} = x_{ij} / \sum_{i=1}^{n} x_{ij} = 1/n$. At this time, H_j reaches its maximum value.

Step 3 Calculate the weights of the *j* dimension attribute:

$$w_j = \frac{1-H_j}{\sum_{j=1}^{m} 1-H_j} \tag{4}$$

In Equation (4), $0 \le w_j \le 1$, $\sum_{j=1}^{m} w_j = 1$, and $j = 1, 2, \cdots, m$. For a given *j*, the smaller the H_j, the larger the w_j and the more important the attribute *j*.

Step 4 Calculate the similarity between the data samples by means of the empowerment Euclidean distance [14]:

$$d_\omega(x_a, x_b) = \sqrt{\sum_{j=1}^{m} w_j (x_{aj} - x_{bj})^2} \tag{5}$$

In Equation (5), $x_a = (x_{a1}, x_{a2}, \cdots, x_{am})$ and $x_b = (x_{b1}, x_{b2}, \cdots, x_{bm})$ represent the two data samples, and w_j is the weight of the attribute *j*. w_j enlarges or shrinks the similarity between the data samples and makes the attributes with larger weight values have a larger clustering effect and the attributes with smaller weight values have a smaller clustering effect.

Step 5 Employ the empowerment category target value function as the empowerment standard deviation function to calculate and arrange σ_i in descending order, and choose the corresponding centres of the top *K* values as the initial clustering centres. The empowerment category target value function is as follows [15]:

$$\sigma_i = \sqrt{\frac{\sum_{x_i \in T_j} d_\omega(x_i, c(T_j))}{|T_j| - 1}} \tag{6}$$

In Equation (6), $T_i (i = 1, 2, \cdots, K)$ represents the i^{th} cluster, σ_i is the empowerment standard deviation of T_i, and $c(T_i)$ is the corresponding clustering centre of T_i. $|T_j|$ is the number of data samples in T_i.

From Equation (6), we can see that the larger the value of σ_i, the greater the similarity of data samples in T_i, the more concentrated data samples in T_i, and the more representative the centre of T_i.

4. Optimized K-Medoids Algorithm

Based on the improved initial clustering centres, the optimized *K*-medoids algorithm is described as follows:

Input

A: The data sample set *D*.
k_1 : The number of centre.

Output *K* clusters.

Step 1 Select the value of *K* between 2 and $\sqrt{N}$, where *N* is the number of all data samples in the data space. Select *K* values one by one within the interval $[2, \sqrt{N}]$, use the clustering validity function to evaluate the clustering effect, and eventually obtain the optimal value of *K*.

Step 2 Use Equation (4) to calculate the weights of the attributes of the data sample.

Step 3 Equally divide the data sample set into $k_1 (k_1 > K)$ data sample subsets, randomly select a data sample from every data sample subset, and then consider the k_1 selected data samples as the seed clustering centres.

Step 4 Scan the data sample set according to the similarity between the data samples and the clustering centres that are obtained by Equation (5), and assign data samples to the respective nearest clustering centre.

Step 5 According to Equation (6), calculate and arrange $\sigma_i (i = 1, 2, \cdots, k_1)$ of k_1 clusters in descending order, and choose the corresponding centres of the top *K* values as the initial clustering centres.

Step 6 Assign the rest of the samples to the respective nearest clustering centre according to the distance between the rest of the samples and the clustering centres.

Step 7 Randomly select *K* samples from unrepresentative samples as the current interim clustering centres O_i.

Step 8 Calculate the current interim absolute error after O_i replaces O_j (O_j represents the precursor cluster centres) using Equation (7) [16]:

$$E_i = \sum_1^k \sum_{p \in C_i} | p - O_i | \tag{7}$$

In Equation (7), *p* is all the data samples of the cluster C_i and O_i is the centre of the cluster C_i.

Step 9 If the current interim absolute error E_i is less than the precursor absolute error E_i, then O_i formally replaces O_j as the current clustering centre, and *K* new clustering centres are formed.

Step 10 Go to step 7 until the termination condition is met.

Step 11 The *K*-medoids algorithm ends and outputs *K* clusters.

5. Proposed Four-Layer Feature Selection Method for Scientific Literature

5.1. The Process Framework

According to the structure characteristics of scientific literature, we put forward a four-layer feature selection method for scientific literature that adopts the improved *K*-medoids algorithm to the top three layers and applies the Apriori algorithm to the fourth layer. The process framework is shown in Figure 1.

The specific processes are as follows. (1) Scientific literature is divided into the following four layers: the title and the keywords form the first layer, the abstract forms the second layer, the introduction and the conclusion form the third layer, and the rest of the main body form the fourth layer. (2) The features from the first layer are obtained by segmenting words and deleting the stop-words as the features of the first layer. (3) Abstracts are translated into the Scientific Literature Corpus (A) by segmenting words and deleting the stop-word, and the improved *K*-medoids algorithm is adopted to get the *K* clusters based the features of the first layer from the Scientific Literature Corpus (A). Some representative features from each cluster are selected as the features of the second layer. (4) The introductions and the conclusions are translated into the Scientific Literature Corpus (B) by segmenting words and deleting the stop-word, and the improved *K*-medoids algorithm is adopted to get the *K* clusters based the features of the second layer from the Scientific Literature Corpus (B). Some representative features from each cluster are selected as the features of the third layer. (5) The rest of the main body is translated into the Scientific Literature Corpus (C) by segmenting words and deleting the stop-word, and the Apriori algorithm is adopted to mine the frequent *k*-item sets from the Scientific Literature Corpus (C). The mined frequent *k*-item sets are selected as the features of the fourth layer. (6) The features of each layer are merged, and duplicate features are eliminated to obtain the final feature set.

5.2. Selection Feature of the Top Three Layers

The title and the keywords of scientific literature are syncopated, and the stop-words are deleted to obtain the features of the first layer. The *K* value of the second layer is determined according to the number of features in the first layer.

The abstract of scientific literature is syncopated, the stop-words are deleted to get the Scientific Literature Corpus (A), the improved *K*-medoids algorithm is employed to obtain *K* clusters, and some reprehensive features are selected from each

cluster to get the features of the second layer. The *K* value of the third layer is determined according to the number of features in the second layer.

The introduction and the conclusion of scientific literature are syncopated, the stop-words are deleted to get the Scientific Literature Corpus (B), the improved *K*-medoids algorithm is employed to obtain *K* clusters, and some reprehensive features are selected from each cluster as the features of the third layer.

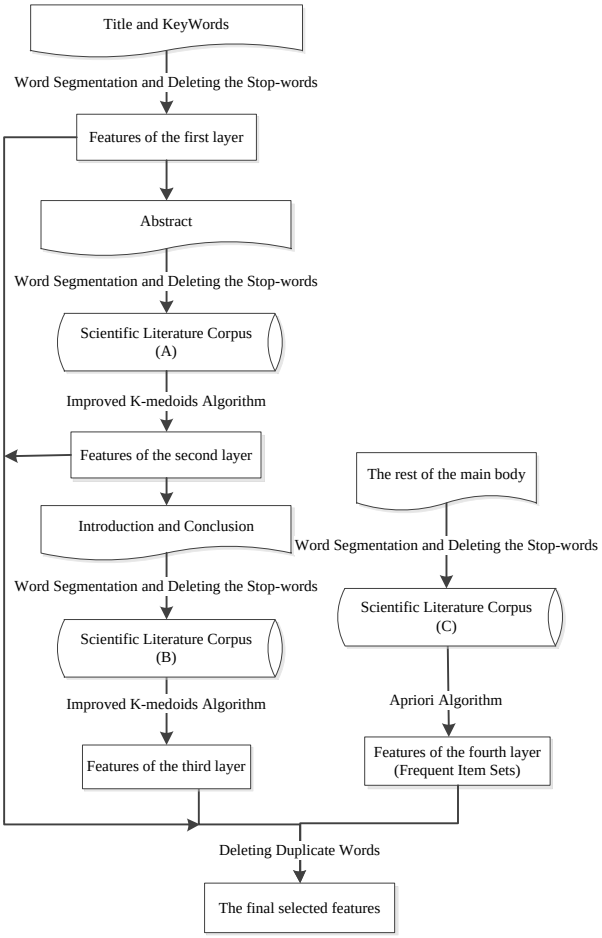

Figure 1. The process framework of proposed four-layer feature selection method for scientific literature

The following is an example of the feature selection of the second layer. The process is described as follows:

Step 1 Determine the *K* value of the second layer according to the number of features in the first layer.

Step 2 Adopt the $tf-idf$ factor commonly used in text categorization to empower the i^{th} feature t_i, the formula of $tf-idf$ is as follows [17]:

$$w_{ij}=\frac{tf_{ij}\times\log(\frac{n}{n_i}+0.01)}{\sum_{i=1}^{m}[tf_{ij}\times\log(\frac{n}{n_i}+0.01)]^2} \tag{8}$$

In Equation (8), n is the number of texts, n_i is the number of texts that contain the feature t_i, df_{ij} represents the word frequency of t_i in the text d_j, $tf_{ij}=\frac{df_{ij}}{\max(df_{ij})}$, and $\max(df_{ij})$ is the maximal frequency of the feature that appears in the text d_j.

Step 3 Adopt the improved *K*-medoids algorithm to obtain the corresponding cluster set $\{C_1, C_2, \cdots, C_K\}$.

Step 4 Calculate the average vector (cluster centre vector) X_i of vectors in the cluster C_i, i = 1, 2, ⋯, K.

Step 5 Calculate the similarity between features of clusters and the cluster centre vector. The formula is as follows [18]:

$$\cos\theta = \frac{\sum_{i=1}^{n} w_{1i} w_{2i}}{\sqrt{(\sum_{i=1}^{n} w_{1i})^2 (\sum_{i=1}^{n} w_{2i})^2}}, \quad 1 \le i \le n \tag{9}$$

In Equation (9), w_{1i} and w_{2i} represent two different vectors and n is the dimension of vector. According to the descending order, we respectively take K features corresponding to the top f values and employ $f \times K$ features to form a feature set of the second layer.

The introduction of scientific literature mainly introduces the application background of the related topic, research status, research achievements, deficiencies, etc., and the conclusion mainly summarizes and outlooks the related topics. The introduction and the conclusion of scientific literature are syncopated, and the stop-words are deleted to get the Scientific Literature Corpus (B). The process is similar to the process of feature selection of the second layer.

5.3. Selection Feature of the Fourth Layer

The rest of the main body of scientific literature is syncopated and the stop-words are deleted to get the Scientific Literature Corpus (C). Due to the large amount of data and the lower feature density in the rest of main body, it is suitable to use the frequent item set mining method (Apriori algorithm) that mines Boolean association rules to select features of the fourth layer. The specific steps are as follows:

Step 1 Scan the Scientific Literature Corpus (C) to generate frequent 1-item sets.

Step 2 Connect the frequent 1-item sets to generate frequent 2-item sets.

Step 3 Connect the frequent (*K*-1)-item sets to generate frequent *K*-item sets, in which *K* >= 3.

If the last elements of the two frequent (*K*-1)-item sets are different and their rest elements are the same, then the two frequent (*K*-1)-item sets can be connected to be a frequent *K*-item set; otherwise, the two frequent (*K*-1)-item sets are abandoned.

Step 4 Weed out infrequent *K*-item sets from candidate *K*-item sets.

If a (*K*-1)-item subset of a candidate *K*-item set is not in the frequent (*K*-1)-item sets, then the candidate *K*-item set is deleted.

Step 5 Scan the Scientific Literature Corpus (C), calculate the support of candidate *K*-item sets, compare it with the minimum support, and obtain the frequent *K*-item sets until the maximum item sets are generated; otherwise, go to step 3.

The mined frequent *K*-items are filtered by word frequency and are pruned by noun as the features of the fourth layer.

6. Comparison Experiments

6.1. Experimental Data

We download 4,600 scientific papers from the China National Knowledge Internet (CNKI), including 1,600 scientific papers of the “Big Data” theme, 1,200 scientific papers of the “Cloud Computing” theme, 1,000 scientific papers of the “Hadoop” theme, and 800 scientific papers of the “AI” theme. Then, we respectively extract half of the scientific papers from each theme to constitute the training data set, and the remaining 2,300 scientific papers form the test data set.

6.2. Experimental Settings

We use the Chinese lexical analysis system (ICTCLAS), which was developed by the Institute of Computing Technology of Chinese Academy of Sciences, to carry out word segmentation. We employ the Weka tool, which was developed at the University of Waikato, as a classification experiment platform (it includes a range of machine learning algorithms in the field of data mining, such as data preprocessing, classification, clustering, association rules, visualization, and so on, and can be downloaded from the following url: http://www.cs.waikato.ac.nz/ml/weka/). We also adopt the KNN classifier to perform the classification experiment and apply MATLAB 7.0 to implement numerical calculation.

The proposed Four-Layer Feature Selection Method (FLFS) is mainly compared with the following three feature selection methods: mutual information (MI), x^2 statistics (CHI), and information gain (IG). We employ the recall ratio, precision ratio, *F*-score [19], micro average F_1, and macro average F_1 to evaluate classification performance.

6.3. Experimental Results and Experimental Analysis

The recall ratio, precision ratio, and F-score can better reflect the overall effect of text classification. As can be seen from Table 1, the overall effect of these four feature selection methods for scientific literature in descending order is the proposed method > CHI > MI > IG.

Table 1. The experimental comparison results of recall ratio, precision ratio and *F* – score on four feature selection methods

		FLFS	MI	CHI	IG
Big data	Recall ratio	100.00%	74.52%	75.29%	74.21%
	Precision ratio	99.35%	66.49%	67.38%	65.91%
	F - score	99.67%	70.28%	71.12%	69.81%
Cloud computing	Recall ratio	100.00%	62.95%	68.47%	58.99%
	Precision ratio	99.78%	58.47%	61.23%	50.37%
	F - score	99.89%	60.63%	64.65%	54.34%
Hadoop	Recall ratio	100.00%	76.94%	82.71%	70.38%
	Precision ratio	99.89%	73.68%	77.59%	69.25%
	F - score	99.94%	75.27%	80.07%	69.81%
AI	Recall ratio	100.00%	74.19%	83.15%	72.47%
	Precision ratio	99.91%	69.42%	76.92%	64.29%
	F - score	99.95%	71.73%	79.91%	68.14%

The micro-average F_1 and the macro-average F_1 can better reflect the stability of text classification. We can see from Tables 2 and 3 that with the changing number of selected features, the micro-average F_1 and the macro-average F_1 both increase gradually and achieve a relatively stable level. The KNN classifier for the proposed method has the best performance in the selected top 900 features; the micro-average F_1 and the macro-average F_1 are about 96% and 93% respectively. The KNN classifier for MI has the best performance on the selected top 1900 features; the micro-average F_1 and the macro-average F_1 are about 77% and 71% respectively. The KNN classifier for CHI has the best performance on the selected top 1700 features; the micro-average F_1 and the macro-average F_1 are about 83% and 75% respectively. The KNN classifier for IG has the best performance on the selected top 2100 features; the micro-average F_1 and the macro-average F_1 are about 71% and 64% respectively. The overall stability of these four feature selection methods for scientific literature in descending order is the proposed method > CHI > MI > IG.

Table 2. The experimental comparison results of the micro average F_1 on four feature selection methods

The number of top features	FLFS	MI	CHI	IG
100	69.27%	21.01%	30.59%	20.67%
300	83.51%	28.59%	36.12%	28.01%
500	87.40%	37.99%	49.73%	33.45%
700	93.61%	50.37%	57.42%	45.19%
900	95.87%	62.49%	62.28%	53.06%
1100	95.86%	67.81%	69.49%	60.28%
1300	95.88%	73.55%	75.09%	65.17%
1500	95.87%	76.89%	80.33%	67.28%
1700	95.86%	77.15%	82.94%	68.59%
1900	95.87%	77.48%	82.93%	69.38%
2100	95.87%	77.49%	82.94%	70.95%
2300	95.88%	77.48%	82.93%	70.94%
2500	95.86%	77.48%	82.93%	70.95%
2700	95.86%	77.49%	82.93%	70.95%
3000	95.87%	77.49%	82.94%	70.95%

Table 3. The experimental comparison results of the macro average F_1 on four feature selection methods

The number of top features	FLFS	MI	CHI	IG
100	55.92%	23.78%	26.41%	18.96%
300	73.98%	30.99%	34.96%	27.35%
500	86.21%	45.21%	49.57%	36.58%
700	90.74%	53.69%	56.29%	46.19%
900	93.48%	60.19%	67.18%	53.28%
1100	93.49%	65.08%	70.04%	57.47%
1300	93.47%	67.32%	73.41%	60.09%
1500	93.47%	68.74%	74.69%	61.57%
1700	93.49%	69.88%	75.38%	63.17%
1900	93.47%	70.68%	75.37%	64.01%
2100	93.48%	70.68%	75.39%	64.19%
2300	93.47%	70.67%	75.37%	64.19%
2500	93.47%	70.67%	75.38%	64.20%
2700	93.48%	70.67%	75.39%	64.18%
3000	93.47%	70.68%	75.38%	64.19%

Through careful analysis, we find the reason lies in that the proposed method can select features layer by layer in a targeted manner according to the structural characteristics of scientific literature, and thus the obtained feature subset is more representative. MI only examines the occurring situation of a feature, and the CHI method considers the occurring probability and the not-occurring probability of a feature, so CHI is superior to MI. Because IG is extremely sensitive to the distribution of data samples, if it is used in the condition of the uneven distribution of data samples, the representation of the selected feature set is poorer. In this paper, the distribution of selected corpus is uneven, so IG is the worst.

7. Conclusions

For the classification of scientific literature, this paper proposes a feature selection method based on hierarchical logic. This method combines a four layer mining mode with the optimized *K*-medoids algorithm and Apriori algorithm. It not only solves the problem that the *K*-medoids algorithm cannot automatically determine the *K* value, but also avoids the impact of the initial clustering centres on the clustering effect. Thus, the performance of the proposed feature selection method has more obvious improvement. However, the proposed feature selection method is also affected by certain factors; for example, the *K*-medoids algorithm is sensitive to isolated point data, and the isolation point data can cause the clustering centres deviation to influence clustering results. Meanwhile, when the Apriori algorithm calculates the support of items, it needs to frequently scan scientific literature corpus. With the increase in literature corpus records, the scan will make the computer system's I/O overhead present geometric progressions. These are the questions we should consider in our future research.

Acknowledgements

The authors would like to thank the editors and the anonymous reviewers for their helpful comments and suggestions, which have improved the presentation. This work was supported in part by the Key Science Research Project of Colleges and Universities in Henan Province of China (No. 19A520009) and the National Science Foundation of China (No. 81501548).

References

1. X. F. Zhang and F. X. Kong, "Research on Scientific Literature Retrieval based on Semantic Concept Analysis," *Journal of Information Theory and Practice*, Vol. 39, No. 8, pp. 115-118, August 2016
2. Q. Li, W. J. Yang, and L. Tan, "Application Research on Constructing a Vector Space Model of Classification based on Thesaurus for the Judgment of Relevance of Chinese Literatures," *Library Journal*, Vol. 35, No. 12, pp. 32-40, December 2016
3. T. Sun, S. Y. Qian, and H. D. Zhu, "Feature Selection Method based on Category Correlation and Discernible Sets," *Journal of Computational Information Systems*, Vol. 11, No. 22, pp. 9687-9698, August 2014
4. M. H. Nguyen and F. D. L. Torre, "Optimal Feature Selection for Support Vector Machines," *Pattern Recognition*, Vol. 43, No. 3, pp. 584-591, March 2010
5. H. D. Zhu, H. C. Li, D. Wu, D. S. Huang, and B. Wang, "Feature Selection Method based on Feature Distinguishability and Fractal Dimension," *Journal of Information and Computational Science*, Vol. 36, No. 5, pp. 6033-6041, May 2015
6. A. Destrero, S. Mosci, C. D. Mol, A. Verri, and F. Odone, "Feature Selection for High-Dimensional Data," *Computational Management Science*, Vol. 6, No. 1, pp. 25-40, January 2009
7. S. Q. Wang and J. M. Wei, "Feature Selection based on Measurement of Ability to Classify Subproblems," *Neurocomputing*, Vol. 224, pp. 155-165, February 2017
8. S. R. Y. Leela, V. Sucharita, B. Debnath, and H. J. Kim, "Performance Evaluation of Feature Selection Methods on Large Dimensional Databases," *International Journal of Database Theory and Application*, Vol. 9, No. 9, pp. 75-82, September 2016
9. A. Rehman, K. Javed, and H. A. Babri, "Feature Selection based on A Normalized Difference Measure for Text Classification,"

Information Processing & Management, Vol. 53, No. 2, pp. 473-489, February 2017
10. D. Lacko, T. Huysmans, J. Vleugels, G. D. Bruyne, M. M. V. Hulle, J. Sijbers, and S. Verwulgen, "Product Sizing with 3D Anthropometry and K-Medoids Clustering," *Computer Aided Design*, Vol. 91, pp. 60-74, October 2017
11. Y. X. Shen, "Benefits Transfer Research of Public Companies Shareholders based on Apriori Algorithm," *Journal of Discrete Mathematical Sciences & Cryptography*, Vol. 20, No. 4, pp. 861-872, April 2017
12. R. Mamoon and K. Lovepreet, "Finding Bugs in Android Application using Genetic Algorithm and Apriori Algorithm," *Indian Journal of Science and Technology*, Vol. 9, No. 23, pp. 1-5, September 2016
13. D. T. Pele, O. E. Lazar, and A. Dufour, "Information Entropy and Measures of Market Risk," *Entropy*, Vol. 19, No. 5, pp. 226-244, May 2017
14. D. P. P. Mesquita, J. P. P. Gomes, A.H. S. Junior, and J. S. Nobreb, "Euclidean Distance Estimation in Incomplete Datasets," *Neurocomputing*, Vol. 248, pp. 11-18, July 2017
15. J. H. Liu, Y. J. Lin, M. L. Lin, S. X. Wu, and J. Zhang, "Feature Selection based on Quality of Information," *Neurocomputing*, Vol. 225, pp. 11-22, February 2017
16. A. Rai and S. H. Upadhyay, "Bearing Performance Degradation Assessment based on A Combination of Empirical Mode Decomposition and K-Medoids Clustering," *Mechanical Systems and Signal*, Vol. 93, pp. 16-29, September 2017
17. B. Tang, S. Kay, and H. B. He, "Toward Optimal Feature Selection in Naive Bayes for Text Categorization," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 28, No. 9, pp. 2508-2521, September 2016
18. T. Basu and C. A. Murthy, "A Supervised Term Selection Technique for Effective Text Categorization," *International Journal of Machine Learning and Cybernetics*, Vol. 7, No. 5, pp. 877-892, May 2016
19. R. H. W. Pinheiro, G. D. C. Cavalcanti, and I. R. Sang, "Combining Dissimilarity Spaces for Text Categorization," *Information Sciences*, Vol. 406, pp. 87-101, September 2017

Hongchan Li received her B.S. degree from Heilongjiang Bayi Agricultural University in 2007 and her M.S. degree from Sichuan University of Science and Engineering in 2010. She is currently a lecturer in the School of Computer and Communication Engineering at Zhengzhou University of Light Industry. Her major research interests include cloud computation, intelligence information processing, computing intelligence, and data mining.

Ni Yao received her M.S. degree from Wuhan University in 2012. She is currently a research assistant in the School of Computer and Communication Engineering at Zhengzhou University of Light Industry. Her major research interests include cloud computation, intelligence information processing, computing intelligence, and data mining.