IPSJ Journal Vol. 51 No. 9 1930–1939 (Sep. 2010)

Regular Paper

Clustering Large Sparse Text Data: A Comparative Advantage Approach

Jie Ji,[†1] Tony Y. T. Chan[†2] and Qiangfu Zhao[†1]

Document clustering is the process of partitioning a set of unlabeled documents into clusters such that documents within each cluster share some common concepts. To analyze the clusters easily, it is convenient to represent the concepts using some key terms. However, by using terms as features, text data is represented in a very high-dimensional vector space, and the computational cost is high. Note that the text data are of high sparsity, and not all weights in the centers are important for classification. Based on this observation, we propose in this study a comparative advantage-based clustering algorithm which can find out the relative strength between clusters, as well as keep and enlarge their strength. Since the vectors are represented by term frequency, the clustering results are more comprehensible compared with dimensionality reduction methods. Experimental results show that the proposed algorithm can keep the characteristic of k-means algorithm, but the computational cost is much lower. Moreover, we also found that the proposed method has a higher chance of getting better results.

1. Introduction

Document clustering is the process of partitioning a set of n unlabeled documents into k categories or clusters. For example, a Web search engine often returns thousands of pages in response to a broad query, making it difficult for users to browse or to identify relevant information. Clustering methods can be used to automatically group the retrieved documents into a list of meaningful categories.

There are two major issues in unsupervised learning research. The first one is how to optimize clustering over the number of clusters, and the second is how to find a good partition with a fixed number of clusters. The first problem is, in general, a very hard problem. Instead of trying to find the optimal solution, we usually find sub-optimal ones using heuristics. The global k-means algorithm (GKMA) is one of the heuristic algorithms used for this purpose[1]. In this algorithm, the number of clusters varies from 1 to K. After finding the centroid for only one cluster, for each k ($k = 2, 3, \ldots, K$), the previous $k-1$ centroids are fixed and a new centroid is selected by examining all data points. This clustering procedure is the second problem. It is known that GKMA is relatively independent of the initial partitions. However, since k-means are used many times, the computational cost is very high. Our purpose is to find a more efficient clustering algorithm to replace k-means. Once the second problem is solved, the first problem can also be solved.

Note that the core procedure in this mechanism could be replaced with any existing clustering algorithms. Existing clustering algorithms can be roughly classified into partitional and hierarchical[2]. The hierarchical algorithms can further be divided into divisive and agglomerative. For the document clustering problem, the number of data points n may be very high and the agglomerative algorithms are usually time consuming because the computational cost is proportional to mn^2.

The well-known k-means algorithm is a partitional algorithm. It performs well on document clustering[3]–[7]. It uses a heuristic search algorithm known as Lloyd's algorithm[8], which is described as follows:

- Step 1: Initialize the clusters.
- Step 2: Calculate the centers from the current partition.
- Step 3: Use the centers to obtain a new partition.

The algorithm will stop if there is no more change in the centers; otherwise, go to step 2. The time complexity of k-means algorithm is $O(rkmn)$, where r is the number of iterations. If $rk < n$, k-means is more efficient than agglomerative hierarchical algorithms. Partitional algorithms can be embedded in the divisive hierarchical algorithms. For example, if we use k-means in the divisive hierarchical algorithm, and if the data are divided into two groups in each hierarchy, the computational cost is proportional to $rmn \log n$ (the constant 2 is omitted here). Thus, if $r \log n < n$, divisive algorithms can be more efficient than agglomerative algorithms.

[†1] The University of Aizu
[†2] The University of New Brunswick, Canada

© 2010 Information Processing Society of Japan

There are some variations of the k-means algorithm to accelerate the clustering speed based on the reduction of n and m [9),10)]. Parallel k-means may also be considered as a speedup method if many processing units are available [11)].

For document clustering, data points are usually presented in terms $d = (v_1, v_2, \ldots, v_m)^t$ where v_i is the term frequency of the ith term in this document. To present documents in this way, even for a moderately-sized set of documents, the dimensionality could be several thousand. However, document data sets naturally have the property of high sparsity, e.g., 95% sparsity [6)]. Our research shows that to define a document's label, not all weights in all centers are necessary [12)]. In this study, we propose a comparative advantage-based algorithm that can compress the k by m weight matrix into an m dimensional indexed weight set, and the cost can be reduced to $O(rmn)$. Since the centers could own the terms exclusively, the important key terms in the clusters can be expected to be more distinct.

The original purpose of this research is to make the clusters as different as possible. Since the computational cost is reduced, the proposed algorithm could also accelerate clustering speed significantly. Moreover, through experiments we also found that the proposed method has a higher chance of producing better results.

2. Comparative Advantage

In this section, we introduce an algorithm based on comparative advantage. The basic idea of comparative advantage is to find the major components in each cluster, and to keep and enlarge them.

2.1 Term Distribution

Comparative advantage is based on the concept of term distribution. Basically, the term distribution for the jth cluster C_j is an m-dimensional vector with the ith element being the frequency of the ith term in this cluster.

To clarify the concept, let's take an example. There are 4 document vectors in **Table 1**. We want to partition these 4 documents into 2 clusters. For example, let the first 2 documents assign to C_1, and the last 2 assign to C_2. We will get two term distributions as shown in **Table 2** by summing up the term frequency for all terms in each cluster. If we put *doc*1 and *doc*3 into C_1, and *doc*2 and *doc*4 into C_2, then we get another two term distributions as shown in **Table 3**.

Table 1 Term matrix for the example.

	birth	liver	life	steam	shock	speed
*doc*1	1	1				
*doc*2	1		1			
*doc*3				1		1
*doc*4				1	1	1

Table 2 Term distributions of the two clusters in the example.

	birth	liver	life	steam	shock	speed
Distribution1	2	1	1	0	0	0
Distribution2	0	0	0	2	1	2

Table 3 Term distributions corresponding to another partition.

	birth	liver	life	steam	shock	speed
Distribution1	1	1	0	1	0	1
Distribution2	1	0	1	1	1	1

The term distributions can be obtained straightforwardly once a partition is given. We represent a partition using a number string of length n, where n is the number of documents. Each number in the string takes value from $[1, k]$. If the ith number of a string is j, this means that the ith document belongs to the jth cluster.

For the above example, the first partition can be represented as [1 1 2 2]. That is, the first two documents are assigned to the first cluster, and the last two are assigned to the second cluster. The second partition is [1 2 1 2], whose meaning can be interpreted straightforwardly.

Our goal is to find a partition so that each cluster uses terms that are as different as possible. In this sense, the partition given in Table 2 is better than the one given in Table 3. This is because in Table 2, "birth", "liver" and "life" are in the same cluster, indicating that this cluster is mainly talking about medical technology, and the second cluster contains terms "steam", "shock" and "speed", which indicates that this is an aeronautical system cluster.

© 2010 Information Processing Society of Japan

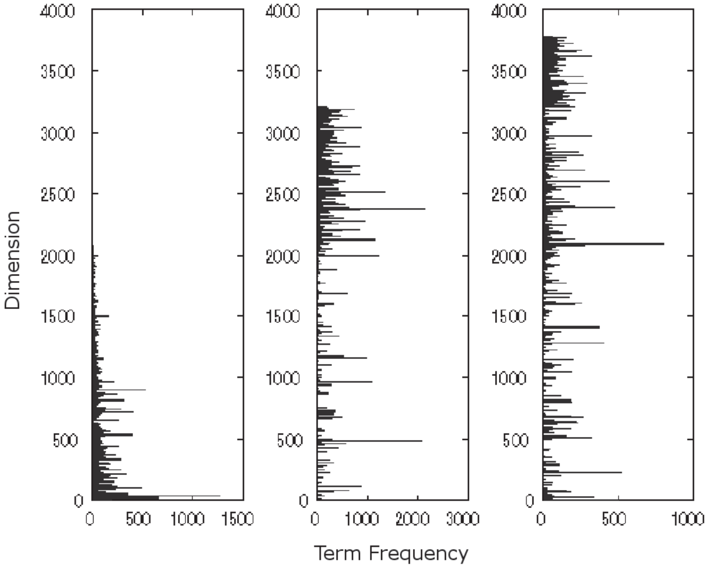

Fig. 1 Term distributions initialized according to teacher signal.

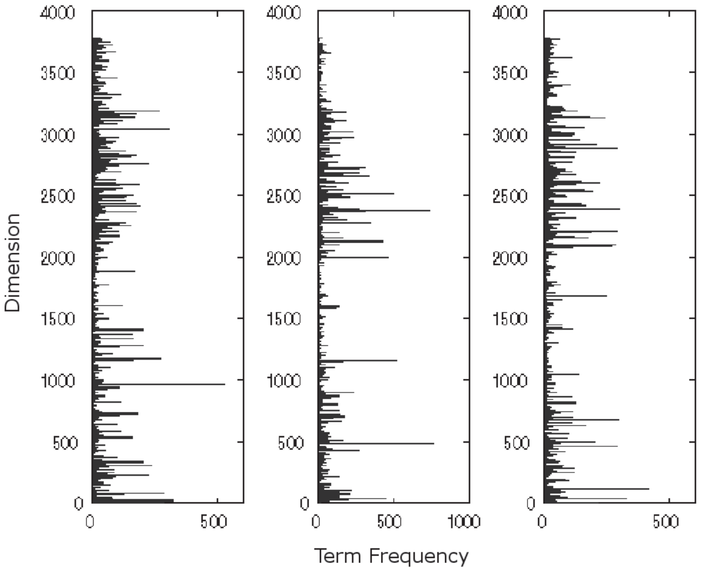

Fig. 2 Term distributions initialized randomly.

Figure 1 shows graphically the result of term distributions of CISI, MEDLINE and CRANFILED of database CLASSIC3. We will give a description of this data collection in Section 3. **Figure 2** shows term distributions by randomly assigning the cluster label for each document. Intuitively speaking, the term distributions initialized by using class label are much better than the second shown in Table 3, because we can not find out distinguishing features from the second term distributions.

2.2 Classification Method

From Fig. 1 we can find out that different clusters feature different terms. Some terms' weights in a certain term distribution are bigger than others. If the weight of ith term in term distribution j is larger than others, term i is the advantage of term distribution j. That is to say, for any given document, its class label definition should satisfy and keep this kind of "advantage".

However, only big weight value does not mean it is an advantage absolutely. Here we introduce the concept of *comparative advantage*[13]. It is a law in economics. The definition is:

The law of comparative advantage refers to the ability of a party (an individual, a firm, or a country) to produce a particular good or service at a lower marginal cost and opportunity cost than another party. It is the ability to produce a product most efficiently given all the other products that could be produced[14),15)]. It can be contrasted with absolute advantage which refers to the ability of a party to produce a particular good at a lower absolute cost than another.

For example, it is obvious that China can earn more money than Switzerland on tourism each year, this is a big weight. However, if the money is divided by China and Switzerland's GDP, we can find that Switzerland can get a higher ratio. Which means Switzerland's tourism industry is comparatively stronger than China and should develop tourism more.

If we take countries' various industry incomes as term distributions, the basic idea of comparative advantage is to scale the term distribution by the total sum of the corresponding cluster, so that the clusters with different sizes can

© 2010 Information Processing Society of Japan

compete equally. Specifically, the term distribution $T_j = (w_{1j}, w_{2j}, \ldots, w_{mj})$ of each cluster is normalized as follows:

$$w'_{ij} = \frac{w_{ij}}{\sum_{i=1}^{m} w_{ij}} = p(t_i \mid C_j), \quad i = 1, 2, \ldots, m; j = 1, 2, \ldots, k \tag{1}$$

where w_{ij} is the term frequency of the term t_i in the cluster C_j, m is the number of terms in the data set and k is the number of clusters. The comparative advantage of the term t_i in the cluster C_j is actually equal to $p(t_i \mid C_j)$, which is the probability that the term t_i appears in the cluster C_j. If

$$I_i = \arg \max_{1 \le j \le k} p(t_i \mid C_j) = \arg \max_{1 \le j \le k} w'_{ij} \tag{2}$$

we say that the I_ith cluster is the most advantageous with regard to the term t_i, compared with other clusters. In this case, I_i is called the *cluster index* of the term t_i. For example, suppose $k = 3$, and the weights of the fifth term "computer" are 0.001, 0.002 and 0.011, respectively for the three cluster centers. The weight $w'_{3,5} = 0.011$ is the largest, and thus the cluster index I_5 of this term is 3.

Based on the above definition, we can represent the k cluster centers using one *indexed weight set* defined by:

$$\Omega = \{(I_1, W_1), (I_2, W_2), \ldots, (I_m, W_m)\} \tag{3}$$

where

$$W_i = w'_{i,I_i}, \quad i = 1, 2, \ldots, m. \tag{4}$$

If we define the *index set* A_j for the jth cluster as follows:

$$A_j = \{\ i\ | 1 \le i \le m; I_i = j\} \tag{5}$$

we can find the similarity between any given datum d and the cluster C_j using:

$$S(d, C_j) = \sum_{i \in A_j} v_i W_i. \tag{6}$$

From the definition of I_i, we can see that A_j has the following property:

$$\sum_{j=1}^{k} |A_j| = m \tag{7}$$

where $|\ |$ is the cardinality of a set. Thus, for any datum d, only m number of calculations are needed for finding the similarity between d and all cluster centers.

Classification of a datum d is based on the above defined similarity. That is, if

$$j_0 = \arg \max_{1 \le j \le k} S(d, C_j) \tag{8}$$

d is classified to the j_0th cluster. Note that although only part of the features are used for finding the similarity (see Eq. (6)), the decision is still based on the nearest neighbor rule.

2.3 Clustering Method

Based on the above discussions, we propose a weighted comparative advantage based algorithm (WCA). The algorithm also follows the same Lloyd's learning procedure as k-means[8], and is described as follows:

- Step 1: Initialize the clusters.
- Step 2: Calculate the indexed weight set and index sets of each clusters from the current partition.
- Step 3: Use the indexed weight set and index sets to get a new partition.

The program will stop if any cluster label does not change any more, otherwise return to Step 2. Note that in Step 1, we may initialize the clusters at random, or select documents randomly as the initial term distributions. Intuitively speaking, after clustering, all documents in the same cluster will share the same key terms.

3. Experimental Results

3.1 General Considerations

To verify the proposed method, we obtained two subsets named as CLASSIC3 and NSF3. CLASSIC3 is a subset from SMART system — one of the most popular test beds where the vector space based algorithm is successfully implemented. It contains 3,893 documents by merging the MEDLINE, CISI and CRANFILED sets. MEDLINE contains 1,033 abstracts from medical journals, CISI contains 1,460 abstracts from information retrieval papers, and CRANFIELD contains 1,400 abstracts from aeronautical systems papers[16]. NSF3 contains 4,303 abstracts of the grants awarded by the Nation Science Foundation. We collected 846 abstracts from astronomy area, 1,954 abstracts from biology area and 1,503 abstracts from computer science area[17].

We removed the title and author and kept abstract information. Then the documents were changed into vectors as in Section 2. After morphological anal-

© 2010 Information Processing Society of Japan

Table 4 Properties of data sets.

	Number of document	Truncation	Dimension
CLASSIC3	3,893	$df/n > 0.2\%$	3,780
Original CLASSIC3	3,893	no	19,929
MEDLINE	1,033		
CISI	1,460		
CRANFILED	1,400		
NSF3	4,303	$df/n > 0.2\%$	5,392
Original NSF3	4,303	no	18,721
ASTRONOMY	846		
BIOLOGY	1,954		
COMPUTER	1,503		

ysis[18], removing common English stop words and merging mutual terms[19], the CLASSIC3 collection contained 19,929 unique terms and the NSF3 collection contained 18,721 unique terms. Then we use a naive truncation method to reduce the dimensionality: To remove the terms whose document frequency is less than 8 (roughly 0.2% of the documents). After dimension reduction, the dimensionality of CLASSIC3 and NSF3 are reduced to 3,780 and 5,392, respectively. **Table 4** shows the properties of data sets.

In this study we mainly concern about given number of clusters, how to find a good partition. Here spherical k-means algorithm is used for comparison[20]. Before applying k-means, document vectors are normalized so that the $norm = 1$[21]. We clustered data to 3 clusters and use criterion functions to evaluate the results, respectively.

For each data collection, we conducted 500 runs and calculated the average value. Since both of the proposed algorithm and k-means algorithm are using Lloyd's searching algorithm, they are initialization sensitive. In order to keep consistency, for each run, they used the same randomly initialized point; that is, to select k documents at random as the centers.

The confusion matrix in **Table 5** shows that the 3 clusters $\pi^\dagger_1$, $\pi^\dagger_2$ and $\pi^\dagger_3$ produced by WCA clustering algorithm. From the tables we can see that the proposed method clusters the given document set in a reasonable way.

Table 5 Confusion matrix of 3 clusters on CLASSIC3.

	CISI	CRANFILED	MEDLINE
$\pi^\dagger_1$	1,448	13	21
$\pi^\dagger_2$	8	1,386	6
$\pi^\dagger_3$	4	1	1,006
Total	1,460	1,400	1,033

Table 6 Confusion matrices of 16 clusters on CLASSIC3.

	CISI	CRANFILED	MEDLINE
$\pi^\dagger_1$	2	0	196
$\pi^\dagger_2$	1	497	3
$\pi^\dagger_3$	415	4	0
$\pi^\dagger_4$	2	0	285
$\pi^\dagger_5$	4	396	1
$\pi^\dagger_6$	0	177	2
$\pi^\dagger_7$	362	0	0
$\pi^\dagger_8$	62	10	9
$\pi^\dagger_9$	0	0	208
$\pi^\dagger_{10}$	6	3	184
$\pi^\dagger_{11}$	0	0	132
$\pi^\dagger_{12}$	108	5	2
$\pi^\dagger_{13}$	205	0	1
$\pi^\dagger_{14}$	256	2	1
$\pi^\dagger_{15}$	37	52	6
$\pi^\dagger_{16}$	0	254	3
Total	1,460	1,400	1,033

The survey precision, survey recall, mutual exclusion rates (MER) and CPU time per iteration are used for criterion functions in this study. To compute precision and recall, each cluster is assigned to the class which is most frequent in the cluster. The class of cluster i is ω_i. For example, cluster $\pi^\dagger_2$ in Table 5 is assigned as cluster 2, which means $\omega_2 = 2$, and cluster $\pi^\dagger_{12}$ in **Table 6** is assigned as class 1, which means $\omega_{12} = 1$. Then the survey precision and recall rate for whole data set are calculated as:

$$P = \frac{\sum_{i=1}^{k} p_i}{k} \tag{9}$$

© 2010 Information Processing Society of Japan

$$R = \frac{\sum_{j=1}^{c} r_j}{c} \tag{10}$$

where k is the number of clusters, c is the number of classes from the teacher signal, p_i is the precision rate of the cluster i, r_j is the recall rate of the class j. A confusion matrix M is a $k \times c$ dimensional matrix with the elements m_{ij} corresponding to the number of documents with class label j belong to cluster i. Based on this matrix, precision of cluster i is calculated as:

$$p_i = \frac{\max m_{ij}}{n_i} \tag{11}$$

and recall of class j is calculated as:

$$r_j = \frac{\sum_{i=1,\omega_i=j}^{k} m_{ij}}{n_j^*} \tag{12}$$

where n_i is the number of documents in cluster i, n_j^* is the number of documents in class j, respectively.

Table 6 shows the clustering result of 16 clusters case. Let's take it as an example. Here $k = 16$, $c = 3$, the precision of cluster 1 $p_1 = \frac{196}{2+196} = 0.9899$ and the precision of cluster 15 $p_{15} = \frac{52}{37+52+6} = 0.5474$. For class 1, the recall rate $r_1 = \frac{415+362+62+108+205+256}{1{,}460} = 0.9644$. The survey precision P is 0.9449 and the survey recall R is 0.9734.

The mutual exclusion rate is a simple way to evaluate the quality of partition. It is defined as:

$$MER = \frac{\left|\bigcup_{i=1}^{k} \Psi_i\right| - N}{(k-1)N} \tag{13}$$

where Ψ_i is the set of top-N important terms for the ith cluster (for example, top-10 weight terms), and N is the number of top important terms used in each cluster, $N < m$. In our experiments, 1% top important terms were chosen, which means top-38 key terms for CLASSIC3 and top-54 key terms for NSF3, respectively. If MER equals to 0, all clusters will share the same key terms. If MER equals to 1, the sets of important key terms in all clusters are completely different.

3.2 Experimental Results

The general experimental results are given in **Table 7**. Both WCA and k-means can get partitions with high precision and recall rates. However, WCA is much faster than k-means algorithm.

For statistic analysis, let us take the precision as the example. For CLASSIC3, although the average value of the precision rates obtained by WCA is slightly better than that obtained by k-means, the result of t-test ($p\text{-}value = 0.18$) showed that this difference is not significant. For NSF3, however, the p-$value$ obtained

Table 7 General experimental results.

	Precision	Recall	MER	CPU Time
WCA-CLASSIC3	0.9492	0.9259	0.9742	3.6646
k-means-CLASSIC3	0.9344	0.9106	0.9689	8.8927
WCA-NSF3	0.9241	0.9121	0.9335	5.005
k-means-NSF3	0.9043	0.8503	0.9187	16.706

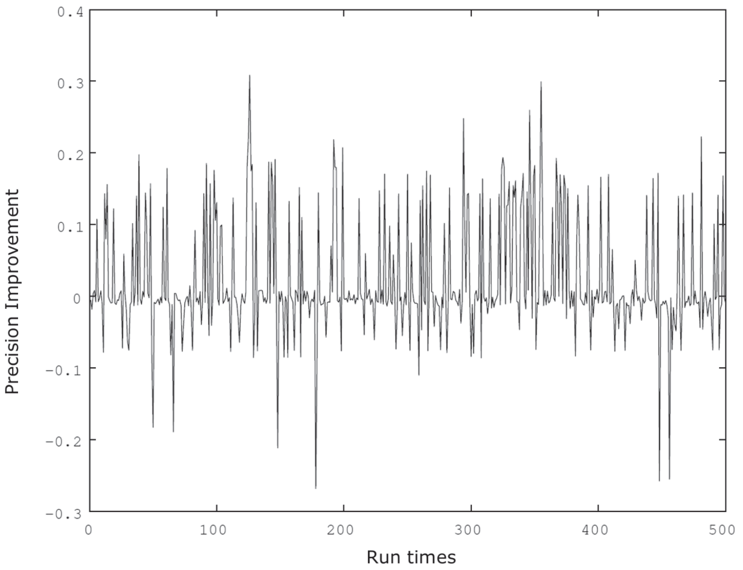

Fig. 3 Precision improvements of CLASSIC3.

© 2010 Information Processing Society of Japan

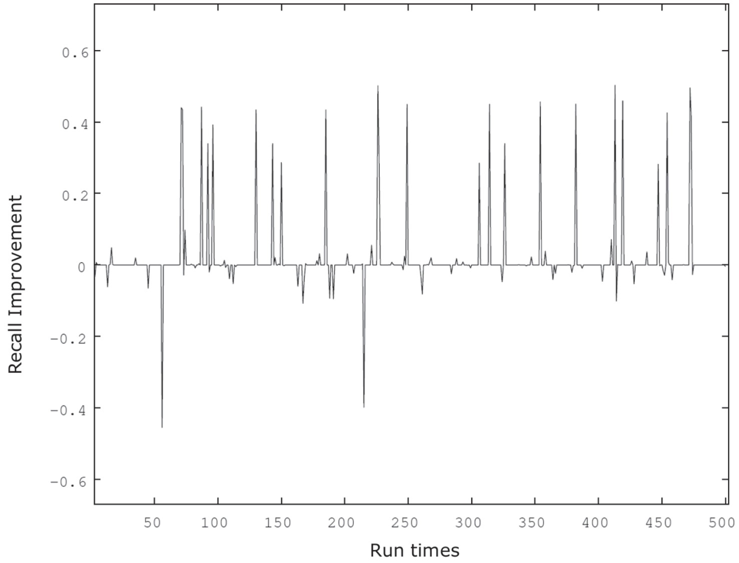

Fig. 4 Recall improvements of CLASSIC3.

by t-test is 0.003, which means that there is big difference between WCA and k-means, and WCA is significantly better.

To give out intuitive images, we use improvement to evaluate the results. Since both of the two algorithms are initialization sensitive, from a random point, they are easily to fall into local optima. If WCA can generate a better result than k-means, we say that WCA is the winner; otherwise, k-means is the winner. A positive 40% precision improvement means $P_{WCA} - P_{k\text{-}means} = 40\%$, e.g., P_{WCA} may be 98.5% and $P_{k\text{-}means}$ is 58.5%, which means WCA gets a much better result than k-means algorithm. **Figures 3–6** show 500 runs' improvements of precision rate and recall rate generate by WCA. The positive improvements in these figures are much more than negative improvements. We can also see this point from **Tables 8** and **9**. Clearly, for NSF3, WCA outperforms k-means in much more runs.

The proposed method has a higher chance to generate a better partition from a bad initialization point. It may due to the proposed method only uses selected

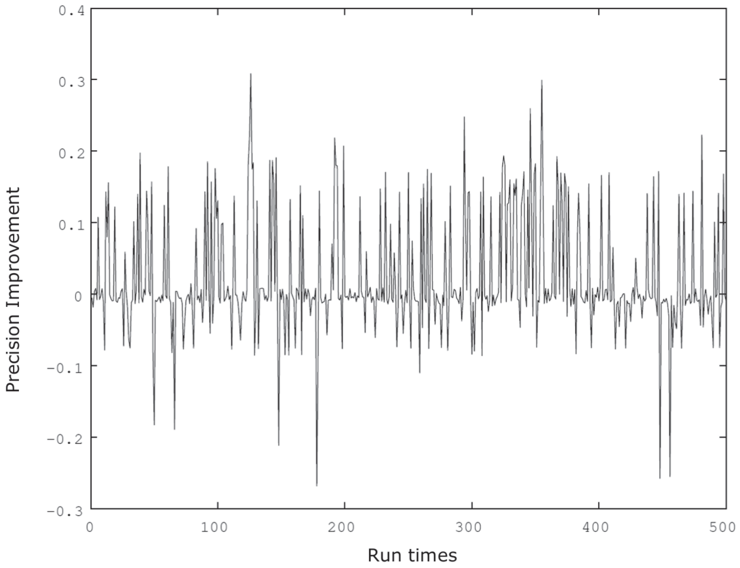

Fig. 5 Precision improvements of NSF3.

Table 8 Winning times of CLASSIC3.

Percent of	Precision		Recall	
Improvement	WCA	k-means	WCA	k-means
> 10%	38	7	23	4
> 20%	18	3	23	2
> 30%	8	1	19	2

Table 9 Winning times of NSF3.

Percent of	Precision		Recall	
Improvement	WCA	k-means	WCA	k-means
> 10%	98	7	107	6
> 20%	8	4	103	5
> 30%	1	0	96	0

© 2010 Information Processing Society of Japan

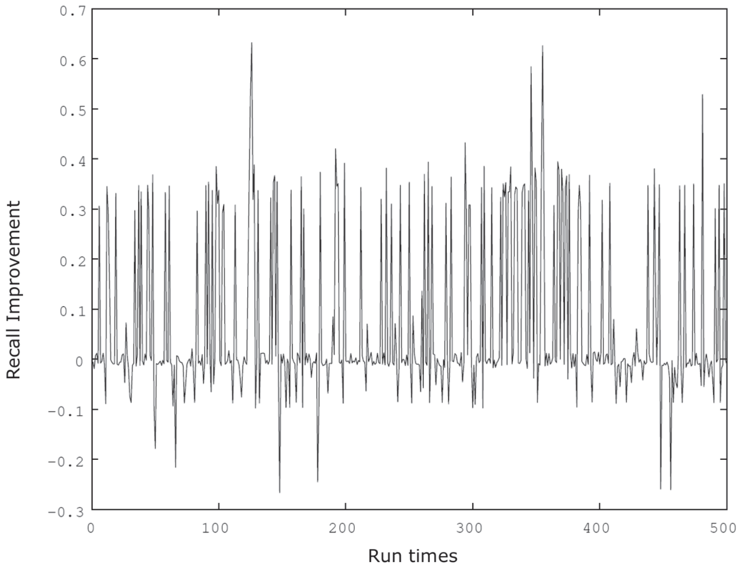

Fig. 6 Recall improvements of NSF3.

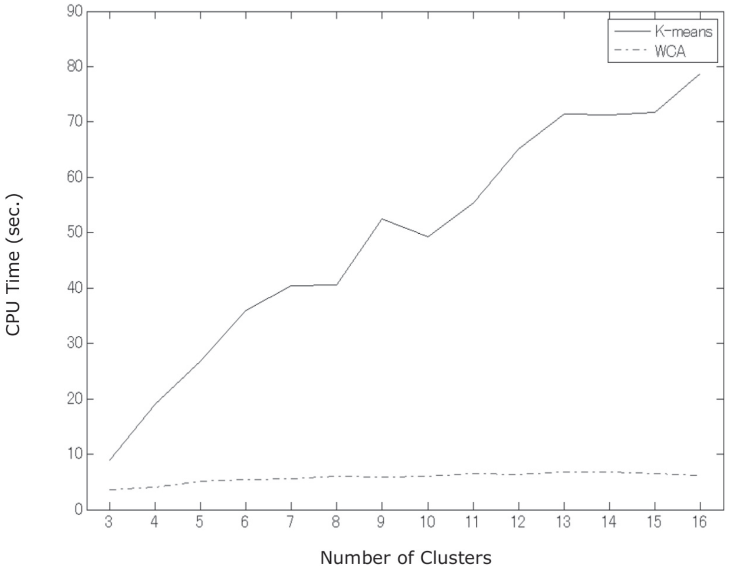

Fig. 7 CPU time of CLASSIC3.

terms to classify documents to k clusters. For each cluster, there are m/k dimensions on average to define a class label. Compared with the high dimensionality of term-space, the number of cluster k is a small enough number. Enough information will be assigned for each cluster. The proposed method made the best use of the sparsity of text data. This kind of dimension reduction only keeps major terms for each cluster.

Since both WCA and k-means use Lloyd's algorithm, the average iteration time of proposed algorithm is similar to k-means. However, the CPU time is different. **Figures 7** and **8** show the results of total CPU time in seconds vs. the number of clusters. As we know, for most clustering algorithms, the major computation involves comparison of two vectors. In WCA, the computation cost for one iteration is $O(mn)$, since one m dimensional indexed weight set represents all $k \times m$ weights. Theoretically speaking, the proposed method can reduce the $O(k)$ time in the classification phase. Thus, faster classification is possible using the partial similarity based classification.

For 3 cluster case, the proposed method accelerated about 2.3 times faster than k-means; and about 15.6 times faster than k-means for 16 cluster case in each iteration. Note that the CPU time in Table 7 is the total CPU time, it is the product of CPU time per iteration and the number of iterations.

4. Conclusion

In this paper, we have proposed a fast clustering algorithm based on weighted comparative advantage (WCA) for text data. The proposed algorithm can figure out the major terms in a text collection, and try to find reasonable clusters by keeping and enlarging these kinds of advantages. Compared with k-means, it extracts indexed weight set from k centroids, resulted in a k time faster classification time. Moreover, through experiments we also found that the proposed method has a higher chance to get better results. The average precision and recall rate of the proposed method are higher than the k-means algorithm.

As a future study, we would like to propose more efficient and effective search

© 2010 Information Processing Society of Japan

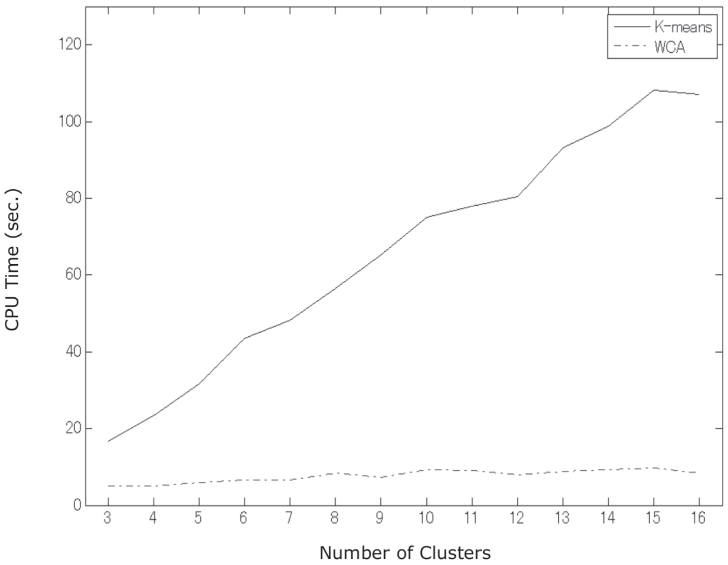

Fig. 8 CPU time of NSF3.

algorithms to get better clustering results. The proposed algorithm will be applied for dimension reduction for text data. The performance will be compared with SVD, PCA or some other dimension reduction methods.

References

1) Likas, A., Vlassis, N. and Verbeek, J.: The global K-means clustering algorithm, *Pattern Recognition*, Vol.36, No.2, pp.451–461 (2003).

2) Jain, A.K. and Dubes, R.C.: *Algorithms for Clustering Data*, Prentice Hall, Englewood Cliffs, New Jersey (1988).

3) MacQueen, J.B.: Some Methods for classification and Analysis of Multivariate Observations, *Proc. 5th Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley, University of California Press, pp.281–297 (1967).

4) Hartigan, J.A.: *Clustering Algorithms*, Wiley (1975).

5) Hartigan, J.A. and Wong, M.A.: A K-Means Clustering Algorithm, *Applied Statistics*, Vol.28, No.1, pp.100–108 (1979).

6) Dhillon, I.S. and Modha, D.S.: Concept Decompositions for Large Sparse Text Data Using Clustering, *Machine Learning*, Vol.42, No.1-2, pp.143–175 (2001).

7) Park, H., Jeon, M. and Rosen, J.B.: Lower Dimensional Representation of Text Data Based on Centroids and Least Squares, *BIT Numerical Mathematics*, Vol.43, No.2 (2003).

8) Lloyd, S.P.: Least Squares Quantization in PCM, *IEEE Trans. Information Theory*, Vol.28, No.2, pp.129–137 (1982).

9) Fodor, I.K.: A Survey of Dimension Reduction Techniques, Technical Report UCRL-ID-148494, Lawrence Livermore Nat'l Laboratory, Center for Applied Scientific Computing (June 2002).

10) Kanungo, T., Mount, D., Netanyahu, N., Piatko, C., Silverman, R. and Wu, A.: An efficient K-means clustering algorithm: Analysis and implementation, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.24, No.7, pp.881–892 (July 2000).

11) Stoffel, K. and Belkoniene, A.: Parallel K-means clustering for large data sets, *Proc. EuroPar'99 Parallel Processing*, pp.1451–1454 (1999).

12) Ji, J. and Zhao, Q.: Comparative Advantage Approach for Sparse Text Data Clustering, *Proc. IEEE 9th International Conference on Computer and Information Technology*, Xiamen, China, pp.3–8 (2009).

13) Hardwick, P., Khan, B. and Langmead, J.: *An Introduction to Modern Economics*, 5th Ed., Financial Times/Prentice Hall (1999).

14) Salton, G. and McGill, M.J.: *Introduction to Modern Information Retrieval*, McGraw-Hill Book Company (1983).

15) O'Sullivan, A. and Sheffrin, S.M.: *Economics, Principles and Tools*, 3rd Ed., Prentice Hall (2002).

16) ftp://ftp.cs.cornell.edu/pub/smart

17) http://www.nsf.gov/awardsearch

18) Bauer, L.: *Introducing linguistic morphology*, 2nd Ed., Georgetown University Press, Washington, D.C. (2003).

19) Frakes, W.B. and Baeza-Yates, R.: *Information Retrieval: Data Structures and Algorithms*, Prentice Hall, Englewood Cliffs, New Jersey (1992).

20) Dhillon, I.S. and Modha, D.S.: Concept Decompositions for Large Sparse Text Data Using Clustering, *Machine Learning*, Vol.42, pp.143–175 (Jan. 2001), doi: 10.1023/A:1007612920971.

21) Salton, G. and Buckley, C.: Term-weighting approaches in automatic text retrieval, *Information Processing and Management*, Vol.24, No.5, pp.513–523 (1988).

(Received December 8, 2009)
(Accepted June 3, 2010)

(Original version of this article can be found in the Journal of Information Processing Vol.18, pp.242–251.)

© 2010 Information Processing Society of Japan

Jie Ji was born in 1983. He received his B.E. degree from Harbin Institute of Technology in 2006 and M.E. degree from the University of Aizu in 2008. Currently he is pursuing his Ph.D. degree. He is receiving Japanese Government Scholarship between 2006 and 2011. His research interests are hierarchical visualization, analysis, classification of very high dimensional data and awareness computing.

Tony Y. T. Chan earned his Ph.D. in computer science at the University of New Brunswick, Canada. Just before that in 1991 he married Lynda Jane Golding from Saint John. Since then, he has taught at universities in Canada, Japan, and Iceland. For over 25 years Dr. Chan's main research interests have been in the fields of artificial intelligence, machine learning, and pattern recognition. He has worked on cooperative projects with universities, research institutes, and commercial companies, in Web mining, cancer bioinformatics, computer game AI, etc. Recently, he and his Icelandic students set up a trial server for restaurants in town so that customers at home could order take-out online.

Qiangfu Zhao received his Ph.D. degree from Tohoku University of Japan in 1988. He joined the Department of Electronic Engineering, Beijing Institute of Technology, China in 1988, first as a post doctoral fellow and then an associate professor. He was an associate professor from October 1993 at the Department of Electronic Engineering, Tohoku University, Japan. He joined the University of Aizu, Japan from April 1995 as an associate professor, and became a tenure full professor in April 1999. Professor Zhao's research interests include image processing, pattern recognition and understanding, computational intelligence, neurocomputing, evolutionary computation and awareness computing.

© 2010 Information Processing Society of Japan