A Clustering Scheme for Large High-Dimensional Document Datasets

Jung-Yi Jiang, Jing-Wen Chen, and Shie-Jue Lee

Dept. of Electrical Engineering,
National Sun Yat-Sen University, Taiwan
{jungyi,a140425,leesj}@water.ee.nsysu.edu.tw

Abstract. Scalability and high dimensionality are two common problems associated with document clustering. We present a novel scheme to deal with these problems. Given a set of documents, we partition the set into several parts. We use one part and cluster the constituent documents into groups. By the obtained groups, we reduce the number of features by a certain ratio. Then we add another part, cluster the documents into groups based on the reduced features, and further reduce the number of the remaining features. This process is iterated until all parts are used. Experimental results have shown that our proposed scheme is effective for clustering large high-dimensional document datasets.

Keywords: Document clustering, scalability, high dimensionality, K-means, information gain.

1 Introduction

Recently, document clustering has attracted more and more attention. However, two challenges are usually encountered, scalability and high dimensionality. In this modern world, more and more documents are generated. Research papers, book chapters, recreation articles, etc., are published everyday. Furthermore, a lot of keywords are used for distinguishing one document from another. For example, more than 15000 keywords are used in the real-world document set Reuters21578. As a result, clustering large high-dimensional document datasets is not an unusual task. Most of the existing clustering algorithms have difficulties in dealing with such datasets.

Document collections are usually represented by a *vector space model*, alternatively known as a *bag-of-words* [5]. The words appear in the documents are treated as features and each document is represented as a vector of certain weighted word frequencies in this feature space. Document clustering is a common unsupervised learning technique used to discover group structure in a set of documents. The challenge for document clustering is to minimize the computation cost as well as to maximize the difference among clusters. Several algorithms based on K-means were proposed [1,2,3,6] for text clustering. These approaches are based on distance or similarity measures. A combination of the batch k-means and the increment k-means algorithms [7] was introduced for document clustering . A spherical k-means algorithm for clustering text data was

L. Kang, Y. Liu, and S. Zeng (Eds.): ISICA 2007, LNCS 4683, pp. 511–519, 2007.

proposed in [6]. A hierarchical clustering algorithm based on divisive partitioning was proposed in [4].

Feature reduction is a useful technique to reduce dimensionality and speed up document processing tasks. Many approaches for feature reduction have shown their good performance for document classification tasks [8,9,10,11,12,13]. Since they work on classification, it is required that the class of each document be known. A well-known feature reduction approach is based on Information Gain [8], which is an information-theoretic measure defined by the amount of reduced uncertainty given a piece of information. However, these approaches cannot be used for document clustering.

We propose a novel scheme to deal with the challenges of scalability and high dimensionality associated with document clustering. Given a set of documents, we partition the set into several parts. We use one part and cluster the constituent documents into groups. By the obtained groups, we reduce the number of features by a certain ratio. Then we add another part, cluster the documents into groups based on the reduced features, and further reduce the number of the remaining features. This process is iterated until all parts are used. Experimental results have shown that our proposed scheme is effective for clustering large high-dimensional document datasets.

2 Background and Related Work

As we mentioned earlier, the bag-of-words model [5]is usually adopted to represent documents for document processing. Each document is expressed as a vector of weighted word frequencies. Let d_i be a document and $D = \{d_1, d_2, \ldots, d_n\}$ be a set containing n documents. Let the word set $W = \{w_1, w_2, \ldots, w_f\}$ be the feature set of the documents. Each document d_i, $1 \leq i \leq n$, can be represented as $d_i =< w_{i1}, w_{i2}, \ldots, w_{if} >$ where each w_{ij} denotes the number of occurrences of w_j in document d_i.

2.1 K-Means Clustering

Let $\{P_j\}_{j=1}^{k}$ be a partition of D where k is a user-specified constant. The goal of the k-means clustering algorithm is to maximize the following objective function:

$$\sum_{j=1}^{k} \sum_{d_i \in P_j} Sim(d_i, P_j) \tag{1}$$

where $Sim(d_i, P_j)$ is the similarity measure between document d_i and P_j. A popular similarity measure is defined as

$$Sim(d_i, P_j) = cos(d_i, m_j) = \frac{d_i \cdot m_j}{\|d_i\| \|m_j\|} \tag{2}$$

where m_j is the centroid of P_j defined as

$$m_j = \frac{1}{|P_j|} \sum_{d \in P_j} d. \tag{3}$$

2.2 Feature Reduction

The feature reduction task is to find a new word set $W' = \{w'_1, w'_2, \ldots, w'_r\}$, $r < f$, such that W and W' work equally well for all the desired properties with D. After feature reduction, each document d_i is converted to a new representation $d'_i =< w'_{i1}, w'_{i2}, \ldots, w'_{ir} >$ and the converted document set is $D' = \{d'_1, d'_2, \ldots, d'_n\}$. If r is very much smaller than f, computation cost can be drastically reduced.

One popular feature reduction approach uses information gain to select W' from W, and W' is a subset of W [8]. This approach only uses the selected features as inputs for classification tasks. It measures the reduced uncertainty by an information-theoretic measure and gives each word a weight. The bigger the weight of a word is, the larger is the reduced uncertainty by the word. Let $\{c_1, c_2, ..., c_p\}$ denote the set of classes. The weight of a word w_i is calculated as follows:

$$\begin{aligned} G(w_i) = & -\sum_{l=1}^{p} Pr(c_l) log Pr(c_l) \\ & + Pr(w_i) \sum_{l=1}^{p} Pr(c_l|w_i) log Pr(c_l|w_i) \\ & + Pr(\overline{w}_i) \sum_{l=1}^{p} Pr(c_l|\overline{w}_i) log Pr(c_l|\overline{w}_i). \end{aligned} \tag{4}$$

The words of top r weights in W are selected as the features in W'. Information gain is applied to compress the complexity of the document set from $O(nf)$ to $O(nr)$. If r is much smaller than f, the computation cost associated with document processing can be drastically reduced.

3 Proposed Method

As mentioned, scalability and high-dimensionality are two challenges for document clustering. Existing clustering algorithms have difficulties in processing all the documents of a document set at one time. Feature reduction algorithms can be used for reducing dimensionality, but they work well only for document classification in which the class label of each document is known. In document clustering, the class labels are not known for documents. We propose a novel approach to deal with these two challenges. A given document set is partitioned into parts. We use one part and cluster the constituent documents into groups. Then we give an appropriate label to each document. Note that all documents have their class labels now. Then we use a feature reduction algorithm to reduce the number of features by a certain ratio. Then we add another part, cluster the documents into groups based on the reduced features, and further reduce the number of the remaining features. This process is iterated until all parts are used. The flowchart of our approach is shown in Fig. 1.

To use feature reduction algorithms efficiently, documents under consideration must be labeled. For this purpose, we cluster parts of documents, and use the assigned cluster labels to perform the feature reduction work. To avoid losing too much information at one time, dimensionality is reduced progressively. We repeat clustering and feature reduction, with more and more documents but less and less features, until all documents are clustered. For clustering, K-means is

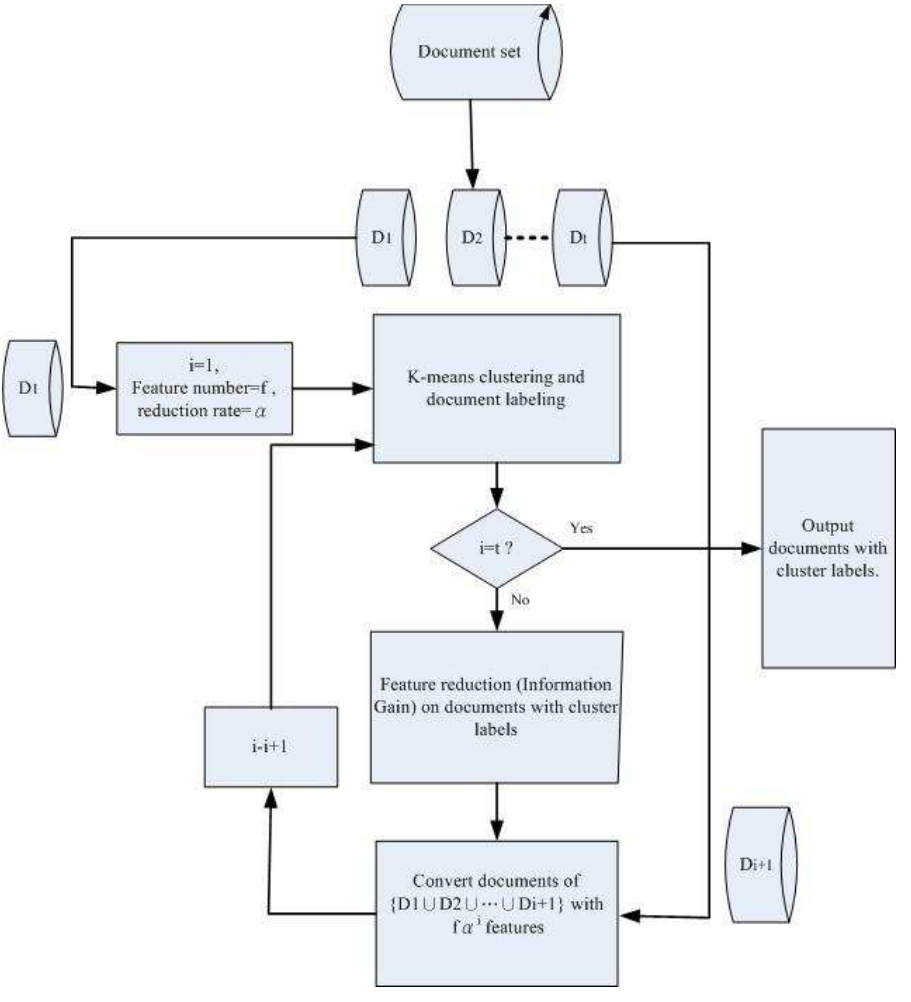


Fig. 1. The flowchart of our approach

adopted, and for feature reduction, information gain is used. The whole process of our approach is detailed as follows:

Input: The document set D, the desired number of cluster K, the feature reduction rate α, the number of partitions t, and the number of original features f.

Step 1: Divide D into partitions$\{D_1, D_2, \ldots, D_t\}$. Let $i = 1$, $g = f$, and $P = \emptyset$.

Step 2: Let $P = P \cup D_i$. Apply K-means on P with g features. K clusters are obtained.

Step 3: Label each document in P with the cluster to which it belongs. Perform feature reduction with these labeled documents, and the number of features is reduced by α, i.e., $g = \alpha g$.
Step 4: If $i == t$, go to Step 5; otherwise $i = i + 1$, and go to Step 2.
Output: The document set D' with g features, and the resulting clusters.

Note that $\alpha \in (0, 1]$. When $\alpha = 1$, feature reduction is not applied. After the ith iteration, the number of features is reduced by $f\alpha^i$.

To evaluate the performance of our proposed method, an entropy based cluster validity measure is adopted. Let K be the number of clusters obtained by our clustering approach and L be the number of classes given by the data source. For each cluster i, we can calculate an entropy e_i of the cluster as follows:

$$e_i = -\sum_{j=1}^{L} p_{ij} log_2 p_{ij} \tag{5}$$

where $p_{ij} = m_{ij}/m_i$ is the probability that a member of cluster i belongs to class j. Note that m_i is the number of objects in cluster i and m_{ij} is the number of objects of class j in cluster i. The **entropy weighted sum** E is defined as the sum of the entropies of each cluster weighted by the size of each cluster, i.e.,

$$E = \sum_{i=1}^{K} \frac{m_i}{M} e_i \tag{6}$$

where M is the total number of data points. If E is smaller, the performance of a clustering method is better.

4 Experimental Results

To show the effectiveness of our proposed method, experiments on a well-known data set, Reuters-21578, are performed. The documents of Reuters-21578 are divided, according to the "ModApte" split, into 9603 training documents and 3299 testing documents. The number of training documents per class varies from 1 to about 4000, with top 10 classes containing 77.5% of the documents and 28 classes have fewer than 10 training documents. Our experiments use the documents of the top 10 classes. In Experiment 1, we use the training documents of Reuters-21578 Top 1. The number of words involved in Experiment 1 is 16283 and the number of documents is 7390. The data set is large and high dimensional. We will show the performance of our method on this large and high-dimensional data set in Experiment 1. In Experiment 3, we use the testing documents of Reuters-21578 Top 10. The number of words involved in Experiment 3 is 16283 and the number of documents is 2787. The data set is small and high-dimensional. We will show the performance of our method on this small and high-dimensional data set in Experiment 3. Data sets in Experiment 1 and 3 are formed by converting documents from files to vectors. To provide data sets

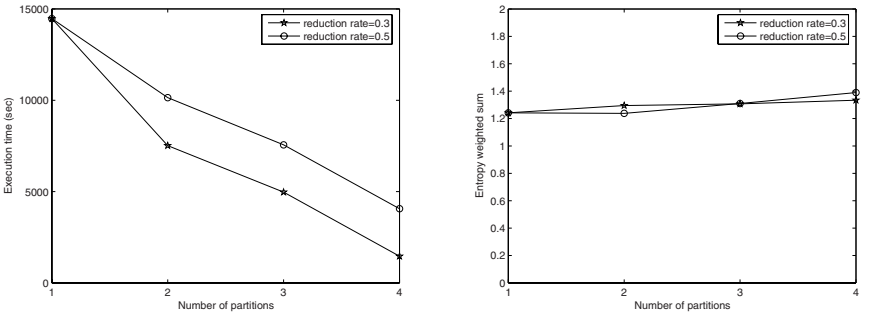
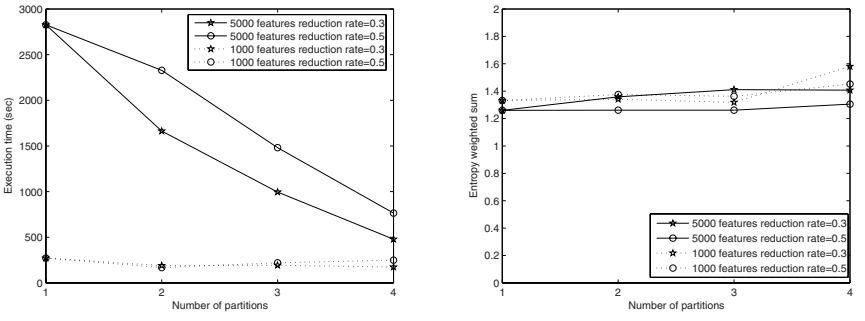


Fig. 2. (a) Execution time for large and high-dimensional data sets. (b) Entropy weighted sum for large and high-dimensional data sets.

with different dimensionality, we use a word clustering approach proposed in [13] to convert the data sets from high dimensionality to lower dimensionality. After converting, two data sets with dimensionality 5000 and 1000, respectively, are obtained from Reuters-21578 Top 10. In Experiment 2, we use the training documents of Reuters-21578 Top 10. The number of words is 5000 and 1000, respectively, and the number of documents is 7390. The data sets are large and lower-dimensional. We will show the performance of our method on large and lower-dimensional data sets in Experiment 2. In Experiment 4, we use the testing documents of Reuters-21578 Top 10. The number of words is 5000 and 1000, respectively, and the number of documents is 2787. The data sets are small and lower-dimensional. We will show the performance of our method on small and lower-dimensional data sets in Experiment 4. In each experiment, two reduction rates, 0.5 and 0.3, are used and the number of partitions is varied among 1, 2, 3, and 4. When the number of partitions is equal to 1, no feature reduction is applied. In other words, K-means clustering with full features is applied when the partition number is 1. For all experiments, K is set to 10.

4.1 Experiment 1: Large and High-Dimensional Data Set

Fig. 2 shows the execution time and entropy weighted sum of our method with different numbers of partitions on a large (7390 documents) and high-dimensional (16283 features) data set. The star-marked line and circle-marked line denote the results with reduction rates 0.3 and 0.5, respectively. As shown in Fig. 2(a), our method reduces the execution time significantly, especially when the partition number is larger and the reduction rate is smaller. When the partition number is 4 and the reduction rate is 0.3, the execution time is reduced from 14474 to 1464 seconds. As shown in Fig. 2(b), our method works well with a little increase in entropy weighted sum.



(a) (b)

Fig. 3. (a) Execution time for large and lower-dimensional data sets. (b) Entropy weighted sum for large and lower-dimensional data sets.

4.2 Experiment 2: Large and Lower-Dimensional Data Set

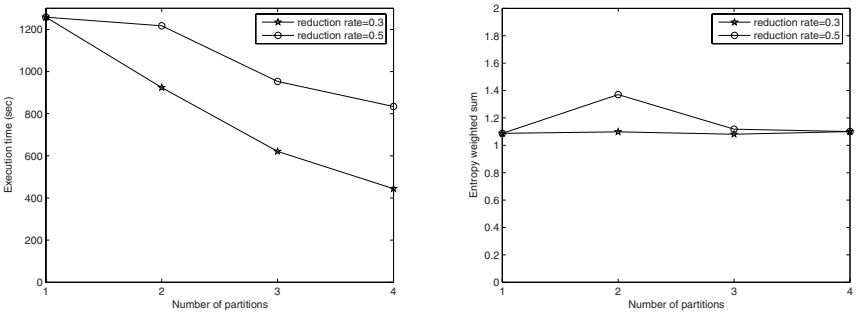
Fig. 3 shows the execution time and entropy weighted sum of our method with different numbers of partitions on large (7390 documents) and lower-dimensional (5000 and 1000 features, respectively) data sets. The star-marked solid lines and the circle-marked solid lines denote the results for the data set with 5000 features, while the star-marked dotted lines and the circle-marked dotted lines denote the results for the data set with 1000 features. As shown in Fig. 3(a), the decrease of execution time our method achieves on the data set with 5000 features is more slower than on full features shown in Fig. 2(a). On the data set with 1000 features, our method spends almost the same time as clustering without feature reduction. As shown in Fig. 3(b), our method works well with a little increase in entropy weighted sum on both data sets with 5000 and 1000 features. Compared to Experiment 1, our method is obviously more suitable for high-dimensional data sets.

4.3 Experiment 3: Small and High-Dimensional Data Set

Fig. 4 shows the execution time and entropy weighted sum of our method with different numbers of partitions on large (7390 documents) and high-dimensional (16283 features) data sets. The star-marked line and circle-marked line denote the results with reduction rate 0.3 and 0.5, respectively. As shown in Fig. 4(a), the execution time reduced on small data sets is more slower than on large data sets shown in Fig. 2(a). As shown in Fig. 4(b), our method works well with a little increase in entropy weighted sum. Compared to Experiment 1, our method is more suitable for large data sets with high-dimensionality.

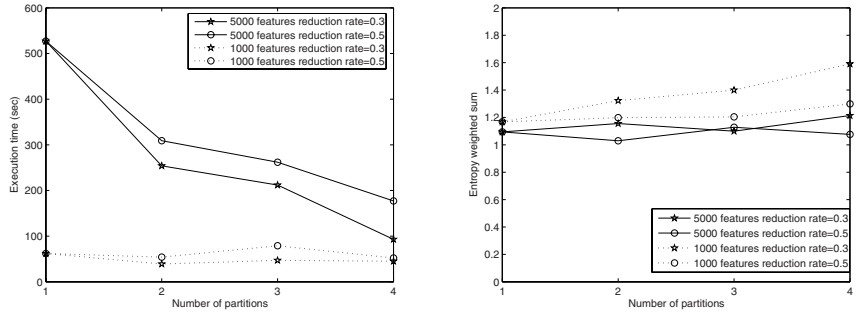
4.4 Experiment 4: Small and Lower-Dimensional Data Set

Fig. 5 shows the execution time and entropy weighted sum of our method with different numbers of partitions on small (2787 documents) and lower-dimensional (5000, 1000 features) data sets. The solid lines with star marks and circle marks



(a) (b)

Fig. 4. (a) Execution time on small and high-dimensional data sets. (b) Entropy weighted sum on small and high-dimensional data sets.



(a) (b)

Fig. 5. (a) Execution time on small and lower-dimensional data sets. (b) Entropy weighted sum on small and lower-dimensional data sets.

denote the results for data sets with 5000 features, while the dotted lines denote the results for data sets with 1000 features. As shown in Fig. 5(a), execution time reduced on the data set with 5000 features is more slower than on large data size (compared to Fig. 3(a)) or large high-dimensional data set (compared to Fig. 2(a)). On the data set with 1000 features, our method spends almost the same time as clustering without feature reduction. As shown in Fig. 5(b), our method works well with a little increase in entropy weighted sum both data sets woth 5000 and 1000 features, respectively. Compared to the previous experiments, our method is more suitable for large high-dimensional data sets.

5 Conclusion

Scalability and high-dimensionality are two common problems encountered in document clustering. We have proposed a novel approach for clustering large high-dimensional document datasets. Traditionally, feature reduction is a powerful tool to reduce dimensionality of data. However, they only work well for classification problems. That is, the information about the class label for each

document has to be known in advance. To apply feature reduction algorithms to the clustering work, we propose a framework which utilizes the results of clustering on a part of the dataset as class labels, and then perform feature reduction based on the obtained cluster labels. Documents are clustered in a progressive way. To test the effectiveness of our approach, K-means and information gain are adopted for clustering and feature reduction, respectively. Experiments on four different datasets have shown that our approach can work efficiently and get acceptable results on large high-dimensional document datasets.

References

1. Dhillon, I.S., Guan, Y., Fan, J.: Efficient Clustering of Very Large Document Collections. In: Data Mining for Scientific and Engineering Applications, pp. 357–381. Kluwer Academic Publishers, Dordrecht (2001)
2. Dhillon, I.S., Kogan, J., Nicholas, M.: Feature Selection and Document Clustering. In: A Comprehensive Survey of Text Mining, pp. 73–100. Springer, Heidelberg (2003)
3. Kogan, J., Teboulle, M., Nicholas, C.: Data Driven Similarity Measures for K-Means Like Clustering Algorithms. Information Retrieval 8(2), 331–349 (2005)
4. Boley, D.: Principal Direction Divisive Partitioning. Data Mining and Knowledge Discovery 2(4), 325–344 (1998)
5. Salton, G., McGill, M.J.: Introduction to Modern Retrieval. McGraw-Hill Book Company, New York (1983)
6. Dhillon, I.S., Modha, D.S.: Concept Decompositions for Large Sparse Text Data using Clustering. Machine Learning 42(1), 143–175 (2001)
7. Kogan, J.: Means Clustering for Text Data. In: Proceedings of the workshop on Text Mining at the First SIAM International Conference on Data Mining, pp. 54–57 (2001)
8. Yang, Y., Pedersen, J.O.: A Comparative Study on Feature Selection in Text Categorization. In: Proceedings of 14th International Conference on Machine Learning, pp. 412–420 (1997)
9. Baker, L.D., McCallum, A.: Distributional Clustering of Words for Text Classification. In: Proceedings of 21st Annual International ACM SIGIR, pp. 96–103 (1998)
10. Slonim, N., Tishby, N.: The Power of Word Clusters for Text Classification. In: Proceedings of 23rd European Colloquium on Information Retrieval Research (ECIR) (2001)
11. Bekkerman, R., El-Yaniv, R., Tishby, N., Winter, Y.: Distributional Word Clusters vs. Words for Text Categorization. Journal of Machine Learning Research 1, 1–48 (2002)
12. Pereira, F., Tishby, N., Lee, L.: Distributional Clustering of English Words. In: 31st Annual Meeting of ACL, pp. 183–190 (1993)
13. Dhillon, I.S., Mallela, S., Kumar, R.: A Divisive Infromation-Theoretic Feature Clustering Algorithm for Text Classification. Journal of Machine Learning Research 3, 1265–1287 (2003)