Document Clustering at NTCIR-4 Workshop: Limiting Search Space of the K-means method using Word Occurrence

Yuji KANEDA   Naonori UEDA   Kazumi SAITO
NTT Communication Science Laboratories, NTT Corporation
2-4, Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0237 Japan
{ykaneda,ueda,saito}@cslab.kecl.ntt.co.jp

Abstract

*In this paper, we propose a new document clustering method based on the K-means method (*kemans*). In our method, we allow only finite candidate vectors to be representative vectors of* kemans*. We also propose a method for constructing these candidate vectors using documents that have the same word. We participated in NTCIR-4 WEB Task D (Topic Classification Task) and experimentally compared our method with* kmeans *on this task.*

Keywords: *Document Clustering, K-means, local optimal solution*

1 Introduction

With the rapid growth in the number of on-line documents, document clustering has become an important task in information retrieval and text mining. The Bag-of-Words (BOW)-based K-means method (kmeans) is commonly used for document clustering [3, 5]. However, kmeans has a drawback in that it often produces poor results on small-sized and high-dimensional data where it tends to get stuck in inferior local optima [2].

NTCIR-4 WEB Task D is a document-clustering task. In this task, a participant is required to classify given web pages retrieved by a meta-search engine. For each query, there are 200 web pages to classify, and each document set consists of diverse web pages. Because this data size is small and many diverse words are used in these documents, kmeans may suffer from the above drawback.

To overcome this problem, we propose a new document clustering method that limits the search space of kmeans by introducing a constraint on representative vectors (RVs). When the number of available solutions decreases, the number of local optima also decreases. Therefore, by limiting the search space, our method can avoid getting stuck in a poor local optima. However, to yield good results by this approach, we must construct appropriate candidate vectors. We also propose a method for constructing the candidates by using a characteristic of text.

We experimentally compared the efficiency of the proposed method with that of kmeans on dry-run and formal-run data sets. We demonstrate that the proposed method with appropriate parameters can outperform kmeans on these data sets.

2 K-means

We briefly explain the BOW-based kmeans. In kmeans, N documents $\{d_n\}_{n=1}^N$ are divided into K clusters. Let $\mathbf{y}_n = (y_{n,1}, \ldots, y_{n,K})$ denote a cluster indicator vector of d_n, where $y_{n,k} = 1$ if d_n belongs to the kth cluster, and $y_{n,k} = 0$ otherwise. In BOW representation, each document d_n is represented by word frequency vector $\mathbf{x}_n(x_{n,1}, \cdots, x_{n,V})$, where $x_{n,i}$ denotes the occurrence frequency of word w_i in d_n among a set of vocabulary, $\mathcal{V} = \{w_1, \ldots, w_V\}$, where V is the total number of words in the vocabulary.

Let $\boldsymbol{\theta}_k = (\theta_{k,1}, \cdots, \theta_{k,V})$ denote the RV of k-th cluster. In kmeans, the following objective function, $\mathcal{J}(\mathbf{Y}, \boldsymbol{\Theta})$, is maximized with respect to $\{\mathbf{Y}, \boldsymbol{\Theta}\}$:

$$\mathcal{J}(\mathbf{Y}, \boldsymbol{\Theta}) = \sum_{n=1}^{N} \sum_{k=1}^{K} y_{n,k}\ s(\mathbf{x}_n, \boldsymbol{\theta}_k), \qquad (1)$$

where $\mathbf{Y} = (\mathbf{y}_1, \cdots, \mathbf{y}_N), \boldsymbol{\Theta} = (\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_K)$, and $s(\cdot, \cdot)$ is a similarity measure. A local optimal solution to this problem can be obtained by iteratively updating $\mathbf{Y}$ and $\boldsymbol{\Theta}$ as follows:

(i) When updating cluster assignments, $\mathbf{Y}$, find the nearest $\boldsymbol{\theta}_k$ to $\mathbf{x}_n$ for each document d_n; then set $y_{n,\hat{k}} = 1$ and set $y_{n,l} = 0, l \neq \hat{k}$, where $\hat{k}$ is the nearest cluster's number.

(ii) When updating RVs, $\boldsymbol{\Theta}$, set $\boldsymbol{\theta}_k$ to the mean of the word frequency vectors in the k-th cluster.

3 Proposed Method

3.1 Approach

We present an algorithm that limits the search space of kmeans by introducing a constraint on $\mathbf{\Theta}$ for overcoming the drawback of kmeans. More specifically, we allow only $M(>K)$ candidate vectors $\mathbf{\Phi} \in \mathbf{R}^{V\times M}$ to be an RV; that is,

$$\boldsymbol{\theta}_k \in \mathbf{\Phi}, \quad k = 1, \cdots, K. \tag{2}$$

In kmeans, $\boldsymbol{\theta}_k$ can take value in among all V-dimensional vectors; in contrast, under this constraint, the size of the search space for $\boldsymbol{\theta}_k$ is only M.

We construct $\mathbf{\Phi}$ using a characteristic of text. More specifically, as a candidate, we use a mean of word frequency vectors of documents in $\mathcal{D}_w$, where $\mathcal{D}_v(\subset \{d_n\}_{n=1}^N)$ is a set of documents that have word v. This mean vector, $\mathbf{m}_v$, is given by:

$$\mathbf{m}_v = \frac{1}{|\mathcal{D}_v|} \sum_{\{n|d_n \in \mathcal{D}_v\}} \mathbf{x}_n, \tag{3}$$

Hence, we construct $\mathbf{\Phi}$ from a set of words $\mathcal{V} = \{v_1, \cdots, v_M\}$ as the following:

$$\mathbf{\Phi} = \{\mathbf{m}_{v_1}, \cdots, \mathbf{m}_{v_M}\}. \tag{4}$$

We use $\mathbf{m}_v$ as a candidate vector for the following two reasons. One reason is that documents in $\mathcal{D}_v$ may have similar contents. Another reason is that the mean frequency vector of similar documents may be a good candidate for an RV.

First, we explain why similarities within $\mathcal{D}_v$ may be high. As an example, suppose v is “baseball”. When a document has “baseball”, this document would tend to also have a word related to “baseball” (ex. “bat” or “strike”). Because of such dependencies among words in a document, $\mathcal{D}_v$ may be similar documents.

Next, we explain why the mean of frequency vectors of highly similar documents is desirable for a candidate vector. When the result of clustering is good, similarities among documents in a cluster are high. Conversely, to achieve a good clustering result, it is necessary for RV to be the mean of frequency vectors of highly similar documents.

3.2 Details

Here, we describe the details of the proposed method. First, we determine M important words, $\mathcal{V} = \{v_1, \cdots, v_M\}$. In this paper, we use the tf-idf score[1] of each word for selecting $\mathcal{V}$. Next, we compute $\mathbf{\Phi}$ by equation (3) and equation (4). Finally, under the constraint of equation (2), we maximize $\mathcal{J}(\mathbf{\Theta}, \mathbf{Y})$; that is, we solve the following maximization problem:

$$\begin{aligned} &\underset{\mathbf{Y},\mathbf{\Theta}}{\text{maximize}}\ J(\mathbf{Y}, \mathbf{\Theta}) \\ &\text{subject to} \\ &\quad y_{n,k} \in \{0,1\},\ n = 1,\cdots,N,\ k = 1,\cdots,K \\ &\quad \sum_{k=1}^{K} y_{n,k} = 1, \quad n = 1,\cdots,N, \\ &\quad \boldsymbol{\theta}_k \in \mathbf{\Phi}, \quad k = 1,\cdots,K. \end{aligned} \tag{5}$$

To solve the above discrete optimization problem, we use a greedy search algorithm w.r.t. $\mathbf{\Theta}$. The procedure is as follows:

(i) Initialize $\mathbf{\Theta}$.

(ii) For each RV $\boldsymbol{\theta}_k, k = 1, \cdots, K$, find the optimal $\boldsymbol{\theta}_k$ that maximizes the objective function while fixing the other RVs $\boldsymbol{\theta}_{k'}, k' \neq k$:

$$\boldsymbol{\theta}_k \leftarrow \underset{\boldsymbol{\phi} \in \mathbf{\Phi}}{\text{argmax}} \left\{F_k(\boldsymbol{\phi}, \mathbf{\Theta})\right\}, \tag{6}$$

where $F_k(\boldsymbol{\phi}, \mathbf{\Theta})$ is given by

$$\begin{aligned} &F_k(\boldsymbol{\phi}, \mathbf{\Theta}) \\ &= \max_{\mathbf{Y}} \mathcal{J}(\mathbf{Y}, (\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_{k-1}, \boldsymbol{\phi}, \boldsymbol{\theta}_{k+1}, \cdots, \boldsymbol{\theta}_K)). \end{aligned} \tag{7}$$

(iii) Repeat step (ii) until the convergence.

(iv) Using a local optimal solution $\hat{\mathbf{\Theta}}$, compute

$$\hat{\mathbf{Y}} = \underset{\mathbf{Y}}{\text{argmax}}\, \mathcal{J}(\mathbf{Y}, \hat{\mathbf{\Theta}}). \tag{8}$$

Then, according to $\hat{\mathbf{Y}} = (\hat{\mathbf{y}}_1, \cdots, \hat{\mathbf{y}}_N)$, divide the documents into K clusters.

When all similarities between $\mathbf{x}_n$ and $\mathbf{m}_v$ are computed, the computational cost of this greedy-search algorithm is $O(NK^2M)$. Hence, the total computational cost is $O(NVM + NK^2M)$. In addition, when updating $\boldsymbol{\theta}_k^{old}$ to $\boldsymbol{\theta}_k^{new}$, we decrease computational cost by search $\boldsymbol{\theta}_k^{new}$ that maximizes $F_k(\boldsymbol{\theta}_k^{new}, \mathbf{\Theta}) - F_k(\boldsymbol{\theta}_k^{old}, \mathbf{\Theta})$.

4 Experimental Setting and Evaluation

4.1 BOW representation

We extracted words from a document by using a morphological analyzer, ChaSen[6]. As the vocabulary of BOW, $\mathcal{W}$, we use noun, adjective, verb, and out-of-vocabulary words, as well as bi-grams and tri-grams that occur more than 3 times in a document; however, we remove words that occur in fewer than 2 documents.

Table 1. Comparison of F-value@20 on dry-run

	rigid				relax			
Query	kmeans	M= 25	M= 50	M= 100	kmeans	M= 25	M= 50	M= 100
18	41.87	55.80	**60.50**	37.20	**45.89**	44.10	38.20	41.20
19	**40.66**	12.50	25.00	23.75	**37.15**	11.40	22.90	21.75
20	**31.66**	23.30	23.30	23.30	**36.67**	25.00	20.80	25.00
28	15.53	13.30	**22.20**	**22.20**	27.72	28.60	28.60	**33.00**
29	5.60	10.40	**16.00**	0.00	8.47	17.71	**23.10**	0.77
33	23.80	**33.30**	28.60	19.00	23.80	**33.30**	28.60	19.00
34	**29.36**	19.40	16.10	22.60	**29.36**	19.40	16.10	22.60
38	**37.97**	16.30	28.60	27.78	**44.01**	18.20	36.40	30.56
47	20.80	**33.60**	24.00	22.40	26.16	**35.90**	33.30	34.62
48	20.60	**27.45**	19.40	26.32	30.16	**30.50**	**30.50**	30.16
58	16.17	23.10	**30.80**	**30.80**	39.51	35.90	**46.20**	43.60
63	9.26	14.30	12.86	**15.73**	14.55	12.10	20.58	**22.99**
Ave.	24.44	23.56	**25.61**	22.59	**30.29**	26.01	28.77	27.10

4.2 Similarity Measure

As similarity measure $s(\mathbf{x}_n, \boldsymbol{\theta}_k)$, we used a log-likelihood when $\mathbf{x}_n$ is generated from a multinomial distribution with parameter $1/(\sum_i \theta_{k,i})\boldsymbol{\theta}_k$; that is, $s(\mathbf{x}_n, \boldsymbol{\theta}_k)$ is given by

$$s(\mathbf{x}_n, \boldsymbol{\theta}_k) = \sum_{i=1}^{V} x_{n,i} \log \frac{\theta_{k,i}}{\sum_{i=1}^{V} \theta_{k,i}}. \qquad (9)$$

In addition, to avoid zero probabilities, we used Laplace smoothing when computing $\boldsymbol{\theta}_k$.

The similarity measure of equation (9) is closely related to the information bottleneck methods [7, 8].

4.3 Evaluation Measure

We use F-value@20 for comparison. First, we sort documents within a cluster by the score of a meta-search engine given by the task organizer. Then, using the above ranking, we sort all documents by the document-belonging cluster's score that is defined as the number of relevance documents within a cluster. Finally, F-value@20 is computed as the harmonic average of *precision* and *recall* when 20 documents are retrieved by this ranking.

4.4 Experimental Results

We evaluated the performance of the proposed method with different M and compared this performance with that of kmeans. Table 1 shows the average F-value@20 on the dry-run for 10 runs of each method. As shown in Table 1, the optimal M varies across the queries and the relevance judgments (rigid or relax). Because the proposed method achieved the highest averaged F-value@20 when $M = 50$, we decided to submit a result with $M = 50$ for the formal-run. When $M = 50$, in both relevance judgments of rigid and relax, the proposed method is superior to kmeans on 7 queries, but inferior on the other 5 queries.

Table 2 shows the same comparison on the formal-run. When $M = 50$, the proposed method outperforms kmeans on 6 queries of the 11 queries for both rigid and relax. As in Table 1, the optimal M varies across the queries and the relevance judgments. If the optimal M is selected for each query, the proposed method outperforms kmeans on 10 queries for rigid and 8 queries for relax.

5 Conclusions and Future Work

We have proposed a new document clustering method that limits the search space of kmeans by introducing a constraint that is computed from documents that have the same word. We have experimentally evaluated the proposed method's efficiency on NTCIR-4 WEB Task D. We have confirmed that the proposed method can achieve superior performance to kmeans when we select appropriate candidate vectors.

One of our future works is development of a more efficient method for determining appropriate important words, $\mathcal{V}$. For determining $\mathcal{V}$, automatic term recognition methods [4] could be a good alternative to the tf-idf score. The selection of M (size of $\mathcal{V}$) is also an important problem. As shown in Tables 1 and 2, the optimal size varied across the document sets. Therefore, to achieve better performance, it may be necessary to estimate optimal M.

References

[1] A. Aizawa. The feature quantity: An information theoretic perspective of tfi df-like measures. In *Proceedings of the 23th annual international ACM SIGIR conference*

Table 2. Comparison of F-value@20 on formal-run

	rigid				relax			
Query	kmeans	$M = 25$	$M = 50$	$M = 100$	kmeans	$M = 25$	$M = 50$	$M = 100$
1	6.02	6.70	**26.70**	18.70	14.83	17.50	**24.70**	21.42
3	11.99	17.10	**28.60**	**28.60**	24.53	28.34	**37.30**	35.00
4	0.87	**17.40**	0.00	0.00	4.21	**7.36**	5.00	7.30
6	29.47	35.90	**38.50**	35.38	33.72	28.90	**35.10**	31.48
8	18.62	20.60	18.30	**22.90**	**48.91**	32.30	43.50	29.20
19	**30.40**	8.00	16.00	24.00	7.33	9.80	8.50	**10.00**
21	3.48	**8.70**	0.87	**8.70**	5.54	**9.50**	6.70	6.30
22	25.52	22.00	15.40	**33.00**	**47.00**	27.50	31.30	36.10
23	0.95	**9.50**	**9.50**	**9.50**	6.11	**8.30**	5.40	6.82
29	10.33	12.90	**32.30**	12.90	**32.12**	26.60	31.60	29.10
45	16.78	19.40	**25.80**	19.40	15.52	22.20	**23.30**	21.50
Ave.	14.04	16.20	19.27	**19.37**	21.80	19.85	**22.95**	21.29

on Research and development in information retrieval (SIGIR'00), pages 104–111, 2000.

[2] I. S. Dhillon, Y. Guan, and J. Kogan. Iterative clustering of high dimensional text data augmented by local search. In *Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM'02)*, page 131, December 2002.

[3] I. S. Dhillon and D. S. Modha. Concept decompositions for large sparse text data using clustering. *Machine Learning*, 42(1):143–175, 2001.

[4] K. Kageura and B. Umino. Methods of automatic term recognition: a review. *Terminology*, 3(2):259–289, 1996.

[5] B. Larsen and C. Aone. Fast and effective text mining using linear-time document clustering. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 16–22, 1999.

[6] Y. Matsumoto, A. Kitauchi, T. Yamashita, Y. Hirano, H. Matsuda, K. Takaoka, and M. Asahara. Morphological analysis system ChaSen version 2.3.0 manual. Technical report, Nara Institute Science and Technology, 2003.

[7] N. Slonim, N. Friedman, and N. Tishby. Unsupervised document classification using sequential information maximization. In *the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 2002.

[8] N. Slonim and N. Tishby. Document clustering using word clusters via the information bottleneck method. In *Research and Development in Information Retrieval*, pages 208–215, 2000.