International Journal of Computer Applications (0975 - 8887)
Volume 62 - No. 19, January 2013

Graph based Text Document Clustering by Detecting Initial Centroids for k-means

Vikas Kumar Sihag
Computer Science & Engineering Department
MNNIT Allahabad, India

Subhash Kumar
Computer Science & Engineering Department
MNNIT Allahabad, India

ABSTRACT

Document clustering is used in information retrieval to organize a large collection of text documents into some meaningful clusters. k-means clustering algorithm of pratitional category, performs well on document clustering. k-means organizes a large collection of items into k clusters so that a criterion function is optimized. As it is sensitive to the initial values of cluster centroids, this paper proposes a graph based method to calculate the appropriate initial cluster centroids. Document collection is represented as a graphical network in which a node represents a document and an edge represents the similarity between two documents. In order to calculate initial centroids, community structure present in graphical network is detected using edge deletion technique. Using community structure, centrality of each node is calculated. Centrality value of a node represents its candidature of being a cluster centroid. Use of community structure assures that calculated centroids have sufficient number of topically related documents and centroids are well separated from each other. k-means with these initial centroids provides a significant improvement over simple k-means for text document clustering.

General Terms:

Data mining, Clustering, Graph theory

Keywords:

Text mining, Document clustering, Cosine similarity, k-means

1. INTRODUCTION

Data mining is a technique to get some useful pattern and information from underlying data. It follows two type of paradigms: one the predictive paradigm which predict the value of an attribute based on the value of other attributes and second the descriptive paradigm which derive patterns, i.e. clusters. Clustering is used to partition a set of objects so that objects in the same cluster are more similar to each other than the objects in other clusters.

As the volume of digital documents has increased rapidly in recent years, there is a requirement of an adequate method to organize this huge collection. Text clustering plays a significant role in text mining to help effectively manage collections of text documents. Document clustering [3] organizes collection of text documents, by grouping documents into clusters to facilitate user's browsing of retrieval results [2]. Primarily two clustering approaches are followed: agglomerative hierarchical and partitional method. In *Agglomerative Hierarchical Clustering* (AHC) algorithm, initially each document is considered as a cluster and similarity among documents is calculated to merge the closest pair. This merging step is repeated until the desired criterion is met. Whereas, in *partitional clustering*, a single-level cluster of documents is initially created, followed by partitioning clusters untill the desired criterion is met. In k-means clustering algorithm a centroid can represent a cluster and when k initial cluster centroids are selected, each document is assigned to a cluster based on a similarity/dissimilarity function and after this k centroids are calculated again. This procedure is repeated until an optimal set of k clusters is obtained. Generally partitional clustering algorithms are more appropriate for the clustering of large text databases due to their relatively low computational cost and high clustering accuracy.

The main feature of partitional clustering is the usage of a global criterion function, whose optimization decides the entire clustering process. This criterion function tries to minimize intra-cluster similarity and maximize inter-cluster dissimilarity. As partitional clustering algorithms are much sensitive to the initial value of cluster centroids, calculating appropriate initial values of these cluster centroids lead to promising results. Cluster centroids are selected in a way that a centroid has enough number of topically related/similar documents and it is well separated from other cluster centroids. In this paper, values of the cluster centroids have been calculated by detecting communities [10] in text document collection. A network graph representation of text document collection is done by representing a document as a node and the edge between two nodes shows the similarity between two documents. In order to detect the communities, recursive deletion of lowest weighted edge is done until desired number of communities are found [10]. After detecting communities, centrality of each node is calculated. The centrality of a node depends on its similarity with the nodes present in the same community and its dissimilarity with all other nodes present in other communities. Further we select the node with maximum centrality value from each community and these nodes serve as the initial centroids for k-means.The calculated cluster centroid values are provided to the initial phase of k-means clustering algorithm to improve the accuracy of document clustering.

Rest of the paper is organized as follows. The next section, discusses the related works for document clustering technique. In section 3, our approach for calculating the initial cluster centroid using the concept of community detection in a graph is proposed. In section 4, experimental results of clustering algorithm are compared with original k-means algorithm in terms of clustering accuracy, followed by connclusion & future work.

2. RELATED WORK

Document clustering has been used to organize a large collection of text documents to make access of a particular document easier [4]. Clustering algorithms of agglomerative hierarchical and partitional types are applied for the clustering of text documents collection. Agglomerative clustering starts with as many number of clusters as there are documents and then merge them based on their pairwise similarity until a desired number of clusters are obtained [5]. Advantage of this method is that a number of clusters need not be supplied in advance. In partition clustering, both k-means and bisecting k-means [6] are applied and reported better than AHC [9] because the runtime and memory requirement are often higher for AHC. Also in case of AHC, if a merge that has taken place is not appropriate, then there is no backtracking to correct the mistakes. k-Means is based on the concept that a center point can represent a cluster and we use the notion of centroid, which is median point or mean of a group of points.

Community detection for clustering similar type of documents had been used using content similarity of documents [8]. Here documents are represented in the form of a complex network and using the edge betweenness, a hierarchical structure of the network is computed. Recently some methods are proposed that use the topics of the documents to find the communities [5]. Graph community detection techniques are applied to partition the graph into cohesive groups of terms.

A good clustering requires a precise definition of the closeness between a pair of data items, in terms of either the pairwise similarity or distance. Before clustering, a similarity or distance measure should be determined. The measure shows the degree of closeness or separation of the target objects. A number of similarity measures [10] are used in calculating the similarity between two documents like cosine similarity, euclidean distance and jaccord coefficient etc. The result of clustering with different similarity measures is different, and that is dependent on the representation of data. In some similarity calculation techniques, the neighbors and links are considered while determining the similarity between two documents [7] and they are used to determine the initial cluster center values used in partitional clustering algorithms. In some clustering approaches, document clustering is performed in Reduced Dimension vector space [11]. Latent Semantic Analysis is used to create a new abstract vector space, which is the appropriate representation of the document collection and using this a significant improvement in cluster quality has been achieved. Some clustering algorithms use bag of words representation of text documents but this has the problem of ignoring the relationship between the important terms that do not occur literally. To resolve this problem, ontology has been introduced in the process of clustering text documents [12] that focuses on the similarity of concepts. Another technique to deal with the problem of bag of word representation, is to consider the sequence of words in documents [13]. Here, the meaning of word is considered which shows concept expressed by synonymous word forms.

3. PROPOSED WORK

This section contains techniques involved in proposed document clustering algorithm.

3.1 Document Preprocessing

The preprocessing phase refines the terms that identify the document collection. The intention of preprocessing phase is to remove the terms with negligible information that can affect the quality of document clusters. The first step of preprocessing removes *stop words*. Stop words are common words with no information and are of no use if considered (eg. pronouns, prepositions, conjunctions etc). Stop words are removed using a list of common words. When a word in the document collection matches a word present in common word list, that word is discarded and is not considered in term document matrix. The second step of preprocessing is *word stemming*. Morphological variants of the words have same meaning and if these words are converted into their root word, the performance of document clustering can be improved. Process of stemming is done in a way that words are converted into their root form by removing their affixes.

3.2 Document Representation

In information retrieval field, *Vector Space Model (VSM)* is a widely used techniques to represent text document. In vector space model, each document is represented as a vector and the terms present in text collection describes the dimensions of these document vectors. Let $d = \{d_1, d_2, ..., d_m\}$ be a set of m documents and $t = \{t_1, t_2, ..., t_n\}$ as a set of n distinct terms present in the document set d.

Then the collection of d documents and t terms is represented by $t \times d$ matrix and the weight W_{ij} of a term t_i in document d_j, is calculated using

$$W_{ij} = tf_{ij} * log(\frac{d}{df_i}) \quad (1)$$

where tf_{ij} is the frequency of term t_i in document d_j and it contains local information about term t_i. d is the total number of documents present in document set and df_i shows the frequency of documents that contains the term i. Here $log\frac{d}{df_i}$ is the global information about term i in the document collection. Equation 1 calculates weight of each term t_i in each document d_j.

3.3 Graded edge Similarity and Community Detection

In order to represent similarity between two documents, a graded scale approach is used. It overcomes the problem of threshold based similarity between two documents as in threshold based similarity either documents are considered similar or non-similar. We have created 8 levels of similarity and the similarity between two documents falls in one of these 8 levels. Initially cosine similarity measure is used to find the similarity between two documents and based on value of cosine similarity we choose level of the graded edge.

Cosine similarity [1] between two documents d_i and d_j is calculated using:

$$cos(d_i, d_j) = \frac{d_i.d_j}{|d_i||d_j|} \quad (2)$$

The reason for not using cosine similarity values directly is their high storage cost as they are floating point values. We are having 8 levels of binary (3 bit) values and each level corresponds for a range of cosine values. Let we have six documents d_1, d_2, d_3, d_4, d_5, and d_6 then similarity among them using graded edge approach can be represented as shown in the table 1. Where 111 represents the maximum similarity between a document pair and 000 depicts that the document pair is completely unrelated.
We have represented document collection as a graph network where a document represents a node and an edge represents the value of similarity between two documents. In order to determine cluster centroids for k-means, a community structure in the network graph is detected. A community structure represents a collection of interrelated nodes. We can calculate communities in a graph by recursively deleting the lowest weight edge. The

Table 1. Similarity Matrix

	d_1	d_2	d_3	d_4	d_5	d_6
d_1	111	001	000	011	110	101
d_2	001	111	001	110	101	110
d_3	000	001	111	001	110	001
d_4	011	110	001	111	101	110
d_5	110	101	110	101	111	101
d_6	101	110	001	110	101	111

algorithm to find the community structure [10] in a graph is as follows-

—*Step 1*. Set the network with values of nodes and edges.

—*Step 2*. Find the edge with lowest wight and remove it from network.(If there is a tie for lowest weight, choose one of them at random.)

—*Step 3*. If the network splits, save the two networks. If the number of
at appropriate place in your TEX file or in bibliography file. desired communities are obtained terminate the algorithm otherwise go to step 2.

After detecting the communities in the graphical network of documents we find a document from each community that can be a potential candidate for a centroid in that community.

3.4 Initial Centroid Detection

In order to detect initial centroid from each community, we calculate centrality of each node present in the community. The centrality of a node in a network represents its cohesiveness with in a community and dissimilarity with all other nodes present in other communities.

The cohesiveness of a node in a community is the sum of similarity values of the given node with all other nodes present in that community. For a node n_i in a community C_i, cohesiveness can be calculated as follows-

$$Cohes_{n_i} = \sum_{j=1}^{n} Similarity(n_i, n_j) \tag{3}$$

where n is number of nodes present in community C_i. The normalized cohesiveness of a node can be calculated by dividing the its cohesiveness value to the number of nodes in that community.

$$NCohes_{n_i} = \frac{Cohes_{n_i}}{n} \tag{4}$$

After calculating the cohesiveness value of a node, its dissimilarity value with all other nodes present in other communities is calculated. The dissimilarity value of a node n_i can be calculated using

$$Dis_{n_i} = \sum_{a=1}^{k} \sum_{j=1}^{n_a} Dis(n_i, n_j) \tag{5}$$

where k is the number of communities and n_a represents the number of nodes in a^{th} community. Normalized dissimilarity of a node n_i can be calculated similarly.

$$NDis_{n_i} = \frac{Dis_{n_i}}{N - n_i} \tag{6}$$

where N is total number of nodes in network and n_i is the number of nodes in i^{th} communities.
Centrality of a node n_i is dependent on its cohesiveness value and dissimilarity value. It can be calculated as:

$$Cent_{n_i} = NCohes_{n_i} \cdot NDis_{n_i} \tag{7}$$

After calculating the centrality of each node, the node with highest centrality value from each community is selected and these nodes serves as the initial centroids for k-means.

3.5 Clustering Algorithm

For clustering, k-means algorithm is used. The values of initial cluster centroids are calculated using proposed approach and provided as input to k-means.

The steps of k-means are:

—Take k initial cluster centroids calculated by the proposed approach.

—For every document present in the document collection, calculate its similarity with all cluster centroids and assign this document to the closest centroid.

—k centroids are recalculated based on the documents assigned to them.

—Repeat steps 2 and 3 until convergence criteria is met.

4. EXPERIMENT AND RESULTS

In order to evaluate the performance of a clustering algorithm, manually assigned cluster labels are used as a baseline criteria to evaluate clusters accuracy. The clusters created by proposed algorithm are compared with the predefined clustering structure, which is normally created by human experts. A clustering result is accurate if the clusters are consistent with predefined clustering structure. To review whether the proposed approach can improve the performance of k-means, the k-means by selecting initial centroids calculated by proposed method is applied on 20 news group dataset and 4 university dataset. The number of clusters provided in k-means algorithm is same as the number of previously assigned categories in the taken data set.

To evaluate the accuracy of our clustering algorithm, F-measure value is used. F-measure is a combination of precision and recall values, used in information retrieval. The value of F-measure is calculated as follows:

$$Fmeasure = \frac{2 \cdot p \cdot r}{p + r} \tag{8}$$

where p is precision and r is recall. The maximum value of F-measure can be 1 with maximal accuracy while minimum value can be 0 with worst quality of clustering. The value of precision and recall can be calculated as:

$$Precision = \frac{x}{x + y} \tag{9}$$

$$Recall = \frac{x}{x + z} \tag{10}$$

where x is the number of total true positives, i.e. the total number of documents clustered together in predefined class and that are indeed found together by the clustering algorithm. y is the total number of false positives, i.e. the number of documents not supposed to be found together but are clustered together and z is the number of total false negatives, i.e. the number of documents which are expected to be found together but not clustered together by the clustering algorithm [14].

For experiment purpose we have used 20 newsgroups [15] and 4 university [16] datasets. In 20 newsgroups dataset, data is organized into 20 different newsgroups, each corresponding to a different topic. In 4 university dataset, pages are manually organized into 8 categories including student, faculty, staff, department, course, project and others. Each category contains the

pages of corresponding topics. Further the f-measure of our clustering is compared with the original k-means clustering algorithm as shown figure 1. The proposed approach show a significant improvement over the the results of simple k-means.

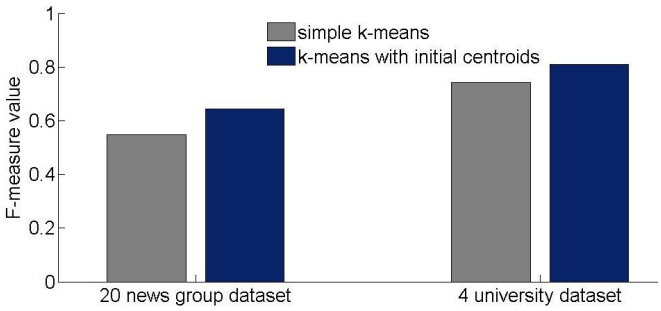

Fig. 1 Comparison of F-measure values.

5. CONCLUSION AND FUTURE WORK

This paper proposes a method to calculate initial cluster centroids for k-means algorithm to cluster text documents. The k-means algorithm performance has been improved by selection of initial centroids using a gaph based technique. The initial centroids calculated by our proposed method are well separated, and each one is close to a sufficient number of topically related documents. k-means with calculated initial centroid is compared with original k-means on real life data sets and results show a significant improvement in terms of clustering accuracy. For future perspective, ontology based document clustering can be employed. Using ontology, one can avoid the problem of polygamy and synonymy present in vector space model, which had been used here for document representation. Another technique that we can apply in our proposed approach to further improve results, is Latent Semantic Analysis (LSA) technique employed projects the high dimensional term-document matrix to a new approximated low-dimensional concept vector space using Singular Value Decomposition (SVD).

6. REFERENCES

[1] Lailil M. and Baharum B., *Document Clustering using Concept Space and Cosine Similarity Measurement*, International Conference on Computer Technology and Development, pp. 58-62, 2009.

[2] Samat A. N., Murad A., Azrifah M., and Atan R., *Malay Documents Clustering Algorithm Based On Singular Value Decomposition* , Journal of Theoretical and Applied Information Technology, pp. 180-186, 2005-2009.

[3] Kaufman L., and Rousseeuw P.J., *Finding Groups in Data: An Introduction to Cluster Analysis.* John Wiley and Sons, 1990.

[4] Van Rijsbergen, and C.J., Information Retrieval, 2nd edition, Butterwoth 1979.

[5] Jain A.K., and Dubes R.C., *Algorithms for Clustering Data.* Prentice Hall, Englewood Cliffs, 1988.

[6] Cutting D.R., Karger D.R., Pedersen J.O., and Tukey J.W., *Scatter/gather: a cluster-based approach to browsing large document collections.* In Proceedings of ACM SIGR Conf. on Research and Development in Information Retrieval, pp. 318-329, 1992 .

[7] Luo Congnan, Li Yanjun, and Chung M Soon., *Text Document Clustering based on Neighbors.* Data and Knowledge Engineering 68, pp. 1271-1288, 2009.

[8] Larsen B., and Aone C., *Fast and effective text mining using linear-time document clustering.* In Proceedings of ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining, pp. 16-22, 1999.

[9] Steinbach M., Karypis G., and Kumar V., *A comparison of document clustering techniques.*, Technical report, Department of Computer Science and Engineering, University of Minnesota, 2000.

[10] Huang A., *Similarity Measures for Text Document Clustering.*, Department of Computer Science and Engineering, The University of Waikato, New Zealand, NZCSRSC 2008, April 2008.

[11] Lerman K., *Document Clustering in Reduced Dimension Vector Space.*, USC Information Science Institute, 4676 Admiralty Way, Marina del Rey, CA 90292.

[12] Hotho A., Staab S., and Stumme G., *Ontologies improve text document clustering.* In Proceedings of IEEE Int'l Conf. on Data Mining, pp. 541-544, 2003.

[13] Li Y., Chung S. M., and Holt j. D., *Text document clustering based on frequent word meaning sequences.* In Proceedings of the 4^{th} International Conference on Business Process Management, pp. 381-404, 2008.

[14] M. Louis, *Evaluating and Comparing Text Clustering Results.* Royal Military College.

[15] 20 Usenet newsgroups dataset, *kdd.ics.uci.edu/databases*

[16] The 4 universities data set, *www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/*