Web Document Clustering for Finding Expertise in Research Area

Anil Kumar Pandey[1] and T. Jaya Lakshmi[2]

Abstract - ***Researchers often need to find expertise in their chosen area of research. Finding expertise is very useful as relevant research papers can be studied and the experts could be identified. Therefore finding expertise in the chosen area of research has always attracted interest among academic community. These days research institutions and individual researchers make their publications and research findings available on web. With the exclusive growth of World Wide Web search engine users are overwhelmed by the huge volume of results returned in response to a simple query, which is far too large to get the desired knowledge. Therefore one of the methods of finding the expertise is by way of efficiently and accurately clustering the web documents, which enhances the integrity of web search engine. Data mining techniques matured making it possible to automate the web document clustering. In this paper, we present mutually exclusive Maximal Frequent Item set discovery based K- Means clustering approach. It has been implemented in JAVA. The common text processing approach is to convert the downloaded web documents into vectors. It is being done by extracting document features and it generates the document-feature data set. For a set of documents, the feature set is composed of all terms appearing in any one of the documents. We call this a document-feature data set. If document m contains feature n, then the corresponding value, in row n and column m of the table, is set to one. Otherwise, it is zero. Then, Apriori algorithm is applied to these document feature data set. The mutually exclusive frequent sets generated by Apriori algorithm are taken as initial points of K-Means algorithm. The output of the K-Means clustering algorithm will be the sets of highly related documents appearing together with same features. This approach enables the clustering of the web documents. It enables researchers to find the documents related to their desired area clustered and displayed together during the web search. It will significantly help them in terms of saving the time and getting all the relevant papers together in a cluster..***

Index Terms - Web Document Clustering, Vector space model, Term frequency, Invert Document frequency, Apriori algorithm, maximal frequent set, k-means clustering.

1. INTRODUCTION

The growth of the Internet has seen an explosion in the amount of information available; Document clustering plays an important role for helping people organize this vast amount of data. It attempts to organize documents into groups such that documents within a group are more similar to each other than documents belonging to different groups Researchers often need to find expertise in their chosen area of research. This is very useful as relevant research papers can be studied and the experts could be identified. Therefore finding expertise in the chosen area of research has always attracted interest among academic community. These days research institutions and individual researchers make their publications and research findings available on web. The first stage in any document clustering technique is document representation model.

The rest of this paper is organized as follows: in section 2, Vector Space Model that is used in literature for document clustering will be briefly introduced. Section 3 presents k-means clustering algorithm and method used to calculate initial centroids in detail. Section 4 describes Web Document Clustering algorithm for finding expertise in Research Area in detail. The experimental results are given in section 5. Finally, conclusion and some future research directions are presented in Sections 6 and 7 respectively.

[1]*Director, GNIT Girl's Institute of Technology, Knowledge Park, Greater Noida*
[2]*Assistant Professor, Raj Kumar Goel Institute of Technology, 5 KM Stone, Delhi - Meerut Road, Ghaziabad*
E-Mail: [1]*dr.anilkpandey@yahoo.co.in and*
[2]*tjayamca@gmail.com*

2. DATA MODEL

Most clustering algorithms expect the data set to be available in the form of a set of vectors

$$X = \{x_1, x_2, \ldots, x_m\}$$

Where the vector x_i, i = 1… m corresponds to a single object in the data set and is called the *feature vector*. Extracting the proper features to represent through the feature vector is highly dependent on the problem domain.

2.1 Document Data Model

Vector Space model is selected to represent document objects. Each document is represented by a vector d, in the term space such that

$$d = \{w_{i1}, w_{i2}, \ldots, w_{in}\} \quad (1)$$

where i= 1,…, n is weight calculated as explained in following paragraph.

Term weighting scheme is employed here to measure the significance of each term [2]. In this scheme, tf_i represents term frequency (TF) and idf_i represents inverse document frequency (IDF). The assumptions behind TF*IDF are based on two empirical observations: First, the more times a term occurs in a document, the more relevant it is to the topic. Second, the more times a term appears throughout all documents in the whole collection, the more poorly it discriminates between documents. Therefore, term frequency is the number of times one term t_k appears in a document *i* and *t f (k, i)* is used to denote it. Inverse document frequency is inversely proportional to df_k, which is the document frequency for term t_k. Given M documents and N terms, the computation of idf *(k)* is as follows [2]:

$$idf\ (k) = \log\left(\frac{M}{df_k}\right) \quad (2)$$

Therefore, the weight is given as

$$w_{ik} = tf(k, i) * idf(k) \quad (3)$$

After the above transformation, the complicated, hard-to-understand documents are converted into machine acceptable, mathematical representations. The problem of measuring the similarity between documents is now converted to the problem of calculating the distance between document vectors. The standard cosine similarity, which defines the angle or cosine of the angle between two vectors, is utilized in our application. It is computed as follows:

$$\cos(d_i, d_j) = \frac{d_i^t d_j}{\|d_i\|\|d_j\|} \quad (4)$$

For a group of vectors A, in K-means, they need to be represented by their "central" vector. This central vector(C_A) is generated by taking the average value of all the points included in this group. It is calculated as follows:

$$C_A = \frac{\sum_{d \in A} d}{|A|} \quad (5)$$

3. CLUSTERING ALGORITHMS

The process of grouping a set of physical or abstract objects into classes of similar objects is called clustering. A cluster is a collection of data objects that are similar to one another within the same cluster and are dissimilar to the objects in other clusters. A cluster of data objects can be treated collectively as one group and so may be considered as a form of data compression.

3.1. K-Means Clustering Algorithm

K-means is one of the simplest unsupervised learning algorithm that solves the well known clustering problem [6]. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori. The main idea is to define k centroids, one for each cluster. These centroids should be placed in an efficient way because different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When no point is pending, the first step is completed and an early groupage is done. At this point we need to re-calculate k new centroids as barycenters of the clusters resulting from the previous step. After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new centroid. A loop has been generated. As a result of this loop we may notice that the k centroids change their location step by step until no more changes are done. In other words centroids do not move any more.

Finally, this algorithm aims at minimizing an objective function, in this case it is cosine distance specified in the previous section.

The algorithm is composed of the following steps:

1. Place K points into the space represented by the objects that are being clustered. These points represent initial group centroids.
2. Assign each object to the group that has the closest centroid.
3. When all objects have been assigned, recalculate the positions of the K centroids.
4. Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated.

3.2. Calculating Initial Cluster Centroids

The Apriori algorithm[5] is the most well known association rule mining algorithm. It uses the following property, which we call the *large item set property*: Any subset of a large item set must be large. The large item sets are also said to be downward closed because if an item set satisfies the minimum support requirements, so do all of its subsets.

The basic idea of the Apriori algorithm is to generate candidate itemsets of a particular size and then scan the database to count these to see if they are large. During scan i, candidates of size i, C_i are counted. Only those candidates that are large are used to generate candidates for next pass. That is, L_i is used to generate C_{i+1}.An itemset is considered as large, if all of its subsets are also large.

We can use this algorithm to generate the initial points of k-means algorithm for document clustering.

4. ALGORITHM DESCRIPTION

Assume that each document in the document-feature data set corresponds to an item in the transactional database; each feature corresponds to a transaction. The aim is to search for highly related documents appearing together with same features. Similarly, the frequent item set discovery in the transaction database serves the purpose of finding items appearing together in many transactions. Therefore, if we apply frequent item set discovery to our document feature data set, "frequent" document set will be discovered.

Here frequent document sets are documents appearing together with the same feature, i.e., document sets which have large amount of feature in common. These documents are considered to be related to a certain extent. Minimum support is the minimum similarity among documents in our application.

The advantage of using frequent item set discovery is that it can capture the relation among more than two documents while the normal similarity measurement, such as cosine similarity mentioned above, can only calculate the proximity between two documents. Moreover, frequent item set discovery is capable of detecting the most related document sets in the whole collection. These document sets can be viewed as having the highest density if we imagine all these document vectors are in a n-dimensional space. The density inside a correctly

defined cluster is normally higher than its outside area. Therefore, these document sets are regarded as the initial clusters and their centroids are the initial points for K-means algorithm.

A maximal frequent item set mining algorithm is employed in this experiment. Suppose that the required cluster number is k. Then we get the maximal frequent item sets with the largest support. The centers of those frequent item sets are the initial points of K-means algorithm.

The clustering process can be summarized as follows:

ALGORITHM

Input: Text files containing abstracts of various research papers, Stop word list and Minimum support.

Step1: Read terms in text files containing abstracts of research papers.

Step2: Remove terms in Stop word list and remove stemming using Porter Stemming Algorithm [8].

Step3: Prepare document feature matrix

Step4: The matrix generated in Step 3 and the minimum support will be given as input to Apriori Algorithm and get the Minimum Frequent Item sets (MFI)as output.

MFI=$\{I_1,I_2 \ldots I_k\}$

Where I=$\{d_a,d_b,\ldots d_c\}$.

Step5: For each document, generate the document vector.

d=$(tf(t_1,d) * idf_{t1}, tf(t_2,d) * idf_{t2}, \ldots tf(t_n,d) * idf_{tn})$

Step6: Calculate the initial centers as follows:

Calculate the center of each item set in MFI

Then IP is:

P_1=Center $_{I1}$

P_2=Center $_{I2}$

Pk=Center Ik

Where $$Center_I = \frac{\sum_{d \in I} d}{|I|}$$

Set the initial points of k-means algorithm as IP

Step7: Set the initial points of K-means algorithm as IP.

Get clustering result.

Output: The sets of highly related documents appearing together with same features.

The algorithm is depicted in Figure 1.

5. EXPERIMENTAL RESULTS

The process is implemented in JAVA.

A simple example is given here to illustrate the whole process of the approach. The data tested consists of twelve abstracts whose names were given as in table 1. The feature set includes six terms: document, cluster, vector, space, model, term. Table 2 shows the details of this document-feature data set. Given the minimum support 50%, two maximal frequent document sets were discovered. This maximal frequent document sets discovery procedure is depicted in Figure 2. Document vectors calculated by using equation 1 and equation 3 are shown in Table 2. They consist of six terms. The discovered maximal frequent document sets are considered to be the highest related documents and they construct the initial clusters. Therefore, their cluster centroids are computed according to equation 5. We set these generated vectors as the initial points in K-means algorithm. Then the algorithm starts to assign each document vector to its nearest cluster centroid and re-compute the new cluster center. This iteration continues until all the clusters do not change any more. Figure 3 illustrates the process and shows the final results. These twelve documents are divided into two groups.

6. CONCLUSION

In this paper, an approach for clustering web documents has been proposed. The experimental results of testing on web documents show that the proposed web document clustering method is clustering the relevant documents is more reliably and simply as compared to other document clustering methods. The proposed web document clustering method clusters the documents and presents to the researcher only those documents, which they intend.

FUTURE SCOPE

Study can be undertaken to assess the possibility of combining this method with clustering algorithms using wavelet analysis. As an extension, similar clustering techniques can be used to find the current trend of a particular research area, and to find the leading journals in a research area and the details about the researchers who are working in the same area.

REFERENCES

[1] Q.T.Tho, S.C. Hui and A.C.M. Fong, "A Web Mining Approach for Finding Expertise in Research Areas", Proceedings of the 2003 International Conference on Cyber worlds (CW'03), IEEE, Page(s):1-8.

[2] His-Cheng, Chang, Chieh Hsu and Yi-Wen Deng, "Unsupervised Document Clustering based on Keyword Clusters", International Symposium on Communications and Information Technologies 2004(ISCIT 2004) Sapporo, Japan, October 26-29, 2004, Page(s):1198-1203.

[3] Gahmoud F. Hussin, Ibrahim El Rube, and Mohamed S. Kamel "Enhanced Document Clustering Using Fusion Of Multiscale Wavelet Decomposition" IEEE/ACS International Conference on Computer M Syatems and Applications,March 31 2008-April 4 2008 Page(s):870 – 874.

[4] Hammouda, K.M., Kamel, M.S. **"**Efficient phrase-based document indexing for Web document clustering" Knowledge and Data Engineering, IEEE Transactions on Volume 16, Issue 10, Oct. 2004 Page(s):1279 – 1296.

[5] Rakesh Agrawal, Ramakrishnan Srikant, "Fast Algorithms for Mining Association Rules" Proceedings of the 20th VLDB Conference Santiago, Chile, 1994. Page(s):487-499.

[6] Manu Konchady, "Text Mining Application programming" Career and Professional Group, Page(s):276-278.

[7] Ling Zhuang, Honghua Dai," A Maximal Frequent Itemset Approach For Web Document Clustering", Proceedings of International Conference on Computer and Information Technology (CIT 2004), 14-16 September 2004, Wuhan, China. IEEE Computer Society 2004, Page(s):970 - 977.
[8] http://tartarus.org/~martin/PorterStemmer
[9] Raymond Kosala, Hendrik Blockeel," Web Mining Research: A Survey", SIGKDD, Volume 2, Issue 1, July'2000. Page(s):119-122.

Doc	docu ment	clu ster	vec tor	spa ce	mod el	term
Abs1	0	1	0	0	0	0
Abs2	1	1	1	0	0	0
Abs3	2	0	1	0	0	0
Abs4	2	1	2	0	3	0
Abs5	2	0	3	0	0	0
Abs6	1	0	2	0	0	0
Abs7	0	0	0	8	1	2
Abs8	0	1	0	4	3	1
Abs9	0	0	0	3	0	2
Abs10	0	0	0	6	3	3
Abs11	0	1	0	4	0	0
Abs12	0	0	0	9	1	1

Table 1: An Example of Document-feature Data

Document	Vector
Abs1	(0, 0.380, 0, 0, 0, 0)
Abs2	(0.380, 0.380, 0.380, 0, 0, 0)
Abs3	(0.760, 0, 0.380, 0, 0, 0)
Abs4	(0.760, 0.380, 0.760, 0, 1.14, 0)
Abs5	(0.760, 0, 1.14, 0, 0, 0)
Abs6	(0.380, 0, 0.760, 0, 0, 0)
Abs7	(0, 0, 0, 2.408, 0.380, 0.760)
Abs8	(0, 0.380, 0, 1.204, 1.14, 0.380)
Abs9	(0, 0, 0, 0.903, 0, 0.760)
Abs10	(0, 0, 0, 1.806, 1.14, 1.14)
Abs11	(0, 0.380, 0, 1.204, 0, 0)
Abs12	(0, 0, 0, 2.709, 0.380, 0.380)

Table 2: Document Vectors

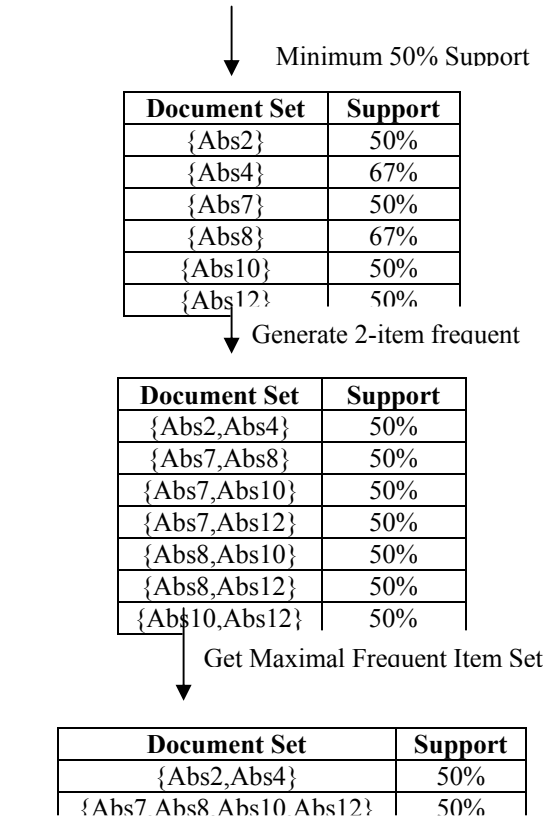

Figure 1: Discovering Initial Points Using

Maximal Frequent Document Set	Vectors
Abs2 Abs4	(0.380,0.380,0.380,0,0,0) (0.760,0.380,0.760,0,1.14,0)
Abs7 Abs8 Abs10 Abs12	(0,0,0,2.408,0.380,0.760) (0,0.380,0,1.204,1.14,0.380) (0,0,0,1.806,1.14,1.14) (0,0,0,2.709,0.380,0.380)

Calculate Cluster Centroids

Cluster	Centroid
1	(0.57,0.38,0.57,0,0.57,0)
2	(0,0.095,0,2.032,0.76,0.665)

Final Clustering of relevant Documents

	Relevant Documents
1	Abs1,Abs2,Abs3,Abs4,Abs5,Abs6
2	Abs7,Abs8,Abs9,Abs10,Abs11,Abs12

Figure 2: Process of K-Means Clustering

Continued on Page No. 146