International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 11 (2018) pp. 9646-9650

Clustering by Enhancing Co-occurrence Frequency

Rathinasabapathy R

Department of Computer Applications, Madurai Kamaraj University,
School of Information Technology, Madurai, Tamil Nadu, India

Abstract

Despite that the methodologies for document clustering are abundant in the research literature, the need for it still persists. Due to the large dimensionality and huge amount of resources to be searched, there is tremendous need for algorithms to be developed to mine the knowledge. In this research, the documents are represented using the term frequency, and it is converted into a vector by augmenting some of the features using frequent sets. Bisecting *k*-means algorithm is used to cluster the data. By changing the bounds on the similarity measures, various clusters can be obtained.

Keywords: Data mining, Document clustering, Algorithm for clustering, Frequent sets, Bisecting *k*-means, similarity measures.

INTRODUCTION

Data mining is a branch of research domain to explore the possibilities to extract the valuable information. Mining relevant information from large collections of documents is difficult and challenging. Clustering, classification, and association rule mining are some of the techniques which are using huge varieties of methodologies in the process. The electronic media has helped to accumulate valuable documents abundantly in a very short duration than one can extract the relevant information from them. The algorithms and methodologies are huge and variety in size and number. The process of extraction of knowledge has many pre-processing steps. It is a complex process involving reading the documents, parsing them into tokens, removing irrelevant tokens, and taking the relevant features. After getting the features representing each document, these are to be placed in appropriate data structures such as document vectors.

Clustering is broadly classified into partition clustering, hierarchical clustering, and density-based clustering. There are two types of hierarchical clustering algorithms, namely divisive and agglomerative clustering algorithms. In partition algorithm, the set of elements are split into several groups using some criteria. Usually, a select set of elements and their neighbourhood elements satisfying an objective function are forming clusters making the selected set of elements as the cluster center. Hierarchical clustering is used to group the elements into different clusters in a hierarchical manner. When whole set of elements is considered as a cluster, and it is repeatedly split hierarchically, it is known as divisive algorithm. When each one element is considered as individual cluster, and these are combined to form cluster as far as needed, the method is called agglomerative hierarchical algorithm. Density-based algorithms use the neighbourhood points of arbitrary point to cluster data[1]. Even though there seems to be fewer algorithms for clustering, there are lot of constraints over all these types of algorithms capable to be considered as individual algorithm, and lot of research has been made over each type of these algorithms [2].

Document clustering is the process of grouping the documents into several groups, and each group identifies certain information. The clustering methodologies are huge in number, and each one of them needs a lot of analysis. The algorithms are not able to cope with one or more of the following: High dimensionality, huge data base, clusters of poor quality, synonym and hypernym problem, inefficient similarity measures, leaving important features during dimension reduction, improper feature selection methods and inefficient document representation. Several algorithms have been developed to alleviate all these problems. The clustering algorithms are using various representations for documents, different criteria to represent the documents, different similarity measures between documents, between document and clusters. .

The foremost important part of document clustering is the appropriate representation of the document so that it can be processed easily and efficiently using the computers. At the same time, the information contained in the documents is to be taken without missing any important features. Many representations have evolved from using the term frequency (TF) representation such as including the measures of inverse document frequency (TF–IDF), inverse cluster frequency(TF–IDF–ICF) [3]. In term frequency representation, the documents are represented as

$d_i = (tf_{i1}, tf_{i2}, tf_{i3}, \ldots, tf_{in})$

wheretf_{ij} represents the frequency of occurrence of the term j in i^{th} document. The representations for document vectors also used term frequency and inverse document frequency (TF-IDF) taking into consideration the number of occurrences of the term in the set of all documents. The frequencies are represented as it appears or they are normalized. The frequencies are also including the semantic measures using WordNet and other measures [4]. Zhao and Karypis [5] discuss various partitions and agglomerative algorithms and their criterion functions. They provide a clustering algorithm called constrained agglomerative algorithm, combining the features of partition as well as agglomerative algorithms resulting in good quality clusters even for large data bases. Frequent terms based algorithms are also used for clustering the documents since each concept occurring in a document will represent the frequent sets. Frequent Item-setbased hierarchical clustering (FIHC) is explained using frequent sets[6]. It initially creates intersecting clusters, and later it is converted into non-

intersecting clusters. It uses cluster frequent sets and global frequent sets. Using a score function, the clusters are made non-intersecting clusters. The elements in more than one cluster are accommodated into a cluster which is maximising the score function. Finding similarity between documents plays an important role in the process of clustering. The similarities can be found in various ways, namely cosine, Jaccard and Euclidean measures [7]. Similarity measures are used to find similarity between documents and between documents and clusters. The minimum and maximum threshold similarities are used to vary the clusters [8].

In this work, the documents are parsed to find document vectors, and it is represented as term weight vectors. Then the vectors are normalised by dividing the frequencies by the sum of the frequencies of the terms occurring in the document. The term weight vector is partitioned using threshold frequency which is to be fixed using statistical measure such as mean frequency of term frequencies occurring in all the documents. The mean frequency is also normalized using the sum of frequencies occurring in the document. Then the terms corresponding to frequencies occurring with less than the threshold are examined to find whether they are co-occurring with high frequency terms in the remaining documents. It can be found using the concept of frequent sets. The term with lower frequency is modified using co-occurrence high frequency term as explained later. Now, the vectors are converted into binary vectors using a threshold value. Then a similarity measure is used to cluster the data using bisecting *k*-means algorithm. It is observed that the bisecting *k*-means hierarchical algorithm works better than *k*-means algorithm [9]. In the remainder of the paper, "FREQUENT SETS AND CLUSTERING" Section describes frequent sets and clustering, "CREATING BINARY VECTOR AND CLUSTERING" Section describes the process of proposed clustering method, "ALGORITHM FOR CLUSTERING" Section describes the algorithm using the proposed method, and "EXAMPLES AND DISCUSSION" Section explains the proposed method with examples and "CONCLUSION" section provides concluding remarks.

FREQUENT SETS AND CLUSTERING

Initially, association rule mining, a datamining technique, is used to do market basket analysis. In a set of transaction T,

$$T=\{t_1,t_2,t_3,\ldots,t_n\}$$

where each t_i is a set of elements from the set $S=\{e_1,e_2,e_3,\ldots\}$.

Then the association rule $A\Rightarrow B$ is said to have support s and confidence c, if $|AUB|/|T|$ is greater than s, and $|AUB|/|A|$ is greater than c.

A subset A of a set S is frequent in a set of transactions, if it exists in more than a specified number of times in the set of transactions T. From the definition of association rules, it can be observed that the frequent sets are to be identified, before finding association rules. Once frequent sets are found, then it is easy to find association rules. There are several algorithms existing to find frequent sets [10].

The association rule mining algorithms are categorised into horizontal, vertical and tree based algorithms. Usually, the transactional data bases hold values either as 0 or 1 notifying the presence of an item or absence of it. Some algorithms are dealing with numeric values for the elements in which case algorithms are converting them into fuzzy values, and the algorithms are known as fuzzy association rule mining.

The terminology used for finding association rule can be applied to document clustering. The documents can be converted into term weight representation d_i,

$$d_i=\{f_{i1},f_{i2},f_{i3},\ldots,f_{im}\}$$

where each f_{ij} is the term weight for word j in document i. The set D will represent set of document vectors,

$$D=\{d_1,d_2,d_3,\ldots,d_n\}$$

The set D will represent the transaction set, and set of terms occurring in each document will represent the element occurring in each transaction. The d_i can be represented as binary vector ignoring the frequency with which they are occurring in a document, or it can be term weight vector (representing the frequency of occurrence of the term) as required by different algorithms

In finding association rules, first, 1-item frequent sets are found by scanning the transactional database. Then they are joined to form candidate 2-item candidate sets and from them frequent 2-item sets are found. Obviously, the documents containing *k*-set frequent will be super set of those containing (*k*+1)-set frequent. Using the concept of frequent sets, hierarchical document clusters are found in various research works [11]. As each document can also be represented as a TF vectors, it can be converted into fuzzy values, and fuzzy association rule mining algorithms can be used to cluster the documents. FMDC provides fuzzy association rule mining by representing each document using TF-IDF(tf_idf_{ij}) by introducing a unique measure [12].

CREATING BINARY VECTOR AND CLUSTERING

In this research work, first, the documents should be pre-processed, and representative features are to taken. The features of the documents are to represented as TF vector known as document vectors. The elements of the document vector are the frequency of representative terms in the document. The document vector for the document is given by

$$d_i=(w_{i1},w_{i2},w_{i3},\ldots,w_{im})$$

where each w_{ij} represents frequency of the word j in the document i.

Lot of algorithms have been proposed using this representation. In this work, first, the vector is converted into binary vector by using the following method. The vector is converted into normalised vector by dividing each element by the total sum of term frequencies occurring in the document as

$$d_{i'}=(w_{i1'},w_{i2'},\ldots,w_{im'})$$

where $w_{ij'} = w_{ij}/\sum_j w_{ij}$.

The keywords with higher frequency are assumed to be representative of the document. Usually, the words with lower frequency are assumed to be not representing the document, and these words are not taken into consideration for clustering. But it may not be so, since the higher abstraction of knowledge may appear only in few places in the document. When terms appearing with lower frequency in a document and but always occur in combination with terms having higher frequency, then they may be assumed to hold a higher frequency of occurrence, or it may also be taken to be representative of the document. Let us name such terms as below threshold words(denoted as below_threshold) and other terms are denoted as above_threshold. The below_threshold term's frequency is modified as above_threshold value by multiplying with a fraction r. The value r is the ratio of occurrence of maximal number of combined occurrence of the above threshold terms with a below threshold term to the number of occurrences of above threshold value individually in the document so that it may also be taken into consideration for clustering.

The value of r is given by the following

r = max{freq(below_threshold, above_threshold)}/freq(above_threshold)}

wherefreq(x,y) denotes the combined occurrence frequency of terms (x,y) in the set of documents, and freq(x) denotes the occurrence frequency of the term x in the set of documents.

The following will illustrate the above discussion:

Suppose that

$w_{il1'}$, $w_{il2'}$, $w_{il3'}$, …$w_{lip'}$>w_{th} and $w_{ik'}$<$w_{th,}$ where $w_{th}$is the assumed threshold frequency.

Find the index l_q for which

w_{ilq} = max{freq(k,l_n)}

where n varies from 1 to p, where freq(k,l_n) is the number of documents in which 2-set (k,l_n) occurs.

r = freq(k,l_q)/freq(l_q)

In the document di, make the value of $w_{ik'}$ which is lower than the threshold to the value of r*$w_{ilq'}$ (if the modified value is lower than $w_{ik'}$, the value is not modified).

Now, the binary vector is created using a threshold value as below.

b_i=(b_{i1},b_{i2},…,b_{im}), where b_{ij} = 1 if $w_{ij'}$>w_{th}

0 otherwise.

This will create a binary vector b_i for each document i. It may be observed that by changing the value of the threshold appropriately, many different binary vectors can be obtained. As the threshold increases, each vector b_i will have less number of 1s, and if the value is decreased, each vector will get many 1s. First choose the high value for the threshold, and cluster the vectors. If there is no sufficient result, the value may be decreased gradually until getting a desired number of clusters.

Let B be a binary matrix, B={b_1,b_2,b_3,…,b_m}. Let us define functions f and g to demonstrate the method of clustering.

Define a function f as,

f: BX{c_i} → R

where R is a set of real numbers, and f(b_i, c_i) = $\sum_j$ b_{ij}*c_{ij}, b_i,c_i ∈B.

Define a function g as

g:B→R, and g(c_i)=$\sum_j$ c_{ij}

Define h:B X {c_i}→R, where R is a set of real numbers, and h(b_i, c_i) = f(b_i,c_i)/g(c_i)

Take any candidate vector c_i in B such that g(c_i) is a minimum. The documents are to be clustered against the candidate vector c_i using the function h. The documents are to be clustered into two sets. One belongs to candidate vector c_i and other vectors that are again to be clustered using one of the remaining vectors. Take another candidate vector c_i in the remaining vectors of B and are clustered against this vector, and the process is repeated in the same fashion. The method to cluster is as shown below:

The cluster D_i is obtained as,

D_i = {b_i/h(b_i , c_p)>μ where g(c_p) is minimum and (b_i, c_p ∈ B) }.

The value h(b_i, c_p) will lie between 0 and 1. For each value of μ, different clusters can be formed. In this way, many clusters can be created. Now we have two sets, one is the cluster D_i of candidate c_p and the remaining as another set. The remaining set of elements are clustered in the same fashion by taking vector c_p for which g(c_p) is minimum. The clusters can be modified by fixing threshold for finding binary vectors and also by fixing the different values for μ.

ALGORITHM FOR CLUSTERING

The algorithm based on the above discussion has been elucidated below. Initially create a set of binary vectors B, as {b_1,b_2,b_3,…,b_m}. The clusters are to be stored into D_1,D_2,D_3,…, respectively. First select $b_i$from B having a minimum value of g(b_i) and name it as $c_{li.}$ The vectors in B are clustered against the vector c_{li}. By fixing the value of μ(>0), the vectors satisfying h(b_i,c_{li}) >μ are moved into cluster D_1, and it is removed from B. If it is not acceptable to the cluster being created, it is left in the set B. The vector c_{li} is added to the cluster $D_1$and also removed from B. The procedure is repeated until all the vectors are exhausted. By fixing different values for μ, different clusters can be obtained.

//Algorithm

/*

B be a binary vector.

D be a set of clusters as, D = {D_1, D_2, D_3, …}

b_i,c_{li} are vectors in B.

*/

i ← 0

While(B is nonempty)

{

i←i+1

c_{li} = {b_i/g(b_i) is a minimum and bi ∈ B }

For every b_q in B

{

if (c_{li} not = b_q)

{

t = h(b_q, c_{li})

If (t > μ)

{

Di = Di ∪ {b_q}

B = B - {b_q}

}

}

}//end of for

}

//end of algorithm

The above algorithm creates several clusters. We can have another type of cluster in which each element of B is used as cluster center. In such a clustering the elements may be in several clusters. Out of all the clusters for an element, the cluster element having the highest h value in a cluster will be retained in the cluster, and the element is removed from all other clusters having the lesser h value.

EXAMPLES AND DISCUSSION

Let us consider the following (Table 5.1) ideal example to illustrate the method described here. Assume that the actual clusters are (1 to 12) and (13 to 16).

Table 5.1 binary matrix for term weight vectors

slno	Term weight vectors
1	1 0 0 0 20 15 0 1 1 0 0 0 0
2	0 0 0 0 20 15 0 1 1 0 0 0 0
3	0 0 0 0 20 15 0 1 1 0 0 0 0
4	0 0 0 0 20 15 0 1 1 0 0 0 0
5	0 0 0 0 20 15 0 1 1 0 0 0 0
6	0 0 0 0 20 15 0 1 1 0 0 0 0
7	0 0 0 0 20 15 0 1 1 0 0 0 0
8	0 0 0 0 20 15 0 1 1 0 0 0 0
9	22 17 0 0 0 0 0 1 1 0 0 0 0
10	22 17 0 0 0 0 0 1 1 0 0 0 0
11	22 17 0 0 0 0 0 1 1 0 0 0 0
12	22 17 0 0 0 0 0 1 1 0 0 0 0
13	0 0 0 0 0 0 0 0 0 25 26 0 2
14	0 0 0 0 0 0 0 0 0 25 26 0 2
15	0 0 0 0 0 0 0 0 0 25 26 0 2
16	0 0 0 0 0 0 0 0 0 25 26 0 2

The above vectors will be converted into the following binary vector(Table 5.2) using the method described by using threshold value.

Table 5.2 binary matrix obtained from term weight vectors using threshold value.

slno	Term weight vectors
1	0 0 0 0 1 1 0 0 0 0 0 0 0
2	0 0 0 0 1 1 0 0 0 0 0 0 0
3	0 0 0 0 1 1 0 0 0 0 0 0 0
4	0 0 0 0 1 1 0 0 0 0 0 0 0
5	0 0 0 0 1 1 0 0 0 0 0 0 0
6	0 0 0 0 1 1 0 0 0 0 0 0 0
7	0 0 0 0 1 1 0 0 0 0 0 0 0
8	0 0 0 0 1 1 0 0 0 0 0 0 0
9	1 1 0 0 0 0 0 0 0 0 0 0 0
10	1 1 0 0 0 0 0 0 0 0 0 0 0
11	1 1 0 0 0 0 0 0 0 0 0 0 0
12	1 1 0 0 0 0 0 0 0 0 0 0 0
13	0 0 0 0 0 0 0 0 0 1 1 0 0
14	0 0 0 0 0 0 0 0 0 1 1 0 0
15	0 0 0 0 0 0 0 0 0 1 1 0 0
16	0 0 0 0 0 0 0 0 0 1 1 0 0

The clusters will be obtained as shown in Table 5.3

Table 5.3 Clusters obtained using initial binary vectors.

centre	μ=0.5	μ>0.5
1	{ (1 to 8) }	{ (1 to 8) }
9	{ (9 to 12 }	{ (9 to 12 }
13	{ (13 to 16) }	{ (13 to 16) }

When the term weights are modified the binary matrix will be as shown in Table 5.5. The clusters will be as shown in Table 5.4

The clusters will be obtained as below

Table 5.4 clusters obtained using the modified binary vectors

centre	μ=0.5	μ>0.5
1	{ (1 to 12) }	{ (1 to 8) }
9	-	{ (9 to 12) }
13	{ (13 to 16) }	{ (13 to 16) }

Table 5.5 modified binary vectors

slno	Modified Term weight vectors
1	0 0 0 0 1 1 0 1 1 0 0 0 0
2	0 0 0 0 1 1 0 1 1 0 0 0 0
3	0 0 0 0 1 1 0 1 1 0 0 0 0
4	0 0 0 0 1 1 0 1 1 0 0 0 0
5	0 0 0 0 1 1 0 1 1 0 0 0 0
6	0 0 0 0 1 1 0 1 1 0 0 0 0
7	0 0 0 0 1 1 0 1 1 0 0 0 0
8	0 0 0 0 1 1 0 1 1 0 0 0 0
9	1 1 0 0 0 0 0 1 1 0 0 0 0
10	1 1 0 0 0 0 0 1 1 0 0 0 0
11	1 1 0 0 0 0 0 1 1 0 0 0 0
12	1 1 0 0 0 0 0 1 1 0 0 0 0
13	0 0 0 0 0 0 0 0 0 1 1 0 1
14	0 0 0 0 0 0 0 0 0 1 1 0 1
15	0 0 0 0 0 0 0 0 0 1 1 0 1
16	0 0 0 0 0 0 0 0 0 1 1 0 1

It can be found that there is a significant change in the clusters formed when using modified binary matrix. When the numbers of documents and terms are huge, there will be enormous differences in getting the clusters.

CONCLUSION

In this work, a document clustering algorithm is demonstrated using term frequency representation. The documents are represented using the term frequency(TF) vectors. The terms having lower frequency in a document are modified using document frequency of co-occurring high frequency term in a document. The method also uses a threshold value to convert the vector into binary vector. Then, a similarity measure has been used to find the similarity between documents, and if the similarity is within the bounds, the documents are accepted to be similar. In this methodology, there are two ways in which the clusters can be modified. The clusters can be modified by fixing threshold for finding binary vectors and also by fixing the different values for μ. The advantages of the proposed methods: the clusters being improved by changing the parameters, efficient similarity method and semantic information being obtained using co-occurring frequent sets. There are many similarity measures available. The clusters can be created using those similarity measures, and the results can be compared and analysed.

REFERENCES

[1] Martin Ester, Hans-Peter Kriegel, Jiirg Sander, Xiaowei Xu A density-based algorithm for discovering clusters in large spatial databases with noise.Proceedings of the Second International Conference on Knowledge Discovery and Data Mining(1996):226-23.

[2] Xu R, Wunsch D. Survey of clustering algorithms.IEEE Transactions on Neural Networks2005;16(3):645–78.

[3] A. Keerthiram Murugesan and B. Jun Zhang, A New Term Weighting Scheme for Document Clustering(2011)

[4] Sedding J, Kazakov D. WordNet-based text document clustering. In: Proceedings of the third workshop on robust methods in analysis of natural language data. Association for Computational Linguistics; 2004.

[5] Zhao Y, Karypis G, FayyadU.Hierarchical clustering algorithms for document datasets. Data Mining and Knowledge Discovery2005;10(2):141–68.

[6] Fung BCM, Wang K, Ester M. Hierarchical document clustering using frequent itemsets. In: Proceedings of the 2003 SIAM international conference on data mining. Society for Industrial and Applied Mathematics; 2003.

[7] Huang A. Similarity measures for text document clustering. In: Proceedings of the sixth New Zealand computer science research student conference (NZCSRSC). Christchurch, New Zealand; 2008.

[8] Carullo M, Binaghi E, GalloI. An online document clustering technique for short web contents.Pattern Recognition Letters2009;30(10):870–6.

[9] Patil R, Khan M.Bisecting *K*-means for clustering web log data.International Journal of Computer Applications2015;116(19).

[10] Srikant R, Agrawal R. Mining generalized association rules;1995. p.407–19.

[11] Beil F, Ester M, XuX. Frequent term-based text clustering.In: Proceedings of the eighth ACM SIGKDD international conference on knowledge discovery and data mining. ACM; 2002.:436-442

[12] Chen CL, Tseng FSC, Liang T.An integration of WordNet and fuzzy association rule mining for multi-label document clustering. Data & Knowledge Engineering2010;69(11):1208–26.