International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 2 Issue 10, October - 2013

# Multiviewpoint Clustering based on Sequential Patterns

**Swetha Ponnoju**[1]
*M.Tech ,Department of Computer Science and Engineering,*
*Kakatiya Institute of Technology and Science,*
*Warangal, Andhra Pradhesh*

**M.Venugopal Reddy**[2]
*Assistant Professor,Department of Computer Science,*
*Kakatiya University,Warangal,Andhra Pradhesh*

**P. Niranjan Reddy**[3]
*Professor,Department of Computer Science and Engineering,*
*Kakatiya Institute of Technology and Science,*
*Warangal, Andhra Pradhesh*

## *Abstract*

*Clustering is the process of grouping the objects such that objects in one group are more similar when compared to the objects in another group. Most clustering techniques pre assumes some cluster relationship before clustering the documents. Similarity among some items is usually defined implicitly or explicitly. In this paper,we introduce some sort of novel approach for clustering the document based on the concept of sequential patterns. Multiviewpoint based similarity measure is used to cluster the documents based on this sequential patterns.This approach improves the efficiency of clustering the documents. Several higher dimension datasets are taken as input to show the resultant improvement in clusters of documents.*

***Keywords*** *— Cosine Similarity , Document Clustering, Multiviewpoint, Sequential Patterns , Similarity measure.*

## 1. Introduction

Clustering is a data mining technique used to place data elements into related groups based on some criterion. The goal of clustering is to find internal structures within data in order to place them in different subgroups.In our paper we implement using Partitioning Clustering Algorithm. Different Algorithms are proposed in data mining out of all those algorithms k-means still remains among the top 10 data mining algorithms[1]. It is the most commonly used partitional clustering algorithm. Its simplicity ,understandability and scalability are the reasons for its tremendous recognition. A common approach to the clustering problem is usually to treat it just as one optimization procedure. A best partition is available by optimizing a unique function with similarity (or distance) amongst the data.

Similarity measure plays an important role in a clustering method. The initial k-means algorithm contains sum-of-squared-error function which uses Euclidean distance as a similarity measure. The objects are mainly clustered based on the similarity measure used. In an exceptionally sparse as well as higher dimensional areas spherical k-means, which uses cosine similarity rather than Euclidean distance is used. Cosine similarity measures the cosine of the angle between two vectors. For finding the similarity between two text documents, the two vectors are usually the term frequency vectors of documents.

## 2. Related Work

Document clustering is the process of organising the documents into clusters based on the similarity between documents. The documents which are placed in one cluster are more similar to each other than the documents that are placed in other cluster. There are different types of

similarity measures introduced earlier for finding the similarity between two documents.

Euclidean distance is the similarity measure which is used in k-means algorithm to cluster the documents.This is the traditional algorithm for clustering the documents. If $d_i$ and $d_j$ are the two documents in a dataset its Euclidean distance is calculated as

$$\text{Distance}(d_i,d_j) = \|d_i - d_j\|$$

K means algorithm tries to minimize the distance between the document and the cluster centroid. So that the document belongs to that cluster[2].

For sparse and high dimensional data spherical k means algorithm is used to cluster the documents. This algorithm uses cosine similarity measure[3]. If $d_i$ and $d_j$ are the two documents its cosine similarity is defined by the cosine of the angle between two documents with respect to orgin. If the cos value is 1 it indicates that those two documents are similar. If the value is 0 then those documents are independent. The other values ranging from 0 to 1 is the similarity value of the two documents. If $d_i$ and $d_j$ are the unit document vectors then its cosine similarity is defined as

$$\cos(d_i , d_j) = d_i . d_j$$

otherwise the similarity between two documents is defined as

$$\cos(d_i,d_j) = d_i . d_j / \|d_i\| . \|d_j\|$$

The min-max cut graph based method uses cosine similarity measure to partition the graph. In this graph each document is a vertex and each edge weight is the similarity value between the two vertex joining the edge[4].

There are also other similarity measures used such as pearson correlation measure, jaccard coefficient measure etc. Pearson correlation measure is used in statistics to find the measure of how two variables are well related. For a sample, pearson correlation which is represented by 'r' is calculated as

$$r = \frac{n(\Sigma xy) - (\Sigma x)(\Sigma y)}{\sqrt{[\,n\Sigma x^2 - (\Sigma x)^2\,]\,[\,n\Sigma y^2 - (\Sigma y)^2\,]}}$$

Related to text documents there are phrase-based and concept based similarity measures which uses conceptual tree similarity measure to identify the similairity measure[6]. Ienco et al. introduced a similar context-based distance learning method for categorical data[7]

We have already seen that cosine similarity is used to calculate the cosine of the angle between two documents with respect to origin. There is another similarity measure named Multiviewpoint Based Similarity measure which finds the similarity between two documents in a cluster with respect to documents in another clusters i.e the similarity between two documents di and dj which are in the same cluster is defined as the average of similarities measured from the views of all other documents outside that cluster[5].Presumption of cluster membership is made prior. The formula for finding the MVS similarity measure is:

$$MVS(d_i , d_j \,|d_i,d_j \varepsilon S_r) = 1/n\text{-}n_r \Sigma \cos(d_i\text{-}d_h , d_j - d_h) \|d_i\text{-}d_h\| \|d_j\text{-}d_h\|$$

Where $d_i$ and $d_j$ are the two documents which are present in the same cluster $S_r$ and $d_h$ is the reference document outside this cluster.

The smaller the distance $\|d_i - d_j\|$ and $\|d_j - d_h\|$ are, the higher the chance that $d_h$ is in fact in the same cluster with $d_i$ and $d_j$.

These distances provide a measure of intercluster dissimilarity, given that points $d_i$ and $d_j$ belong to cluster Sr, whereas $d_h$ belongs to another cluster. The overall similarity between $d_i$ and $d_j$ is determined by taking average over all the viewpoints not belonging to cluster Sr.we use this multiviewpoint based similarity measure in our proposed work for clustering the documents.

Before finding the similarity between the documents we need to find the document vectors for all the documents in the dataset. Document vectors are subjected to some weighing schemes such as standard Term Frequency-Inverse Document frequency (TF-IDF) and these values are normalized to have unit length. In our paper we use

sequential pattern values instead of tf-idf values to cluster the documents efficiently. This approach is explained in detail in next section.

## 3. Proposed Work

Documents clustering is done previously by using the Term frequency- inverse document frequency (TF-IDF) values of the terms involved in the documents. Here we introduce a concept called sequential patterns which is used to find out the sequential patterns frequencies . This is calculated by using the term frequencies of documents as input to it.

***Sequential patterns :*** sequential patterns technique finds the sequential patterns with in a document and frequencies of the terms involved in the patterns . These frequency values of terms are used to cluster the documents. First by using these values , document vectors of all documents are constructed. Based on these document vectors similarities between the documents is calculated by using multiviewpoint based similairity measure. Finally clustering of documents is done . The algorithm for finding the sequential patterns and frequencies of the terms is as follows:

**SequentialPatterns:** ( Documents $d_i$ in dataset , Termfrequencies $tf_i$ )

**Input :** documents in the dataset $d_i$ , Termfrequencies $tf_i$.
**Output :** Sequential Pattern values of all the documents ($SP_i$ )

1. initialize 'n' , where n indicates number of documents in the dataset.
2. **For** i = 1 to n **do**
3. Initialize infile, intok as bufferedReader variables for storing the document $d_i$ and termfrequency file $tf_i$ respectively for each iteration.
4. Initialize altoken, alFreq as Arraylist and strTemp, str as String // altoken stores the results of all tokens(terms in file) and alfreq stores the frequency of token.
5. **While** ((strTemp = read the line from intok)!=NULL) **do**
6. **If** ( strTemp.length = = 0 ) **then**
7. continue next iteration of this loop
8. **End if**
9. String tokenizer is used to separate the tokens in strTemp
10. Read the next token into strToken from strTemp
11. Read the tokenFrequency into strFreq from strTemp
12. **If** ( strToken.count > 2 ) **then**
13. add strToken to alToken
14. add strFreq to alFreq
15. **End if**
16. **End While**
17. **While** ((str= read the Line from infile) != NULL ) **do**
18. **For** p = 0 to alToken.size **do**
19. **For** q = p to alToken.size **do**
20. **if** ( p = q ) **then**
21. Continue
22. **if** ((Str contains altoken(p)) and (Str contains altoken(q) )) **then**
23. k1= frequnecy value of p //which is known from alFreq(p)
24. k2= frequency value of q //which is known from alFreq(q)
25. update alFreq(p) with k1+k2
26. **Endif**
27. **End For**
28. **End For**
29. **End While**
30. Initialize bufferedWriter variable $SP_i$ //to write the sequential pattern values of document $d_i$
31. **For** k=0 to alToken.size **do**
32. Assign alToken(k) to strToken.
33. Assign alFreq(k) to strFreq.
34. Write ( strToken, strFreq) into $SP_i$ file.
35. New line in $SP_i$ file
36. **End For**
37. **End For**

In the above algorithm each iteration takes one document and its corresponding termfrequency file as input to find out the sequential patterns and frequency of terms in the document.
The while loop in step 5 is used to separate the terms and its frequencies of termfrequency file. This can be done by using string tokenizer. The terms are stored in alToken array list and its frequencies are stored in alFreq array list.
The while loop in step 17 is used to find out the sequential patterns in document by using the terms in alToken. The frequency of the terms is updated by adding the frequency

of all the terms in a pattern and assigning it to the first term in the pattern. This frequency values is updated for each term in alFreq array list by using index values of terms.
The for loop in step 31 is used to assign the terms in alToken to strToken and its corresponding values to StrFreq and those values are written in $SP_i$ file. This is an output file which consists of terms and its final frequency values after finding the sequential patterns. These frequency values is used to cluster the documents based on the similarity measure.

We have already known that the formula for calculating the Multiviewpoint Based Similarity measure is

$$MVS(d_i , d_j |d_i,d_j \varepsilon S_r) = 1/n\text{-}n_r \Sigma \cos(d_i\text{-}d_h , d_j – d_h) \|d_i\text{-}d_h\| \|d_j\text{-}d_h\|$$

Instead of cosine similarity in the above formula , Jaccard coefficient similarity can also be used. For text document, the Jaccard coefficient compares the sum weight of shared terms to the sum weight of terms that are present in either of the two document but are not the shared terms. If Jaccard Similarity is expressed over a bit vector, then it can be written as:

$$SIM(d_i,d_j)= d_i . d_j / \|d_i\|^2 + \|d_j\|^2 – (d_i , d_j)$$

Where $d_i$ and $d_j$ are the bit vectors whose similarity is measured by using the above formula.

Its similarity measure when we consider another reference document $d_h$ as a reference is given by:

$$MVS(d_i , d_j \mid d_i , d_j \varepsilon S_r) = 1/n\text{-}n_r \Sigma SIM(d_i,d_j) \|d_i\text{-}d_h\| \|d_j\text{-}d_h\|$$

Here the similarity of documents is found by considering other documents outside the cluster $S_r$ as reference. Finally the similarity is assigned by considering average of all the similarities.

Instead of finding the average over all reference documents for finding the final similarity value , we can take the mode of all the values of similarities which takes the value that appears most often and assigns it to the final similarity(MVS). If there is no mode value in all the similarity values, maximum value of all the similarities is considered and it is assigned to the final similarity.Even though the average value gives an efficient similarity value we use this mode and maximul values of similarities as an alternative.

## 4. Experimentation

We implement our paper by taking the Reuters TranscribedSubset dataset as input. This dataset consists of text documents which was created by selecting 20 files each from the 10 largest classes in the Reuters-21578 collection. These text documents are taken as input and clustering is done by using the multiviewpointbased similarity measure based on sequential patterns. We also implement the paper by taking higher dimensional basketball dataset which has information regarding the players and number of matches played. This dataset consists of data in xml format . Clustering of these xml documents is done using multiviewpoint based similarity measure based on sequential patterns.

## 5. Results

The concept of this paper is implemented and different results are shown below, The proposed paper is implemented in Java technology on a Pentium-IV PC with 20 GB hard-disk and 256 MB RAM. The propose paper's concepts shows efficient results and has been efficiently tested on different Reuters Datasets. The following figures shows the evaluation results.

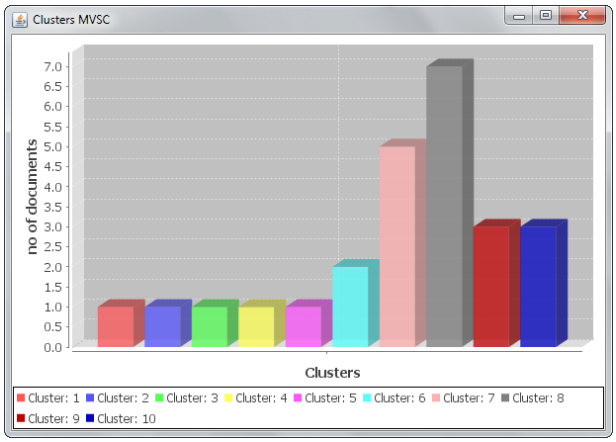


**Fig. 1**

The above graph represents 25 number of documents which are clustered, we have taken some documents from reuters dataset and clustered them in 10 clusters. The above graph comes when we use terms frequency instead of sequential patterns to cluster the documents.

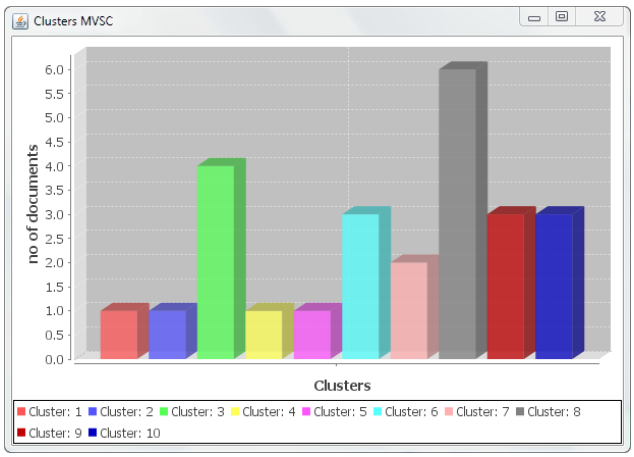


**Fig. 2**

The above graph represents the 25 number of documents which are clustered, we used sequential patterns concept to cluster the documents.we can observe that the documents are changed from one cluster to another cluster after using sequential patterns.

### 5.1. Validity Test

The improvement in clustering by using sequential patterns is clearly shown by using the validity graph. This graph is the result of validity test. The purpose of this test is to check how much a similarity measure coincides with the true class labels. It is based on one principle: if a similarity measure is appropriate for the clustering problem, for any of a document in the corpus, the documents that are closest to it based on this measure should be in the same cluster with it.

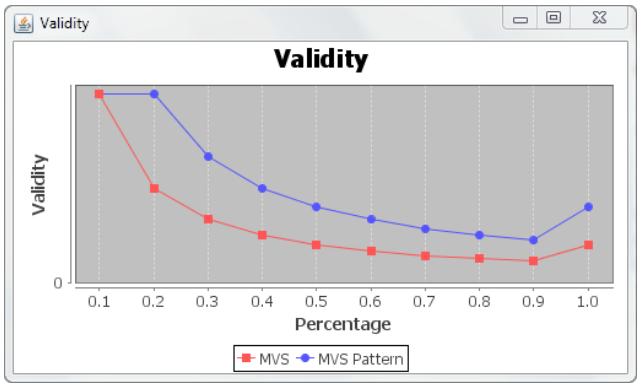


**Fig 3**

The above graph shows the validity scores of multiviewpoint based similarity measure using term frequency (red colour) and multiviewpoint based similarity measure using sequential patterns(blue colour). The validity score of mvs using sequential patterns is clearly better than mvs using term frequencies. Hence by using the sequential patterns the clustering of documents is done efficiently.

## 6. Conclusion

Within this paper, we propose a sequential patterns concept for Multiviewpoint based Similarity measuring procedure for clustering the documents. And we have made few changes to the MVS by introducing a new formula replacing the cosine for calculating the multiviewpoint based similarity measure. Instead of finding the average of all the similiarites of documents we considered the mode value and maximum value of the similarities of the reference document. And finally we have implemented the MVS using high dimensional xml dataset as input.Compared along with other state-of-the-art clustering techniques that use unique variations of similarity gauge, on a large number of document datasets and under different evaluation metrics, the proposed algorithm produce significantly enhanced clustering effectiveness.

## 7. Future Directions

The key contribution of the paper is based on the fundamental concept of similarity measure from several reference factors. Future strategies could makes use of the same theory, but determine alternative forms for the

relative likeness, or usually do not use average but have got other solutions other than max or mode to combine this relative similarities based on the different referrals points. Aside from, this paper targets partitioned clustering involving documents. Down the road, it would certainly also become possible to apply the proposed criterion capabilities for hierarchical clustering algorithms.

## References

[1] X. Wu, V. Kumar, J. Ross Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P. S. Yu, Z.-H. Zhou, M. Steinbach, D. J. Hand, and D. Steinberg, "Top 10 algorithms in data mining," Knowl. Inf. Syst., vol. 14, no. 1, pp. 1–37, 2007.

[2] Soumi Ghosh ,Sanjay Kumar Dubey, "Comparative Analysis of K-Means and Fuzzy C-Means Algorithms", International Journal of Advanced Computer Science and Applications, Vol. 4, No.4, 2013

[3] S. Zhong, "Efficient online spherical K-means clustering," in *IEEE IJCNN*, 2005, pp. 3180–3185.

[4] C. Ding, X. He, H. Zha, M. Gu, and H. Simon, "A Min-Max Cut Algorithm for Graph Partitioning and Data Clustering," Proc.IEEE Int'l Conf. Data Mining (ICDM), pp. 107-114, 2001.

[5] Duc Thang Nguyen, Lihui Chen, Chee Keong Chan, "Clustering with Multiviewpoint-Based Similarity Measure",in IEEE, VOL. 24, NO. 6, JUNE 2012.

[6] P. Lakkaraju, S. Gauch, and M. Speretta, "Document similarity based on concept tree distance," in Proc. of the 19th ACM conf. on Hypertext and hypermedia, 2008, pp. 127–132.

[7] D. Ienco, R. G. Pensa, and R. Meo, "Context-based distance learning for categorical data
clustering," in Proc. of the 8th Int. Symp. IDA, 2009, pp. 83–94.