I.J. Information Engineering and Electronic Business, 2017, 5, 44-49
Published Online September 2017 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijieeb.2017.05.06

A Top-Down Partitional Method for Mutual Subspace Clusters Using K-Medoids Clustering

***B. Jaya Lakshmi**
Department of Information Technology, GVPCOE(A), Visakhapatnam,530048,India
Email: meet_jaya200@gvpce.ac.in

K.B. Madhuri
Department of Information Technology, GVPCOE(A), Visakhapatnam,530048,India
Email: kbmcst1@yahoo.com

Received: 09 February 2017; Accepted: 25 July 2017; Published: 08 September 2017

***Abstract*—**In most of the applications, data in multiple data sources describes the same set of objects. The analysis of the data has to be carried with respect to all the data sources. To form clusters in subspaces of the data sources the data mining task has to find interesting groups of objects jointly supported by the multiple data sources. This paper addresses the problem of mining mutual subspace clusters in multiple sources. The authors propose a partitional model using k-medoids algorithm to determine k-exclusive subspace clusters and signature subspaces corresponding to multiple data sources, where k is the number of subspace clusters to be specified by the user. The proposed algorithm generates mutual subspace clusters in multiple data sources in less time without the loss of cluster quality when compared to the existing algorithm.

Index Terms—Mutual subspace clustering, Multiple data sources, Partitional clustering, Signature subspaces, Subspace.

I. Introduction

Subspace clustering is an extension of traditional clustering [1]. It finds set of objects that are homogeneous in subspaces of high-dimensional datasets. Mutual subspace clustering is the process of finding mutual subspace clusters from multiple sources. A single data source is used in subspace clustering whereas in mutual subspace clustering multiple data sources are used.

For example, consider an application of analyzing cancer patients [2]. For this purpose, both clinical data and genomic data have to be collected to develop effective therapies for cancers. Examining clinical data or genomic data individually might not expose the inherent patterns and correlations present in both data sets. Therefore, it is important to integrate clinical and genomic data and mining knowledge from both data sources. Clustering is a powerful tool for uncovering underlying patterns without requiring much prior knowledge about data [3-6]. To determine phenotypes of cancer, subspace clustering has been broadly used to explore such data.

To understand the clusters on clinical attributes well, and to find out the genomic explanations, it is highly appropriate to find clusters which are manifested in subspaces in both the clinical attributes and the genomic attributes. To check whether the cluster is mutual in a clinical subspace and a genomic subspace, the clinical attributes and genomic attributes are used to verify and justify. The mutual clusters are more understandable and interpretable. Mutual subspace clustering is used to integrate multiple sources and mine the related clusters [7].

Consider a data source as a set of points in a clustering space. Let S_1 and S_2 be two clustering spaces formed by subset of attributes. And $S_1 \cap S_2 = \Phi$, and O be a set of points in space $S_1 \cup S_2$ on which the clustering analysis is applied. A mutual subspace cluster is a triplet (C, U, V) such that $C \subseteq O$, $U \subseteq S_1$, $V \subseteq S_2$ and C is a cluster in both U and V respectively. U and V are called the signature subspaces of cluster C in S_1 and S_2 respectively. To make this simple, only two clustering spaces will be considered. However, this model can be easily extended to situations where more than two clustering spaces present.

This paper is organized as follows. Section II discusses the recent developments of the subspace clustering techniques and mutual subspace clustering techniques. The proposed methodology id detailed in section III. The results are analyzed in section IV. The paper is concluded in section V.

II. Related Work

Based on the strategy of subspace clustering there are two approaches namely, top-down approach and bottom-up approach [8]. Top-down approach finds an initial clustering in the full dimensional space and evaluates the subspaces of each cluster that iteratively improves the clustering results. The bottom-up approach finds dense regions in low-dimensional spaces and candidate low dimensional clusters are combined them to form clusters

in higher dimensional spaces [1]. The redundancy of subspace clusters is eliminated either as post pruning step or in the methodology itself as a wrapper approach.

The top-down methods of locality is determined by some approximation of clustering based on weights for the dimensions obtained so far [9-10]. The algorithms like PROCLUS, ORCLUS, FIND-IT, and δ – clusters determine the weights of instances for each cluster [11-14]. The algorithm COSA is unique in that uses the k-nearest neighbors for each instance in the dataset to determine the weights for each dimension for that particular instance [14].

The monotonicity of weak density is used by DUSC (Dimensionality on Biased Subspace Clustering) [15]. DUSC overcomes the problem of density divergence. The density divergence refers to the phenomenon of the data objects being spread farther apart with the increase in the number of dimensions. The process of DUSC is helpful in pruning the search space. Since the density threshold is not same for all the dimensions, the pruning criterion cannot be applied on the search space. The property of monotonicity no more holds. So, if a lower dimensional subspace does not yield any subspace cluster with a specified density threshold, a higher dimensional subspace may yield a subspace cluster with a different threshold [15].

Instead of subspace clusters being generated and then removing the redundant ones, the approach of INSCY (Indexing Subspace Clusters with-in-process-removal of redundancY) finds only subspace clusters which are non-redundant [16]. To accomplish this process a special index called SCY- tree is used to store the regions which are likely to hold subspace clusters. This technique reduces the repeated scanning of database cost for frequent pattern information which in turn stores the whole dataset in the SCY-tree data structure in a compact form with respect to all the projections with one scan of the database only.

Top down k-means method is appropriate where larger mutual subspace clusters exist [7]. That is, in a particular dataset most the points belong to a single mutual subspace cluster. The main goal of mutual subspace clustering is to derive the mutual subspace clusters that supports the multiple data sources. The process of mutual subspace clustering starts with arbitrary k points $c_1 \ldots c_k$ in the clustering space S_1 as the temporary centers of clusters $C_1 \ldots\ldots C_k$ respectively. The k centers do not necessarily belong to object set, O.

The data points in O will be assigned to the clusters according to their distances to the centers in space $S_1$and that point will be assigned to the closest center of the cluster. For each cluster in the subspace the signature subspace is found and the center of each subspace cluster. In this process firstly, we find the signature subspace and the center each subspace cluster in S_2. The Average pair wise distance is used to estimate the signature subspace in S_2. The Average pair wise distance is used to measure the compactness of the cluster.

This iterative process will be repeated until the clustering gets stable with low miss-assignment rate and the removal of conflict points. Once the conflict points are removed the clustering gets stable and finally the mutual subspace clusters are derived. The algorithm k-means which is sensitive to outliers may substantially distort the distribution of data because of an object with extremely large value [6,8].

III. Proposed Methodology

In top down k-medoids mutual subspace clustering, the representative data point for a given subspace cluster can be considered as a medoid instead of a cluster mean [17-19]. The cluster members are most similar to the its medoid or a representative object. Based on the clustering principle of minimizing the sum of the dissimilarities between each object and its corresponding reference point, the partitioning method generates the clusters after sufficient number of iterations. The initial representative objects will be chosen arbitrarily.

The process of replacing representative objects by non-representative objects will be done iteratively as long as the quality of the resulting clustering is improved. To measure the average dissimilarity between an object and the representative object of its cluster, the quality is estimated by using a cost function. A non-representative object o_{random} is determined which is a good replacement for current representative object o_j. The following four cases are examined for each of the non- representative objects, p.

Case 1: p currently belongs to representative object, o_j. If o_j is replaced by o_{random} as a representative object and p is closest to one of the representative objects, $o_{i,}$ then p is reassigned to o_i.

Case 2: p currently belongs to representative object, o_j. If o_j is replaced by o_{random} as a representative object and p is closest to $o_{random,}$ then p is reassigned to $o_{random.}$

Case 3: p currently belongs to representative object, o_i. If o_j is replaced by o_{random} as a representative object and p is still closest to o_i, then the assignment does not change.

Case 4: p currently belongs to representative object, $o_{i.}$ If o_j is replaced by o_{random} as a representative object and p is closest to $o_{random,}$ then p is reassigned to $o_{random.}$

A reassignment will be occurred each time, and a difference in absolute error, E will be contributed to the cost function. If a current representative object is replaced by a non-representative object, the cost function is calculated by the difference in absolute error-value. The total cost of swapping is the sum of costs incurred by all non-representative objects. To reduce the actual absolute error E, if the total cost is negative, then o_j is replaced or swapped with o_{random}. The current representative object, o_j is considered acceptable, if the total cost is positive and cluster members are not changed in the iteration.

A. Algorithm

Input: a set of points O in clustering spaces S_1 and $S_{2,}$ the number of clusters k, and parameters θ;

Output: a set of k mutual subspace clusters.

METHOD

1. select arbitrary k medoids $c_1 \ldots c_k$ in S_1;
2. Apply k-medoids clustering algorithm to find mutual subspace clusters;
3. assign each data point in O to a cluster of the closest medoid;
4. DO
5. Calculate Cost C and C'
6. If C' < C
7. FOR EACH cluster C_i DO
8. Find the signature subspace in S_2 and the medoid;
   END-DO
9. FOR EACH cluster C_i DO
10. Find the signature subspace in S_1 and the medoid;
    END-DO
11. IF cluster medoids are stable THEN remove conflict points;
12. UNTIL the clustering is stable;
13. FOR EACH cluster C_i DO
14. Output (C_i, U_i, V_i) where C_i is the set of points in C_i, U_i and V_i are the signature subspaces in S_1 and S_2 respectively.
15. END- FOR

The two subspaces S_1 and S_2 and the number of clusters taken as input to the k-medoids. The process starts by selecting arbitrary centers of clusters $C_1 \ldots C_k$ respectively in the clustering space S_1. The points O will be assigned to the clusters based on their distance from the center to the point in subspace S_1. Choose a random medoid for each cluster.

The cost C (old cost) and C' (new cost) will be calculated. If the new cost is lesser than old cost, then the refinement stops. To refine the clusters in the clustering space $S_2$the mutual subspace clusters will be described. In order to improve the cluster assignment, we need to find out the signature subspaces in S_2. The distance should be checked for each point and o will be assigned to the closest medoid in the signature subspace of the cluster for the improvement of the cluster assignment. The clustering refinement will be generated in this process. For the refinement process the information of $S_2$is used in S_1. To adjust the cluster assignment, the signature subspaces will be computed and the medoids for each cluster in S_1 are chosen. A Mutual subspace cluster gets stable when the clustering spaces of the signature subspaces agree with each other. It means that the medoids of the clustering spaces attract to the same set of points approximately for a cluster.

The portion of data points in O which are assigned to different cluster in each iteration has been defined as the miss-assignment rate. The clustering of mutual subspace clusters gets stable when the miss-assignment rate and the signature subspaces of the clusters gets stable. When the signature subspaces of the clustering spaces do not change and the miss-assignment rate will be lower than θ% in two consecutive rounds of refinement, then the iterative refinement stops. Here θ is a user-specified threshold value.

On the other side approximate points might not belong to mutual subspace clusters then the iterative refinement might fall into an infinite loop, subsequently the two clustering spaces does not agree with each other on those points. To identify potential infinite loop, the cluster assignments has been compared in two consecutive rounds of two clustering spaces. The mis-assigned point each which is assigned in different clustering spaces for different clusters, and it is repeatedly assigned to the same clusters in same clustering spaces, then the cluster centers gets stable and the point that does not belong to any mutual cluster is removed. The removed point is named as a conflict point. When these conflict points are removed. Then the centers and the clusters gets stable, thus deriving mutual subspace clusters.

IV. Experimental Results

In the experimental results, the execution time has been compared between top-down k-means and top-down k-medoids methods on the datasets taken from UCI machine learning repository [20]. Some of the datasets namely Housing dataset, Wine recognition dataset, Seeds dataset and Column3weka dataset are made used for the purpose of analysis. The experiments are conducted on a PC with 8-bit core i5 processor, 8GB RAM.

Table 1. Comparison of execution time in milli-seconds between top-down k-means and top-down k-medoids for housing dataset.

No. of Clusters(k)	Top-down k-means method	Top-down k-medoids method
K=2	64580	64520
K=4	339880	339631
K=6	291588	291511
K=8	319048	309042

The Execution time expressed in milli-seconds is the time required to generate the subspace clusters. With the increase in k value the time required to generate the subspace clusters increases. Fig.1 depicts the execution time of top-down k-means and top-down k-medoids method when run on Housing dataset. The proposed method i.e. Top-down k-medoids method has marginally better performance compared to the existing method

Table 1, 2, 3 and 4 show the corresponding values of execution time for the increased k value when run on different datasets. Figure 2, 3 and 4 shows the performance of the proposed methods in terms of execution time for the other datasets and the corresponding values are tabulated in Table 2, 3 and 4.

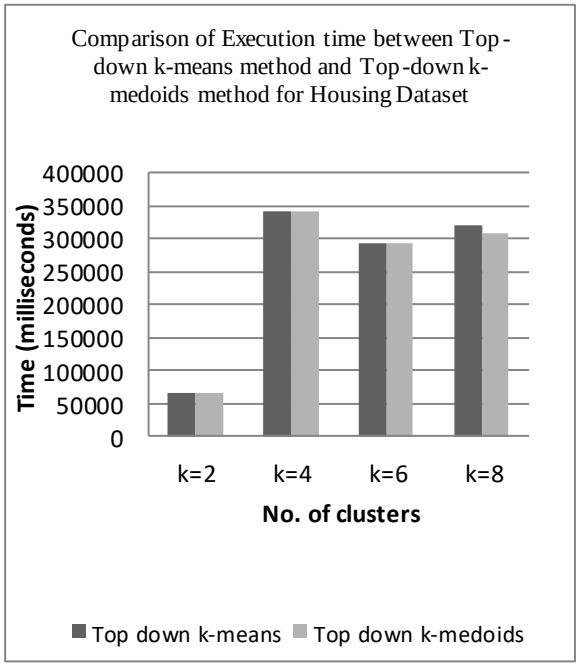

Fig.1. Comparison of Time between Top-down Method k-means and Top-down k-medoids in bar charts for Housing Dataset.

Table 2. Comparison of execution time in milli-seconds between top-down k-means and top-down k-medoids for wine recognition dataset.

No. of Clusters(k)	Top-down k-means method	Top-down k-medoids method
K=2	69250	69050
K=4	11511	10520
K=6	9968	8689
K=8	9462	9263

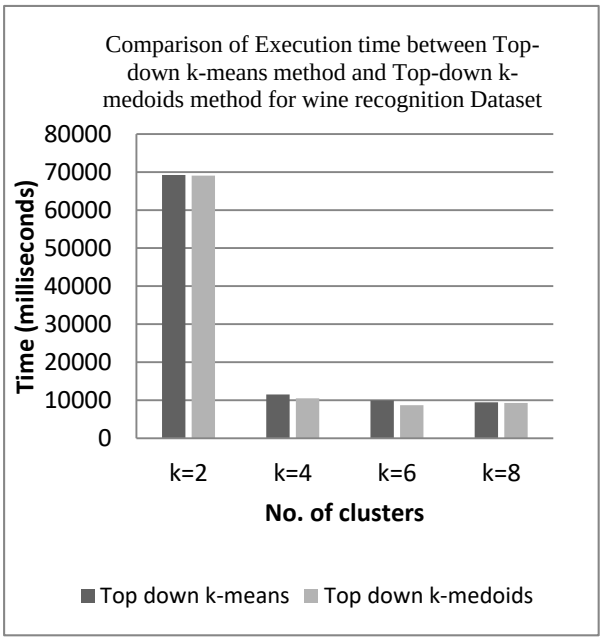

Fig.2. Comparison of Execution time between Top-down k-means method and Top-down k-medoids method in bar charts for wine recognition Dataset

Table 3. Comparison of execution time in milli-seconds between top-down k-means and top-down k-medoids for seeds dataset.

No. of Clusters(k)	Top-down k-means method	Top-down k-medoids method
K=2	776344	775323
K=4	10530	10030
K=6	9859	8583
K=8	11170	11063

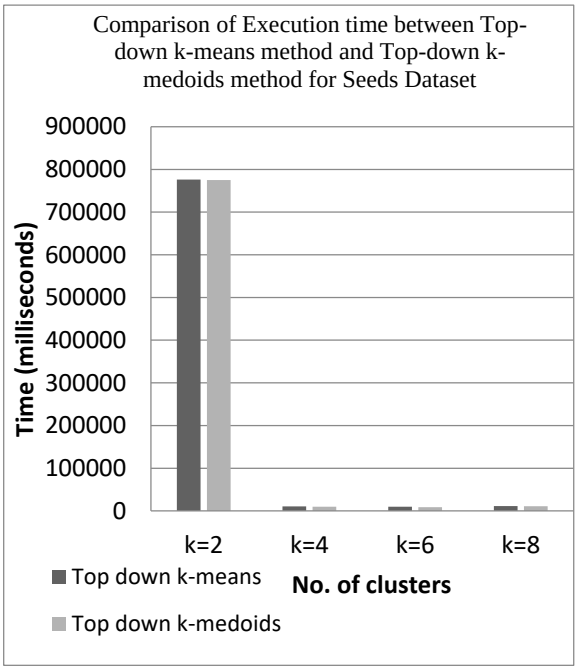

Fig.3. Comparison of Execution time between Top-down k-means method and Top-down k-medoids method in bar charts for Seeds Dataset.

The proposed algorithm converges sooner when compared to the existing algorithm. The medoids in each iteration is a data point through which the cluster assignment is done. The final clusters are obtained when the clustering result of two iterations is the same. With the increase in k value, the proposed algorithm converges to the output results in faster way as the distance computations and comparisons would reduce drastically.

Table 4. Comparison of execution time in milli-seconds between top-down k-means and top-down k-medoids for colum3weka dataset

No. of Clusters(k)	Top-down k-means method	Top-down k-medoids method
K=2	14135	14020
K=4	41063	41020
K=6	40962	39625
K=8	29539	28453

Purity is the most common metric used for measuring the quality of the clusters [6,8]. The Purity of a cluster is defined as the ratio of the number of data objects belonging to a maximum class to the total number of its cluster members. In this research work, the subspace

clusters with respect to multiple data sources are identified. The purity of a subspace cluster is computed with respect to the signature spaces. The class labels of the abovementioned datasets are compared while computing their purity.

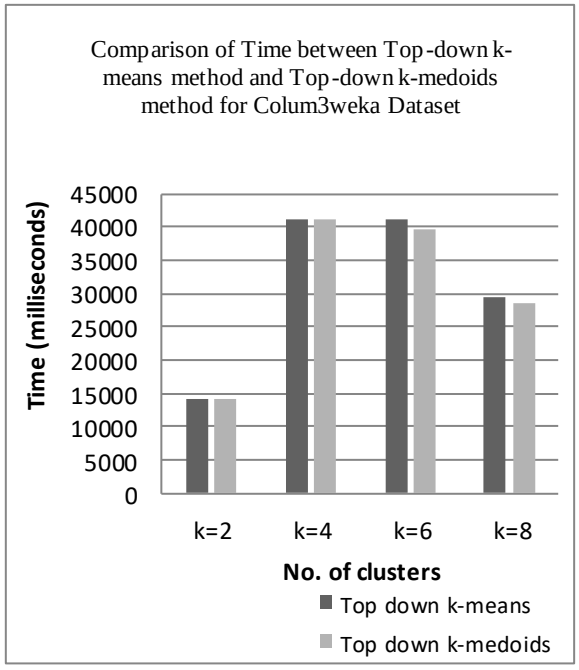

Fig.4. Comparison of Execution time between Top-down k-means method and Top-down k-medoids method in bar charts for Colum3weka Dataset

Table. 5, 6, 7 and 8 depict the purity of the resulted mutual subspace clusters when run datasets Housing dataset, Wine recognition dataset, Seeds dataset, Column3weka dataset respectively for different values of k. It could be observed that, as the k value increases, the purity of the mutual subspace clusters is improved.

Table 5. Comparison of purity between top-down k-means method and top-down k-medoids method for housing dataset

No. of Clusters(k)	Top-down k-means method	Top-down k-medoids method
K=2	0.461165	0.473526
K=4	0.582258	0.599543
K=6	0.687097	0.796432
K=8	0.696774	0.705603

Table 6. Comparison of purity between top-down k-means method and top-down k-medoids method for wine recognition dataset

No. of Clusters(k)	Top-down k-means method	Top-down k-medoids method
K=2	0.518258	0.536962
K=4	0.581461	0.597641
K=6	0.651685	0.675448
K=8	0.707865	0.714275

Table 7. Comparison of purity between top-down k-means method and top-down k-medoids method for seeds dataset

No. of Clusters(k)	Top-down k-means method	Top-down k-medoids method
K=2	0.65	0.6758439
K=4	0.761905	0.784592
K=6	0.714286	0.723955
K=8	0.728571	0.725694

Table 8. Comparison of purity b/w top-down k-means and top-down k-medoids method for colum3weka dataset

No. of Clusters(k)	Top-down k-means method	Top-down k-medoids method
K=2	0.675	0.683455
K=4	0.625806	0.635964
K=6	0.63871	0.641329
K=8	0.651613	0.661435

V. CONCLUSION

In this research work, a new data mining problem of mining mutual subspace clusters from multiple sources is studied. Most of the real time applications deal with multiple data sources describing the data objects in various contexts. There is a high need for efficient and effective techniques for carrying data analytics in a more meaningful way. This is helpful for the data analysts to make sound decisions.

We have developed an interesting partitional model that makes use of k-medoids clustering method for mutual subspace clustering. Experiments are conducted on synthetic data sets and real data sets to examine the effectiveness and the efficiency of the Top-down k-medoids method. In Section IV, The results are analyzed and found that the proposed method performs marginally better in terms of execution time and purity.

REFERENCES

[1] Kelvin Sim, Vivekanand Gopalkrishnan, Arthur Zimek, Gao Cong, "A survey on enhanced subspace clusterin," *Data Mining and Knowledge Discovery*, vol. 26, no.2(2013) pp.332-397. "doi: 10.1007/s10618-012-0258-x"

[2] Hichem Benfriha, Fatiha Barigou, Baghdad Atmani, "A text categorisation framework based on concept lattice and cellular automata", International Journal of Data Science(IJDS),Vol.1,No.3(2016) pp.227- 246. "doi: 10.154/IJDS.2016.075933"

[3] Semire Yekta, "A Qualitative Research of Online Fraud Decision-Making Process", *International Journal of Computer and Information Engineering-World Academy of Science, Engineering and Technology* Vol.4, No.5, 2017.

[4] Martin Böhmer, Agatha Dabrowski, Boris Otto, " Conceptualizing the Knowledge to Manage and Utilize Data Assets in the Context of Digitization: Case Studies of Multinational Industrial Enterprises", International *Journal of Computer and Information Engineering-World*

Academy of Science, Engineering and Technology, Vol.4, No.4, 2017.
[5] Sangeeta Yadav, Mantosh Biswas, " Improved Color-Based K-Mean Algorithm for Clustering of Satellite Image", *International Journal of Computer and Information Engineering-World Academy of Science, Engineering and Technology*, Vol.4, No.2, 2017.
[6] Pang-Ning Tan, Michael Steinbach, Vipin Kumar, "Introduction to Data Mining," Addison-Wesley Longman Publishing Co., Inc., Boston, MA, 2005.
[7] Ming Hua, Jian Pei "Clustering in applications with multiple data sources—A mutual subspace clustering approach" N*eurocomputing,*Volume.92, no.1, 2012, pp133–144. "doi: 10.1016/j.neucom.2011.08.032".
[8] Jiawei Han, Micheline kamber, "Data Mining: Concepts and Techinques", Second Edition, The Morgan Kaufmann Series in Data Management System.
[9] Zhang Huirong, Tang Yan, He Ying, Mou Chunqian, Xu Pingan, Shi Jiaokai, "A novel subspace clustering method based on data cohesion," *in ModelOptik - International Journal for Light and Electron Optic*, vol. 127, issue 20,(2016) pp. 8513-8519."doi: 10.1016/j.ijleo.2016.06.004".
[10] L.Parsons, E.Haque, H.Liu, "Subspace clustering for high dimensional data: a review," *SIGKDD Explor.Newsl.*vol.6, no.,1 90–105, 2004. (Online: http:// dl.acm.org/ citation.cfm? id=1007731).
[11] C. C. Aggarwal, C. M. Procopiuc, J. L. Wolf, P. S. Yu, and J. S. Park, "Fast algorithms for projected clustering", *In Proc. of ACM SIGMOD Intl. Conf. Management of Data* (1999) pp 61–72 (Online:https:// citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.108.5164).
[12] C. C. Aggarwal, and P. S. Yu, "Finding generalized projected clusters in high dimensional space", *In Proc. of ACM SIGMOD Intl. Conf. Management of Data*, (2000) pp 70–81. Online:http:// dl.acm.org/citation. cfm?id=335383)
[13] K.-G.Woo,J.-H.Lee,M.-H.Kim,Y.-J.Lee, "Findit: a fast and intelligent subspace clustering algorithm using dimension voting*,", Inf. Software Technol.* vol.46, no.4 (2004) pp 255–271."doi: 10.1016/j.infsof.2003.07.003".
[14] J. H. Friedman and J. J. Meulman, "Clustering objects on subsets of attributes", *Royal Statistical Theory*, pp. 815–84 2004."doi: doi=10.1.1.116.7209".
[15] Assent I, Krieger R, M ̈uller E, and Seidl T,"DUSC: Dimensionality unbiased subspace clustering," *In proceedings International Conference on Data Mining*, pp.409-414,2007 (Online: https://www.ipd.kit.edu/~muellere/publications/ICDM2007.pdf)
[16] Assent, I., Krieger, R., Muller, E., and Seidl, T, "INSCY: Indexing subspace clusters with in-process-removal of redundancy", *Proceeding of IEEE international conference on data mining*, Pisa, Italy, pp. 719-724, 2008. (Online: http://ieeexplore.ieee.org/abstract/document/4781168/)
[17] Jin, X. and Han, J, "K-Medoids Clustering, Encyclopedia of Machine Learning", *Springer US*, pp. 564-565, 2010. "doi: 10.1007/978-0-387-30164-8_426"
[18] Siddu P. Algur, Prashant Bhat, "Web Video Object Mining: Expectation Maximization and Density Based Clustering of Web Video Metadata Objects", *International journal of Information Engineering and Electronic Business*, Vol.8, no.1, pp.69-77, 2016."doi: 10.5815/ijieeb.2016.01.08".
[19] N. Krishnaiah, G. Narsimha, "Web Search Customization Approach Using Redundant Web Usage Data Association and Clustering*", International journal of Information Engineering and Electronic Business*, Vol.8, no.4, pp. 35-42, 2016."doi: 10.5815/ijieeb.2016.04.05".
[20] Lichman. M., UCI Machine Learning Repository. [online] http://archive.ics.uci.edu/ml. Irvine, (Accessed on 20 August 2016).

Authors' Profiles

B. Jaya Lakshmi received M.Tech. degree in Computer Science and Technology (Specialization-Artificial Intelligence & Robotics) from AU College Of Engineering(A), Andhra University, Visakhapatnam in 2009. She is pursuing Ph.D in JNTUK, Kakinada. Presently she is working as Assistant Professor in department of Information Technology at Gayatri Vidya Parishad College of Engineering(A), Visakhapatnam, Andhra Pradesh, India. Her research interests include Data Mining and Pattern Recognition. She published research papers in International Journals. She obtained UGC minor research project No.F:MRP-4554/14(SERO/UGC) in 2014.

K.B. Madhuri received M.Tech. degree in Computer Science and Technology from Andhra University in 1999. She obtained Ph.D from JNTU, Hyderabad in 2009. Presently she is working as Professor and Head of the department in department of Information Technology at Gayatri Vidya Parishad College of Engineering (A), Visakhapatnam, Andhra Pradesh, India. Her research interests include Data Mining, Pattern Recognition, Data warehousing and RDBMS. She is currently guiding two Ph.D scholars. She published research papers in National and International Journals. She is a member of IEEE and associate member of Institute of Engineers (India).

How to cite this paper: B. Jaya Lakshmi, K.B. Madhuri, "A Top-Down Partitional Method for Mutual Subspace Clusters Using K-Medoids Clustering", International Journal of Information Engineering and Electronic Business(IJIEEB), Vol.9, No.5, pp. 44-49, 2017. DOI: 10.5815/ijieeb.2017.05.06

