PATTERN BASED SUBSPACE CLUSTERING: A REVIEW

Debahuti Mishra[1], Shruti Mishra[2], Sandeep Satapathy[3], Amiya Kumar Rath[4] and Milu Acharya[5]
[1,2,3,5] Department of Computer Science and Engineering,
Institute of Technical Education and Research
Siksha 'O' Anusandhan University, Bhubaneswar, Odisha, INDIA
[4]Department of Computer Science and Engineering
College of Engineering Bhubaneswar
Bhubaneswar, Odisha, INDIA
debahuti@iter.ac.in, shruti_m2129@yahoo.co.in, sandeepkumar04@gmail.com,
amiyamaiya@rediffmail.com and milu_acharya@yahoo.com

***Abstract*- The task of biclustering or subspace clustering is a data mining technique that allows simultaneous clustering of rows and columns of a matrix. Though the definition of similarity varies from one biclustering model to another, in most of these models the concept of similarity is often based on such metrics as Manhattan distance, Euclidean distance or other L_p distances. In other words, similar objects must have close values in at least a set of dimensions. Pattern-based clustering is important in many applications, such as DNA micro-array data analysis, automatic recommendation systems and target marketing systems. However, pattern-based clustering in large databases is challenging. On the one hand, there can be a huge number of clusters and many of them can be redundant and thus makes the pattern-based clustering ineffective. On the other hand, the previous proposed methods may not be efficient or scalable in mining large databases. The objective of this paper is to perform a comparative study of all subspace clustering algorithms in terms of efficiency, accuracy and time complexity.**

Keywords*: *Subspace clustering; Biclustering; p-cluster; z-cluster

I. INTRODUCTION

Some recent researches [7] indicate that pattern based clustering is useful in many applications. In general, given a set of data objects, a subset of objects form a pattern based clusters if these objects follow a similar pattern in a subset of dimensions. Comparing to the conventional clustering, pattern-based clustering is a more general model and has two distinct features. On the one hand, it does not require a globally defined similarity measure. Different clusters can follow different patterns on different subsets of dimensions.

On the other hand, the clusters are not necessary exclusive. That is, an object can appear in more than one cluster. The generality and flexibility of pattern-based clustering may provide interesting and important insights in some applications where conventional clustering methods may meet difficulties. Much active research has been devoted to various issues in clustering, such as scalability, the curse of high-dimensionality, etc. However, clustering in high dimensional spaces is often problematic. Theoretical results [1] have questioned the meaning of closest matching in high dimensional spaces. Recent research work [2, 3] has focused on discovering clusters embedded in subspaces of a high dimensional data set. This problem is known as subspace clustering. In this paper, we explore a more general type of subspace clustering which uses pattern similarity to measure the distance between two objects.

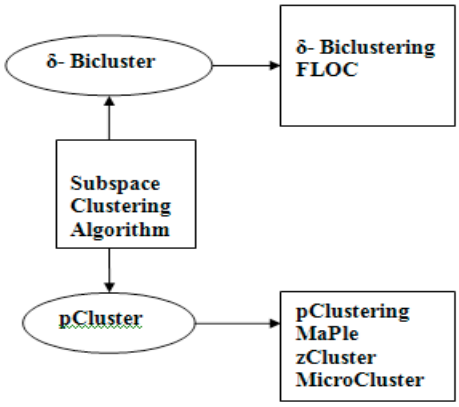

Figure 1: Subspace Clustering Methods

A. GOAL OF PAPER

In this paper we present a comparative study of all the subspace clustering algorithms present along with the method of how these algorithms are described. Our aim is just to chalk out the better subspace clustering algorithm in terms of accuracy, efficiency and time consumed.

B. PAPER LAYOUT

Section I gives the introductory concepts of subspace clustering, Section II, we present the abstracted view of the entire subspace clustering algorithm. We have also made a comparative discussion regarding the issues in each algorithm and to depict which is better. Section III gives the conclusion and the future work.

II. ALGORITHMS

Most clustering models, including those used in subspace clustering, define similarity among different objects by distances over either all or only a subset of the dimensions. Some well-known distance functions include Euclidean distance, Manhattan distance, and cosine distance. However, distance functions are not always adequate in capturing correlations among the objects. In fact, strong correlations may still exist among a set of objects, even if they are far apart from each other as measured by the distance functions. Some well-known subspace clustering algorithms are based on the main categories of approximate answers and complete answers.

A. δ -BICLUSTERING

In case of pattern based clustering major of grouping in objects shows similar patterns. Here we are considering the attribute values for which we have to go through the detailed features of objects. Hence the result we obtain is more accurate. In case of a data matrix clustering can be in the direction of a row or column. Simultaneous clustering of row and column of a matrix is called bi-clustering. Bi-clustering algorithms generate bi-clusters which is nothing but the similar behavior of a subset of rows across a subset of columns and vice versa. If some objects are similar in several dimensions (a subspace), they will be clustered together in that subspace. This is very useful, especially for clustering in a high dimensional space where often only some dimensions are meaningful for some subsets of objects.

Cheng et al. introduced the bi-cluster concept [3] as a measure of the coherence of the genes and conditions in a sub matrix of a DNA array. A sub matrix A_{IJ} is called a δ bi-cluster if $H(I, J)$ for some. Let X is the set of genes and Y the set of conditions. Let $I \subseteq X$ and $J \subseteq Y$ be subset of genes and conditions, respectively. The pair (I, J) specifies a sub matrix A_{IJ}. $H(I, J)$ is the mean squared residue score. Mean squared residue is the variance of the set of all elements in the bi-cluster plus mean row variance and the mean column variance.

$$H(I,J) = \frac{1}{|I|}\frac{}{|J|}\sum_{i\in I, j\in J}(d_{ij} - d_{iJ} - d_{Ij} + d_{IJ})^2$$

Where

d_{ij} = data value at row i and column j

d_{iJ} =mean of the i^{th} row in the sub matrix

d_{Ij} = mean of the j^{th} row in the sub matrix

d_{IJ} =mean of all elements in the sub matrix

A sub matrix A_{IJ} is called a δ-bi-cluster if $H(I, J) \leq \delta$ (where $\delta > 0$ is user defined some threshold value)

Yang et al. [5] proposed a move-based algorithm to find biclusters more efficiently. It starts from a random set of seeds (initial clusters) and iteratively improves the clustering quality. It avoids the cluster overlapping problem as multiple clusters are found simultaneously. However, it still has the outlier problem, and it requires the number of clusters as an input parameter.

There are several limitations of this work like the means squared residue used in [4, 5] is an averaged measurement of the coherence for a set of objects. But the most undesirable property is that a sub matrix of a δ bi-cluster is not necessarily a δ bi-cluster which creates a lot of difficulty in designing efficient algorithms.

If we set δ=2, the bi-cluster shown in Figure 2, contains an obvious outlier but it still has a fairly small mean squared residue (4.238).If we get rid of such outliers by reducing the δ threshold, it will exclude many bi-clusters which do exhibit similar patterns.

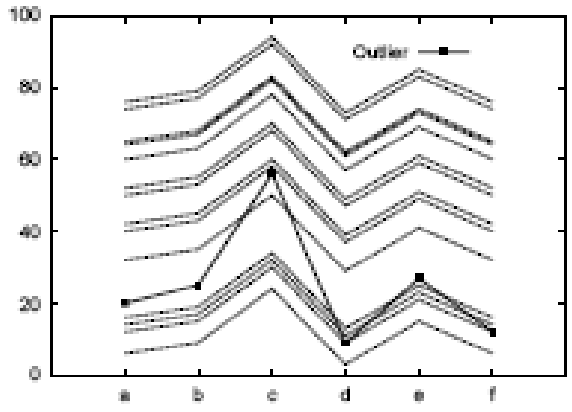

Figure 2: Data set with Residue 4.238

The below figure 3 shows mean square residue can not exclude outliers in a δ- bi-cluster.

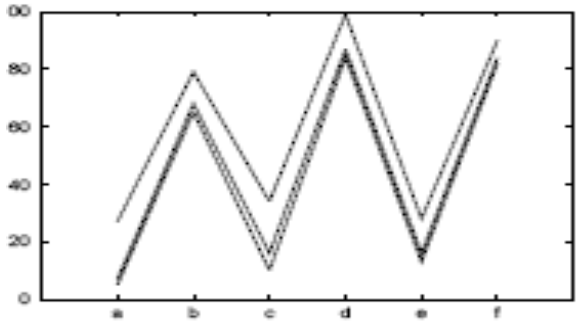

Figure 3: Data set with Residue 5.722

B. p-CLUSTERING

Unlike the bi-clustering algorithm and the δ-clusters algorithm the pCluster algorithm simultaneously detects

multiple clusters that satisfy the user-specified δ threshold. Under the pCluster model it has been proposed that, two objects are similar if they exhibit a coherent pattern (patterns that are related) on a subset of dimensions. Moreover, since the pCluster algorithm provides the complete answer, they will not miss any qualified subspace clusters, while random algorithms, e.g., the biclustering algorithm and the δ-clusters algorithm provide only an approximate answer.

Wang et al. proposed a clustering model, namely the pCluster, to capture not only the closeness of objects, but also the similarity of the patterns exhibited by the objects. We are generally interested in objects that exhibit a coherent pattern on a subset of attributes of A.

Let D be a set of objects, A be a set of attributes in D, (O, T) be a sub matrix where $O \subseteq D$ and $T \subseteq A$. If $x, y \in O$ and $a, b \in T$, then pScore of the 2×2 matrix is:

$$pScore\left(\begin{bmatrix} d_{xa} & d_{xb} \\ d_{ya} & d_{yb} \end{bmatrix}\right) = \left|\left(d_{xa} - d_{xb}\right) - \left(d_{ya} - d_{yb}\right)\right|$$

Again ,if pScore of the 2×2 matrix $\leq \delta$ for some $\delta \geq 0$ is said to form δ-p-Cluster. Where as, in a bi-cluster model a sub matrix of a δ-bi-cluster is not necessarily a δ – bi-cluster. However one important property of pCluster is anti – monotonicity which says that if (O, T) be a δ-pCluster then any of its sub matrix, (O', T') is also a δ-pCluster. Hence, from the definition we can infer that pCluster is symmetric. However, since a pCluster requires that every 2 objects and every 2 attributes conform to the inequality, it models clusters that are more homogeneous.

Basically, p-Cluster algorithms are a little bit slow but are very efficient and accurate for clinical purpose etc. It also mines the cluster simultaneously. The bi-cluster algorithm, on the other hand, finds clusters one by one, and the discovery of one cluster might obstruct the discovery of other clusters. This is time consuming and the cluster they find depend on the order of their search. Also, the pCluster model gives us many opportunities of pruning, that is, it enables us to remove many objects and columns in a candidate cluster before it is merged with other clusters to form clusters in higher dimensions.

The entire p-Cluster algorithm is achieved in three steps. They are mainly:

a) *Pair-Wise Clustering:* Based on the maximal dimension set Principle we find the largest (column) clusters for every two objects, and the largest (object) clusters for every two columns. Clusters that span a larger number of columns (objects) are usually of more interest, and finding larger clusters interest also enables us to avoid generating clusters which are part of other clusters.

b) *Pruning Unfruitful Pair-Wise Clusters*: Not every column (object) cluster found in pair wise clustering will occur in the final p-Clusters. To reduce the combinatorial cost in clustering, we remove as many pair-wise clusters as early as possible by using the Pruning Principle.

c) *Forming δ- p-Cluster:* In this step, we combine pruned pair-wise clusters to form p-Clusters.

C. z-CLUSTERING

Yoon et al[9] proposed the z-Cluster algorithm based on the pCluster model that exploits the zero-suppressed binary decision diagrams (ZBDDs) data structure to cope with the computational challenges. The ZBDDs have been used widely in other domains, namely, the computer-aided design of very large-scale integration (VLSI) digital circuits, and can be useful in solving many practical instances of intractable problems. The zCluster algorithm exploits this property of ZBDDs, and can find all the subspace clusters that satisfy specific input conditions without exhaustive enumeration. In order to generate MDSs, zCluster uses an approach similar to that used in the pCluster algorithm. The zCluster algorithm differs in the remaining steps after constructing the prefix tree used in pCluster. The zCluster algorithm efficiently utilizes ZBDDs [9] in the remaining steps. This ZBDD-based representation is crucial to keeping the entire algorithm computationally manageable set of condition-pair MDSs can be regarded as a set of combinations and represented compactly by the ZBDDs. Therefore, the symbolic representation using ZBDDs is more compact than the traditional data structures for sets. Moreover, the manipulation of condition-pair MDSs, such as union and intersection, is implicitly performed on ZBDDs, thus resulting in high efficiency.

Although the pCluster algorithm [6] and the zCluster algorithm [8] provide the complete answer, they contain some time-consuming steps. First, the pCluster algorithm and the zCluster algorithm equally use the clusters containing only two genes or two conditions to construct larger clusters having more genes and conditions, which are called gene-pair and condition-pair MDSs. However, this step of measuring the difference of each gene-pair on the conditions of a DNA microarray is really time consuming, since the number of genes in the real life microarray is usually very large. Thus, the time complexity of constructing the gene-pair MDSs is much higher than the time complexity of constructing the condition-pair MDSs in those previous proposed clustering algorithms. Also, the pCluster algorithm [5] proposes a prefix tree structure using the depth-first algorithm to mine the final subspace clusters. The zCluster algorithm [10] contains the similar step of mining. However, this step is the bottleneck of the mining. For each node, the pCluster algorithm has to examine the possible combinations of genes on the conditions registered in the path. The algorithm distributes the gene information

in each node to other nodes which represent subsets of the condition set along the path of this node. This distributing operation is the major cause that the pCluster algorithm may not be efficient or scalable for large databases.

D. MaPle

MaPle enumerates all the maximal pClusters systematically. It guarantees both the completeness and the non-redundancy of the search, i.e., every maximal pCluster will be found, and each combination of attributes and objects will be tested at most once. For each subset of attributes D, MaPle finds the maximal subsets of objects R such that (R,D) is δ- pCluster. If (R,D) is not a sub-cluster of another pCluster (R' ,D) such that R⊆R', then (R, D) is a maximal δ- pCluster. There can be a huge number of combinations of attributes. MaPle progressively refines the search step by step. Moreover, MaPle also prunes searches that are unpromising to find maximal pClusters. It detects the attributes and objects that can be used to assemble a larger pCluster from the current pCluster. If MaPle finds that the current subsets of attributes and objects as well as all possible attributes and objects together turn out to be a sub cluster of a pCluster having been found before, then the recursive searches rooted at the current node are pruned, since it cannot lead to a maximal pCluster.

Comparing to p-Clustering, MaPle has several advantages. First, in one of the step of p-Clustering, for each node in the prefix tree, the combinations of the objects registered in the node will be explored to find pClusters. This can be expensive if there are many objects in a node. In MaPle, the information of pClusters is inherited from the "parent node" in the depth-first search and the possible combinations of objects can be reduced substantially. Moreover, once a subset of attributes D is determined hopeless for pClusters, the searches of any superset of D will be pruned. Second, MaPle prunes non-maximal pClusters. Many unpromising searches can be pruned in their early stages. Third, new pruning techniques are adopted in the computing and pruning MDSs. That also speeds up the mining.

E. FLOC

The FLOC method also follows the δ-bi-cluster model. Its move-based algorithm, FLOC [6] which can efficiently and accurately approximate the _k δ clusters with the lowest average residue. The FLOC algorithm starts from a set of seeds (initial clusters) and carries out an iterative process to improve the overall quality of the clustering. At each iteration, each row and column is moved among clusters to produce a better clustering in terms of a lower average residue [7]. The best clustering obtained during each iteration will serve as the initial clustering for the next iteration. The algorithm terminates when the current iteration fails to improve the overall clustering quality.

III. CONCLUSION

Out of all the algorithms, pCluster Model captures the closeness of objects and pattern similarity among the objects in subsets of dimensions. It is found that it discovers all the qualified pClusters. The depth-first clustering algorithm avoids generating clusters which are part of other clusters. This is more efficient than other current algorithms. It is resilient to outliers. Our future work would be to hybridize pCluster model with any soft computing technique.

REFERENCES

[1] Beyer K, Goldstein J, Ramakrishna R, Shaft U,"When is nearest neighbors meaningful," in Proc. of the International Conference in Database Theories, 1999, pp.217 {235).

[2] Aggarwal C C, Procopiuc C, Wolf J, Yu P S, Park J S. ,"Fast algorithms for projected clustering," in Proc. of SIGMOD, Philadephia, USA, 1999, pp.61 {72).

[3] Cheng C H, Fu A W, Zhang Y. ,"Entropy-based subspace clustering for mining numerical data," in Proc. of SIGKDD, San Diego, USA, 1999, pp.84 (93).

[4] Cheng Y, Church G.," Biclustering of expression data," in Proc. of 8th International Conference on Intelligent System for Molecular Biology, 2000, pp.93 (103).

[5] Yang J, Wang W, Wang H, Yu P S,"δ -clusters: Capturing subspace correlation in a large data set," in Proc. of ICDE, San Jose, USA, 2002, pp.517 {528.

[6] H. V. Jagadish, J. Madar, and R. T. Ng,"Semantic compression and pattern extraction with fascicles," in the Proc. of 25th VLDB, 1999, pp. 186- 198..

[7] Wang H., "Clustering by pattern similarity in large data sets," in the Proc. of SIGMOD, 2002.

[8] Minato, S.,"Zero-Suppressed BDDs for Set Manipulation in Combinatorial Problems," in Proc. of IEEE/ACM Design Automation Conf., 1993,pp. 272–277.

[9] Yoon, S., Nardini, C., Benini, L., Micheli, G. D.," Discovering Coherent Biclusters from Gene Expression Data Using Zero-Suppressed Binary Decision Diagrams," in the Proc. of IEEE/ACM Trans. on Computational Biology and Bioinformatic 2 (4), 2005, pp.339–354.

AUTHORS PROFILE

Debahuti Mishra is an Assistant Professor and research scholar in the department of Computer Sc. & Engg, Institute of Technical Education & Research (ITER) under Siksha `O` Anusandhan University, Bhubaneswar. She received her Masters degree from KIIT University, Bhubaneswar.Her research areas include Data mining, Bio-informatics Software Engineering, Soft computing . She is an author of a book Aotumata Theory and Computation by Sun India Publication (2008).

Shruti Mishra is a scholar of M.Tech(CSE) Institute of Technical Education & Research (ITER) under Siksha `O` Anusandhan University, Bhubaneswar.Her research areas include Data mining, Parallel Algorithms etc.

Sandeep Kumar Satapathy is Lecturer in the department of Computer Sc. & Engg, Institute of Technical Education & Research (ITER) under Siksha `O` Anusandhan University, Bhubaneswar . He has received his Masters degree from Siksh `O` Anusandhan University, Bhubaneswar. His research areas include Web Mining, Data mining etc.

Dr.Amiya Kumar Rath obtained Ph.D in Computer Science in the year 2005 from Utkal University for the work in the field of Embedded system. Presently working with College of Engineering Bhubaneswar (CEB) as

Professor of Computer Science & Engg. Cum Director (A&R) and is actively engaged in conducting Academic, Research and development programs in the field of Computer Science and IT Engg. Contributed more than 30 research level papers to many national and International journals. and conferences Besides this, published 4 books by reputed publishers. Having research interests include Embedded System, Adhoc Network,Sensor Network ,Power Minimization, Biclustering, Evolutionary Computation and Data Mining.

Dr. Milu Acharya obtained her Ph.D (Utkal University), she is Professor in Department of Computer Applications at Institute of Technical Education and Research (ITER)) under Siksha `O` Anusandhan University, Bhubaneswar r. She has contributed more than 20 research level papers to many national and International journals and conferences Besides this, published three books by reputed publishers. Her research interests include Biclustering,Data Mining , Evaluation of Integrals of analytic Functions , Numerical Analysis , Complex Analysis , Simulation and Decision Theory.