A COMPARATIVE STUDY IN SUBSPACE AND PATTERN BASED CLUSTERING

Debahuti Mishra[1], Shruti Mishra[2], Sandeep Kumar 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, Odisha, INDIA
[4]Department of Computer Science and Engineering College of Engineering, Bhubaneswar, Odisha, INDIA
Email: [1]debahuti@iter.ac.in

ABSTRACT

Subspace clustering is an extension of traditional clustering that seeks to find clusters in different subspaces within a dataset. Traditional clustering algorithms consider all of the dimensions of an input dataset in an attempt to learn as much as possible about each instance described. In very high dimensions it is common for all of the instances in a dataset to be nearly equidistant from each other, completely masking the clusters. Subspace clustering algorithms localize the search for relevant dimensions allowing them to find clusters that exist in multiple, possibly overlapping subspaces. For this they basically use certain distance functions like 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. Pattern-based clustering, a kind of subspace clustering methods, is effective in discovering such clusters. Conceptually, in given a dataset a subset of objects form a pattern-based cluster if these objects follow a similar pattern in a subspace. Some well-known subspace clustering algorithms are based on the main categories of approximate answers and complete answers. Moreover, the p-Cluster algorithm provides the complete answer; they will not miss any qualified subspace clusters, while random algorithms, e.g., the bi-clustering algorithm and the δ-clusters algorithm, provide only an approximate answer. This paper tries to compare the above two clustering methods as per scalability, structure, correlations and efficiency.

I. INTRODUCTION

In the knowledge discovering process, clustering aims at detecting groups of similar objects while separating dissimilar ones. Traditional clustering approaches compute a partition of the data, grouping each object in at most one cluster or detecting it as noise. However, it is not always the case that an object is part of only one cluster. Multiple meaningful groupings might exist for each object. As multiple concepts described by different attributes are mixed in the same data set, clusters are hidden in subspace projections and do not appear in all dimensions. Subspace clustering aims at detecting such clusters in any projection of the database. Researches indicate [1] that pattern based clustering is far useful in many applications. Basically, 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.

In comparison, to the conventional clustering, pattern-based clustering is a more general model. It does not require a globally defined similarity measure. Here, different clusters can follow different patterns on different subsets of dimensions. Also, the clusters are not necessarily exclusive. That is, an object can appear in more than one cluster. The generality and flexibility of pattern-based clustering may also provide interesting and important insights in some applications where conventional clustering methods may meet difficulties. Fig.1 shows various categories of subspace clustering algorithm.

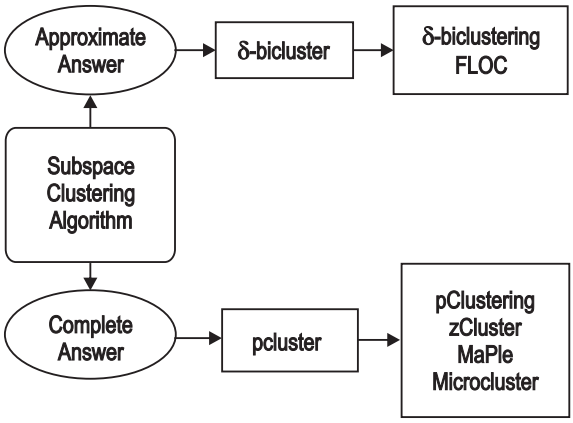

Fig. 1. Subspace clustering algorithm

A. Goal of paper

This paper gives a comparative study of subspace clustering and pattern based clustering. This paper gives a detail study on both the clustering methods. The comparison is done by taking many parameters as shown in table II.

B. Paper layout

The layout of the paper is as follows: Section I gives the introductory concepts of subspace as well as pattern based clustering. Section II deals with preliminary concepts of subspace clustering and section III describes the pattern based clustering. Section IV gives comparison on both the techniques. Finally, section V gives the conclusion and future work.

II. SUBSPACE CLUSTERING

Subspace clustering is an extension of traditional clustering; it aims to find clusters embedded in subspaces of a high dimensional dataset. We can classify the existing methods of subspace clustering into two types according to their similarity measures. One type is based on distance similarity; the other one is based on pattern similarity. 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.

According to Wang et al, 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. Early work on subspace clustering mainly employed distance based similarity. According to Agrawal et al, CLIQUE [4] may be the first known subspace clustering algorithm. It works in a level-wise manner, by using an Apriori style approach. According to Jones et al, DOC is a medoid-based subspace clustering algorithm, which is better than Clique in scalability and clustering quality. It selects a number of good candidate medoids and explores the clusters around them. According to Goil et al, Mafia is another extension of Clique, it uses an adaptive grid based on the distribution of data to improve efficiency and cluster quality. Yang et al. also suggested that bi-clustering is a generalized case of traditional subspace clustering (i.e., the CLIQUE algorithm).

Cheng and Church introduced the bi-cluster model [3] based on mean squared reside scores with a threshold. Yang et al. expanded on the work of Cheng and Church by introducing the concept of α-occupancy, which allows missing values in a bi-cluster up to a threshold. The residue score of a missing value is then defined to be zero. The bi-clustering algorithm [3] and the δ-clusters algorithm [5] provide only an approximate answer. The projected clustering [2] [3], still a kind of subspace clustering, in that it allows each cluster to have only a subset of relevant dimensions. Though distance-based clustering methods have been widely used, they have an obvious drawback, which inspires the research on pattern-based clustering.

III. PATTERN BASED CLUSTERING

Pattern-based clustering algorithms determine clusters based on the similarities of the patterns among objects across the relevant dimensions, instead of the absolute distance values among objects. The *p*-Cluster model is a generalization of subspace clustering. Basically, *p*-Cluster algorithm provides the complete answer; they will not miss any qualified subspace clusters. Many methods based on the *p*-Cluster model have been proposed. Most of the methods, e.g., *p*-Clustering [5], MaPle [6] and *z*-Cluster [7] are based on the calculation of MDS (maximum dimension sets).

To illustrate pattern-based clustering, we give an example in Fig. 2. Fig. 2(a) is a dataset consists of five objects with five attributes. Fig. 2(b) shows the values of the objects in full space (five attributes), where no obvious pattern is visible. However, if we just select attributes { *a*, *b*, *d*, *e* } as in Fig. 2(c) for objects { 2, 3, 5 }, we can observe the following pattern: for all the three objects, from attribute '*a*' to attributes '*b*'; '*d*'

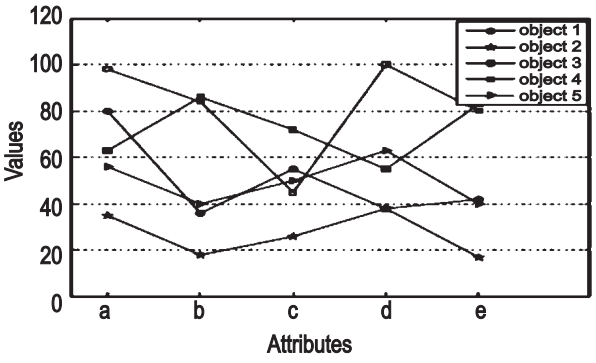

Fig. 2 (a) Data in full space

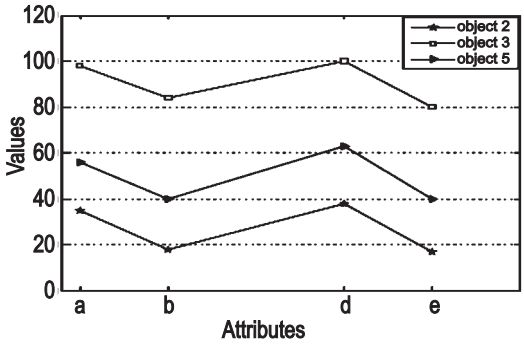

Fig. 2 (b) Pattern in subspace

Fig. 2 An example of Pattern based cluster

and '*e*', the values first go down, and then up and finally down. We can assign these three objects into the same subspace cluster as they show similar pattern. Likewise, similar patterns may exist with other objects in other subspaces.

TABLE 1. The synthetic data set

Attributes / Objects	*a*	*b*	*c*	*d*	*e*
1	80	36	55	38	42
2	35	18	25	38	17
3	98	84	45	100	80
4	63	86	72	55	83
5	56	40	50	63	40

A. p-Clustering

Wang et al. (2002), noticed several limitations of the δ-bi-cluster model for bi-clustering, and proposed a new model, *p*-Cluster to capture not only the closeness of objects, but also the similarity of the patterns exhibited by the objects. 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 *p*-Score of the 2×2 matrix is:

$$p\,\text{Score}\left(\begin{bmatrix} d_{xa} & d_{xb} \\ d_{ya} & d_{yb} \end{bmatrix}\right) = |(d_{xa} - d_{xb}) - d_{ya} - d_{yb})| \quad ..(1)$$

Again, if *p*-Score of the 2×2 matrix $\leq \delta$ for some $\delta \geq 0$ is said to form $\delta - p$ Cluster. Where as, in a bi-cluster model a sub matrix of a δ-bicluster is not necessarily a δ-bicluster. However one important property of *p*-Cluster is anti – monotonicity which says that if (*O*, *T*) be a $\delta - p$ Cluster then any of its sub matrix, (*O′ T′*) is also a $\delta - p$ Cluster. Hence, from the definition we can infer that *p*-Cluster is symmetric. However, since a *p*-Cluster 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. Also, the *p*-Cluster 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.

B. z-Clustering

The cluster search problem is in general NP-hard [1], and the subspace clustering problem is no exception [3][5][9]. To cope with this computational challenge, the *z*-Cluster algorithm [7] exploits a compact data structure called zero-suppressed binary decision diagrams (ZBDDs) [8] to implicitly represent and manipulate massive data. 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 *z*-Cluster algorithm exploits this property of ZBDDs and

TABLE 2: Comparison of Subspace and Pattern Based Clustering

Parameters	Subspace Clustering	Pattern Based Clustering		
		p - cluster	*z* - cluster	MaPle
Distance Measure	Use certain distance measures to find the certain similarity among clusters (but in case of high dimensional) data detecting meaningful cluster becomes difficult)	Based on similarity of pattern along with some threshold function	Same as *p* - cluster but uses the zero - suppressed pecision diagrams (ZBDDS) data structure	For each subset of attributes, it finds the maximal subsects of objects such that it is a opcluster
Cluster types	It avoids cluster overlapping problems as multiple clusters re found simultaneously	Detects multiple clusters that satisfy the user specified δ threshold	Finds all subspace cluster that statisfy specific input conditions without exhaustive enumeration	It prones non-maximal clusters
Time consumes	Finds cluster one by one hence discovery of one cluster constructs the other. This problem is time consuming	If finds clusters using Maximum Dimension Sets (MDS) for every two attributes which is time consuming	Finds clusters without much exhaustive enumerative search. But it is time consuming as they provide complete answer	Uses new pruning technique for computing and proning MDS that speeds up the mixing. Hence, it consumes less time.
Cluster structure	Determines homogeneous clusters	It discovers shifting and scaling patterns which gives rise to pattern based clusters		
Clusters	Prone to outliers	Resilent to outliers	Resilient to outliers	Resilient to outliers
Efficiency	Exponents search space of arbitrary subspaces pose challenges for an efficient computation. It is NP-haro problem	Quite efficient, but it performs a complex duplicating process at each node to create a post fix tree (using MDS) because of which the complexity seems to the no. of conditions which reduces the efficiency.	It uses ZBDDS in its steps which makes it efficient	As it guarantee both completeness and the non-redunancy of the search (every maximal *p* - cluster is found) which makes it efficient.
Antimorocity property	Violates this property. It says that "the sub matrix of δ - bi - cluster is not necessarily a δ - bi - cluster". This creates difficulty in designing efficient algorithm.	Follows the property of "Antimonocity" i.e sub matrix of any δp cluster is a δp cluster. (Hence it is symmertic in nature)		
Result pattern	Provides an approximate answer	Provides complete answer as they do not miss any qualified cluster	Provides an complete answer	Provides an complete answer

can find all the subspace clusters that satisfy specific input conditions without exhaustive enumeration.

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.

C. MaPle

MaPle enumerates all the maximal *p*-Clusters systematically. It guarantees both the completeness and the non-redundancy of the search, i.e., every maximal *p*-Cluster 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-*p*-Cluster. If (*R*, *D*) is not a sub-cluster of another *p*-Cluster (*R'*, *D*) such that *R*⊆*R'*, then (*R*, *D*) is a maximal δ- *p*-Cluster. 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 *p*-Clusters. It detects the attributes and objects that can be used to assemble a larger *p*-Cluster from the current *p*-Cluster. 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 *p*-Cluster having been found before, then the recursive searches rooted at the current node are pruned, since it cannot lead to a maximal *p*-Cluster.

IV. COMPARISON AMONG SUBSPACE & PATTERN BASED CLUSTERING

Many techniques have been proposed to find subspace clusters. Some well-known subspace clustering algorithms are based on the main categories of approximate answers and complete answers. Cheng and Church [3] proposed the bi-cluster model which captures the coherence of the genes and conditions in a sub matrix of a DNA microarray. Next, based on the same bi-cluster model, Yang et al [5] proposed a move-based algorithm, δ-cluster, to improve the performance of the bi-clustering

Compared with conventional clustering methods, pattern-based clustering is a more general model and has obvious advantages, which makes pattern-based clustering approaches gain much popularity in many application domains.

We have chalked out a clear comparison between subspace clustering algorithm and various pattern based clustering algorithms (Table II) like*p*-Cluster, *z*-Cluster and MaPle on the basis of certain parameters like distance measure, cluster type, time consumed, cluster structure, outliers, efficiency, anti-monotonicity property and result pattern.

V. CONCLUSION AND FUTURE WORK

Subspace clustering aims at finding multiple clusters embedded in subspaces of a high dimensional dataset based on some similarity measure. However, pattern based clustering helps in determining clusters based on the similarities of the patterns among objects across the relevant dimensions, instead of the absolute distance values among objects. Out of the two above algorithms it has been found that pattern based clustering is highly preferred.

Out of the entire pattern based clustering algorithms, *p*-Cluster 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 *p*-Clusters. 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 *p*-Cluster model with any soft computing technique.

REFERENCES

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

[2] Aggarwal C.C, Procopiuc C, Wolf J, Yu P S, Park J S., 1999 "Fast algorithms for projected clustering". *In Proc. of SIGMOD, Philadelphia, USA*, 1999, pp.61-72.

[3] Cheng Y, Church G., 2000 "Biclustering of expression data". *In Proc. of 8thInternational Conference on Intelligent System for Molecular Biology*, pp.93 {103}.

[4] Agrawal R, Gehrke J, Gunopulos D, Raghavan P, 1998 "Automatic subspace clustering of high dimensional data for data mining applications." *In Proc. of SIGMOD.*

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

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

[7] Pei, J., Zhang, X., Cho, M.,Wang, H., Yu, P.S., 2003 "Maple: a fast algorithm for maximal pattern-based clustering". *In the Proceedings of the 3rd IEEE International Conference on Data Mining, pp 259– 266.*

[8] 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*

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