ISSN:2231-0711
www.ijcset.net

The Data Mining Triclustering algorithm for mining Real Valued Datasets -A Review

Preshita S. Mahiskar[#1], Prof. A. W. Bhade[*2] and Dr. P. N. Chatur[#3]

[1, 3] *Department Computer Science and Engineering,*
[2] *Department of Information Technology,*
Government College of Engineering, Amravati, Maharashtra, India - 444604.

[1]preshita1808@gmail.com
[2]bhade.archana@gcoea.ac.in
[3]chatur.prashant@gcoea.ac.in

Abstract - Cluster analysis has been widely used in several disciplines, such as statistics, software engineering, biology, psychology and other social sciences, in order to identify natural groups in large amounts of data. These data sets are constantly becoming larger, and their dimensionality prevents easy analysis and validation of the results. The subspace pattern mining has been tailored to microarray data clustering to find biclusters and triclusters. We focus on deterministic clustering algorithm: Triclusters, which can mine arbitrarily positioned and overlapping biclusters/triclusters. Depending on different parameter values, they can mine different types of clusters, including those with constant or similar row/column values, as well as scaling and shifting expression patterns. We also give a useful set of metrics to evaluate the clustering quality, and show their effectiveness on real data.

Keywords- Data mining, data analysis, clustering, biclustering, Triclustering .

I. INTRODUCTION

Data mining is an iterative process within which progress is defined by discovery, through either automatic or manual methods. It is most useful in an exploratory analysis scenario in which there are no predetermined notions about what will constitute an "interesting" outcome [1]. It is also the search for new, valuable, and nontrivial information in large volumes of data. It is a cooperative effort of humans and computers. Best results are achieved by balancing the knowledge of human experts in describing problems and goals with the search capabilities of computers.

The process of grouping similar objects in the given dataset is known as clustering. A large variety of clustering algorithms have been proposed to find clusters in the given dataset. Not many real-life datasets are available for testing the proposed algorithms. Moreover the existing datasets do not have actual clustering result [1]. This leads to the idea of generating benchmarking datasets with high dimensionality and noise, which can evaluate clustering algorithms on various aspects like scalability, accuracy and robustness to noise.

Traditional clustering algorithms try to find clusters in all dimensions of the dataset [2]. When the dimensionality of the dataset increases, some dimensions could be irrelevant for few data points. There could be clusters which are spread in subset of dimensions of the dataset; these clusters may not be visible when seen in all the dimensions of the dataset. A number of subspace clustering algorithms are proposed to find such clusters. We propose methods to generate datasets which are useful for the subspace clustering algorithms.

Conventional clustering algorithms group similar data points together along one dimension of a data table. Bi-clustering simultaneously clusters both dimensions of a data table. 3-clustering goes one step further and aims to concurrently cluster two data tables that share a common set of row labels, but whose column labels are distinct [4]. The concept of Triclusters has been investigated recently in the context of two relational datasets that share labels along one of the dimensions.

II. THE CONCEPT OF CLUSTERING

Clustering is the process of grouping a set of objects into classes with maximum intra-class similarity and minimum inter-class similarity. This similarity criterion is based upon the entire set of attributes. For example two objects will be considered similar to each other if they exhibit strongly co-related values over the entire set of attributes [3]. This method restricts us to finding global patterns only and leaves out any chance of finding local patterns where two objects may be similar to each other based upon only a subset of attributes. This subset similarity criterion has been named as biclustering [4,8], co-clustering [2] or block clustering [5].

A. Biclustering

Due to recent technological advances in such areas as IT and biomedicine, the researchers face ever-increasing

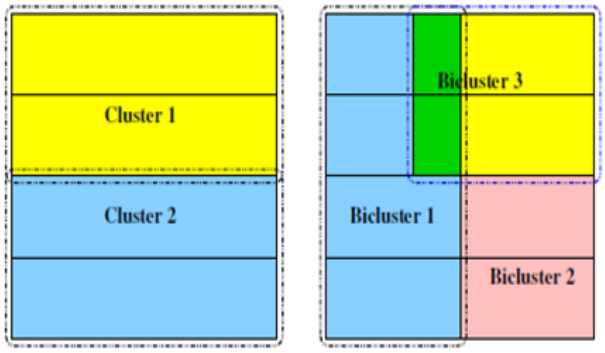
challenges in extracting relevant information from the enormous volumes of available data [1]. The so-called data avalanche is created by the fact that there is no concise set of parameters that can fully describe a state of real-world complex systems studied nowadays by biologists, ecologists, sociologists, economists, etc.

Biclustering consists in simultaneous partitioning of the set of samples and the set of their attributes (features) into subsets (classes). Samples and features classified together are supposed to have a high relevance to each other. It refers to a class of clustering algorithms that perform row-column clustering simultaneously [5]. The detected bicluster pattern contains a subset of whole rows and a subset of whole columns. The concept of biclustering has already been widely applied in other research fields besides bioinformatics since it was first proposed. For example, in document clustering, biclustering algorithms are applied to cluster documents and words simultaneously.

The great majority of the algorithms perform simultaneous clustering on both dimensions of the data matrix in order to find biclusters of the previous four classes. However, we also analyzed two-way clustering approaches that use one-way clustering to produce clusters on both dimensions of the data matrix separately. These one-dimension results are then combined to produce subgroups of rows and columns whose properties allow us to consider the final result as biclustering [6]. When this is the case, the quality of the bicluster is not directly evaluated. One-way clustering metrics are used to evaluate the quality of the clustering performed on each of the two dimensions separately and are then combined, in some way, to compute a measure of the quality of the resulting biclustering. The type of biclusters produced by two-way clustering algorithms depends, then, on the distance or similarity measure used by the one-way clustering algorithms. Biclustering can be applied whenever the data to analyze has the form of a real-valued matrix *A*, where the set of values *aij* represent the relation between its rows *i* and its columns *j*.

Thus, the biclusters define as: A Bicluster B is a pair consisting of a subset of rows r and a subset of columns c taken from a dataset D with row-set R and column-set C. We use the term B = {r, c} to denote a Bicluster and use s_B to denote the standard deviation of the cell values in bicluster B; here r ⊆ R and c ⊆ C.

Biclustering finds local patterns where a subset of objects might be similar to each other based on only a subset of attributes. The comparison between biclusters and clusters is illustrated in Figure 1(a) demonstrates the concept of clusters and 1(b) demonstrates the biclusters in input data matrix. Note that biclusters can cover just part of rows or columns and may overlap with each other as shown in figure 1(b). Biclustering process can generate a wide variety of object groups that capture all the significant correlation information present in a data set. Biclustering has become very popular in discovering patterns from gene microarray experiments data [7].



(a) Clusters of Objects (b) Biclusters of Objects.
Figure 1: Illustration of Clusters and Biclusters

For example, in Figure2 (a), if points p2, p4 and p5 are similar in the whole space, they can be clustered together. Biclustering, however, does not have such a strict requirement. If some points are similar in several dimensions (a subspace), they will be clustered together in that subspace. For example, in Figure 2 (b), if points p2, p4 and p5 are similar in the subspace composed of dimensions d2, d3 and d5, they form a bicluster. Biclustering is very useful, especially for clustering in a high dimensional space where often only some dimensions are meaningful for some subset of points.

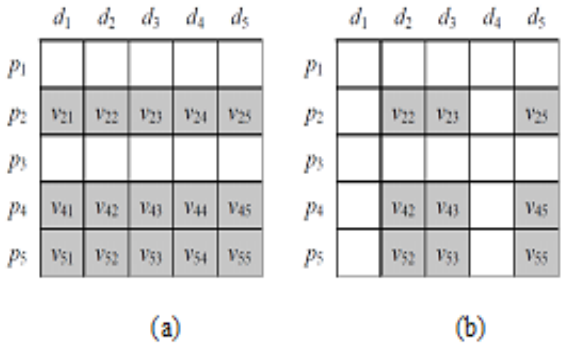


Figure 2. Difference between (a) Fullspace Clustering and (b) Biclustering

The complexity of computing bi-clusters depends much on the way the bi-clustering problem is defined and how its quality is measured. In most interesting formulations the bi-clustering problem is NP-hard problem. The computational complexity for discovering k biclusters is of the order of O (mn× (m+ n) ×k), where m and n are the number of sample and conditions, respectively.

B. Triclustering

The idea of Triclusters is being increasingly applied in data mining situations where two distinct datasets need to be mined simultaneously [8]. The algorithm can handle situations involving:

- Datasets in which a few data objects may be present in only one dataset and not in both datasets,
- The two datasets may have different numbers of objects and/or attributes, and
- The cell-value distributions in two datasets may be different.

Algorithms for generating triclusters whose cell-values demonstrate simple well known statistical properties, such as upper bounds on standard deviations, are needed for many applications. In our formulation the cell-values of each selected triclusters, formed by two independent biclusters, are such that the standard deviations in each bicluster obeys an upper bound and the sets of objects in the two biclusters overlap to the maximum possible extent [9]. Each component bicluster (sub-matrix in a dataset) obeys the constraint of a low-variance distribution of cell values (with specified upper-bound on the variance) and the intersection set of the rows of two biclusters (sub matrices) is the largest possible while permitting each bicluster to contain some non-shared rows.

The algorithm accommodates data distribution disparities between datasets and focuses on satisfying simple to understand standard deviation bounds on biclusters from real valued datasets. Even though there are many biclustering algorithms that could search for maximal sized low-variance biclusters within individual datasets [10]. A Low-Variance 3-Cluster may be formed by using non-maximal local biclusters which contain only subsets of attributes forming the locally optimal biclusters, and also consequently, larger number of rows, increasing chances of larger sets of shared rows. Our search algorithm for triclusters traverses the concept lattices implicit in the two data tables, enumerating the promising parts, and pruning away the non-promising parts, using information from both datasets to make the pruning decisions.

The key features of approach include:

1) Mine only the maximal triclusters satisfying certain homogeneity criteria.

2) The clusters can be arbitrarily positioned anywhere in the input data matrix and they can have arbitrary overlapping regions.

3) We use a flexible definition of a cluster which can mine several types of triclusters, such as triclusters having identical or approximately identical values for all dimensions or a subset of the dimensions, and triclusters that exhibit a scaling or shifting expression values

4) Triclusters is a deterministic and complete algorithm, which utilizes the inherent unbalanced property in data, for efficient mining [9].

5) Triclusters can optionally merge/delete triclusters that have large overlaps, and can also automatically relax the similarity criteria. It can thus tolerate some noise in the dataset, and lets the user focus on the most important clusters.

6) We present a useful set of metrics to evaluate the clustering quality.

III. CONCLUSION

Our algorithm employs a number of pruning strategies to make the algorithms more efficient. Generally our algorithm conducts heuristic beam search in two prefix-tree defined search spaces which are the same as the lattices of all possible biclusters in the two datasets. This pruning strategy will not influence the results, because for each 3-Cluster or candidate bicluster we maintain the interestingness measure and always retain the largest k candidates. We also employ another pruning strategy to help obtain distinct 3-clusters from the datasets. . Low-Variance 3 Clusters open the door for new direction of cross-domain research and knowledge discovery.

REFERENCES

[1] Bouguettaya A. "On Line Clustering", *IEEE Transaction on Knowledge and Data Engineering* Volume 8, No. 2 1996.

[2] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. “Automatic subspace clustering of high dimensional data for data mining applications”. In *ACM SIGMOD Conference on Management of Data*, 1998.

[3] Huang Z. “Clustering Large Data Sets with Mixed Numeric and Categorical Values” *Proceedings of the 1st Pacific-Asia Conference on Knowledge Discovery and Data Mining, (PAKDD)* Singapore 1997 21–34p.

[4] M. J. Zaki and C.-J. Hsiao. “Efficient algorithms for mining closed itemsets and their lattice structure”. *IEEE Transactions on Knowledge and Data Engineering*, 17(4):462-478, 2005.

[5] Cheng Y. and Church G. M. “Biclustering of expression data”. *8th International Conference on Intelligent Systems for Molecular Biology* 2000 93-103p

[6] Zhao L. and Zaki M. J., “TRICLUSTER: an effective algorithm for mining coherent clusters in 3D microarray data” *International conference on Management of data*, 2005

[7] Ahmad W. and Khokhar A. “An Architecture for Privacy Preserving Collaborative Filtering on Web Portals” *Third International Symposium on Information Assurance and Security* 2007

[8] Alqadah F. and Bhatnagar R., “An effective algorithm for mining 3-clusters in vertically partitioned data” *17th ACM conference on Information and knowledge management*, 2008

[9] Li A. and Tuck D. “An Effective Tri-Clustering Algorithm Combining Expression Data with Gene Regulation Information” *Gene Regulation and System Biology* 3:49-64, 2009.

[10]Mojarad M. R.et al. “A Survey on Biological Data Analysis by Biclustering” *International Conference on Educational and Information Technology* 2010