ISSN: 2229 – 6956(ONLINE)
DOI: 10.21917/ijsc.2011.0019

DENSITY CONSCIOUS SUBSPACE CLUSTERING USING ITL DATA STRUCTURE

C. Palanisamy[1] and S. Selvan[2]
[1]*Department of Information Technology, Bannari Amman Institute of Technology, Tamil Nadu, India*
E-mail: cp_samy@yahoo.com
[2]*Alpha Engineering College, Chennai, Tamil Nadu, India*
E-mail: drselvan@ieee.org

Abstract
Most of the subspace clustering algorithms uses monotonicity property to generate higher dimensional subspaces. But this property is not applicable here since different subspace cardinalities have varying densities ie., if a k-dimensional unit is dense, any (k-1) dimensional projection of this unit may not be dense. So in DENCOS a mechanism to compute upper bounds of region densities to constrain the search of dense regions is devised, where the regions whose density upper bounds are lower than the density thresholds will be pruned away in identifying the dense regions. They compute the region density upper bounds by utilizing a data structure, DFP-tree to store the summarized information of the dense regions. DFP-Tree employs FP-Growth algorithm and builds an FP-Tree based on the prefix tree concept and uses it during the entire subspace identification process. This method performs repeated horizontal traversals of the data to generate relevant subspaces which is time consuming. To reduce the time complexity, we employ ITL data structure to build Density Conscious ITL (DITL) tree to be used in the entire subspace identification process. ITL reduces the cost by scanning the database only once, by significantly reducing the horizontal traversals of the database. The algorithm is evaluated through experiments on a collection of benchmark data sets datasets. Experimental results have shown favourable performance compared with other popular clustering algorithms.

Keywords:
Subspace Clustering, ITL Tree, Recall, Precision

1. INTRODUCTION

A critical problem, called the density divergence problem is ignored by most of the subspace clustering algorithms. Due to high sparsity of higher dimensional data, different subspace cardinalities require varying region densities as thresholds to qualify. Clusters in higher dimensional subspaces require lower density thresholds and vice versa. Otherwise we may lose true clusters in the dataset and the trade-off between recall and precision will be certainly faced. To get variable density thresholds DENCOS [1] have considered the clusters in a subspace as the regions which have relatively high densities as compared to the average region density in the subspace. To identify such clusters, they introduce a novel density parameter α for users to specify their expected relative rate of the densities of the dense regions and the average region density in a subspace. The higher dimensional data is very sparse, cluster densities vary in different subspace cardinalities. This is referred to as the density divergence problem. This implies that extracting clusters in higher subspaces should be with a lower density requirement and vice-versa; otherwise true clusters may be lost. The requirement of varying density thresholds for clusters in different subspace cardinalities makes subspace clustering very challenging in simultaneously achieving high precision and recall for clusters in different subspace cardinalities [1]. For a cluster, recall is defined as the percentage of the data points in a true cluster that are identified in this cluster. Precision is defined as the percentage of the data points in this cluster that really belong to the true cluster [13].

2. PROBLEM STATEMENT

We adopt the grid-based approach [2], [3] to discover subspace clusters, where the data space is partitioned into a number of non-overlapping rectangular units by dividing each attribute into optimal number of equal-length intervals. Dividing each dimension into optimal number of intervals has a profound effect on the clustering accuracy. The density of each unit is calculated as the number of data points contained in it. For identifying relevant dense units, we use different density thresholds for different subspace cardinalities.

In our subspace clustering model, we adopt the method proposed in DENCOS [1] to calculate different density threshold of different subspace cardinalities. To discover clusters, they introduce a density parameter α specified by the user. Let τ_k denote the density threshold for the subspace cardinality k, and let N be the total number of data points. Then, the density threshold τ_k is defined as [1],

$$\tau_k = \alpha(N/\delta_k) \quad (1)$$

When the data are uniformly distributed in a k-dimensional subspace, the number of data points in each of the interval δ_k k-dimensional units in this subspace will be N/δ_k, i.e. the average unit density. In this case, no clusters are discovered because each point in this space has almost the same density. On the other hand if the data has more compacted clusters, the units within clusters will be much denser and would have a larger count value than the average density.

Given the unit strength factor α and the maximal cardinality k_{max}, the subspace clustering problem can be stated as follows: Find the clusters in as a maximal set of connected dense k-dimensional units whose count values exceed the density threshold τ_k.

3. ITL DATA STRUCTURE

Different data representation schemes proposed for association rule mining is broadly classified as horizontal data layout, vertical data layout, and a combination of the two [4], [5], [6], [7]. Most candidate generation and test algorithms [4] use the horizontal data layout and most pattern-growth algorithms like FP-Growth [5] and H-Mine [6] use a combination of vertical and horizontal data layouts. HMine scans the database twice. FP-Growth performs repeated horizontal traversals of the database while generating frequent

itemsets. If these costs are reduced, the mining process will be improved further. Item-Trans Link (ITL) proposed by Raj et al [7] reduces the cost by scanning the database only once and significantly reducing the horizontal traversals of the database and keeping the links between samples unchanged during the mining process. A short description of the ITL Data structure is included for clarity. It consists of an item table and the databases linked to it, the TransLinks. ItemTable contains all the items and the support of each item. It also has a link to the first occurrence of each item in the databases of TransLink described below. TransLink represents the items of every instance for all the instances in the database. The items of a sample are sorted. For each item in a database, it contains a link to the next occurrence of that item in another sample so that the counting can be done quickly.

ITL data structure is illustrated with the sample data in Fig.1.

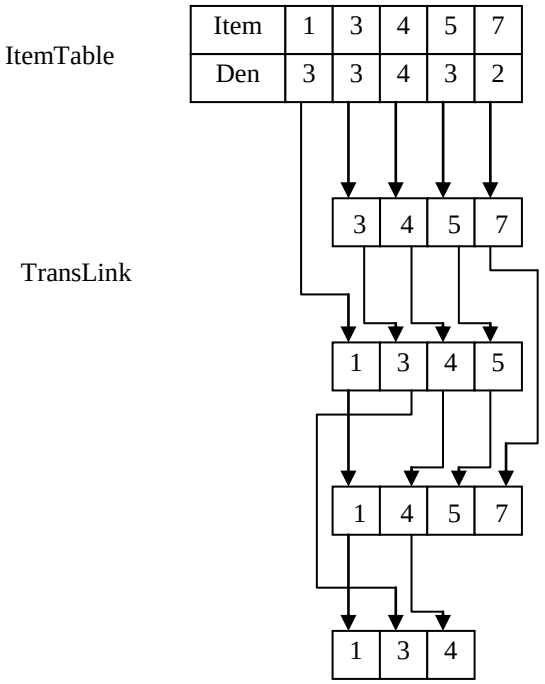

Fig.1. Illustration of an ITL Data structure

4. FRAMEWORK FOR CLUSTERING

The problem here considers different density thresholds in different subspace cardinalities. So the monotonicity property used to find the dense units by most of the subspace clustering algorithms is not applicable. That is, if a k-dimensional unit satisfies the threshold τ_k, any (k − 1) dimensional projection of this unit may not satisfy the threshold τ_k−1. Without the monotonicity property, the Apriori-like candidate generate-and-test scheme adopted in most previous works cannot be adopted for discovering the dense units.

To solve this, our problem is modeled as a problem similar to frequent itemset mining in association rule mining. The intervals in all dimensions are considered as a set of unique items in frequent itemset mining problem. Any k- dimensional unit is regarded as a k-itemset, i.e., an itemset of cardinality k. Hence identifying the dense units satisfying the density thresholds in subspace clustering is similar to mining the frequent itemsets satisfying the minimum support in frequent itemset mining.

4.1 PRUNING

The brute-force generation of the all possible candidate units from each path may result in less dense candidate units. So we should find out the paths which have the possibilities to contain dense units, thus resulting in a smaller set of candidate units for the dense unit discovery. An effective scheme to identify the dense units from the ITL data structure is described. For the nodes with node counts satisfying the thresholds for some set of subspace cardinalities, we will take their prefix paths to generate the dense units of their satisfied subspace cardinalities. Let $k(n_i)$ denote the subspace cardinality, then the relation $\tau_{k(ni)} \leq n_i.count < \tau_{k(ni)-1}$ must hold since the thresholds are decreased while identifying higher dimensional subspaces. If so, the prefix path is retained and used to generate candidate dense units. Otherwise the prefix path of the node n_i is discarded and proceed to the next node.

The steps involved in finding dense units are summarized as follows:

1. Read the input dataset with k dimensions and d instances.
2. Using the number of dimensions, k and user defined parameter α as inputs; calculate the different density thresholds τ_k for various subspace cardinalities using equation 1.
3. Represent the k-dimensional unit by the set of k 1-dimensional units, corresponding to the k intervals.
4. Find the count value of k 1-dimensional units by directly counting the occurrences of the set of k, 1- dimensional units in the transformed dataset.
5. The 1-dimensional dense units are directly discovered from the header table by identifying the 1-dimensional units with the stored total unit counts exceeding τ_1.
6. To identify higher dimensional dense units represent the d instances in different intervals with their appropriate density counts using. Item-Trans Link (ITL) data structure.
7. We compute the region density upper bounds by utilizing the ITL data structure, where we store the summarized information of the dense regions.
8. Generate candidate subspaces by applying pruning described in section 4.1. Identify relevant dense units using the threshold on the candidate subspaces thus found.
9. Repeat the procedure until all relevant maximal dimensional subspaces are found.

Once the dense units are discovered, the procedure proposed in [2] is used to group the connected dense units into clusters.

5. EXPERIMENTAL EVALUATION

To evaluate the algorithm we have used two real datasets with higher data dimensionalities namely Corel Image Features dataset and Letter Recognition dataset in UCI machine learning repository [8]. To compare the time complexity and test the scalability of the algorithm we use artificially generated synthetic datasets.

5.1 SYNTHETIC DATA SETS

Several synthetic datasets shown in Table.1 are used to assess the qualitative performance of our approach. Synthetic datasets are generated by using the data generation method utilized in CLIQUE [2]. In a dataset, the clusters are generated by specifying the following terms: (1) the dimensions of the subspace in which the cluster is embedded, and (2) for each attribute A_j of the subspace, the range $[A_{j.start}, A_{j.end}]$ of A_j the cluster is embedded in. Then, we generate the dataset such that the average densities of the data points inside the clusters are much larger than their surrounding regions. The data points assigned into a cluster are generated with uniform distribution. For a data point p assigned to a cluster, its attribute values are assigned as follows. For each attribute A_j of the subspace in which the cluster is embedded, we randomly determine the value of attribute A_j of p from the range $[A_{j.start}, A_j.end]$. For the remaining attributes, the value is drawn randomly from the entire range of the attribute. The number of data points is set to be equal to 25,000.

Table.1. Specifications of synthetic datasets

Data set	Number of Dimensions	Dimension of subspaces
Dataset 1	5	4
Dataset 2	10	4,5
Dataset 3	25	5,5
Dataset 4	40	3,3,5
Dataset 5	55	5,5,7
Dataset 6	85	4,9,16,31

5.1.1 Algorithm Accuracy on Synthetic Datasets

In this subsection, we utilize the synthetic datasets shown in Table.1 to compare the clustering results of our approach with the ones of CLIQUE [2] and SUBCLU [9]. To evaluate the clustering results, we take the dense regions generated by the data generator as the known clusters, and evaluate the quality of these known clusters discovered in the clustering algorithm by two matrices, the precision and recall. For a cluster discovered, "precision" is defined as the percentage of the data points in this cluster that really belong to the known cluster. "Recall" is defined as the percentage of the data points in a known cluster that are identified in this cluster. For each dataset, a number of α values are used to find the best clustering result, and the results are reported for α value = 15. In all datasets, CLIQUE and SUBCLU are studied with a broad range of parameter settings and the best results are taken for comparison.

We have executed CLIQUE [2], SUBCLU [9], DENCOS [1] and our approach DITL (Density Conscious ITL) on the first three datasets listed in Table.1. The results reveal that they all accurately discover the clusters with both precision and recall close to unity. In Table.2, we show the effect of changing the number of dimensions of subspace clusters on precision for various algorithms using dataset 6 of Table.1. The dimensionality of the dataset is equal to 85 and number of dimensions of the subspaces in which clusters exist are 4,9,16 and 31. By analysing the precision values in Table.2, it is observed that, both DENCOS and DITL performs better in all cases compared with other algorithms. Although the performance of DITL is similar to that of DENCOS, the execution time is reduced as demonstrated in Table.4. CLIQUE(4) indicates the specific parameters set for CLIQUE to discover clusters in 4 dimensional subspaces. Similar assumption is applicable for other CLIQUE and SUBCLU algorithms listed in the first column of Table.2. CLIQUE and SUBCLU achieve high precision values when the appropriate parameters are supplied and fail in cases where appropriate parameters are not adequate.

Table.2. Effect of changing the number of dimensions of subspace clusters on precision for various algorithms

Algorithm	Number of Dimensions of Subspace Cluster			
	4	9	16	31
D ITL	100%	100%	100%	100%
DENCOS	100%	100%	100%	100%
CLIQUE (4)	96.07%	42%	19.14%	25.94%
CLIQUE (9)	49.09%	97.24%	39.14%	49.33%
CLIQUE (16)	23.15%	56.07%	96.03%	49.33%
CLIQUE (31)	37.91%	98.09%	49.21%	84.55%
SUBCLU (4)	96.36%	49.09%	49.21%	24.55%
SUBCLU (9)	37.80%	94.09%	49.21%	34.55%
SUBCLU (16)	23.57	49.09%	89.21%	35.15%
SUBCLU (31)	37.42%	49.09%	49.21%	87.55%

In Table.3, we show the effect of changing the number of dimensions of subspace clusters on recall for various algorithms. This evaluation corresponds to dataset 6 listed in Table.1. The dimensionality of the dataset is equal to 85 and numbers of dimensions of the subspaces in which clusters exist are 4,9,16 and 31. From Table.3 it is observed that, the recall values of DENCOS are slightly higher than that of DITL. The trade off is that DITL executes faster than DENCOS as demonstrated in Table.4. CLIQUE [2] and SUBCLU [9] have difficulties in simultaneously discovering these clusters with high quality.

Table.3. Effect of changing the number of dimensions of subspace clusters on recall for various algorithms

Algorithm	Dimension of Subspace Cluster			
	4	9	16	31
D ITL	100%	100%	90%	89%
DENCOS	100%	100%	91%	92%
CLIQUE (4)	91.54%	34.5%	35.14%	49.33%
CLIQUE (9)	98%	100%	25.14%	25.14%

CLIQUE (16)	52.01%	61.54%	95.14%	25.14%
CLIQUE (31)	100%	100%	100%	98.73%
SUBCLU (4)	100%	50%	46.94%	47.73%
SUBCLU (9)	100%	100%	40.57%	37.73%
SUBCLU (16)	100%	100%	100%	45.73%
SUBCLU (31)	100%	100%	100%	77.33%

In CLIQUE (4), we use the threshold to find only 4 dimensional clusters, and so in finding 9, 16 and 31 dimensional clusters, we find that the algorithm fails, which is reflected in the low recall. The reason is that the high threshold for discovering the 4 dimensional clusters cannot identify the 9, 16, 31 dimensional low-density units. On the other hand, in CLIQUE (9) in discovering 9 dimensional clusters, the threshold which is set for discovering the 9-dimensional clusters with high quality may be too high as compared to the low density of the 16, 31-dimensional clusters, resulting in poor quality of the 16, 31-dimensional clusters. In SUBCLU [9], clusters are discovered by identifying the core objects, where a point is a core object if the number of data points in its ε-neighborhood is larger than m. The same thresholds ε and m are imposed to define core objects in all subspace cardinalities. As shown in SUBCLU [9] in discovering 9 dimensional clusters, the two thresholds ε and m are relaxed to identify core objects in higher subspaces because data are more sparsely populated in higher subspaces. However, the relaxed thresholds would make some data points between the two clusters be also identified as core objects, and these core objects will be linked together with the core objects in the two 4 dimensional clusters, resulting in only one cluster, resulting in low recall. Similar arguments hold for other dimensional subspace clusters of CLIQUE and SUBCLU.

5.1.2 Scalability with Dataset Size

We vary the size of the datasets to assess the scalability against the dataset size. We have generated synthetic datasets with number of data objects varying from 10,000 to 1,00,000 with their dimension fixed to be equal to 50. The performance of Density conscious ITL (DITL) is compared with CLIQUE [2], SUBCLU [9] and DENCOS [1] and the results are shown in Fig.2.

As the dataset size increases, the execution time of our approach DITL is almost maintained constant. This is because for different dataset size, the structures of the constructed ITL-tree are not changed much such that the execution time of mining the dense units from the ITL Tree do not increase much.

In the other case, we vary the dataset dimensionality from 10 to 100 and maintain the number of objects to be equal to 1,25,000. The performance of Density conscious ITL (DITL) is compared with CLIQUE [2], SUBCLU [9] and DENCOS [1] and the results are shown in Fig.3.

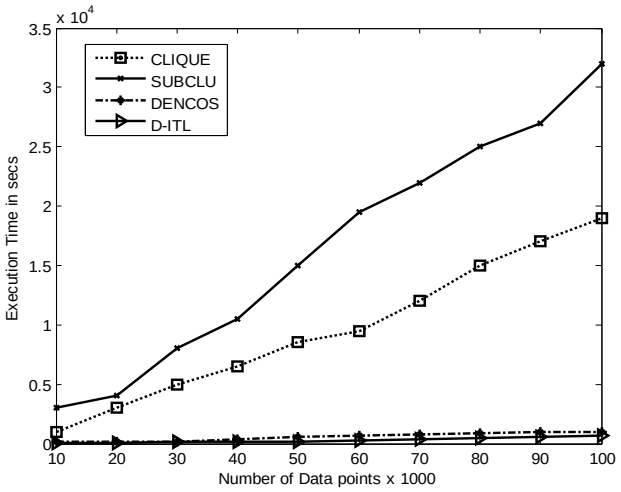

Fig.2. Effect of varying the number of data points on execution time for various algorithms

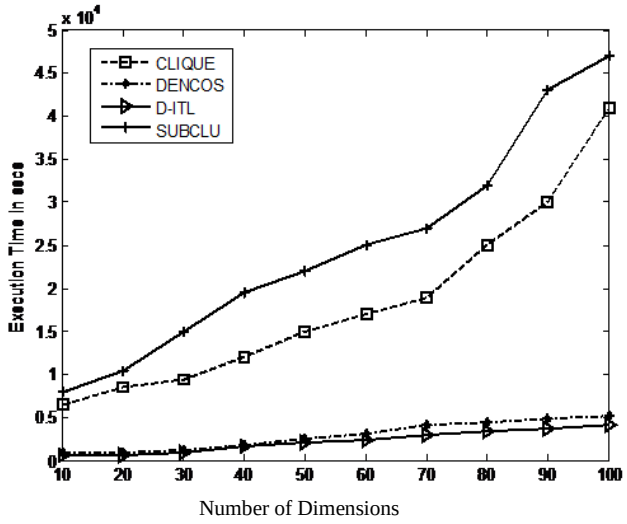

Fig.3. Effect of varying the number of dimensions on execution time for various algorithms

As the dataset size increases, the execution time of Density adaptive ITL is almost maintained constant. This is because for different dataset size, the structures of the constructed ITL-tree are not changed much such that the execution time of mining the dense units from the ITL Tree do not increase much.

The execution times of DENCOS and our approach DITL in identifying various dimensional subspace clusters are recorded for a dataset with number of objects d = 1,25,000. Then the dimensions are varied from 20 to 100 and the results are as shown in Table.4. It is observed that the execution time required by Density Conscious ITL (DITL) is very low when compared with DENCOS. This is because we carry forward only the relevant subspaces to generate higher dimensional relevant subspaces by applying an effective pruning strategy. We find the best paths which have the possibilities to contain dense units, thus resulting in a smaller set of candidate units for the dense unit discovery. The scheme uses the node counts as threshold. For the nodes with node counts satisfying the thresholds for some set of subspace cardinalities, we will take their prefix paths

to generate the dense units of their satisfied subspace cardinalities.

Table.4. Comparison of execution times on different dimensions of a dataset for various algorithms

	Execution Time in secs				
Dimensionality / Algorithm	20	40	60	80	100
DENCOS	2000	2600	3520	4200	5100
DITL	1200	1800	2300	3200	4180

5.2 REAL DATA SETS

Two real datasets with higher data dimensionalities are used to assess the effectiveness of our approach. These real datasets are (1) "Corel Image Features" dataset in UCI KDD archive [10], and (2) "Letter Recognition" dataset in UCI machine learning repository [8]. The "Corel Image Features" dataset contains image features (co-occurrence texture) extracted from a Corel image collection, and the "Letter Recognition" dataset contains the numerical attributes (statistical moments and edge counts) extracted from the stimulus images with English alphabets. These two real datasets are both with 16 data attributes.

5.2.1Density ratio

The quality of the clustering result is evaluated in terms of density ratio, abbreviated as DR. For a subspace S', the average density ratio, is defined as,

$$DR(S') = \frac{\text{average region density of the regions inside the clusters of S'}}{\text{average region density of the regions outside the clusters of S'}} \quad (2)$$

Higher DR(S') value means that the regions of higher densities can be better separated from the regions of lower densities indicating a better-quality.

For Corel Image Features dataset α is set to 2. The variation of average density ratio with different subspace cardinality for Corel Image Features dataset is shown in Fig.4. From Fig.4, it is observed that both DENCOS and DITL show almost same performance upto subspace cardinality 6 and there is a sudden variation from subspace cardinality 7, indicating good discrimination in higher subspace cardinalities.

The variation of average density ratio with different subspace cardinality for Letter Recognition dataset is shown in Fig.5. For the Letter Recognition dataset α is set to 3. From Fig.5, it is observed that our approach DITL out performs DENCOS in subspace cardinalities due to the inherent characteristics of the dataset. This indicates that the clusters discovered by our approach are real regions that are of high densities in the subspaces. It is also observed from the figure that, the clusters in the higher subspace cardinalities are well discriminated by our approach DITL.

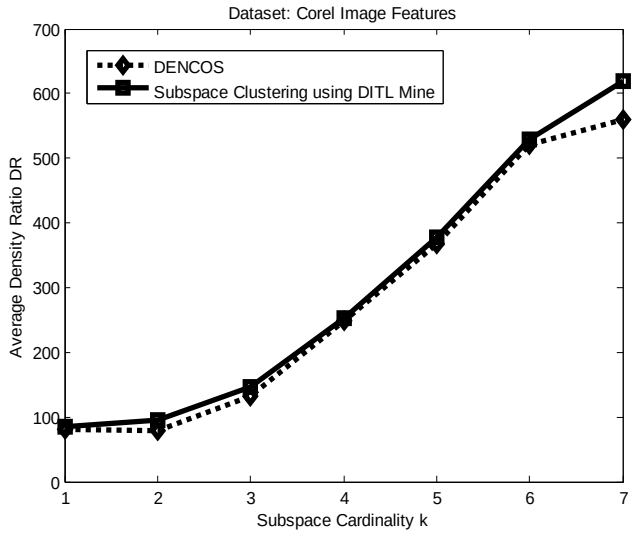

Fig.4. Subspace Cardinality Vs Average Density Ratio on Corel Image Features Dataset

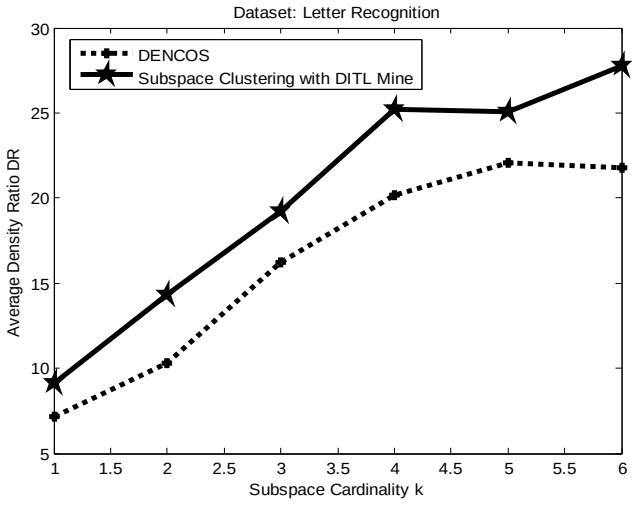

Fig.5. Subspace Cardinality Vs Average Density Ratio on Letter Recognition Dataset

5.3 APPLICATION TO GENE EXPRESSION DATA

The gene expression data appears as a matrix where the rows represent genes, and the columns represent samples [11]. The value of the i-th feature of a particular gene is the expression level of this gene in the i-th sample. Clustering the genes in subspaces may help to identify the genes whose expression levels are similar in a subset of samples, where co-expressed genes usually are functionally correlated. The performance of a clustering algorithm on the public yeast genome data set is evaluated by comparing its generated clusters with the original clusters. We employ performance measures namely; Subspace Clustering Error (SCE), the Coverage Index (CI) and the discrepancy in the number of clusters (DNC) [12]. The performances of various algorithms are shown in Table.5. The ITL tree based adaptive density algorithm has the best SCE, CI and DNC scores, which indicate that it has the best ability to detect consensus gene distribution patterns implied by its

detected subspaces and genome clusters without including excessive unnecessary clusters.

Table.5. Different performance measures using the public yeast genome data set for various algorithms

Algorithm	Subspace clustering error (SCE)	Coverage index (CI)	Discrepancy in the number of clusters (DNC)
ITL tree Based Approach	0.320	0.271	1
DENCOS	0.452	0.351	1
HARP	0.491	0.420	3
SAMBA	0.962	0.863	8
Cheng-Church	0.784	0.717	7

6. CONCLUSION

In this paper we have presented the density divergence problem to find true subspace clusters in different subspace cardinalities. This approach discovers true clusters efficiently in different subspace cardinalities. To efficiently discover dense units, a practicable way would be to store the complete information of the dense units in all subspace cardinalities into a compact structure such that the mining process can be directly performed in memory without repeated database scans. We proposed to construct a compact structure using ITL data structure from which the clustering is performed. To further improve the performance of the algorithm, effective pruning strategy is also suggested. We have demonstrated the performance of the proposed algorithm using a variety of real and synthetic data sets. The algorithm is also applied on genome dataset. The algorithm has shown reasonable performance improvement compared with other popular algorithms.

REFERENCES

[1] Chu Y., Huang J., Chuang K., Yang D. and Chen M, 2010, "Density Conscious Subspace Clustering for High-Dimensional Data", IEEE Transactions on Knowledge and Data Engineering, Vol.22, No.1, pp.16 – 30.

[2] Agrawal R., Johannes G., Dimitrios G. and Prabhakar R., 1998, "Automatic Subspace Clustering of High Dimensional Data for Data Mining applications", Proceedings of ACM SIGMOD International Conference on Management of Data, ACM Press, pp. 94-105.

[3] Glomba M. and Markowska-Kaczmar U., 2006, "IBUSCA: A Grid-based Bottom-up Subspace Clustering Algorithm", Sixth International Conference on Intelligent Systems Design and Applications, ISDA, Vol. 1, No. 16-18, pp.671-676.

[4] Agrawal R., Imielinski T. and Swami A.N. 1993, "Mining Association Rules between Sets of Items in Large Databases", SIGMOD, Vol. 22, No. 2, pp. 207-216.

[5] Han J. and Kamber M. 2000, "Data Mining: Concepts and Techniques", Morgan Kaufmann.

[6] Pei J., Han J., Lu H., Nishio S., Tang S. and Yang, D. 2001, "H-Mine: Hyper-Structure Mining of Frequent Patterns in Large Databases", Proc. of the 2001 IEEE ICDM, San Jose, California, pp. 441-448.

[7] Raj P.G. and Yudho G.S. 2002, "ITL-Mine: Mining Frequent Itemsets More Efficiently", FSKD, pp. 167-171.

[8] Asuncion A. and Newman D.J. 2007, "UCI Machine Learning Repository" - http://archive.ics.uci.edu/ml/, Irvine, CA: University of California, School of Information and Computer Science.

[9] Kailing K., Kriegel H.P., Kröger P. and Wanka S. 2003, "Ranking Interesting Subspaces for Clustering High Dimensional Data", Proceedings of 7th European Conference on Principles and Practice of Knowledge Discovery in Databases, pp. 241-252.

[10] Hettich S. and Bay S. 1999, "The UCI KDD archive" - http://kdd.ics.uci.edu.

[11] Jiang D., Tang C. and Zhang A. 2004, "Cluster analysis for gene expression data: A survey", IEEE Transactions on Knowledge and Data Engineering, Vol. 16, No. 11, pp. 1370-1386.

[12] Kriegel, Hans-Peter; Kröger, Peer; Zimek, Arthur, 2009, "Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering", ACM Transactions on Knowledge Discovery from Data, Vol. 3, No.1, pp.1–58.

[13] Houle, Michael E.; Kriegel, Hans-Peter; Kröger, Peer; Schubert, Erich; Zimek, Arthur, 2010, "Can Shared-Neighbor Distances Defeat the Curse of Dimensionality?", Proceedings of the 21th International Conference on Scientific and Statistical Database Management, Vol.6187, pp.482-500.