IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 15, Issue 3 (Nov. - Dec. 2013), PP 58-63
www.iosrjournals.org

Clustering Algorithm Based On Correlation Preserving Indexing

P.Sudhher[1], V.Kesav Kumar[2]
[1] *M.Tech, SMCE, Guntur, AP.*
[2]*Asso.prof., SMCE, Guntur, AP.*

Abstract: *Fast retrieval of the relevant information from the databases has always been a significant issue. Different techniques have been developed for this purpose; one of them is Data Clustering. In this paper Data Clustering is discussed along with the applications of Data Clustering and Correlation Preserving Indexing. We proposed a CPI (Correlation Preserving Indexing) algorithm and relate it to structural differences between the data sets.*

Keywords*: Data Clustering, Data Mining, Clustering techniques, Correlation Preserving Indexing.*

I. Introduction:

Cluster analysis is a convenient method for identifying homogenous groups of objects called clusters. Objects (or cases, observations) in a specific cluster share many characteristics, but are very dissimilar to objects not belonging to that cluster.

The most important task of clustering gave important attention to various branches such as machine learning and artificial intelligence. The clustering has the broad appeal and usefulness as basic steps in exploratory data analysis are an unsupervised learning method that constitutes a corner stone of an intelligent data analysis process. The analysis used for exploiting the various inters relationships among a collection of patterns which are divided and organized in homogeneous clusters. With a good clustering method the computers, automatically organized in a meaningful clustering hierarchy.

In many traditional approaches to machine learning, a target function is estimated using labeled data.

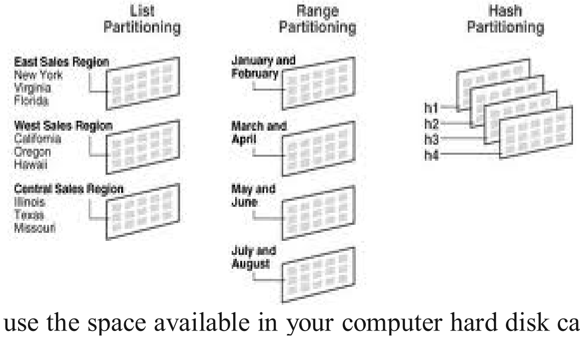
II. Existing System & Analysis :

Document clustering is one of the most crucial techniques to organize the documents in an unsupervised manner. It has received a lot of attention in recent years.

2.1 PARTITIONING MATHODS:

Partitioning methods are divided into two major subcategories, the centroid and the mediod algorithms. The centroid algorithms represent each cluster by using the gravity Centre of the instances. The mediod algorithms represent each cluster by means of the instances closest to the gravity Centre.

The most well-known centroid algorithm is the k-means. The k-means method partitions the data set into k subsets such that all points in a given subset are closest to the same Centre. In detail, it randomly selects k of the instances to represent the clusters. Based on the selected attributes, all remaining instances are assigned to their closer Centre. K-means then computes the new centers by taking the mean of all data points belonging to the same cluster. The operation is iterated until there is no change in the gravity centers. If k cannot be known ahead of time, various values of k can be evaluated until the most suitable one is found. The effectiveness of this method as well as of others relies heavily on the objective function used in measuring the distance between instances. The difficulty is in finding a distance measure that works well with all types of data. There are several approaches to define the distance between instances.



Knowing how to optimally use the space available in your computer hard disk can be the difference between enjoying your personal computer or not. Computer hard disks come in different sizes, and it is possible to divide the hard disk into different sections, slices or storage units -- a process commonly referred to as partitioning. A computer can have a fixed or dynamic partition. A fixed partition has a constant size while a dynamic partition can change its size based on the need. Most operating systems have methods and software inbuilt for the managing of fixed partitions.

1.1.1 PREVENT DATA LOSS:

One of the advantages of fixed partitions is that you can prevent data loss when there is a software malfunction or loss of power. Fixed partitions also help you increase your chances of data recovery in worse situations. The computer hard disk should be partitioned into at least two major sections to enjoy this advantage. One partition should have the programs installed while the other partition should have the data. In the event of a program malfunctions when working on your computer, you can access, retrieve or recover your data intact from the data partition.

2.1.2 RESRICTIONS:

A disadvantage of fixed partitions is the severe restriction that comes through the fixed or allocated space in the particular partition. This means you cannot install a file, folder or program that is bigger than the space provided in the partition. This, therefore, limits the operation or work you can do in any given partition.

2.1.3 LOSS OF SPACE:

Another disadvantage of fixed partitions is the loss of disk space from the total disk space available when operating different operating systems in the same hard disk. This especially happens when you are forced to duplicate various files, folders or programs from one operating system to another operating system in order to perform certain work or functions in that particular operating system. This duplication of content from one partition to another reduces the overall space you can use in the computer hard disk for other files, folders or programs.

2.2. HIERARCHICAL CLUSTERING:

The hierarchical methods group data instances into a tree of clusters. There are two major methods under this category. One is the agglomerative method, which forms the clusters in a bottom-up fashion until all data instances belong to the same cluster. The other is the divisive method, which splits up the data set into smaller cluster in a top-down fashion until each cluster contains only one instance. Both divisive algorithms and agglomerative algorithms can be represented by dendrograms.

Both agglomerative and divisive methods are known for their quick termination. However, both methods suffer from their inability to perform adjustments once the splitting or merging decision is made.

Other advantages are:
1) Does not require the number of clusters to be known in advance
2) Computes a complete hierarchy of clusters
3) Good result visualizations are integrated into the methods
4) A "flat" partition can be derived.

Hierarchical clustering techniques use various criteria to decide "locally" at each step which clusters should be joined (or split for divisive approaches). For agglomerative hierarchical techniques, the criterion is typically to merge the "closest" pair of clusters, where "close" is defined by a specified measure of cluster proximity. There are three definitions of the closeness between two clusters: single-link, complete-link and average-link.

The single-link similarity between two clusters is the similarity between the two most similar instances, one of which appears in each cluster. Single link is good at handling non-elliptical shapes, but is sensitive to noise and outliers. The complete-link similarity is the similarity between the two most dissimilar instances, one from each cluster. Complete link is less susceptible to noise and outliers, but can break large clusters, and has trouble with convex shapes. The average-link similarity is a compromise between the two.

Some of the hierarchical clustering algorithms are:
Balanced Iterative Reducing and clustering using Hierarchies - BIRCH, Clustering Using representatives - CURE and CHAMELEON.

2.2.1 ADVANTAGES OF HIERARCHICAL CLUSTERING:

It is sometimes meaningful to cluster data at the experiment level rather than at the level of individual genes. Such experiments are most often used to identify similarities in overall gene-expression patterns in the context of different treatment regimens—the goal being to stratify patients based on their molecular-level responses to the treatments. The hierarchical techniques outlined earlier are appropriate for such clustering, which is based on the pairwise statistical comparison of complete scatterplots rather than individual gene sequences. The data are represented as a matrix of scatterplots, ultimately reduced to a matrix of correlation coefficients. The correlation coefficients are then used to construct a two-dimensional dendrograms in the exact same way as in the gene-cluster experiments previously described.

The overall process of constructing a two-dimensional dendrograms using hierarchical clustering data is depicted. The example in the figure embodies all the principles of the technique but in a vastly simplified form; expression-profile experiments typically include hundreds, sometimes thousands of genes, and the analysis is almost always more complex than this illustration.

Construction of a two-dimensional dendrograms representing a hierarchical cluster of related genes. Each column represents a different experiment, each row a different spot on the microarray. The height of each link is inversely proportional to the strength of the correlation. Relative correlation strengths are represented by integers in the accompanying chart sequence. Genes 1 and 2 are most closely regulated, followed by genes 4 and 5. The regulation of gene 3 is more closely linked with the regulation of genes 4 and 5 than any remaining link or combination of links. The strength of the correlation between the expression levels of genes 1 and 2 and the cluster containing genes 3, 4, and 5 is the weakest (relative score of 10).

Messenger *RNA* profiling techniques have become a cornerstone of modern disease classification. These advances are especially significant in areas such as oncology and neuroscience, where complex phenotypes have recently been found to correlate with specific changes in gene expression, the result being more precise patient stratification both for clinical trials and treatment. A significant example that illustrates the utility of hierarchical clustering involves the identification of distinct tumor subclasses in diffuse large B-cell lymphoma (*DLBCL*). Two distinct forms of *DLBCL* have been identified using hierarchical clustering techniques, each related to a different stage of B-cell differentiation. The fact that the cluster correlates are significant is demonstrated by direct relationships to patient survival rates

2.2.2 DISADVANTAGES OF HIERARCHICAL CLUSTERING:

Despite its proven utility, hierarchical clustering has many flaws. Interpretation of the hierarchy is complex and often confusing; the deterministic nature of the technique prevents reevaluation after points are grouped into a node; all determinations are strictly based on local decisions and a single pass of analysis; it has been demonstrated that the tree structure can lock in accidental features reflecting idiosyncrasies of the clustering rules; expression patterns of individual gene sequences become less relevant as the clustering process progresses; and an incorrect assignment made early in the process cannot be corrected . These deficiencies have driven the development of additional clustering techniques that are based on multiple passes of analysis and utilize advanced algorithms borrowed from the artificial intelligence community. Two of these techniques, k-means clustering and self-organizing maps (SOMs), have achieved widespread acceptance in research oncology where they have been enormously successful in identifying meaningful genetic differences between patient populations.

When discussing clustering algorithms, it is essential to recognize the limitations of two- and three-dimensional representations of individual gene-expression values across a collection of experiments depicts a simple analysis composed of two experiments. Each experiment is represented by a dimension in the grid, and clusters of the genes are readily apparent. Visual representation of two gene-clustering experiments. For each transcript, the fluorescence ratio (Cy5/Cy3) is plotted on one of the two axes; y-axis for experiment 1 and x-axis for experiment 2. Genes with high fluorescence ratios in both experiments appear farthest from the origin. Genes with similar behavior across the two experiments are closely clustered on the graph. Three different clusters are evident in diagram. Representations of two or three experiments are relatively straightforward to visualize because they can be plotted on the axes of a simple graph (either x and y axes or x, y, and z axes).

Results from more than four experiments are difficult to represent because they cannot be visualized in three dimensions. which depicts the results from three experiments, represents the most complex case that can be easily visualized on the axes of a graph. As in the two-dimensional case, each gene occupies a unique position on the graph determined by its fluorescence ratio in each of the three experiments. For example, a gene that exhibits Cy5/Cy3 fluorescence ratios of 3,5, and 4.5 in the three experiments would be represented by a single point plotted at the coordinate 3,5,4.5 in the graph. As in the two-experiment model, absolute distance from the origin correlates with the Cy5/Cy3 ratio, and the distance between points in three-dimensional space is representative of the likelihood that the genes they represent are regulated across the three experiments.

A 3D(three experiment) gene-clustering analysis containing ten different sequences. Each axis in the drawing represents a different experiment, and each set of expression levels is represented by a single vector defined in three dimensions. As in the two-dimensional case, grouping of the sequences is accomplished by determining the geometric distance between each vector. Higher-dimensional models representing more than three experiments cannot be visualized as single vectors, and so different graphical techniques must be used.

The ten genes used in these experiments are clustered into readily recognizable groups. As mentioned previously, higher-dimensional representations, those containing more than three sets of experimental results, are much more complex to imagine because absolute distances between individual genes and gene clusters do not lend themselves to visual representation. However, despite the complexities associated with visual representation of microarray data across large numbers of experiments, it is always possible to calculate a single vector to represent all the expression values for any gene sequence regardless of the number of dimensions/experiments. It is the distance between these vectors that determines the degree to which a pair of genes is regulated.

2.3 DENITY-BASED CLUSTERING:

Density-based clustering algorithms try to find clusters based on density of data points in a region. The key idea of density-based clustering is that for each instance of a cluster the neighborhood of a given radius (*Eps*) has to contain at least a minimum number of instances (*minpts)*. One of the most well-known density-based clustering algorithms is the DBSCAN.

DBSCAN separate data points into
Three classes:

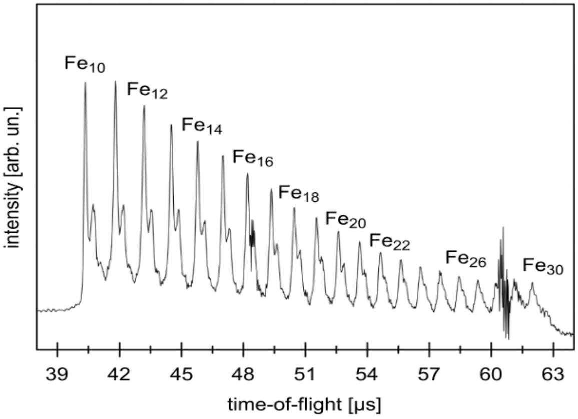
- Core points. These are points that are at the Interior of a cluster. A point is an interior point if there are enough points in its neighborhood.
- Border points. A border point is a point that is not a core point, i.e., there are not enough points in its neighborhood, but it falls within the neighborhood of a core point.
- Noise points. A noise point is any point that is not a core point or a border point

To find a cluster, DBSCAN starts with an arbitrary instance (*p*) in data set (*D*) and retrieves all instances of *D* with respect to *Eps* and *minpts*. The algorithm makes use of a spatial data structure - R*tree [24] - to locate points within Eps distance from the core points of the clusters.

2.4 GRID-BASED CLUSTERING:

Grid-based clustering algorithms first quantize the clustering space into a finite number of cells (hyper-rectangles) and then perform the required operations on the quantized space. Cells that contain more than certain number of points are treated as dense and the dense cells are connected to form the clusters. Some of the grid-based clustering algorithms are: statistical information Grid-based method - STING, wave cluster, and clustering Inquest - CLIQUE.

STING first divides the spatial area into several levels of rectangular cells in order to form a hierarchical structure. The cells in a high level are composed from the cells in the lower level. It generates a hierarchical structure of the grid cells so as to represent the clustering information at different levels. Although STING generates good clustering results in a short running time, there are two major problems with this algorithm. Firstly, the performance of STING relies on the granularity of the lowest level of the grid structure. Secondly, the resulting clusters are all bounded horizontally or vertically, but never diagonally. This shortcoming might greatly affect the cluster quality.



CLIQUE is another grid-based clustering algorithm. CLIQUE starts by finding all the dense areas in the one-dimensional spaces corresponding to each attribute. CLIQUE then generates the set of two-dimensional cells that might possibly be dense, by looking at dense one-dimensional cells, as each two-dimensional cell must be associated with a pair of dense one-dimensional cells. Generally, CLIQUE generates the possible set of k-dimensional cells that might possibly be dense by looking at dense (k - 1) dimensional cells. CLIQUE produces identical results irrespective of the order in which the input records are presented. In addition, it generates cluster descriptions in the form of DNF expressions [1] for ease of comprehension. Moreover, empirical evaluation shows that CLIQUE scales linearly with the number of instances, and has good scalability as the number of attributes is increased.

Unlike other clustering methods, wave cluster does not require users to give the number of clusters. It uses a wavelet transformation to transform the original feature space. In wavelet transform, convolution with an appropriate function results in a transformed space where the Natural clusters in the data become distinguishable. It is a very powerful method; however, it is not efficient in high dimensional space.

III. Proposed System:

Now, introducing the

- Correlation Preserving Indexing (CPI).
- Clustering algorithm based on CPI.

CPI ALGORITHEM

INPUT: Consistent ontology 0 ←(c, p, I, a, F) 01 U 02.axiom **a** representing the current integration candidate (e.g. (c1 out equivalent class c2)), initial similarity score for alignment candidate **a** (e.g. the similarity score between c1 & c2)

Output: New similarity score normalized with respect to the consequences of inference **0** if the candidate alignment causes an inconsistency.

```
A←A U {a}
f ← 1
For  I ←1 to N do
    If a is a class equivalence/Subsumption axiom then
        P ←choose a property incident to one of the class in a
    else if a is a property equivalence/subsumption axiom then
        P←choose a property referenced by a
    else if a is an instance equivalence axiom then
        P←choose a property incident to an instances in a
    End if
    Choose an axiom $^{ar} \neq$ a from A that involves p
    $(\mu^i, e^i) \leftarrow \Delta(\mu^{i-1}, e^{i-1})$ by applying $\Delta$ for $a^t$
    if there is an inconsistency then
        Return 0
    End if
    Recompute the effected instance set closure
    For all pairs of X, Y € I that have become equivalent do
        f ← f similarity(x, y)/ 1-similarity(x, y)
    End for
End for
Return min (1, f, score)
```

CPI ALGORITHEM

Input: Consistent ontology 01 and 02

Output: Integration of 01 and 02.

```
Compare 0 ← 01 and 02
Initialize clusters to single classes, properties and individuals in 0
Compute initials similarity scores for all pairs of clusters
Repeat
    Determine equivalence/Subsumption axioms for all pairs of clusters with similarity greater than $\mu$
    Run CPI interference in parallel for all pairs of clusters with similarity greater than $\lambda$ t
    Choose pair of clusters with best similarity measure and merge them into a single cluster
    Recompute similarity scores for affected clusters
Until there are no clusters with similarity measure greater than $\lambda$ t
Return 0
```

Scope:

- Can be applied on blog analysis.
- Can be applied on web pages classification.
- Can be applied on news classification.

IV. Conclusion:

With the further advancements of the data mining algorithms, the efficiency of analyzing large amounts of the data must be improved for achieving further efficiency. The demand for less retrieval period of data extraction from a large data warehouses points the efficiency of data retrieval of present clustering techniques, here we must use further advancements to the present clustering techniques. And the new clustering algorithms have to improve for a good efficient data handling using clustering techniques.

- CPI method is the better approach than of LSI and CPI.
- CPI method is better than of traditional K-means.
- CPI method has good generalization capability.

References:

[1]. Jeffrey Augen, "Bioinformatics and Data Mining in Support of Drug Discover," Handbook of Anticancer Drug Development. D. Bud man, A. Calvert, E. Rowinsky, editors. Lippincott Williams and Wilkins.2003).

[2]. Document Clustering in Correlation Similarity Measure Space Taiping Zhang, Member, IEEE, Yuan Yan Tang, Fellow, IEEE,Bin Fang, Senior Member, IEEE, and Yong Xiang.

[3]. Effective and efficient Clustering methods for spatial data mining-Raymont.T.Ng, Jaiwei Han.

[4]. Data Mining Concepts and Techniques-Jaiwei Han, M. Kamber, Jian Pei.

[5]. Document Clustering usinglocality Preserving IndexingDeng CAI, Xiaofei He, And Jiawei Han.

[6]. Data Clustering: A reviewK. JainMichigan State UniversityM.N. MurtyIndian Institute of sciencesand P.J. FLYNN the Ohio State University.

[7]. Indexing by Latent Semantic Analysis Scott Deerwester,Susan T. Dumais, George W. Furnas, and Thomas K. Landauer,Richard HarshmanUniversity of Western Ontario, London, Ontario Canada. Document Clustering with Cluster Refinement and Model Selection Capabilities -Xin Liu, Yihong Gong, Wei Xu.,Shenghuo Zhu.

[8]. Recent Advances in Clustering: A Brief Survey- S.B. KOTSIANTIS, P. E. PINTELAS.

P.Sudheer was born in HYD, Ranga Reddy d.t, Andhra Pradesh,India.he received B.tech in Computer Science&Engineering from jntu Kakinada university,Kakinada,Andhra Pradesh,India.presently he is doing M.tech in Sri Mittapalli College of Engineering, NH-5,Thumalapalem Guntur,Andhrapradesh,India.He areas of research interests include Data Mining.

Kesava Kumar Vemula received the M.Sc Degree in Computer Science from Madras University, Chennai, in 2000, and M.Tech in Computer Science & Engineering from Pondicherry Central University, Pondicherry, in 2007. His research interests include denial-of-service attacks in wireless networks, privacy protection in wireless sensor networks, as well as secure routing algorithms in multichannel and cognitive radio networks.