

International Journal of Computer Science and Mobile Computing



A Monthly Journal of Computer Science and Information Technology

ISSN 2320-088X

IJCSMC, Vol. 3, Issue. 2, February 2014, pg.86 – 99

RESEARCH ARTICLE

Evaluating and Analyzing Clusters in Data Mining using Different Algorithms

N. Sunil Chowdary¹, D. Sri Lakshmi Prasanna², P. Sudhakar³

^{1,2,3} Assistant Professor, Department of CSE

^{1,2,3} Sri Sarathi Institute of Engineering and Technology, Nuzvid,-521201, Krishna District, A.P

Abstract:- Cluster analysis or clustering is the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense or another) to each other than to those in other groups (clusters). It is a main task of exploratory data mining, and a common technique for statistical data analysis, used in many fields, including machine learning, pattern recognition, image analysis, information retrieval, and bioinformatics.

1. Introduction:

Cluster analysis itself is not one specific algorithm, but the general task to be solved. It can be achieved by various algorithms that differ significantly in their notion of what constitutes a cluster and how to efficiently find them. Popular notions of clusters include groups with small distances among the cluster members, dense areas of the data space, intervals or particular statistical distributions. Clustering can therefore be formulated as a multi-objective optimization problem. The appropriate clustering algorithm and parameter settings (including values such as the distance function to use, a density threshold or the number of expected clusters) depend on the individual data set and intended use of the results. Cluster analysis as such is not an automatic task, but an iterative process of knowledge discovery or interactive multi-objective optimization that involves trial and failure. It will often be necessary to modify data preprocessing and model parameters until the result achieves the desired properties.

Besides the term *clustering*, there are a number of terms with similar meanings, including *automatic classification*, *numerical taxonomy*, *botryology* (from Greek βότρυς "grape") and *typological analysis*. The subtle differences are often in the usage of the results:

While in data mining, the resulting groups are the matter of interest, in automatic classification primarily their discriminative power is of interest. This often leads to misunderstanding between researchers coming from the fields of data mining and machine learning, since they use the same terms and often the same algorithms, but have different goals.

Cluster analysis was originated in anthropology by Driver and Kroeber in 1932 and introduced to psychology by Zubin in 1938 and Robert Tryon in 1939 and famously used by Cattell beginning in 1943 for trait theory classification in personality psychology.

2. Clusters and clustering

According to Vladimir Estivill-Castro, the notion of a "cluster" cannot be precisely defined, which is one of the reasons why there are so many clustering algorithms. There is a common denominator: a group of data objects. However, different researchers employ different cluster models, and for each of these cluster models again different algorithms can be given. The notion of a cluster, as found by different algorithms, varies significantly in its properties. Understanding these "cluster models" is key to understanding the differences between the various algorithms. Typical cluster models include:

- Connectivity models: for example hierarchical clustering builds models based on distance connectivity.
- Centroid models: for example the k-means algorithm represents each cluster by a single mean vector.
- Distribution models: clusters are modeled using statistical distributions, such as multivariate normal distributions used by the Expectation-maximization algorithm.
 - Density models: for example DBSCAN and OPTICS defines clusters as connected dense regions in the data space.
- Subspace models: in Biclustering (also known as Co-clustering or two-mode-clustering), clusters are modeled with both cluster members and relevant attributes.
- Group models: some algorithms do not provide a refined model for their results and just provide the grouping information.
- Graph-based models: a clique, i.e., a subset of nodes in a graph such that every two nodes in the subset are connected by an edge can be considered as a prototypical form of cluster. Relaxations of the complete connectivity requirement (a fraction of the edges can be missing) are known as quasi-cliques.

A "clustering" is essentially a set of such clusters, usually containing all objects in the data set. Additionally, it may specify the relationship of the clusters to each other, for example a hierarchy of clusters embedded in each other.

Clustering can be roughly distinguished as:

- Hard clustering: whether each object belongs to a cluster or not
- Soft clustering (also called fuzzy clustering): each object belongs to each cluster to a certain degree. (e.g. a likelihood of belonging to the cluster)

There are also finer distinctions possible, for example:

- Strict partitioning clustering: here each object belongs to exactly one cluster.
- Strict partitioning clustering with outliers: objects can also belong to no cluster, and are considered outliers.
- Overlapping clustering (also: alternative clustering, multi-view clustering): while usually a hard clustering, objects may belong to more than one cluster.
- Hierarchical clustering: objects that belong to a child cluster also belong to the parent cluster.
- Subspace clustering: while an overlapping clustering, within a uniquely defined subspace, clusters are not expected to overlap.

3. Clustering algorithms

Clustering algorithms can be categorized based on their cluster model, as listed above. The following overview will only list the most prominent examples of clustering algorithms, as there are possibly over 100 published clustering algorithms. Not all provide models for their clusters and can thus not easily be categorized

There is no objectively "correct" clustering algorithm, but as it was noted, "clustering is in the eye of the beholder." The most appropriate clustering algorithm for a particular problem often needs to be chosen experimentally, unless there is a mathematical reason to prefer one cluster model over another. It should be noted that an algorithm that is designed for one kind of model has no chance on a data set that contains a radically different kind of model. For example, k-means cannot find non-convex clusters.

3.1 Connectivity based clustering (hierarchical clustering):

Connectivity based clustering, also known as *hierarchical clustering*, is based on the core idea of objects being more related to nearby objects than to objects farther away. These algorithms connect "objects" to form "clusters" based on their distance. A cluster can be described largely by the maximum distance needed to connect parts of the cluster. At different distances, different clusters will form, which can be represented using a dendrogram, which explains where the common name "hierarchical clustering" comes from: these algorithms do not provide a single partitioning of the data set, but instead provide an extensive hierarchy of clusters that merge with

each other at certain distances. In a dendrogram, the y-axis marks the distance at which the clusters merge, while the objects are placed along the x-axis such that the clusters don't mix.

Connectivity based clustering is a whole family of methods that differ by the way distances are computed. Apart from the usual choice of distance functions, the user also needs to decide on the linkage criterion to use. Popular choices are known as single-linkage clustering (the minimum of object distances), complete linkage clustering (the maximum of object distances) or UPGMA ("Unweighted Pair Group Method with Arithmetic Mean", also known as average linkage clustering). Furthermore, hierarchical clustering can be agglomerative (starting with single elements and aggregating them into clusters) or divisive (starting with the complete data set and dividing it into partitions).

These methods will not produce a unique partitioning of the data set, but a hierarchy from which the user still needs to choose appropriate clusters. They are not very robust towards outliers, which will either show up as additional clusters or even cause other clusters to merge (known as "chaining phenomenon", in particular with single-linkage clustering). In the general case, the complexity is $O(n^3)$, which makes them too slow for large data sets. For some special cases, optimal efficient methods (of complexity $O(n^2)$) are known: SLINK for single-linkage and CLINK for complete-linkage clustering. In the data mining community these methods are recognized as a theoretical foundation of cluster analysis, but often considered obsolete. They did however provide inspiration for many later methods such as density based clustering.

- **Linkage clustering examples:**

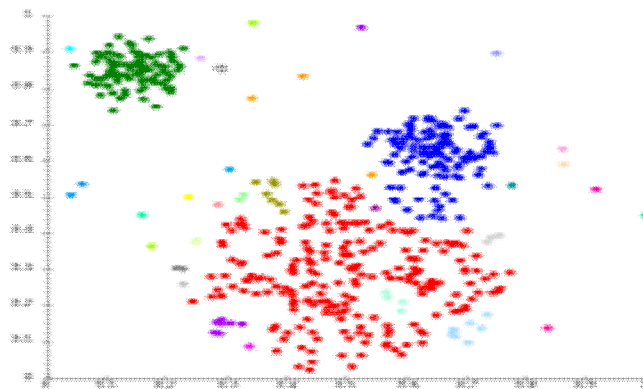


Figure: Single-linkage on Gaussian data. At 35 clusters, the biggest cluster starts fragmenting into smaller parts, while before it was still connected to the second largest due to the single-link effect.

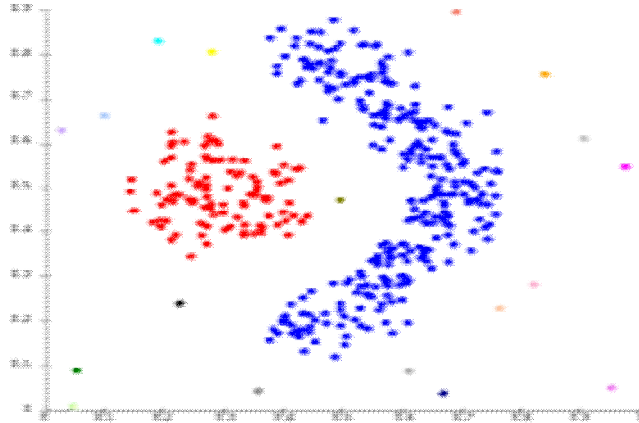


Figure: Single-linkage on density-based clusters. 20 clusters extracted, most of which contain single elements, since linkage clustering does not have a notion of "noise".

3.2 Centroid-based clustering (k-means clustering):

In Centroid-based clustering, clusters are represented by a central vector, which may not necessarily be a member of the data set. When the number of clusters is fixed to k , k -means clustering gives a formal definition as an optimization problem: find the k cluster centers and assign the objects to the nearest cluster center, such that the squared distances from the cluster are minimized.

The optimization problem itself is known to be NP-hard, and thus the common approach is to search only for approximate solutions. A particularly well known approximative method is Lloyd's algorithm, often actually referred to as " k -means algorithm". It does however only find a local optimum, and is commonly run multiple times with different random initializations.

Most k -means-type algorithms require the number of clusters - k - to be specified in advance, which is considered to be one of the biggest drawbacks of these algorithms. Furthermore, the algorithms prefer clusters of approximately similar size, as they will always assign an object to the nearest centroid. This often leads to incorrectly cut borders in between of clusters

K -means has a number of interesting theoretical properties. On the one hand, it partitions the data space into a structure known as a Voronoi diagram. On the other hand, it is conceptually close to nearest neighbor classification, and as such is popular in machine learning. Third, it can be seen as a variation of model based classification, and Lloyd's algorithm as a variation of the Expectation-maximization algorithm for this model discussed below.

- **k-Means clustering examples:**

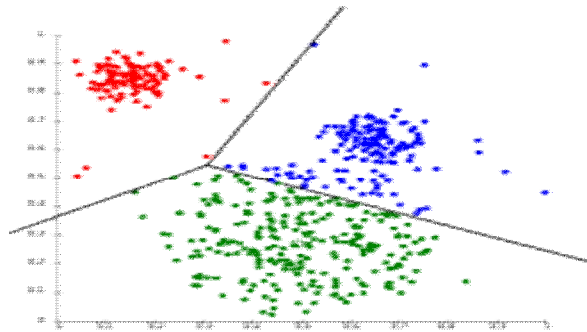


Figure: K-means separates data into Voronoi-cells, which assumes equal-sized clusters (not adequate here)

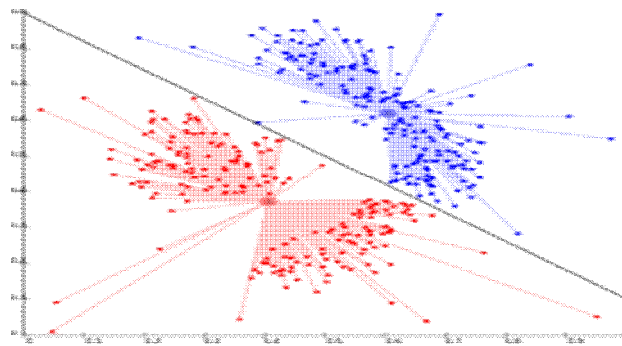


Figure: K-means cannot represent density-based clusters

3.3 Distribution-based clustering:

The clustering model most closely related to statistics is based on distribution models. Clusters can then easily be defined as objects belonging most likely to the same distribution. A nice property of this approach is that this closely resembles the way artificial data sets are generated: by sampling random objects from a distribution.

One prominent method is known as Gaussian mixture models (using the expectation-maximization algorithm). Here, the data set is usually modeled with a fixed (to avoid overfitting) number of Gaussian distributions that are initialized randomly and whose parameters are iteratively optimized to fit better to the data set. This will converge to a local optimum, so multiple runs may produce different results.

Distribution-based clustering is a semantically strong method, as it not only provides you with clusters, but also produces complex models for the clusters that can also capture correlation and dependence of attributes. However, using these algorithms puts an extra burden on the user: to choose appropriate data models to optimize, and for many real data sets, there may be no mathematical model available the algorithm is able to optimize.

● **Expectation-Maximization (EM) clustering examples:**

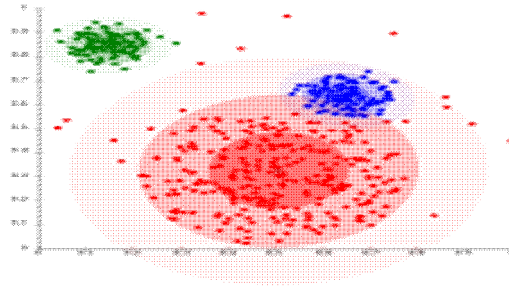


Figure: On Gaussian-distributed data, EM works well, since it uses Gaussians for modelling clusters

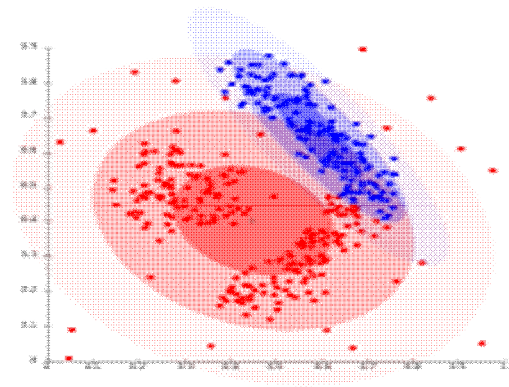


Figure: Density-based clusters cannot be modeled using Gaussian distributions

3.4 Density-based clustering:

In density-based clustering, clusters are defined as areas of higher density than the remainder of the data set. Objects in these sparse areas - that are required to separate clusters - are usually considered to be noise and border points.

The most popular density based clustering method is DBSCAN. In contrast to many newer methods, it features a well-defined cluster model called "density-reachability". Similar to linkage based clustering; it is based on connecting points within certain distance thresholds. However, it only connects points that satisfy a density criterion, in the original variant defined as a minimum number of other objects within this radius. A cluster consists of all density-connected objects. Another interesting property of DBSCAN is that its complexity is fairly low - it requires a linear number of range queries on the database - and that it will discover essentially the same results in each run, therefore there is no need to run it multiple times. OPTICS is a generalization of DBSCAN that removes the need to choose an appropriate value for the range parameter ϵ , and produces a hierarchical result related to that of linkage clustering. DeLi-Clu, Density-Link-Clustering combines ideas from single-linkage clustering and OPTICS, eliminating the ϵ parameter entirely and offering performance improvements over OPTICS by using an R-tree index.

The key drawback of DBSCAN and OPTICS is that they expect some kind of density drop to detect cluster borders. Moreover, they cannot detect intrinsic cluster structures which are prevalent in the majority of real life data. On a data set consisting of mixtures of Gaussians, these algorithms are nearly always outperformed by methods such as EM clustering that are able to precisely model this kind of data.

● **Density-based clustering examples:**

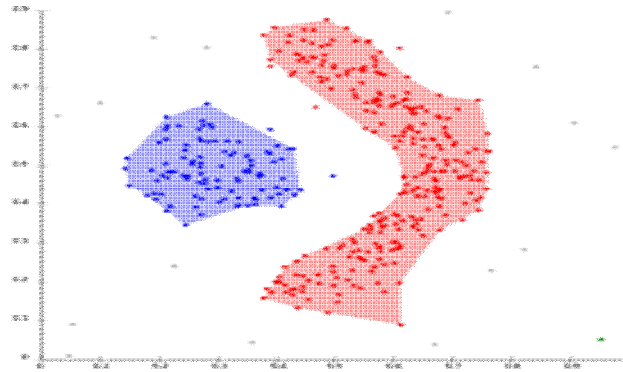


Figure: Density-based clustering with DBSCAN.

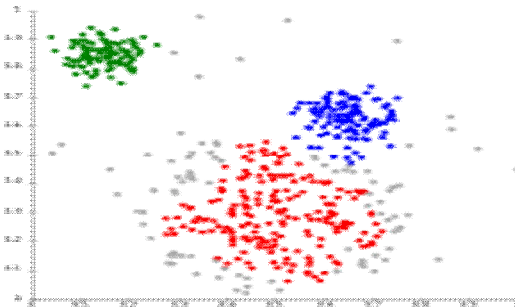


Figure: DBSCAN assumes clusters of similar density, and may have problems separating nearby clusters

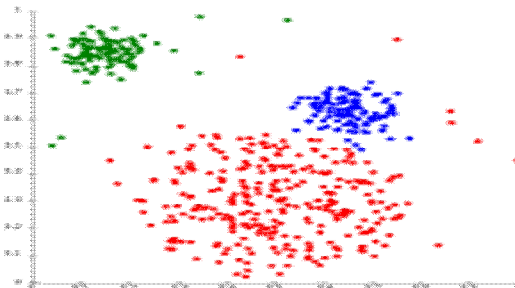


Figure: OPTICS is a DBSCAN variant that handles different densities much better

4. Evaluation of clustering results

Evaluation of clustering results sometimes is referred to as cluster validation.

There have been several suggestions for a measure of similarity between two clusterings. Such a measure can be used to compare how well different data clustering algorithms perform on a set of data. These measures are usually tied to the type of criterion being considered in assessing the quality of a clustering method.

4.1 Internal evaluation

When a clustering result is evaluated based on the data that was clustered itself, this is called internal evaluation. These methods usually assign the best score to the algorithm that produces clusters with high similarity within a cluster and low similarity between clusters. One drawback of using internal criteria in cluster evaluation is that high scores on an internal measure do not necessarily result in effective information retrieval applications. Additionally, this evaluation is biased towards algorithms that use the same cluster model. For example k-Means clustering naturally optimizes object distances, and a distance-based internal criterion will likely overrate the resulting clustering.

Therefore, the internal evaluation measures are best suited to get some insight into situations where one algorithm performs better than another, but this shall not imply that one algorithm produces more valid results than another. Validity as measured by such an index depends on the claim that this kind of structure exists in the data set. An algorithm designed for some kind of models has no chance if the data set contains a radically different set of models, or if the evaluation measures a radically different criterion. For example, k-means clustering can only find convex clusters, and many evaluation indexes assume convex clusters. On a data set with non-convex clusters neither the use of k-means, nor of an evaluation criterion that assumes convexity, is sound.

The following methods can be used to assess the quality of clustering algorithms based on internal criterion:

- **Davies–Bouldin index**

The Davies–Bouldin index can be calculated by the following formula:

$$DB = \frac{1}{n} \sum_{i=1}^n \max_{i \neq j} \left(\frac{\sigma_i + \sigma_j}{d(c_i, c_j)} \right)$$

Where n is the number of clusters, c_x is the centroid of cluster x , σ_x is the average distance of all elements in cluster x to centroid c_x , and $d(c_i, c_j)$ is the distance between centroids c_i and c_j . Since algorithms that produce clusters with low intra-cluster distances (high intra-cluster similarity) and high inter-cluster distances (low inter-cluster similarity) will have a low Davies–Bouldin index, the clustering algorithm that produces a collection of clusters with the smallest Davies–Bouldin index is considered the best algorithm based on this criterion.

- **Dunn index**

The Dunn index aims to identify dense and well-separated clusters. It is defined as the ratio between the minimal inter-cluster distance to maximal intra-cluster distance. For each cluster partition, the Dunn index can be calculated by the following formula:

$$D = \min_{1 \leq i \leq n} \left\{ \min_{1 \leq j \leq n, i \neq j} \left\{ \frac{d(i, j)}{\max_{1 \leq k \leq n} d'(k)} \right\} \right\}$$

Where $d(i, j)$ represents the distance between clusters i and j , and $d'(k)$ measures the intra-cluster distance of cluster k . The inter-cluster distance $d(i, j)$ between two clusters may be any number of distance measures, such as the distance between the centroids of the clusters. Similarly, the intra-cluster distance $d'(k)$ may be measured in a variety of ways, such as the maximal distance between any pair of elements in cluster k . Since internal criteria seek clusters with high intra-cluster similarity and low inter-cluster similarity, algorithms that produce clusters with high Dunn index are more desirable.

4.2 External evaluation

In external evaluation, clustering results are evaluated based on data that was not used for clustering, such as known class labels and external benchmarks. Such benchmarks consist of a set of pre-classified items, and these sets are often created by human (experts). Thus, the benchmark sets can be thought of as a gold standard for evaluation. These types of evaluation methods measure how close the clustering is to the predetermined benchmark classes. However, it has recently been discussed whether this is adequate for real data, or only on synthetic data sets with a factual ground truth, since classes can contain internal structure, the attributes present may not allow separation of clusters or the classes may contain anomalies. Additionally, from a knowledge discovery point of view, the reproduction of known knowledge may not necessarily be the intended result.

Some of the measures of quality of a cluster algorithm using external criteria include:

- **Rand measure**

The Rand index computes how similar the clusters (returned by the clustering algorithm) are to the benchmark classifications. One can also view the Rand index as a measure of the percentage of correct decisions made by the algorithm. It can be computed using the following formula:

$$RI = \frac{TP + TN}{TP + FP + FN + TN}$$

Where TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives, and FN is the number of false negatives. One issue with the Rand index is that false positives and false negatives are equally weighted. This may be an undesirable characteristic for some clustering applications. The F-measure addresses this concern.

- **F-measure**

The F-measure can be used to balance the contribution of false negatives by weighting recall through a parameter $\beta \geq 0$. Let precision and recall be defined as follows:

$$P = \frac{TP}{TP + FP}$$

$$R = \frac{TP}{TP + FN}$$

Where P is the precision rate and R is the recall rate. We can calculate the F-measure by using the following formula:

$$F_{\beta} = \frac{(\beta^2 + 1) \cdot P \cdot R}{\beta^2 \cdot P + R}$$

Notice that when $\beta = 0$, $F_0 = P$. In other words, recall has no impact on the F-measure when $\beta = 0$, and increasing β allocates an increasing amount of weight to recall in the final F-measure.

- **Pair-counting F-Measure:**

It is the F-Measure applied to the set of object pairs, where objects are paired with each other when they are part of the same cluster. This measure is able to compare clusterings with different numbers of clusters.

- **Jaccard index**

The Jaccard index is used to quantify the similarity between two datasets. The Jaccard index takes on a value between 0 and 1. An index of 1 means that the two dataset are identical, and an index of 0 indicates that the datasets have no common elements. The Jaccard index is defined by the following formula:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{TP}{TP + FP + FN}$$

This is simply the number of unique elements common to both sets divided by the total number of unique elements in both sets.

- **Fowlkes–Mallows index**

The Fowlkes-Mallows index computes the similarity between the clusters returned by the clustering algorithm and the benchmark classifications. The higher the value of the Fowlkes-Mallows index the more similar the clusters and the benchmark classifications are. It can be computed using the following formula:

$$FM = \sqrt{\frac{TP}{TP + FP} \cdot \frac{TP}{TP + FN}}$$

Where TP is the number of true positives, FP is the number of false positives, and FN is the number of false negatives. The FM index is the geometric mean of the precision and recall P and R , while the F-measure is their harmonic mean. Moreover, precision and recall are also known as Wallace's indices B^I and B^{II} .

● Confusion matrix

A confusion matrix can be used to quickly visualize the results of a classification (or clustering) algorithm. It shows how different a cluster is from the gold standard cluster.

● Mutual Information

It is an information theoretic measure of how much information is shared between a clustering and a ground-truth classification that can detect a non-linear similarity between two clusterings. Adjusted mutual information is the corrected-for-chance variant of this that has a reduced bias for varying cluster numbers.

Conclusion:

The research community is searching for a new environment in which to generate innovative ideas and prove their effectiveness. A new paradigm beyond clustering in data mining and types of Clusters and algorithms that support in Data mining and are describing yields the demand in Data mining.

The task of grouping a set of objects in such a way that objects in the same group (called a cluster) is more similar to each other than to those in other clusters. The main task of exploratory data mining, and a common technique for statistical data analysis, used in many fields, including machine learning, pattern recognition, image analysis, information retrieval, and bioinformatics.

Clustering provides a challenging platform in which to investigate applications that can be performed in Data Mining. Some of the measures such as Internal Evaluation and External Evaluation have been provided to evaluate Cluster analysis based on data that was not used for clustering, such as known class labels and external benchmarks for assigning the quality of measure in Clustering.

References:

- [1] Tryon, Robert C. (1939). *Cluster Analysis: Correlation Profile and Orthometric (factor) Analysis for the Isolation of Unities in Mind and Personality*. Edwards Brothers.
- [2] L. Parsons, E. Haque and H. Liu, Subspace Clustering for High Dimensional Data: A Review, SIGKDD Explorations, 6(1), June 2004
- [3] E. Schikuta. Grid clustering: An efficient hierarchical clustering method for very large data sets. Proc. 1996 Int. Conf. on Pattern Recognition,.

- [4] G. Sheikholeslami, S. Chatterjee, and A. Zhang. WaveCluster: A multi-resolution clustering approach for very large spatial databases. VLDB'98.
- [5] A. K. H. Tung, J. Han, L. V. S. Lakshmanan, and R. T. Ng. *Constraint-Based Clustering in Large Databases, ICDT'01*.
- [6] A. K. H. Tung, J. Hou, and J. Han. *Spatial Clustering in the Presence of Obstacles , ICDE'01*
- [7] H. Wang, W. Wang, J. Yang, and P.S. Yu. Clustering by pattern similarity in large data sets, *SIGMOD' 02*.
- [8] W. Wang, Yang, R. Muntz, STING: A Statistical Information grid Approach to Spatial Data Mining, VLDB'97.
- [9] T. Zhang, R. Ramakrishnan, and M. Livny. BIRCH : an efficient data clustering method for very large databases. SIGMOD'96.

Web References:

- [1] <http://en.wikipedia.org/wiki/File:DBSCAN-density-data.svg>
- [2] <http://www.acm.org/sigs/sigkdd/explorations>

Authors Bibliography



Mr. Sunil Chowdary Nandamala, excellent teacher, Received B.Tech., (CSE) from LBR College of Engineering, Mylavaram, JNTU Hyderabad and M.Tech, (CSE) from Akula Sriramulu Institute of Engineering and Technology, Tadepalligudem, JNTU Kakinada is working as an Assistant Professor in Department of C.S.E, Sri Sarathi Institute of Engineering and Technology, Nuzvid, Krishna District, Andhra Pradesh. He has 4 years of teaching experience. He has published many papers in both National & International Journals, Guided students in preparing Technical Presentations and Posters. His area of Interest includes Data Communications & Networks, Advanced Programming Technologies like ASP .Net, Java, Database Management Systems, **Software Quality Assurance** and other advances in computer Applications.



Mrs. D.Srilakshmiprasanna, Received B.Tech., (IT) from Prakasam Engineering College, Kandukuru, Prakasam District, JNTUK, Kakinada and M.Tech, (SE) from Godavari Institute of Engineering and Technology, Rajahmundry, E.G. District, JNTU Kakinada is working as an Assistant Professor in Department of C.S.E, Sri Sarathi Institute of Engineering and Technology, Nuzvid, Krishna District, Andhra Pradesh. She has 5 years of teaching experience. She has published papers in Journals, Guided students in projects in fulfillment of B.Tech. His area of Interest includes Data Mining and Networks, and other advances in computer Applications.



Mr. P.SUDHAKAR, Received B.Tech., (CSE) from Sri Sarathi Institute of Engineering and Technology, Nuzvid, Krishna District, JNTUH, Kakinada and M.Tech, (CSE) from Gokul Institute of Technology and Sciences, Bobbili, Vizianagaram. District, JNTU Kakinada is working as an Assistant Professor in Department of C.S.E, Sri Sarathi Institute of Engineering and Technology, Nuzvid, Krishna District, Andhra Pradesh. He has 6 years of teaching experience. He has published papers in Journals, Guided students in projects in fulfillment of B.Tech. His area of Interest includes Networking and soft computing and other advances in computer Applications.