

# Recent Advances in Clustering: A Brief Survey

S.B. KOTSIANTIS, P. E. PINTELAS  
Department of Mathematics  
University of Patras  
Educational Software Development Laboratory  
Hellas

*Abstract:* - Unsupervised learning (clustering) deals with instances, which have not been pre-classified in any way and so do not have a class attribute associated with them. The scope of applying clustering algorithms is to discover useful but unknown classes of items. Unsupervised learning is an approach of learning where instances are automatically placed into meaningful groups based on their similarity. This paper introduces the fundamental concepts of unsupervised learning while it surveys the recent clustering algorithms. Moreover, recent advances in unsupervised learning, such as ensembles of clustering algorithms and distributed clustering, are described.

*Key-Words:* - Pattern Analysis, Machine Intelligence, Intelligent Systems

## 1 Introduction

Cluster analysis is an unsupervised learning method that constitutes a cornerstone of an intelligent data analysis process. It is used for the exploration of inter-relationships among a collection of patterns, by organizing them into homogeneous clusters. It is called unsupervised learning because unlike classification (known as supervised learning), no a priori labeling of some patterns is available to use in categorizing others and inferring the cluster structure of the whole data. Intra-connectivity is a measure of the density of connections between the instances of a single cluster. A high intra-connectivity indicates a good clustering arrangement because the instances grouped within the same cluster are highly dependent on each other. Inter-connectivity is a measure of the connectivity between distinct clusters. A low degree of interconnectivity is desirable because it indicates that individual clusters are largely independent of each other.

Every instance in the data set is represented using the same set of attributes. The attributes are continuous, categorical or binary. To induce a hypothesis from a given data set, a learning system needs to make assumptions about the hypothesis to be learned. These assumptions are called biases. Since every learning algorithm uses some biases, it behaves well in some domains where its biases are appropriate while it performs poorly in other domains.

A problem with the clustering methods is that the interpretation of the clusters may be difficult. In addition, the algorithms will always assign the data to clusters even if there were no clusters in the data.

Therefore, if the goal is to make inferences about its cluster structure, it is essential to analyze whether the data set exhibits a clustering tendency. In a real-world application there may be errors (called noise) in the collected data set due to inaccurate measurement or due to missing values therefore a pre-processing is needed (e.g. choose a strategy for handling missing attribute values). The choice of which specific learning algorithm to use is a critical step, too. The issue of relating the learning algorithms to the type of data and to the nature of the problem to be solved still remains an open and fundamental problem [21]. An evaluation criterion of clustering quality is the unknown attribute prediction accuracy. The first step is by taking an unseen instance, removing the value of one of its attributes and then trying to classify it. The missing attribute on the unseen instance is predicted to be the same as the value of the attribute on the closest matching instance. This value can be then compared to the actual value of the removed attribute and so can be judged to be correct or not. This process is repeated for each attribute. The number of attributes correctly predicted is then totaled up and divided by the number of attributes in order to give the average prediction accuracy.

Cluster analysis is a difficult problem because many factors (such as effective similarity measures, criterion functions, algorithms and initial conditions) come into play in devising a well tuned clustering technique for a given clustering problem. Moreover, it is well known that no clustering method can adequately handle all sorts of cluster structures (shape, size and density).

Sometimes the quality of the clusters that are found can be improved by pre-processing the data. It is not uncommon to try to find noisy values and eliminate them by a preprocessing step. Another common technique is to use post-processing steps to try to fix up the clusters that have been found. For example, small clusters are often eliminated since they frequently represent groups of outliers (instances with noise). Alternatively, two small clusters that are close together can be merged. Finally, large clusters can be split into smaller clusters.

Outlier detection is one of the major technologies in data mining, whose task is to find small groups of data objects that are exceptional when compared with rest large amount of data. Outlier mining has strong application background in telecommunication, financial fraud detection, and data cleaning, since the patterns lying behind the outliers are usually interesting for helping the decision makers to make profit or improve the service quality. In recent years, outliers themselves draw much attention, and outlier detection is studied intensively by the data mining community [4; 30].

The missing value problem can occur due to some occasional sensor failures. One simple but wasteful method to cope with this problem is to throw away the incomplete attribute vectors. Another more logical method is the missing attribute's values for numeric attributes to instantiate with the median value of that attribute across all training instances. Missing attribute values for categorical attributes can be replaced by the mode value for that attribute across all training instances. Comparisons of various methods for dealing with missing data are found in [20].

Usually, from statistical point of view, instances with many irrelevant input attributes provide little information. Hence, in practical applications, it is wise to carefully choose which attributes to provide to the learning algorithm. Different algorithms have been developed for this purpose. For example, this can be accomplished by discarding attributes that show little variation or that are highly correlated with other attributes [35].

Generally, clustering algorithms can be categorized into partitioning methods, hierarchical methods, density-based methods, grid-based methods, and model-based methods. An excellent survey of clustering techniques can be found in (Jain et al., 1999). Thus, in this work apart from the brief description of the clustering techniques we refer to some more recent works than those in Jain's article as well as few articles that were not referred by Jain. The reader should be cautioned that a single article

couldn't be a comprehensive review of all learning algorithms. Rather, our goal is to provide a representative sample of the research in each of learning technique. In each of the areas, there are many other papers that describe relevant work. Some typical applications of the clustering in many fields can be found in (Han & Kamber, 2001).

Partitioning algorithms that construct various partitions and then evaluate them by some criterion are described in section 2. Hierarchical algorithms that create a hierarchical decomposition of the instances using some criterion are covered in section 3. The section 4 explains the density-based algorithms that are based on connectivity and density functions. The section 5 describes the grid-based methods, which are based on a multiple-level granularity structure. The model-based algorithms are covered in section 6, while, recent advances in clustering techniques, such as ensembles of clustering algorithms, are described in section 7. The final section concludes this work.

## 2 Partitioning Methods

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

The most well-known centroid algorithm is the k-means [21]. 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 centres. 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 [21].

Generally, the k-means algorithm has the following important properties: 1. It is efficient in processing large data sets, 2. It often terminates at a local optimum, 3. The clusters have spherical shapes, 4. It is sensitive to noise. The algorithm described

above is classified as a batch method because it requires that all the data should be available in advance. However, there are variants of the k-means clustering process, which gets around this limitation [21]. Choosing the proper initial centroids is the key step of the basic K-means procedure.

The k-modes algorithm [20] is a recent partitioning algorithm and uses the simple matching coefficient measure to deal with categorical attributes. The k-prototypes algorithm [20], through the definition of a combined dissimilarity measure, further integrates the k-means and k-modes algorithms to allow for clustering instances described by mixed attributes. More recently, in [6] another generalization of conventional k-means clustering algorithm has been presented. This new one applicable to ellipse-shaped data clusters as well as ball-shaped ones without dead-unit problem, but also performs correct clustering without pre-determining the exact cluster number.

Traditional clustering approaches generate partitions; in a partition, each pattern belongs to one and only one cluster. Hence, the clusters in a hard clustering are disjoint. Fuzzy clustering extends this notion to associate each pattern with every cluster using a membership function. Larger membership values indicate higher confidence in the assignment of the pattern to the cluster. One widely used algorithm is the Fuzzy C-Means (FCM) algorithm [32], which is based on k-means. FCM attempts to find the most characteristic point in each cluster, which can be considered as the “center” of the cluster and, then, the grade of membership for each instance in the clusters.

Other soft clustering algorithms have been developed and most of them are based on the Expectation-Maximization (EM) algorithm [26]. They assume an underlying probability model with parameters that describe the probability that an instance belongs to a certain cluster. The strategy in this algorithm is to start with initial guesses for the mixture model parameters. These values are then used to calculate the cluster probabilities for each instance. These probabilities are in turn used to re-estimate the parameters, and the process is repeated. A drawback of such algorithms is that they tend to be computationally expensive. Another problem found in the previous approach is called overfitting. This problem might be caused by two reasons. On one hand, a large number of clusters may be specified. And on the other, the distributions of probabilities have too many parameters. In this context, one possible solution is to adopt a fully Bayesian approach, in which every parameter has a prior probability distribution.

Hierarchical algorithms that create a hierarchical decomposition of the instances are covered in the following section.

### 3 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 afterwards (e.g. via a cut through the dendrogram).

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 [39], Clustering Using REpresentatives – CURE [18] and CHAMELEON [23].

BIRCH [39] uses a hierarchical data structure called CF-tree for partitioning the incoming data points in an incremental and dynamic way. CF-tree is

a height-balanced tree, which stores the clustering features and it is based on two parameters: *branching factor*  $B$  and *threshold*  $T$ , which refer to the diameter of a cluster (the diameter (or radius) of each cluster must be less than  $T$ ). A CF tree is built as the data is scanned. As each data point is encountered, the CF tree is traversed, starting from the root and choosing the closest node at each level. When the closest “leaf” cluster for the current data point is finally identified, a test is performed to see if adding the data item to the candidate cluster will result in a new cluster with a diameter greater than the given threshold,  $T$ . *BIRCH* can typically find a good clustering with a single scan of the data and improve the quality further with a few additional scans. It can also handle noise effectively. Moreover, because *BIRCH* is reasonably fast, it can be used as a more intelligent alternative to data sampling in order to improve the scalability of other clustering algorithms. However, *BIRCH* has one drawback: it may not work well when clusters are not “spherical” because it uses the concept of radius or diameter to control the boundary of a cluster. In addition, it is order-sensitive as it may generate different clusters for different orders of the same input data. Bubble and Bubble-FM [13] clustering algorithms are extensions of *BIRCH* to general metric spaces (categorical values in attributes).

In *CURE*, instead of using a single centroid to represent a cluster, a constant number of representative points are chosen to represent a cluster. The number of points chosen, is a parameter,  $c$ , but it was found that a value of 10 or more worked well. The similarity between two clusters is measured by the similarity of the closest pair of the representative points belonging to different clusters. Unlike centroid/medoid based methods, *CURE* is capable of finding clusters of arbitrary shapes (e.g. ellipsoidal, spiral, cylindrical, non-convex) and sizes, as it represents each cluster via multiple representative points. Shrinking the representative points towards the centroid helps *CURE* in avoiding the problem of noise present in the single link method. However, it cannot be applied directly to large data sets. For this reason, *CURE* takes a random sample and performs the hierarchical clustering on the sampled data points.

*ROCK* [17], is another clustering algorithm for categorical data using the Jaccard coefficient to measure similarity. It accepts as input the set  $S$  of  $n$  sampled points to be clustered (that are drawn randomly from the original data set), and the number of desired clusters  $k$ . *ROCK* samples the data set in the same manner as *CURE*.

*CHAMELEON* [23] finds the clusters in the data set by using a two-phase algorithm. In the first step it

generates a  $k$ -nearest neighbor graph [12] that contains links only between a point and its  $k$ -nearest neighbors. After, *CHAMELEON* uses a graph-partitioning algorithm to cluster the data items into a large number of relatively small sub-clusters. During the second phase, it uses an agglomerative hierarchical clustering algorithm to find the genuine clusters by repeatedly combining together these sub-clusters. None cluster can contain less than a user specific number of instances.

More recently, a novel incremental hierarchical clustering algorithm (*GRIN*) for numerical data sets based on gravity theory in physics is presented in [7]. One main factor that makes the *GRIN* algorithm able to deliver favorite clustering quality is that the optimal parameters settings in the *GRIN* algorithm are not sensitive to the distribution of the data set.

## 4 Density-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* [9]. *DBSCAN* separate data points into three classes (Fig. 2):

- 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.

An incremental version of *DBSCAN* (incremental *DBSCAN*) is presented in [10]. It was proven that this incremental algorithm yields the same result as *DBSCAN*. In addition, another clustering algorithm (*GDBSCAN*) generalizing the density-based algorithm *DBSCAN* is presented in [31]. *GDBSCAN* can cluster point instances to both, their numerical and their categorical attributes. Moreover, in [37] the

PDBSCAN, a parallel version of DBSCAN is presented. Furthermore, DBCLASD (Distribution Based Clustering of Large Spatial Data sets) eliminates the need for MinPts and Eps parameters [38]. DBCLASD incrementally augments an initial cluster by its neighboring points as long as the nearest neighbor distance set of the resulting cluster still fits the expected distance distribution. While the distance set of the whole cluster might fit the expected distance distribution, this does not necessarily hold for all subsets of this cluster. Thus, the order of testing the candidates is crucial. In [2] a new algorithm (OPTICS) is introduced, which creates an *ordering* of the data set representing its density-based clustering structure. It is a versatile basis for interactive cluster analysis.

Another density-based algorithm is the DENCLUE [19]. The basic idea of DENCLUE is to model the overall point density analytically as the sum of influence functions of the data points. The influence function can be seen as a function, which describes the impact of a data point within its neighbourhood. Then, by determining the maximum of the overall density function can identify clusters. The algorithm allows a compact mathematical description of arbitrarily shaped clusters in high-dimensional data sets and is significantly faster than the other density based clustering algorithms. Moreover, DENCLUE produces good clustering results even when a large amount of noise is present. As in most other approaches, the quality of the resulting clustering depends on an adequate choice of the parameters. In this approach, there are two important parameters, namely  $\sigma$  and  $\xi$ . The parameter  $\sigma$  determines the influence of a point in its neighborhood and  $\xi$  describes whether a density-attractor is significant. Density-attractors are local maxima of the overall density function.

FDC algorithm (Fast Density-Based Clustering) is presented in [40] for density-based clustering defined by the density-linked relationship. The clustering in this algorithm is defined by an equivalence relationship on the objects in the database. The complexity of FDC is linear to the size of the database, which is much faster than that of the algorithm DBSCAN.

More recently, the algorithm SNN (Shared Nearest Neighbors) [8] blends a density based approach with the idea of ROCK. SNN sparsifies similarity matrix by only keeping K-nearest neighbors, and thus derives the total strength of links for each x.

## 5 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 [36], WaveCluster [33], and CLustering In QUEst – CLIQUE [1].

STING [36] 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 [1] 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, WaveCluster [33] 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.

## 6 Model based Methods

AutoClass [5] uses the Bayesian approach, starting from a random initialization of the parameters, incrementally adjusts them in an attempt to find their maximum likelihood estimates. Moreover, in [28] it is assumed that, in addition to the observed or predictive attributes, there is a hidden variable. This unobserved variable reflects the cluster membership for every case in the data set. Therefore, the data-clustering problem is also an example of supervised learning from incomplete data due to the existence of such a hidden variable [22]. Their approach for learning has been called RBMNs (Recursive Bayesian Multinets).

Another model based method is the SOM net [25]. The SOM net can be thought of as two layers neural network. Each neuron is represented by  $n$ -dimensional weight vector,  $m = (m_1, \dots, m_n)$ , where  $n$  is equal to the dimension of the input vectors. The neurons of the SOM are themselves cluster centers; but to accommodate interpretation the map units can be combined to form bigger clusters. The SOM is trained iteratively. In each training step, one sample vector  $x$  from the input data set is chosen randomly, and the distance between it and all the weight vectors of the SOM is calculated using a distance measure, e.g., Euclidean distance. After finding the Best-Matching Unit (the neuron whose weight vector is closest to the input vector), the weight vectors of the SOM are updated so that the Best-Matching Unit is moved closer to the input vector in the input space. The topological neighbors of the BMU are also treated in a similar way. An important property of the SOM is that it is very robust. The outlier can be easily detected from the map, since its distance in the input space from other units is large. The SOM can deal with missing data values, too.

Many applications require the clustering of large amounts of high dimensional data. However, most automated clustering techniques do not work effectively and/or efficiently on high dimensional data, i.e. they are likely to miss clusters with certain unexpected characteristics. There are various reasons for this. First, it is difficult to find the necessary parameters for tuning the clustering algorithms to the specific applications characteristics. Second, it is hard to verify and interpret the resulting high dimensional clusters and third, often the concept of clusters inspired from low dimensional cases cannot be extended to high dimensional cases. A solution could be instead of integrating all the requirements into a single algorithm, to try to build a combination of clustering algorithms (ensembles of clustering

algorithms).

## 7 Ensembles of Clustering Algorithms

The theoretical foundation of combining multiple clustering algorithms is still in its early stages. In fact, combining multiple clustering algorithms is a more challenging problem than combining multiple classifiers. In [29] the reason that impede the study of clustering combination has been identified as various clustering algorithms produce largely different results due to different clustering criteria, combining the clustering results directly with integration rules, such as sum, product, median and majority vote can not generate a good meaningful result.

Cluster ensembles can be formed in a number of different ways [34], such as (1) the use of a number of different clustering techniques (either deliberately or arbitrarily selected). (2) The use of a single technique many times with different initial conditions. (3) The use of different partial subsets of features or patterns. In [11] a split-and-merge strategy is followed. The first step is to decompose complex data into small, compact clusters. The K-means algorithm serves this purpose; an ensemble of clustering algorithms is produced by random initializations of cluster centroids. Data partitions present in these clusterings are mapped into a new similarity matrix between patterns, based on a voting mechanism. This matrix, which is independent of data sparseness, is then used to extract the natural clusters using the single link algorithm.

More recently, the idea of combining multiple different clustering algorithms of a set of data patterns based on a Weighted Shared nearest neighbors Graph WSnnG is introduced in [3].

Due to the increasing size of current databases, constructing efficient distributed clustering algorithms has attracted considerable attention. Distributed Clustering assumes that the objects to be clustered reside on different sites. Instead of transmitting all objects to a central site (also denoted as server) where we can apply standard clustering algorithms to analyze the data, the data are clustered independently on the different local sites also denoted as clients). In a subsequent step, the central site tries to establish a global clustering based on the local models, i.e. the representatives. Generally, as far as distributed clustering is concerned, there are different scenarios:

- Feature-Distributed Clustering (FDC), consists in combining a set of clusterings obtained from clustering algorithm having partial view of the data features.

- Object-Distributed Clustering (ODC), consists in combining clusterings obtained from clustering algorithm that have access to the whole set of data features and to a limited number of objects.
- Feature/Object-Distributed Clustering (FODC), consists in combining clusterings obtained from clustering algorithm having access to limited number of objects and/or features of the data.

## 8 Conclusion

We should remark that the list of references is not a comprehensive list of papers discussing unsupervised methods: our aim was to produce a critical review of the key ideas, rather than a simple list of all publications which had discussed or made use of those ideas. Despite this, we hope that the references cited cover the major theoretical issues, and provide routes into the main branches of the literature dealing with such methods.

Generally, we will say that partitioning algorithms typically represent clusters by a prototype. An iterative control strategy is used to optimize the whole clustering such that, e.g., the average or squared distances of instances to its prototypes are minimized. Consequently, these clustering algorithms are effective in determining a good clustering if the clusters are of convex shape, similar size and density, and if the number of clusters can be reasonably estimated.

In general, the disability to identify the appropriate number of clusters is one of the most fundamental shortcomings of non-hierarchical techniques [15]. Hierarchical clustering algorithms decompose the data set into several levels of partitioning which are usually represented by a dendrogram – a tree which splits the data set recursively into smaller subsets. Although hierarchical clustering algorithms can be very effective in knowledge discovery, the cost of creating the dendrograms is prohibitively expensive for large data sets.

Density-based approaches apply a local cluster criterion and are very popular for the purpose of data set mining. Clusters are regarded as regions in the data space where the instances are dense, and they are separated by regions of low instance density (noise). These regions may have an arbitrary shape and the points inside a region may be arbitrarily distributed. A performance comparison [38] shows that DBSCAN is slightly faster than DBCLASD and both, DBSCAN and DBCLASD are much faster than

hierarchical clustering algorithms and partitioning algorithms.

Generally, grid-based clustering algorithms first separate 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.

A solution for better results could be instead of integrating all the requirements into a single algorithm, to try to build a combination of clustering algorithms. However, the theoretical foundation of combining multiple clustering algorithms is still in its early stages and thus more work is needed in this direction. In addition, one can also study the impact of the coordinated sub-sampling strategies on the performance and quality of object distributed clustering. The question is to determine what types of overlap and object ownership structures lend themselves particularly well for knowledge reuse.

## References:

- [1] Agrawal R., Gehrke J., Gunopulos D. and Raghavan P. (1998). Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications. In Proc. of the 1998 ACM-SIGMOD Conf. On the Management of Data, 94-105.
- [2] Ankerst M., Breunig M., Kriegel H., Sander J., OPTICS: Ordering Points to Identify the Clustering Structure, Proc. ACM SIGMOD'99 Int. Conf. on Management of Data.
- [3] H. Ayad and M. Kamel. Finding natural clusters using multi-clusterer combiner based on shared nearest neighbors. In Multiple Classifier Systems: Fourth International Workshop, MCS 2003, Guildford, Surrey, United Kingdom, June 11–13.
- [4] M. Breunig, H.-P. Kriegel, R. Ng, and J. Sander. Lof: Identifying density-based local outliers. In Proc. of SIGMOD'2000, pages 93–104, 2000.
- [5] Cheeseman P. & Stutz J., (1996), Bayesian Classification (AutoClass): Theory and Results, In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smith, and R. Uthurusamy, editors, Advances in Knowledge Discovery and Data Mining, pages 153-180, AAAI/MIT Press.
- [6] Yiu-Ming Cheung, k\*-Means: A new generalized k-means clustering algorithm, Pattern Recognition Letters 24 (2003) 2883–2893.
- [7] Chien-Yu Chen, Shien-Ching Hwang, and Yen-Jen Oyang, An Incremental Hierarchical

- Data Clustering Algorithm Based on Gravity Theory gravity theory, *Advances in Knowledge Discovery and Data Mining: 6th Pacific-Asia Conference, PAKDD 2002*, Taipei, Taiwan, May 6-8, 2002. Springer-Verlag LNCS 2336.
- [8] L. Ertöz, M. Steinbach, and V. Kumar. Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data. In *Proceedings of Second SIAM International Conference on Data Mining*, San Francisco, CA, USA, May 2003.
  - [9] Ester, M., Kriegel, H.-P., Sander, J., and Xu X. (1996), A density-based algorithm for discovering clusters in large spatial data sets with noise. *Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining*. Portland, OR, pp. 226–231.
  - [10] Ester, M., Kriegel, H.-P., Sander, J., Wimmer M. and Xu X. (1998), Incremental Clustering for Mining in a Data Warehousing Environment, *Proceedings of the 24th VLDB Conference* New York, USA, 1998.
  - [11] A. Fred and A.K. Jain. Data clustering using evidence accumulation. In *Proceedings of the 16th International Conference on Pattern Recognition. ICPR 2002*, volume 4, pages 276–280, Quebec City, Quebec, Canada, August 2002.
  - [12] Gaede V. and Gunther O. (1998), Multidimensional Access Methods, *ACM Computing Surveys*, Vol. 30, No. 2.
  - [13] Ganti V., Ramakrishnan R., Gehrke J., Powell A., French J., (1999), Clustering Large Datasets in Arbitrary Metric Spaces, *ICDE 1999*, pp. 502–511.
  - [14] Goebel M., Gruenwald L. (1999), “A Survey Of Data Mining And Knowledge Discovery Software Tools”, *SIGKDD Explorations*, Vol. 1, No. 1, P. 20–33, June 1999.
  - [15] Gordon, A.D. (1998), How many clusters? An investigation of five procedures for detecting nested cluster structure. In, *Data Science, Classification, and Related Methods*, edited by C. Hayashi, N. Ohsumi, K. Yajima, Y. Tanaka, H. Bock, and Y. Baba. Tokyo: Springer-Verlag.
  - [16] Jerzy W. Grzymala-Busse, Ming Hu, A Comparison of Several Approaches to Missing Attribute Values in Data Mining, *Rough Sets and Current Trends in Computing : Second International Conference, RSCTC 2000 Banff*, Canada.
  - [17] Guha, S, Rastogi, R., Shim K. (1999), "ROCK: A Robust Clustering Algorithm for Categorical Attributes", In the *Proceedings of the IEEE Conference on Data Engineering*.
  - [18] Guha, S., Rastogi, R., Shim K. (1998), "CURE: An Efficient Clustering Algorithm for Large Data sets", Published in the *Proceedings of the ACM SIGMOD Conference*.
  - [19] Hinneburg A. and Keim D. (1998), An efficient approach to clustering in large multimedia data sets with noise, In *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining*, pages 58–65 (1998).
  - [20] Huang Z. (1998), Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values, *Data Mining and Knowledge Discovery* 2, 283–304 (1998).
  - [21] Jain A.K., Murty M.N., Flynn, P.J. (1999), “Data Clustering: A Review”, *ACM Computing Surveys*, Vol.31, No3.
  - [22] Jensen F. (1996), *An Introduction to Bayesian Networks*. Springer.
  - [23] Karypis G., Han E. H. and Kumar V. (1999), CHAMELEON: A hierarchical clustering algorithm using dynamic modeling, *Computer* 32(8): 68–75, 1999.
  - [24] Katayama N. & Satoh S. (1997), The SR-tree: An Index Structure for High-Dimensional Nearest Neighbor Queries, *Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data*, Tucson, Arizona.
  - [25] Kohonen T. (1997), *Self-Organizing Maps*, Second Extended Edition, Springer Series in Information Sciences, Vol. 30, Springer, Berlin, Heidelberg, New York, 1995, 1997.
  - [26] McLachlan, G. J., & Krishnan, T. (1997), *The EM algorithm and extensions*, JohnWiley & Sons.
  - [27] Ng, R.T. and Han, J. (1994), Efficient and effective clustering methods for spatial data mining. *Proc. 20th Int. Conf. on Very Large Data Bases*. Santiago, Chile, pp. 144–155.
  - [28] Pena J., Lozano J., Larranaga P. (2002), Learning Recursive Bayesian Multinets for Data Clustering by Means of Constructive Induction, *Machine Learning*, 47, 63–89, 2002.
  - [29] Y. Qian and C. Suen. Clustering combination method. In *International Conference on Pattern Recognition. ICPR 2000*, volume 2, pages 732–735, Barcelona, Spain, September 2000.
  - [30] S. Ramaswamy, R. Rastogi, and K. Shim. Efficient algorithms for mining outliers from large data sets. In *Proc. of SIGMOD’2000*, pages 427–438, 2000.



- [31] Sander J. O., Martin Ester, Hans-Peter Kriegel, Xiaowei Xu (1998), Density-Based Clustering in Spatial Data sets: The Algorithm GDBSCAN and Its Applications, *Data Mining and Knowledge Discovery* 2, 169–194 (1998), Kluwer Academic Publishers.
- [32] Sato M., Sato Y., Jain L. (1997), *Fuzzy Clustering Models and Applications (Studies in Fuzziness and Soft Computing Vol. 9)*, ISBN: 3790810266
- [33] Sheikholeslami, C., Chatterjee, S., Zhang, A. (1998), "WaveCluster: A-MultiResolution Clustering Approach for Very Large Spatial Data set". *Proc. of 24th VLDB Conference*.
- [34] Strehl and J. Ghosh. Cluster ensembles – a knowledge reuse framework for combining partitionings. In *Conference on Artificial Intelligence (AAAI 2002)*, pages 93–98, Edmonton, July 2002. AAAI/MIT Press.
- [35] Talavera L. (1999), Dependency-based dimensionality reduction for clustering symbolic data. In *Proceedings of the Workshop on Pre- and Post-Processing in Machine Learning and Data Mining, Advanced Course on Artificial Intelligence (ACAI '99)*.
- [36] Wang W., Yang J. and Muntz.R. (1997), STING: A Statistical Information Grid Approach to Spatial Data Mining, *Proceedings of the 23rd VLDB Conference Athens, Greece, 1997*.
- [37] Xu X., Jager J., Hans-Peter Kriegel (1999), A Fast Parallel Clustering Algorithm for Large Spatial Data sets, *Data Mining and Knowledge Discovery*, 3, 263–290 (1999).
- [38] Xu X., Ester M., Kriegel H.-P., Sander J. (1998), A Nonparametric Clustering Algorithm for Knowledge Discovery in Large Spatial Data sets, *Proc. IEEE Int. Conf. on Data Engineering*, IEEE Computer Society Press, 1998.
- [39] Zhang, T., Ramakrishnan, R., and Linvy, M. (1997), BIRCH: An efficient data clustering method for very large data sets. *Data Mining and Knowledge Discovery*, 1(2): 141–182.
- [40] Bo Zhou, David W. Cheung, and Ben Kao, A Fast Algorithm for Density-Based Clustering in Large Database, *PAKDD'99, LNAI 1574*, pp. 338-349, 1999.