Data Clustering

Yanchang Zhao
University of Technology, Sydney, Australia
Phone: +61 2 6155 1550
E-mail: yczhao@it.uts.edu.au
Postal address: PO Box 123, Broadway, NSW 2007, Australia

Longbing Cao
University of Technology, Sydney, Australia
Phone: +61 2 9514 4477
E-mail: lbcao@it.uts.edu.au
Postal address: PO Box 123, Broadway, NSW 2007, Australia

Huafeng Zhang
University of Technology, Sydney, Australia
Phone: +61 2 6155 1562
E-mail: hfzhang@it.uts.edu.au
Postal address: PO Box 123, Broadway, NSW 2007, Australia

Chengqi Zhang
University of Technology, Sydney, Australia
Phone: +61 2 9514 7941
E-mail: chengqi@it.uts.edu.au
Postal address: PO Box 123, Broadway, NSW 2007, Australia
Data Clustering

Yanchang Zhao†, Longbing Cao, Huaiyang Zhang, Chengqi Zhang

University of Technology, Sydney, Australia

PO Box 123, Broadway, NSW 2007, Australia

{yczao, lbcao, hfzhang, chengqi}@it.uts.edu.au

INTRODUCTION

Clustering is one of the most important techniques in data mining. This chapter presents a survey of popular approaches for data clustering, including well-known clustering techniques, such as partitioning clustering, hierarchical clustering, density-based clustering and grid-based clustering, and recent advances in clustering, such as subspace clustering, text clustering and data stream clustering. The major challenges and future trends of data clustering will also be introduced in this chapter.

The remainder of this chapter is organized as follows. The background of data clustering will be introduced in Section 2, including the definition of clustering, categories of clustering techniques, features of good clustering algorithms, and the validation of clustering. Section 3 will present main approaches for clustering, which ranges from the classic partitioning and hierarchical clustering to recent approaches of bi-clustering and semi-supervised clustering. Challenges and future trends will be discussed in Section 4, followed by the conclusions in the last section.

† This work was supported by the Australian Research Council (ARC) Linkage Project LP0775041 and Discovery Projects DP0449535, DP0667060 & DP0773412, and by the Early Career Researcher Grant from University of Technology, Sydney, Australia.

† Corresponding author.
BACKGROUND

Data clustering is sourced from pattern recognition (Theodoridis & Koutroumbas, 2006), machine learning (Alpaydin, 2004), statistics (Hill & Lewicki, 2007) and database technology (Date, 2003). *Data clustering* is to partition data into groups, where the data in the same group are similar to one another and the data from different groups are dissimilar (Han & Kamber, 2000). More specifically, it is to segment data into clusters so that the intra-cluster similarity is maximized and that the inter-cluster similarity is minimized. The groups obtained are a partition of data, which can be used for customer segmentation, document categorization, etc.

Clustering techniques can be “clustered” into groups in multiple ways. In terms of the membership of objects, there are two kinds of clustering, *fuzzy clustering* and *hard clustering*. Fuzzy clustering is also known as *soft clustering*, where an object can be in more than one cluster, but with different membership degree. In contrast, an object in hard clustering can belong to one cluster only. Generally speaking, clustering is referred to as hard clustering implicitly. In terms of approaches, data clustering techniques can be classified into the following groups: partitioning clustering, hierarchical clustering, density-based clustering, grid-based clustering and subspace clustering. In terms of the type of data, there are spatial data clustering, text clustering, multimedia clustering, time series clustering, data stream clustering and graph clustering.

For a good clustering algorithm, it is supposed to have the following features: 1) the ability to detect clusters with various shapes and different distributions; 2) the capability of finding clusters with considerably different sizes; 3) the ability to work when outliers are present; 4) no or few parameters needed as input; and 5) scalability to both the size and the dimensionality of data.
How to evaluate the results is another important problem for clustering. For the validation of clustering results, there are many different measures, such as Compactness (Zait & Messatfa, 1997), Conditional Entropy (CE) and Normalized Mutual Information (NMI) (Strehl & Ghosh, 2002; Fern & Brodley, 2003). The validation measures can be classified into three categories, 1) internal validation, such as Compactness, Dunn’s validation index, Silhouette index and Hubert’s correlation with distance matrix, which is based on calculating the properties of result clusters, 2) relative validation, such as Figure of merit and Stability, which is based on comparisons of partitions, and 3) external validation, such as CE, NMI, Hubert’s correlation, Rand statistics, Jaccard coefficient, and Folkes and Mallows index, which is based on comparing with a known true partition of data (Halkidi et al., 2001, Brun et al., 2007).

**DATA CLUSTERING TECHNIQUES**

The popular clustering techniques will be briefly presented in this section. More detailed introduction and comparison of various clustering techniques can be found in books on data mining and survey papers on clustering (Berkhin, 2002; Grabmeier & Rudolph, 2002; Han & Kamber, 2000; Jain, Murty, & Flynn, 1999; Kolatch, 2001; Xu & Wunsch, 2005; Zait & Messatfa, 1997).

**Partitioning Clustering**

The idea of partitioning clustering is to partition the data into $k$ groups first and then try to improve the quality of clustering by moving objects from one group to another. A typical method of partitioning clustering is $k$-means (Alsabti, Ranka, & Singh, 1998; Macqueen, 1967), which randomly selects $k$ objects as cluster centers and assigns other objects to the nearest cluster centers, and then improves the clustering by iteratively
updating the cluster centers and reassigning the objects to the new centers. *k-medoids* (Huang, 1998) is a variation of *k*-means for categorical data, where the medoid (i.e., the object closest to the center), instead of the centroid, is used to represent a cluster. Some other partitioning methods are PAM and CLARA proposed by Kaufman & Rousseeuw (1990) and CLARANS by Ng and Han (1994).

The disadvantage of partitioning clustering is that the result of clustering is dependent on the selection of initial cluster centers and it may result in a local optimum instead of a global one. A simple way to improve the chance of obtaining the global optimum is to run *k*-means multiple times with different initial centers and then choose the best clustering result as output. Another disadvantage of *k*-means is that it tends to result in sphere-shaped clusters with similar sizes. Moreover, how to choose a value for *k* also remains as a non-trivial question.

**Hierarchical Clustering**

With *hierarchical clustering* approach, a hierarchical decomposition of data is built in either bottom-up (agglomerative) or top-down (divisive) way (see Figure 1). Generally a dendrogram is generated and a user may select to cut it at a certain level to get the clusters. With *agglomerative clustering*, every single object is taken as a cluster and then iteratively the two nearest clusters are merged to build bigger clusters until the expected number of clusters is obtained or when only one cluster is left. AGENS is a typical agglomerative clustering algorithm (Kaufman & Rousseeuw, 1990). *Divisive clustering* works in an opposite way, which puts all objects in a single cluster and then divides the cluster into smaller and smaller ones. An example of divisive clustering is DIANA (Kaufman & Rousseeuw, 1990). Some other popular hierarchical clustering algorithms are BIRCH (Zhang, Ramakrishnan, & Livny, 1996), CURE (Guha, Rastogi, & Shim,
In hierarchical clustering, there are four different methods to measure the distance between clusters: centroid distance, average distance, single-link distance and complete-link distance. *Centroid distance* is the distance between the centroids of two clusters. *Average distance* is the average of the distances between every pair of objects from two clusters. *Single-link distance*, also known as *minimum distance*, is the distance between the two nearest objects from two clusters. *Complete-link distance*, also referred to as *maximum distance*, is the distance between the two objects which are the farthest from each other from two clusters.

![Figure 1. Hierarchical Clustering](image)

**Density-Based Clustering**

The rationale of *density-based clustering* is that a cluster is composed of well-connected dense regions. DBSCAN is a typical density-based clustering algorithm, which works by expanding clusters to their dense neighborhood (Ester, Kriegel, Sander, & Xu, 1996).
Although a user is not required to guess the number of clusters before clustering, he has to provide two other parameters, the radius of neighborhood and the density threshold, to run DBSCAN. AGRID (Zhao & Song, 2003) is an efficient density-based algorithm in that it uses grid to reduce the complexity of distance computation and cluster merging. By partitioning the data space into cells, only neighboring cells are taken into account when computing density and merging clusters. Some other density-based clustering techniques are OPTICS (Ankerst, Breunig, Kriegel, & Sander, 1999) and DENCLUE (Hinneburg & Keim, 1998). The advantage of density-based clustering is that it can filter out noise and find clusters of arbitrary shapes (as long as they are composed of connected dense regions). However, most density-based approaches utilize the indexing techniques for efficient neighborhood inquiry, such as R-tree and R*-tree, which do not scalable well to high-dimensional space.

**Grid-Based Clustering**

*Grid-based clustering* works by partitioning the data space into cells with grid and then merging them to build clusters. Some grid-based methods, such as STING, WaveCluster and CLIQUE, use regular grids to partition data space, while some employ adaptive or irregular grids, such as adaptive grid (Goil, Nagesh, & Choudhary, 1999) and optimal grid (Hinneburg & Keim, 1999).

STING (Wang, Yang, & Muntz, 1997) is a grid-base multi-resolution clustering technique in which the spatial area is divided into rectangular cells and organized into a statistical information cell hierarchy. Statistical information, such as the count, mean, minimum, maximum, standard deviation and distribution, is stored for each cell. Thus, the statistical information of cells is captured and clustering can be performed without recourse to the individual objects. WaveCluster (Sheikholeslami, Chatterjee, & Zhang,
1998) is proposed to look at the multi-dimensional data space from a signal processing perspective. The objects are taken as a $d$-dimensional signal, and the high frequency parts of the signal correspond to the boundary of clusters, where the distribution of objects changes rapidly. The low frequency parts with high amplitude correspond to clusters, where data are concentrated. CLIQUE (Agrawal, Gehrke, Gunopulos, & Raghavan, 1998) works like APRIORI (Agrawal & Srikant, 1994), a famous algorithm for association mining. It partitions each dimension into intervals and computes the dense units in all dimensions. Then the dense units are combined to generate the dense units in higher dimensions.

The advantage of grid-based clustering is that the processing time is very fast, because it is independent on the number of objects, but dependent on the number of cells. Its disadvantage is that the quality of clustering is dependent on the granularity of cells and that the number of cells increases exponentially with the dimensionality of data. Therefore, adaptive grid and optimal grid are designed to tackle the above problems. MAFIA (Goil et al., 1999) uses adaptive grids to partition a dimension depending on the distribution of data in the dimension, in contrast to partition every dimension evenly. OptiGrid (Hinneburg & Keim, 1999) uses contracting projections of data to determine the optimal cutting hyper-planes for data partitioning, where the grid is “arbitrary”, as compared with equidistant, axis-parallel grids.

**Model-Based Clustering**

*Model-based clustering* assumes that the data are generated by a mixture of probability distributions, and it attempts to learn statistical probability models from data, with each model representing one particular cluster (Zhong & Ghosh, 2003). The model type is often set as Gaussian or hidden Markov models (HMMs), and then the model structure is
determined by model selection techniques and parameters are estimated using maximum likelihood algorithms. Some examples of model-based clustering methods are DMBC (Distributed Model-Based Clustering) (Kriegel et al., 2005), and model-based clustering based on data summarization (bEMADS and gEMADS) (Jin et al., 2005). Another kind of model-based clustering is neural network approaches, such as SOM (self-organizing feature maps) (Kohonen, 1988).

**Fuzzy Clustering**

Generally speaking, an object is either a member of a cluster or not a member of it. *Fuzzy clustering* is an extension by allowing an object to be a member of more than one cluster. For example, an object is the member of three clusters with membership as 0.5, 0.3 and 0.2, respectively. Fuzzy clustering is also known as *soft clustering*, as compared with *hard clustering* where the membership can be either 1 or 0 only. Two examples of fuzzy clustering are fuzzy k-means (Karayiannis, 1995) and *EM (Expectation Maximization) algorithm* (Dempster, Laird, & Rubin, 1977). EM algorithm generates fuzzy clustering in two steps. The *expectation* (E) step calculates for each object the expectation of probabilities in each cluster, and then the *maximization* (M) step computes the distribution parameters of clusters, i.e., maximizing the likelihood of the distributions given the data. The two steps are repeated to improve the clustering, until the improvement (say, the increase in log-likelihood) becomes negligible. EM algorithms may result in a local maximum instead of the global maximum. Similar to k-means, the chance to get the global maximum can be improved by running EM for multiple times with different initial guesses and then choosing the best clustering.
**Subspace Clustering**

In some real-world applications, the dimensionality of data is in hundreds or even thousands, and more often than not, no meaningful clusters can be found in the full dimensional space. Therefore, *subspace clustering* is proposed for high dimensional data, where two clusters can be in two different subspaces and the subspaces can be of different dimensionalities. Well-known subspace clustering algorithms are CLIQUE (Agrawal et al., 1998), MAFIA (Goil et al., 1999), Random Projection (Fern & Brodley, 2003), Projected clustering (Aggarwal, Wolf, Yu, Procopiuc, & Park, 1999), Monte-Carlo (Procopiuc, Jones, Agarwal, & Murali, 2002), and Projective clustering (E. K. K. Ng, Fu, & Wong, 2005). With most techniques for subspace clustering, a cluster is defined as an axis-parallel hyper-rectangle in a subspace.

A technique for subspace clustering proposed by Fern and Brodley (2003) is to find the subspaces in a random projection and ensemble way. The dataset is first projected into random subspaces, and then EM algorithm is used to discover clusters in the projected dataset. The algorithm generates several groups of clusters with the above method and then combines them into a similarity matrix, from which the final clusters are discovered with an agglomerative clustering method.

MAFIA (Goil et al., 1999) is an efficient algorithm for subspace clustering using a density and grid based approach. It uses adaptive grids to partition a dimension depending on the distribution of data in the dimension. The bins and cells that have low density of data are pruned to reduce computation. The boundaries of the bins are not rigid, which improves the quality of clustering.
**Bi-Clustering**

*Bi-clustering*, also known as *co-clustering* or *two-way clustering*, is to group objects for a subset of attributes by performing simultaneous clustering of both rows and columns (Cheng & Church, 2000; Madeira & Oliveira, 2004). It is a kind of subspace clustering. Bi-clustering is widely used for clustering microarray data to analyze the activities of genes under many different conditions. Microarray data can be viewed as a matrix, where each row represents a gene, each column stands for a condition and each entry gives the expression level of a gene under a condition. From microarray data, four major types of biclusters are to discover: 1) biclusters with constant values, 2) biclusters with constant values on rows or columns, 3) biclusters with coherent values, and 4) biclusters with coherent evolutions (Madeira & Oliveira, 2004).

**Text Clustering**

*Text clustering*, also referred to as *document clustering*, is to group documents based on the similarity in the terms used and is widely used for document categorization, information retrieval and web search engine (Beil, Ester, & Xu, 2002; Zhong & Ghosh, 2005). Most algorithms for text clustering are based on a vector space model, where each document is represented by a vector of frequencies of terms. At first, a bag of words is collected for each document by filtering tags, stemming and pruning. Then the similarity of two documents is measured by how many words they share in common, and the documents can be clustered into groups with one of the clustering algorithms introduced above. Some examples of text clustering algorithms are Suffix Tree Clustering (Zamir & Etzioni, 1998), FTC (Frequent Term-based Clustering) and HFTC (Hierarchical FTC) (Beil, Ester, & Xu, 2002).
**Data Stream Clustering**

As tradition clustering deals with static data, *data stream clustering* is to cluster data streams where new data arrive continuously, such as click-streams, retail transactions and stock prices (C. C. Aggarwal, J. Han, J. Wang, & P. S. Yu, 2003; Guha, Meyerson, Mishra, Motwani, & O'Callaghan, 2003). For *data stream* clustering, the clusters are adjusted dynamically according to new data, and emerging clusters are also detected. New data can come in two ways, either as new records, such as transaction data, or as new dimensions, such as stock price data and other time series data. In either way, new data can bring changes in clusters as time elapses. Aggarwal et al. (2003) proposed the concepts of pyramid time frame and micro-clustering to cluster evolving *data streams*. The statistical information of data is stored as micro-clusters, and the micro-clusters are stored at snapshots in time following a pyramidal pattern. The above are then used in an offline process to explore stream clustering over different horizons.

**Semi-Supervised Clustering**

Generally speaking, clustering is unsupervised learning. However, sometimes there is a small amount of knowledge which can be used to guide clustering, and such kind of clustering is referred to as *semi-supervised clustering*. The knowledge available is normally not enough for a supervised learning to classify the data. The knowledge can be either pairwise constraints, such as must-link and cannot-link, or class labels for some objects. Some examples of semi-supervised clustering techniques are COP-COBWEB (Constraint-Partitioning COBWEB) (Wagstaff & Cardie, 2000), CCL (Constrained Complete-Link) (Klein et al., 2002), MPC-KMeans (Metric Pairwise Constrained K-Means) (Basu et al., 2003), semi-supervised clustering with user feedback (Cohn et al.,
2003), and a probabilistic model for semi-supervised clustering (Basu et al., 2004).

**FUTURE TRENDS**

Data mining is confronted with larger volume of data, higher dimensionality, more complex data and new types of applications. The above are also challenges for clustering. More scalable algorithms are needed to clustering data in Gigabytes or even in Terabytes and of dimensionality in hundreds and even in thousands. In addition to scalability, the other problem introduced by high dimensionality is the meaningfulness of similarity, the definition of clusters and the meaning of clustering. Another challenge is from new types of data and more complex data, such as multimedia data, semi-structured/unstructured data and stream data. The visualization of clusters and the change/trend analysis of clusters is also a trend of future research. More challenges will also be brought by new applications of clustering, such as bioinformatics, astronomy and meteorology.

**CONCLUSION**

We have presented a survey of popular data clustering approaches, including both classic methods and recent advanced algorithms. The basic ideas of the approaches have been introduced and their characteristics analyzed. The techniques are designed for different applications and for different types of data, such as numerical data, categorical data, spatial data, text data and microarray data. The definitions of clusters in the algorithms are not always the same, and most of them favor certain types of clusters, such as sphere-shaped clusters, convex clusters and axis-parallel clusters. New definitions of clusters and novel techniques for clustering keep emerging as data mining is applied in new applications and in new fields.
BIBLIOGRAPHY


Ng, R. T., & Han, J. (1994). *Efficient and Effective Clustering Methods for Spatial Data Mining*. Paper presented at VLDB'94: the 20th International Conference on Very Large Data Bases, San Francisco, CA, USA.


International Conference on Very Large Data Bases, August 25-29, 1997, Athens, Greece.


**KEY TERMS**
Bi-clustering – Also known as co-clustering, it is to group objects for a subset of attributes by performing simultaneous clustering of both rows and columns.

Data Clustering – Data clustering is to partition data into groups, where the data in the same group are similar to one another and the data from different groups are different from one another.

Data Stream Clustering – It is to group continuously arriving data, instead of static data, into groups based on the similarity.

Density-Based Clustering – Density-based clustering takes densely populated regions as clusters, while objects in sparse areas are removed as noises.

Fuzzy Clustering – Also known as Soft Clustering. For fuzzy clustering, an object can be classified with fractional membership into multiple groups, in contrast to Hard Clustering where an object can be classified into one group only.

Grid-Based Clustering – It is to partition the whole space into cells with grids and then merge the cells to build clusters.

Hierarchical Clustering – It is to build a hierarchical decomposition of data in either bottom-up or top-down way. Generally a dendrogram is generated and a user may select to cut it at a certain level to get the clusters.
Model-Based clustering – Model-based clustering assumes that the data are generated by a mixture of probability distributions, and attempts to learn statistical probability models from data, with each model representing one particular cluster.

Partitioning Clustering – It is a clustering approach which uses centers to represent clusters and then improves the partitioning by moving objects from group to group.

Semi-Supervised Clustering – Semi-supervised clustering is a partly supervised clustering which is guided with a small amount of knowledge, such as pairwise constraints and class labels for some objects.

Subspace Clustering – It is to find clusters in subspaces, where two clusters may exist in two different subspaces and the subspaces may also have different dimensionalities.

Text Clustering – It is to group documents based on the similarity in their topics and text.