Evaluation of clustering methods for adaptive learning systems

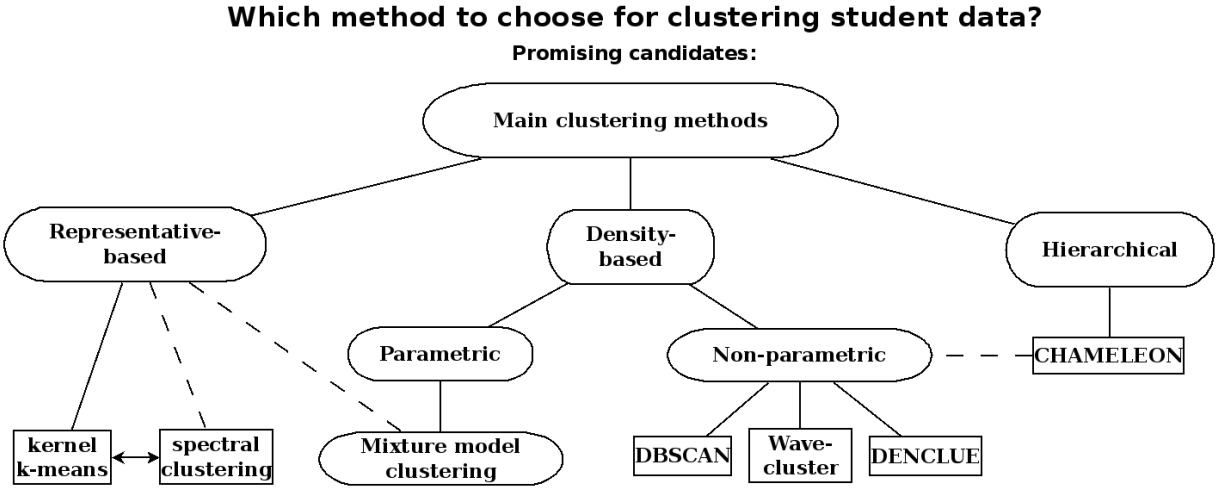
Wilhelmiina Hämäläinen
School of Computing, University of Eastern Finland, Finland

Ville Kumpulainen
School of Computing, University of Eastern Finland, Finland

Maxim Mozgovoy
School of Computer Science and Engineering, University of Aizu, Japan

Abstract

Clustering student data is a central task in the educational data mining and design of intelligent learning tools. The problem is that there are thousands of clustering algorithms but no general guidelines which method to choose. The optimal choice is of course problem- and data-dependent and can seldom be found without trying several methods. Still, the purposes of clustering students and the typical features of educational data make certain clustering methods more suitable or attractive. In this chapter, we evaluate the main clustering methods from this perspective. Based on our analysis, we suggest the most promising clustering methods for different situations.



Introduction

Clustering student data is a central task in the educational data mining and design of intelligent learning tools. Dividing data into natural groups gives a good summary how students are learning and helps to target teaching and tutoring. This is especially topical in the domain of online adaptive learning systems due to larger amount of students and their greater diversity. Clustering can also facilitate the design of predictive models, which are the heart of intelligent tutoring systems.

Indeed, a number of scholars report successful examples of clustering (for various purposes) in actual educational environments. However, the problem of selecting the most appropriate clustering method for student data is rarely addressed. There is a plenty of mainstream clustering methods and literally thousands of specialized clustering algorithms available (Jain, 2010), and choosing the right method for the given task is not easy. In practice, researchers often just pick up the most popular k-means method without a second thought whether its underlying assumptions suit the data. In practice, this means that one may end up with an artificial partition of data instead of finding natural clusters.

The aim of the present work is to evaluate a variety of clustering methods from the perspective of clustering student data. We analyze the main approaches to clustering and see how useful models they produce and how well their underlying assumptions fit typical student data. We do not try to list as many algorithms as possible, but instead our emphasis is to describe the underlying clustering principles and evaluate their properties. Our main goal is to cover those clustering methods which are generally available in the existing data mining and statistical analysis tools, but we introduce also some promising "future methods". Based on this analysis, we suggest the most promising clustering methods for different situations.

The rest of the chapter is organized as follows: First, we give the basic definitions, analyze domain-specific requirements for clustering methods, and survey related research. Then, we introduce the main approaches for clustering and evaluate their suitability for typical student data. Finally, we discuss future research directions and draw the final conclusions. The basic notations used in this chapter are introduced in Table 1.

Table 1. Basic notations.

Notation	**Meaning**
M	number of dimensions (variables)
N	number of data points
K	number of clusters
$\mathbf{p_i} = (p_{i1}, \ldots, p_{im})$	data point in m-dimensional data space
$D = \{\mathbf{p_1}, \ldots, \mathbf{p_n}\}$	data set of n points
C_i	cluster
$\mathbf{c_i}$	cluster centroid (representative point of cluster C_i)
$d(\mathbf{p_i}, \mathbf{p_j})$	distance between points $\mathbf{p_i}$ and $\mathbf{p_j}$
$D(C_i, C_j)$	distance between clusters C_i and C_j; inter-cluster distance
$\lvert C_i \rvert$	size of cluster C_i; number of points belonging to C_i

Background

In this section, we define the basic concepts related to clustering, discuss the goals and special requirements of clustering educational data, and survey related research.

Basic definitions

The main problem of clustering is how to define a cluster. There is no universally accepted precise definition of clustering. Intuitively, clustering means a grouping of data points, where points in one group are similar or close to each other but different or distant from points in the other groups. One may also describe clusters as denser regions of the data space separated by sparser regions or as homogeneous subgroups in a heterogeneous population. Here, we give only a very generic definition of clustering and then describe its different aspects.

Definition 1 (Clustering). Let $D = \{\mathbf{p_1}, \ldots, \mathbf{p_n}\}$ be a data set of n points, $\mathcal{C} = \{C_1, \ldots, C_k\}$ a set of k clusters, and M some clustering criterion. A *hard clustering* assigns each point $\mathbf{p_j}$ into exactly one cluster C_i according to M. A *soft clustering* defines for each point-cluster pair $(\mathbf{p_j}, C_i)$ a degree of membership according to M.

The above definition emphasizes one important aspect of clustering methods, their "hardness" or "softness". Hard methods produce always separate clusters, even if cluster boundaries were not clear, while soft methods allow overlapping clusters. Usually, the degree of membership is defined as the probability of point $\mathbf{p_j}$ to belong into cluster C_i, but fuzzy membership values are also used.

A related issue is how the clusters are represented. One approach is to define only cluster *centroids* (representative points, typically mean vectors) and use the distance from centroids to assign other points into clusters. If cluster probabilities are required, then other parameters like variance are needed to define cluster density functions. Alternatively, clusters can be represented by their boundaries or simply by listing their members. In *conceptual clustering* clusters are described by logical statements, like $(X \leq 5) \wedge (Y = 1)$.

Another aspect of cluster representation is whether the clustering is *flat* ("partitional" by Jain and Dubes (1988, Ch.3)) or *hierarchical*. A flat clustering is a single partition of data points, while a hierarchical clustering represents a sequence of nested clusterings (partition of larger clusters into subclusters).

The most important aspect of clustering methods is the clustering criterion (the *inductive principle* by Estivill-Castro (2002)). While the representational aspects describe *what kind of clustering models are possible*, the clustering criterion defines *which model fits the data best*. Usually, the clustering criterion is defined in the terms of a *similarity measure*, which tells how close or similar two points are. Given a similarity measure, one can also define inter-cluster similarity and express a clustering criterion that tries to maximize intra-cluster similarity and/or minimize inter-cluster similarity. If the clustering criterion is expressed explicitly, as a mathematical optimization problem (like a goodness function to be maximized), then the clustering problem can be tackled with different search algorithms. However, often the clustering criterion is expressed only implicitly, like rules for merging or splitting clusters based on similarity. In this case, the clustering criterion is usually bound to a certain search algorithm, which complicates an analytic comparison of clustering methods.

In the following, we discuss only the most commonly used similarity measures. For a comprehensive overview of different measures we refer to (Gan et al., 2007).

For numeric data, the most common choice for the similarity measure is the L_p metric or *Minkowski distance*:

$$L_p(\mathbf{p_i}, \mathbf{p_j}) = \left(\sum_{l=1}^{m} (p_{il} - p_{jl})^p \right)^{1/p},$$

where $p \in \mathbb{R}^+$ is a user-given parameter and $\mathbf{p_i} = (p_{i1}, ..., p_{im})$ and $\mathbf{p_j} = (p_{j1}, ..., p_{jm})$ are two (m-dimensional) data points. The best known Minkowski measures are *Euclidean distance* L_2 and *Manhattan distance* L_1. For high dimensional data (at least when $m \geq 20$), L_1 and fractional L_p metrics ($p < 1$) are recommended, because they offer better contrast between distances (Aggarwal et al., 2001). They are also more robust measures for data containing several outliers (Agrawal et al., 1998). The reason is that with small p the distance distribution becomes more homogeneous and clusters can be better separated.

Generally, L_p metrics work well, if the clusters are compact and well-separated, but they fail, if the variables are in different scales. This is often the case with educational data, where variables can measure very different things like age, exercise points, or average studying time per week. As a solution, the data should be standardized to one norm or some variables should be weighed. If the variables are strongly correlated, then one should first remove correlations (by principal component analysis) or use *Mahalanobis metric*, which takes the dependencies into account (Jain et al., 1999). Detecting and

removing irrelevant variables is also important, because they can dominate the distance measure and distort the results.

For categorical data, alternative similarity measures have been proposed (see e.g. Kantardzic (2011, Ch.9.2)). Two common choices are the *overlap metric* that is simply the number of common variable values which two data points share and *mutual neighbourhood distance* (*MND*), which reflects how close neighbours two points are to each other.

String values are a special case of categorical data and several distance measures have been developed for measuring distance or similarity between two strings. *Hamming distance* is a special case of the overlap metric, which calculates the number of character positions, where the two strings differ. Another popular measure is *minimum edit distance* that defines the minimum number of edit operations (e.g., insertion, deletion, substitution) needed to transform one string to another.

Mixed data, containing both numerical and categorical variables, is very tricky for clustering, because different types of variables are not comparable. Often, the easiest solution is to discretize numerical variables and use similarity measures for categorical data. Similarity measures for mixed data have also been developed (e.g., (Wilson and Martinez, 1997; Cheng et al., 2004; Ichino and Yaguchi, 1994)), but they are seldom available in general clustering tools.

Domain-specific requirements for clustering methods

Our main objective is to evaluate different clustering methods in the educational context, especially in distance learning. For this purpose, we should know the special requirements of the domain. These are imposed by two factors: the goals of clustering students and the characteristics of typical data.

According to our literature survey, there are two main reasons to cluster student data. The first reason is purely descriptive: to understand the data, to see if the students fall into subgroups with different characteristics and how this affects learning. This information can reveal successful learning patterns (Käser et al., 2013) or effective ways of using learning tools (Perera et al., 2009), but it can also be used for allocating students into different teaching groups (Huikkola et al., 2008) or targetting tutoring (Schmitt et al., 2007). In this context, outliers which do not belong to any cluster, are also interesting. They represent somehow exceptional students who may need extra concern, like individualized teaching or more challenges.

The second reason is to facilitate construction of predictive models for intelligent tutoring systems or other adaptive learning tools. In the intelligent tools, the main problem is usually classification — the tool should first classify the student and/or the current situation before it can select an optimal action (Hämäläinen and Vinni, 2010). Clustering is often a useful preprocessing step for the classifier construction. It can be used to select representative features which separate classes well (such that clusters have homogeneous class distributions), to select natural classes, or even as an initialization for a K-nearest neighbour style classifiers (Lopez et al., 2012; Käser et al., 2013). Cluster-based linear regression models are also possible (Trivedi et al., 2011a).

Common to these goals is that clustering is used to find a natural grouping of data, irrespective of cluster sizes or shapes. This is in contrast to many applications, where clustering is actually used for segmentation, i.e., partitioning the data in some convenient way (Hand et al., 2002, Ch.9.3). In segmentation, partitioning into balanced hyperspherical clusters may well be convenient, even if it would not capture the real clusters. For this reason, methods which work well in other domains (when used for segmentation) may fail in educational data mining.

In student data, the clusters are often unclear, arbitrary-shaped, or overlapping. Therefore, the clustering method should also be flexible. For overlapping clusters, a soft clustering would be the most natural. It could also help to select optimal actions similarly to probabilistic classifiers (see e.g. (Hämäläinen and Vinni, 2010)).

Characteristics of typical data have a strong impact on the selection of clustering method. In this chapter, our main focus is clustering student data (demographic and performance data). The following characterization of typical student data (available in distance learning and adaptive learning systems) is based on a meta-analysis in the previous research (Hämäläinen and Vinni, 2010) and confirmed by our own and other researchers' experiences (Perera et al., 2009). We list six characteristics and the requirements they impose on the clustering methods.

First, the typical data sets are quite small, usually only 100-300 rows or even less. The reason is that data sets are usually from one course during one semester and thus restricted by the number of students. Sometimes, it is possible to pool data from several years, if the recorded variables (like exercise tasks and their assessment) have remained unchanged. Still, it is unlikely that the data size would exceed 1000 rows. In practice, this means that one can select computationally demanding (and often better quality) methods for clustering.

Second, the data usually consists of both categorical and numerical variables. Numerical variables are typically discrete, like task scores, which can obtain only a few possible values. Continuous numerical variables are rarer. One reason is that physical devices, which are the main source of continuous variables, are seldom used to gather data for educational systems. Binary variables, like gender, whether the student has certain preliminary knowledge or skills, is working aside studies, etc. are relatively common. Answers to multiple choice queries are often recorded as nominal or binary variables (correct or incorrect), but sometimes they can also be ordinal (like a selfassessment of programming skills into excellent, good, satisfactory, or poor). Ideally, the clustering method should be able to handle mixed data, but, in practice, one may have to transform all variables to categorical or cluster only numerical variables.

Third, the number of variables can be very large in proportion to the number of rows (even > 50 variables), if it contains all available demographic features and comprehensive questionnaire data, including scores, error types, habits, preferences, and measures for student activity. However, it is hard to give reliable estimates based on previous research articles, because most of them report only a handful (5-10) variables which they have used for final modelling. Anyway, it seems that the original data can be quite sparse, and some feature selection or combining is required before clustering.

Fourth, the numeric variables are seldom normally distributed. In fact, the distribution may be even valley-shaped instead of hill-shaped. This means that one should be cautious when using methods which assume normality or other fixed distribution families.

Fifth, student data contains often a relatively large number of outliers (exceptional students). Therefore, the clustering method should be robust to outliers and preferably help to detect them.

Sixth, the variables tend to be mutually dependent. This means that the data is not uniformly distributed and we are likely to find clusters. However, some methods (or similarity measures, like L_p metrics) do not allow correlations. If one wants to use such methods, then the correlations (or dependencies, in general) should be removed before clustering.

Related research

There are many good survey papers which have evaluated and compared different clustering methods and algorithms. Jain et al. (1999) give a very comprehensive survey of the whole clustering process, including similarity measures, clustering methods, applications, and even practical tips. Different clustering criteria and methods are well evaluated in the papers by Berkhin (2006) and Halkidi et al. (2001), although the latter concentrates more on validation. Ghosh (2004) gives an overview of different clustering approaches. The paper by Estivill-Castro (2002) discusses different aspects of clustering methods (models, clustering criteria, and algorithms) and compares some wellknown clustering criteria. All these papers compare only the traditional clustering methods. The survey by Xu and Wunsch (2005) covers also newer methods, like neural network- and kernel-based methods. It gives a comprehensive

overview of the most important algorithms, but it can be hard to see which algorithms actually implement the same clustering criteria. In addition, there are survey papers on certain specific topics, like subspace clustering (Parsons et al., 2004) or text clustering (Aggarwal and Zhai, 2012).

In addition, there are domain-specific papers which have either compared different clustering methods empirically or evaluated their suitability for the typical data and problems of the domain. This kind of papers are especially popular in bioinformatics (e.g., (Yona et al., 2009; Andreopoulos et al., 2009)). However, they are of little use in the educational domain, where the clustering purposes and data sets are very different.

In the educational domain, we have not been able to find any evaluations or comparisons between different clustering methods. Still, many interesting use cases and applications have been reported. In most of them, the researchers have just picked up one clustering method (typically the *k*-means algorithm). In this sense, the work by Lopez et al. (2012) is exceptional, because several algorithms were actually compared. In addition, the researchers constructed a successful cluster-based classifier for predicting course outcomes. Trivedi et al. (2011a) went even further and developed a new prediction method (combining several cluster-based linear regression models) for predicting course grades. Pardos et al. (2012) applied the same model for knowledge tracing. Nugent et al. (2010) modified the *k*-means clustering algorithm for clustering students according to their skill profiles. The work by Huikkola et al. (2008) is a good example of careful feature extraction and selection when clustering large-dimensional questionnaire data. There is also an excellent overview on spectral clustering with some discussion how it could be applied in educational data mining (Trivedi et al., 2011b).

Evaluation of the main clustering methods

In this section, we introduce the main approaches for clustering and evaluate their suitability for typical educational (student) data. The number of different clustering methods is so enormous that it is impossible to cover all of them. Therefore, we have grouped the methods into four categories according to what kind of clustering models they produce: Representative-based and hierarchical methods, mixture model clustering, and density-based methods. The goal is to cover clustering methods that are usually available in data mining and statistical analysis tools.

Representative-based methods

In *representative-based clustering methods*, each cluster C_i is represented by its centroid $\mathbf{c_i}$. In the case of numerical data, centroid is usually the mean vector of cluster points, although it may be also the median vector or a data point closest to the median, called a *medoid*. Each data point $\mathbf{p_j}$ is assigned to the cluster whose centroid $\mathbf{c_i}$ is closest, i.e. which minimizes distance $d(\mathbf{p_j}, \mathbf{c_i})$ with the given distance measure d. The clustering problem is to select centroids $\mathbf{c_1}$, …, $\mathbf{c_k}$ such that some score function measuring the clustering quality is optimized. The most common criterion is to minimize the total *sum of squared errors*

$$SSE = \sum_{i=1}^{k} \sum_{\mathbf{p_j} \in C_i} d^2(\mathbf{p_j}, \mathbf{c_i}),$$

where distance measure d is typically Euclidean distance, L_2. In the following, we call this as the *k-means criterion*. This should not be confused with the *k-means algorithm* which is only one way to optimize the *k*-means criterion. Many properties associated to the *k*-means algorithm are actually due to the underlying clustering criterion and thus shared by the alternative search methods.

The *k*-means criterion works well, if the clusters are hyperspherical, compact and well-separated, but otherwise it can produce misleading results. It also tends to produce equal-sized clusters which is desirable in segmentation but seldom makes justice to natural clusters. The main problem of the *k*-means criterion is its sensitivity to outliers. The reason is that a couple of extreme points can have a strong

impact on the mean. It is also clear from the definition that the *k*-means criterion can be used only for a strict clustering of numerical data. In addition, the SSE score function does not offer means to determine the optimal number of clusters, *k*. (Jain et al., 1999; Estivill-Castro, 2002; Jain and Dubes, 1988, Ch.3.3; Hand et al., 2002, Ch.9.4).

Some of these problems can be alleviated by modifying the *k*-means criterion. When the Euclidean distance is replaced by the Mahalanobis distance, it is possible to detect also hyperellipsoidal clusters (Jain et al., 2000). The *k-medians* criterion uses medians instead of means as cluster centroids. The criterion is more robust to outliers because medians are less affected by extreme points than means (Estivill-Castro, 2002). However, determining medians is computationally much more costly. As a compromise, it is required that the centroids are data points closest to the medians, known as the *k-medoids* criterion. The criterion allows also ordinal categorical variables, if a suitable distance measure has been defined. For categorical data, one can use modes (the most common values) to define centroids, with an appropriate distance measure (the *k-modes algorithm* (Huang, 1998)). There are even attempts to extend the idea for mixed data (the *k-prototypes algorithm* (Huang, 1998)).

A fuzzy version of the k-means criterion, known as *fuzzy k-means*, defines a soft clustering, where each point can belong to several clusters with different degrees. The minimized score function is

$$FSSE = \sum_{i=1}^{k} \sum_{j=1}^{n} mem_i^b(\mathbf{p_j}) d(\mathbf{p_j}, \mathbf{c_i}),$$

where $mem_i(\mathbf{p_j})$ is the grade of membership with which $\mathbf{p_j}$ belongs to the cluster C_i, $\Sigma_{i=1}^{k} mem_i(\mathbf{p_j}) = 1$, and parameter $b \geq 1$ regulates the degree of fuzziness.

Soft clustering is quite appealing in the educational context, because the clusters are seldom well-separated and one would not like to force data points into artificial clusters. Fuzzy membership values do also offer extra information, but their interpretation can be difficult, because they are not probabilities. The biggest problem in fuzzy clustering is how to define the membership functions which have a crucial effect on results. Some solutions are discussed e.g. in (Jain and Dubes, 1988, Ch.3.3.8).

Clustering methods based on statistical mixture models could also be considered as soft representative-based methods. However, they are actually more general, and the *k*-means criterion can be considered as a special case of the mixture model clustering, as we will see later.

So far, we have discussed only the clustering criteria of representative-based clustering. These criteria determine what kind of clustering models we could find, if we had a globally optimal search algorithm. However, the optimization problem is intractable even for small data sets, and in practice one has to use heuristic search algorithms. These algorithms have their own bias which can affect the results.

The most popular clustering algorithm for the *k*-means criterion and its variants is the iterative search (*k*-means, *k*-medoids, and fuzzy *c*-means algorithms). The basic *k*-means algorithm begins from some initial set of centroids and assigns all points to their nearest clusters. Then it iteratively calculates new cluster means (centroids) and assigns points to their nearest clusters, until some stopping criterion (e.g. convergence) is met. The *k*-means algorithm is easy to implement and quite efficient even with large data sets. In practice, it can produce surprisingly good results, if the clusters are compact, hyperspherical, and well-separated (Jain et al., 2000). However, it is very sensitive to the selection of the initial partition (centroids) and can easily converge to a local minimum, if the initial partition is far from the final solution (Jain et al., 1999). It is also very sensitive to noise and outliers, which is mostly due to the clustering criterion. An interesting dilemma is that while the *k*-means criterion itself tends to favour equal-sized clusters (Hand et al., 2002, Ch. 9.4), the *k*-means algorithm can produce very different sized clusters, even empty ones (Berkhin, 2006).

In common implementations, the user should define the number of clusters in advance, but more sophisticated algorithms (like *x*-means (Pelleg and Moore, 2000)) can select the number automatically according to some goodness criterion. The fuzzy *c*-means algorithm (Bezdek, 1981) shares the same drawbacks as *k*-means, but is has been said to be less prone to converge into a local minimum (Jain et al., 1999).

More advanced search algorithms can produce better results, but at the cost of computational cost. In addition, these algorithms usually require several user-specified parameters, which affect the results.

The most straight-forward solution is to optimize the *k*-means clustering criterion or its fuzzy variant with the classical optimization methods, like simulated annealing, tabu search, genetic algorithms, or other evolutionary methods. In empirical comparisons, all these have performed better than the basic *k*-means, but none of them has been consistently superior to others. It seems that problem-specific evolutionary methods are potentially very good at discovering the global optimum, if the data set is small-dimensional and small-sized (less than 1000 rows) (Jain et al., 1999).

Another approach is to search for an optimal clustering using neural networks. Several architectures and algorithms have been suggested for both the *k*-means and the fuzzy *k*-means criteria (for an overview see e.g. (Xu and Wunsch, 2008, Ch.5). Common to these methods is that they suit only to numerical data, the results depend on several parameters, and the clustering can be unstable (Jain et al., 1999). In this group, we should also mention *self-organizing maps* (SOMs) (Kohonen, 1982) that perform a *k*-means style clustering with extra constraints, but also produce a visual 2-dimensional representation of the clusters and their topological relations. This representation is potentially very informative, but it should be interpreted with caution, because it does not always represent the space density correctly (Xu and Wunsch, 2005). Like most neural network methods, SOMs are also sensitive to initial parameters and can produce unstable results (Jain et al., 1999).

Kernel-based clustering methods, like kernel k-means (Schölkopf et al., 1998), are potentially very robust methods for numerical data. The basic idea of kernel-based methods is to map the original data implicitly (with the kernel trick) into a higher-dimensional space, where clusters become linearly separable. This is useful especially when the clusters are linearly inseparable in the original space and, thus, cannot be separated with the common *k*-means. The kernel-methods can detect arbitrary-shaped clusters and they are robust to noise and outliers. However, they are too inefficient for large data sets and deciding the parameters can be difficult (Xu and Wunsch, 2005).

In this connection, we should also mention *spectral clustering methods*, because they are closely related to kernel-based methods, in spite of their different looking algorithms. It has been shown that the underlying clustering criteria (optimized objective functions) of the spectral clustering and a particular form of the *weighed kernel k-means* clustering are equivalent (Dhillon et al., 2004). The main idea of spectral clustering methods is the following: First, a similarity matrix, based on point-wise distances between nearest neighbours, is constructed and the corresponding Laplacian matrix is computed. The *k* first eigenvectors of this matrix are selected as new variables. Then, the data is clustered with the *k*-means (or any other method) in this new smaller-dimensional space. The spectral methods can detect arbitrary-shaped clusters, which could not be detected in the original data space. In addition, they suit to any data type, because they require only point-wise distances. The results can be very good, if the original similarity matrix was well chosen, but choosing a good similarity matrix is not a trivial task. In addition, the methods do not scale up to large data sets. This can be partially alleviated by considering only the nearest neighbours (i.e., using a sparse similarity matrix), but this can have a strong impact on the clustering results. (Luxburg, 2007; Dhillon et al., 2007). A more promising solution is to solve the spectral clustering problem with a faster weighed kernel *k*-means method (Dhillon et al., 2007).

For the educational data, the representative-based methods, like the *k*-means clustering and its neural equivalents, are seldom the optimal choice, even if the data were numerical. The clusters are often unclear or arbitrary-shaped and there are typically several outliers. In this sense, the kernel-based methods

sound the most promising, especially when the data sets tend to be small. For categorical and mixed data, the spectral clustering and its solutions with the weighed kernel *k*-means offer a good alternative. Unfortunately, these new clustering methods are not yet available in common data mining tools.

Hierarchical methods

Hierarchical clustering methods do not produce just a single partition of data, but a sequence of partitions. The result is a hierarchy of clusters which can be represented visually as a *dendrogram*. The dendrogram is a binary tree structure, whose root corresponds a single large cluster containing all data points and leaves correspond to *n* small clusters, each containing just one data point. Children of a node show how a cluster is divided into subclusters. An example is shown in Figure 1. In practice, the dendrogram can be constructed until the desired number of clusters is left or one can select the best clustering afterwards, after analyzing the dendrogram.

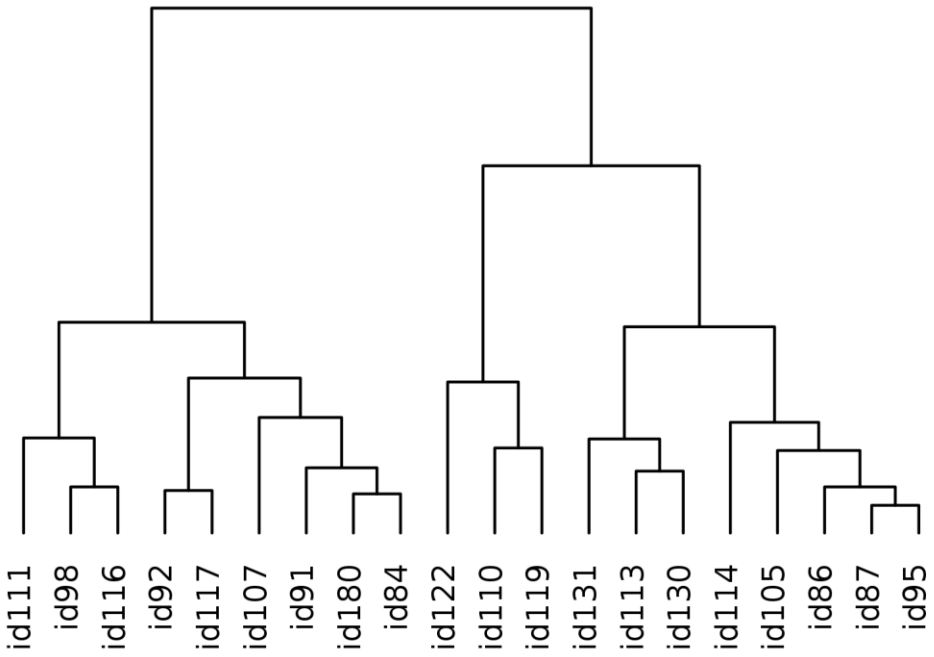


Figure 1: An example dendrogram
(using the complete-link metric).

Hierarchical clustering methods can be divided into two main approaches, *agglomerative* and *divisive* methods. Agglomerative methods proceed in a bottom-up manner, beginning from clusters of single data points and merging neighbouring clusters until only one cluster is left. Divisive methods work in the opposite way, in a top-down manner, by dividing superclusters into neighbouring subclusters until all points are in their own clusters. In practice, agglomerative methods are more popular, because they are computationally cheaper and easier to implement.

The neighbouring clusters to be merged or split are defined by some score function *D*, which measures the similarity of clusters. This inter-cluster distance is often called a *linkage metric*. It is calculated from the distances between some or all points in two clusters. The choice of the linkage metric has a strong impact on the clustering results, because it defines the shape, size, and density of clusters which can be found. Most metrics tend to produce hyperspherical (isotropic) clusters. In Table 2, we have listed the most common inter-cluster measures and types of clusters they tend to produce.

The *single-link* metric is flexible in the sense that it can find non-isotropic clusters (with unsymmetrical shapes) and the clusters can be even concentric. On the other hand, it has tendency to produce elongated and straggly clusters. This is called the "chaining effect": the measure combines clusters through other external points and the clusters become chain-like. As a result, the single-link metric works best for well-separated, non-spherical clusters. One advantage of the single-link metric compared to other linkage metrics is that it is independent from the data order and the resulting clustering is always unique. The single-link metric is also quite efficient to compute. In empirical comparisons, the single-link metric has usually performed poorly, although it may sometimes surprise with good results (Jain and Dubes, 1988, Ch.3.5.2). This is also our own experience, although we have found that the single-link metric is very good at detecting outliers.

The *complete-link* metric works usually better than the single-link metric, but it is slower to calculate and does not suit for large data sets. It tends to produces small, compact, equal-sized clusters, but it cannot separate concentric clusters. The main disadvantage of the complete-link metric is its dependency on the data order. In empirical comparisons, the complete-link metric has usually performed well (Jain and Dubes, 1988, Ch.3.5.2). Our experiences with the complete-link metric have also been quite good, although it has not been able to detect even obvious outliers.

Table 2. Common measures for the inter-cluster distance *D*, given the distance between points *d*. Cluster type describes what kinds of clusters the measure tends to produce.

Metric	***D*(C_1, C_2)**	**Cluster type**
Single-link	$\min_{\mathbf{p_1}\in C_1, \mathbf{p_2}\in C_2} \{d(\mathbf{p_1}, \mathbf{p_2})\}$	elongated, straggly, also concentric clusters
Complete-link	$\max_{\mathbf{p_1}\in C_1, \mathbf{p_2}\in C_2} \{d(\mathbf{p_1}, \mathbf{p_2})\}$	small, compact, hyperspherical, equal-sized
Average-link	$\frac{\sum_{\mathbf{p_1}\in C_1, \mathbf{p_2}\in C_2} d(\mathbf{p_1}, \mathbf{p_2})}{\lvert C_2\rvert\lvert C_2\rvert}$	quite compact clusters; allows different sizes and densities
Minimum variance (Ward)	$SSE(C_1 \cup C_2) - SSE(C_1) - SSE(C_2)$	compact, quite well-separated, hyperspherical; cannot find elongated clusters or clusters of very different sizes
Distance of centroids	$d(\boldsymbol{c_1}, \boldsymbol{c_2})$	hyperspherical, equal-sized clusters; cannot detect elongated clusters

The *average-link* metric produces clusters which are between the single-link and the complete-link metrics in their compactness. It produces dense clusters, letting larger clusters to be sparser than the smaller ones. Its main disadvantage is the dependency on the data order. In addition, the metric is quite inefficient to compute, which may be a burden for really large data sets. In empirical comparisons, the average-link metric has usually performed quite well (Jain and Dubes, 1988, Ch.3.5.2). In our own experience, the average metric has produced good and stable results, although it has not been able to detect outliers. One possible explanation for the stability is that the metric is based on all points in measured clusters and is thus less affected by outliers and noise than metrics based on single points.

The *minimum variance* metric is famous and it is used e.g. in the classical *Ward's method* (Ward, 1963). It minimizes the variance in the clusters through the *SSE* score function. The resulting clusters are hyperspherical and quite compact, but it is not possible to find elongated clusters. The metric is computationally efficient, because it depends only on the centroids and sizes of C_1 and C_2 (Jain and Dubes, 1988, Ch.3.2.7). Like the previous two metrics, the minimum variance metric is also dependent on the data order. In empirical comparisons, the metric has generally performed well, especially if clusters

are of equal size (Jain and Dubes, 1988, Ch.3.5.2). Our own experiences have been mixed. In some data sets, the metric has produced very good results, while in others it has produced strange results or showed instability (strong dependence on the data order).

The *distance of centroids* is sometimes used to approximate the minimum variance metric. The metric is really efficient to calculate, but results can be quite poor. It works well (like most methods) if clusters are hyperspherical and well-separated, but if they are arbitrary-shaped, like elongated, the results can be quite insensible (Guha et al., 1998). In empirical comparisons, its performance has not been impressive (Jain and Dubes, 1988, Ch.3.5.2). According to our experiences, this metric has worked relatively well. The results have been quite similar to the average-link metric, although not always as good. In addition, we have observed that the metric seems to be less sensitive to the data order than the minimum variance metric.

In addition to these classical linkage metrics, some algorithms use their own metrics. For example, CURE (Guha et al., 1998) selects several representative points to capture the shape and extent of a cluster, moves them towards the cluster center, and determines the cluster distances by these new points. The metric has many advantages: it can detect nonspherical (like elongated) clusters of different sizes and is robust to outliers (Guha et al., 1998). Another example is CHAMELEON (Karypis et al., 1999), which combines graph-partitioning and agglomerative hierarchical clustering. In the first phase, the algorithm constructs a nearest-neighbour graph, whose edge weights reflect how close neighbours the points are to each other. The graph is partitioned into a large number of small clusters. In the second phase, the algorithm performs an agglomerative hierarchical clustering and merges clusters using a special metric. The metric is actually a combination of two metrics, which measure the *relative inter-connectivity* between two clusters and their *relative closeness*. The resulting clustering criterion is quite flexible and CHAMELEON can detect arbitrary-shaped clusters of different sizes and densities. In practice, CHAMELEON has also performed very well (Han and Kamber, 2006, Ch.7.5.4). However, both CURE and CHAMELEON require user-specified parameters which can be difficult to select.

Another interesting option is to base the metric on the probability of data given the clustering, like in the statistical mixture model methods. These metrics try to evaluate how much the log likelihood of data changes when two clusters are combined. Different variations of this metric and their relations to classical linkage metrics have been described in (Zhong and Ghosh, 2003).

In general, hierarchical methods have several attractive features and suit well to educational data. They can be applied to numerical, categorical, and even mixed data, if the point-wise distances have been defined. The dendrogram contains useful information on the hierarchy and relationships of clusters. In addition, there are several similarity measures to try with different shapes of clusters.

However, there are also drawbacks which should be kept in mind. We have already noted that most hierarchical methods are dependent on the data order and can produce different clusterings with different data orders. In practice, it is always advisable to test the stability of results with different data orders. Deciding the number of clusters or the final level of the dendrogram can also be difficult. Even for a moderate size of data (more than a couple of hundred rows), dendrograms can be too messy to inspect visually. As a solution, one can use methods, like BIRCH (Zhang et al., 1997), which first precluster the data into a large number of small clusters, and then construct the hierarchy. However, these methods may sometimes produce unnatural clusterings (Halkidi et al., 2001).

Other problems are related to the search algorithms and not to the hierarchical methods per se. The main disadvantage of the classical greedy algorithms is the static nature of cluster allocations. When a point has once been assigned to a cluster, it cannot be moved elsewhere. Therefore, the algorithm is not able to correct its earlier misassignments (Xu and Wunsch, 2005). This restriction has been solved in the more sophisticated implementations. For example, when hierarchical clustering is implemented by neural networks, the points can be moved from a cluster to another during clustering (Khan and Luo, 2005). In

addition, hierarchical algorithms are not as efficient as the k-means algorithm for large data sets (a large number of rows or dimensions), although this is seldom a problem with the educational data.

Mixture model clustering

Mixture model clustering refers to methods which perform clustering by fitting a statistical mixture model into data. They have also been called "clustering by mixture decomposition" (Jain and Dubes, 1988), "probabilistic model-based clustering" (Hand et al., 2002), or simply "probabilistic clustering" (Berkhin, 2006).

Mixture model clustering methods differ from the previous methods in one fundamental aspect: they do not try cluster the existing data points into crisp clusters, but instead they try to learn a statistical density model for the whole data space. From this model, one can then derive for each data point $\mathbf{p_j}$ the probabilities $P(C_i \mid \mathbf{p_j})$ by which it belongs to clusters C_i, $i = 1, \ldots, k$. In addition, the model enables us to predict clusters for new data points and update the model when new data points are added or removed. In the 2-dimensional case, the model can be represented visually, by density contours (see Figure 2). The underlying assumption is that the data has been generated by a mixture of probabilistic models (multivariate distributions). Each cluster has a prior probability and its own probability distribution. The whole model is typically represented as a multivariate mixture model.

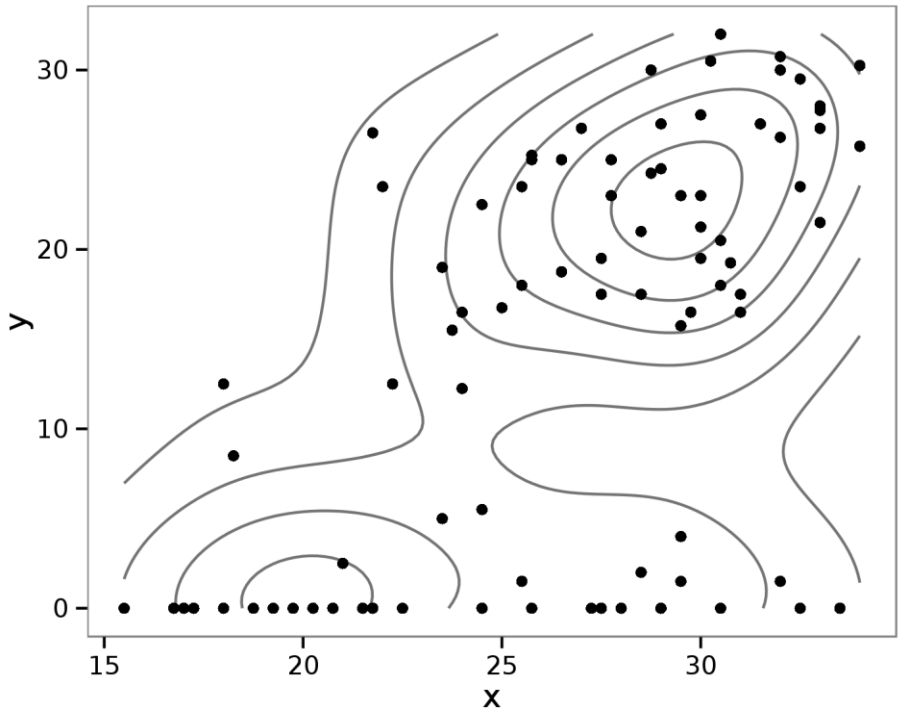


Figure 2: An example contour map that describes data densities.

Definition 2 (Multivariate mixture model). Let S be a numeric data space and k the number of clusters. Let $f_i(\mathbf{p}, \Theta_i)$ be be the density function of cluster C_i with parameters Θ_i and π_i the prior probability of cluster C_i. Then the multivariate mixture model is defined by density function $f : S \rightarrow [0, 1]$ such that for all $\mathbf{p} \in S$

$$f(\mathbf{p}) = \sum_{i=1}^{k} \pi_i f_i(\mathbf{p}, \Theta_i).$$

Here $f_i(\mathbf{p}, \Theta_i)$ defines the probability that data point p belongs to cluster C_i and $f(\mathbf{p})$ describes the posterior probability of data point **p** given the whole model. If $f(\mathbf{p})$ is very low, the point does not fit the model, and it can be interpreted as an outlier.

The density function f_i describes the data distribution in cluster C_i. In principle, we can define a different type of distribution for each cluster. This is useful, when the data is very skewed. However, in practice, it is very difficult to define the appropriate distributional form without any prior knowledge. That is why it is usually assumed that the density in all clusters has the same distributional form, and only the distribution parameters are different. A common choice is to assume the normal distribution.

The whole clustering process consists of four steps:

1. Determine the number of clusters k (optional).
2. Choose the density functions f_i for all clusters C_i.
3. Determine the cluster probabilities π_i and parameters Θ_i (and potentially k) such that the probability of the data given the model is maximized (according to some principle).
4. Assign each point **p** to the most probable cluster, i.e. select such C_i that $P(C_i \mid \mathbf{p})$ is maximal.

Usually, the parameters are selected either by the *Maximum likelihood* (ML) principle, which maximizes the data (log-)likelihood given the model, or the *Maximum a posteriori probability* (MAP) principle, which maximizes the posterior probability of the model, given data. If the prior probabilities of clusters are equal, then these two principles coincide (Fraley and Raftery, 2000). The simplest and most effcient way to approximate the parameters is the *Expectation Maximization* (EM) algorithm, which performs an iterative *k*-means style search. In Bayesian approaches, *Markov Chain Monte-Carlo methods* (MCMC) have also been used, but they are computationally much more demanding (Fraley and Raftery, 2000).

The best number of clusters is usually selected automatically, according to some score function like *Minimum Description Length* (MDL) or *Bayesian Information Criterion* (BIC) (Fraley and Raftery, 2000). Both of them maximize the log-likelihood of data with some penalty terms for large k. Techniques based on cross-validation have also been applied successfully (Smyth, 2000). According to Stanford and Raftery (2000), both BIC and cross-validation methods converge to the global optimum, when the sample size grows, but BIC is faster to compute.

The mixture model clustering has several advantages. The clustering is not strict, but each data point can belong to several clusters with different probabilities. This means that the clusters can be overlapping, which is often the case with educational data. In addition, the probabilities themselves are useful extra information which can be utilized in adaptive learning systems (e.g., when selecting an optimal action). In the two-dimensional case, the densities have a nice visual representation as contour maps, and outliers are easily recognized.

The mixture models are very flexible and can describe even complex structures. For example, every cluster can have different size and density, even different type of distribution. In addition, it has been observed that several other clustering methods are special cases of mixture model clustering (Fraley and Raftery, 2000; Kamvar et al., 2002). For example, the *k*-means criterion is equivalent to the mixture model, where density functions are Gaussian, all variables are assumed to be mutually independent, and all variables in all components have the same variance (Fraley and Raftery, 2000). Mixture model clustering can also be applied to categorical data, using e.g. multinomial distributions, but mixed data is problematic.

The flexibility of the mixture model clustering has also a drawback: the resulting clustering depends strongly on the selected form of the distribution which can be hard to define without any prior knowledge. Assuming a wrong parametric model can lead to poor or misleading results. In the worst case, wrong model assumptions can impose a structure into data, instead of finding one. (Duda et al., 2000,

Ch.10.6) In principle, unknown distributions can be approximated with a sufficiently large number of Gaussian components (Fraley and Raftery, 2000), but such models do not reveal the real clusters directly. In addition, the most adaptive models are also the most complex and can require a lot of parameters. These can be impossible to determine accurately, if the data is sparse (Fraley and Raftery, 2000). Therefore, many implementations allow only the normal distribution with the assumption of variable independence and uniform variances. This is usually unrealistic with educational data. The normal distribution makes the method also sensitive to outliers, because extreme data points have a strong effect on variance.

There are also problems which are related to the search algorithm. The optimization problem is computationally demanding and even the simple iterative EM algorithm does not scale to large data sets. Fortunately, this is seldom a problem with educational data sets. The EM algorithm is also very sensitive to the initial parameters and can get stuck at a local optimum. As a solution, the data can be first clustered with another method, which defines the initial cluster means and variances. For example, initialization by hierarchical clustering has produced good results and at the same time the ideal number of clusters could be determined (Dasgupta and Raftery, 1998; Fraley and Raftery, 1998). Another alternative is to implement the entire probabilistic clustering in a hierarchical way (Murtagh and Raftery, 1984; Banfield and Raftery, 1993).

In practice, mixture model clustering with the EM algorithm has produced varying results. Usually the results have been good, but the algorithm can fail if some clusters contain only a few data points or there are redundant variables (Fraley and Raftery, 2000). Our own experiences are restricted to clustering numerical data using Gaussian mixture models, with the unrealistic assumption of variable independence. This kind of a model was clearly unsuitable for our student data and could not detect the natural clusters. Still, the composed density function could describe the data well, if the number of clusters was sufficiently large. We have also tested how well the method suits to outlier detection, i.e., whether the most improbable points capture real outliers. The results have been diverse. It seems that if the model is sufficiently simple (only a few clusters), the most improbable points correspond well the real outliers. However, if the model is very complex (many clusters), then all points are relatively probable and no outliers can be detected by their probability. These observations hint that if one wants to get a strict clustering and detect outliers, then a small number of clusters is preferable, but if one prefers an accurate description of the data by densities, then a more complex model with several clusters should be used.

According to this evaluation, the mixture model clustering has a great potential for clustering educational data, but there are still many practical problems to be solved.

Density-based methods

Density-based methods regard clusters as dense regions in the data space separated by sparser regions. Some of these methods approximate the overall density function, like mixture model methods, while others use only local density information to construct clusters. In this sense, the mixture model methods could also be classified as (parametric) density-based methods, although the term usually refers to non-parametric methods, which are discussed here. The advantage of the non-parametric density-based methods is that one does not have to know the number or distributional form of the subpopulations (clusters).

In the following, we introduce three famous algorithms which represent different approaches to density-based clustering. Many of the other density-based algorithms are only improved (better-scalable) variants of these algorithms. In addition, there are hybrid methods (like previously mentioned CHAMELEON and BIRCH) which use density-based techniques only in some phase of the algorithm, like preclustering.

Most density-based algorithms proceed in a two-phase manner: First, they detect the cluster cores in the densest areas and then they expand the clusters, as long as the density remains sufficiently high. DBSCAN (Ester et al., 1996) can be considered as a prototype of this type of algorithms. DBSCAN defines cluster cores as points whose surroundings (inside a certain radius, *Eps*) contain sufficiently many (at least *MinPts*) other points. Then the clusters are expanded, by adding all sufficiently close points, which may lead to merging of clusters. Points which are too far from any cluster cores remain as outliers. The main advantages of DBSCAN are its ability to detect arbitrary-shaped clusters and robustness to outliers. In addition, it can be applied to categorical or even mixed data, given a suitable measure for point-wise distances. However, it is very sensitive to the input parameters. Wrong parameter values can lead to a similar chaining effect as seen with the single-link metric. Another problem is how to detect clusters of different densities, because they would require different parameter values. In addition, DBSCAN does not suit to large-dimensional or large data sets (Halkidi et al., 2001; Guha et al., 1998; Wang and Hamilton, 2003).

DENCLUE (Hinneburg et al., 1998) represents another approach of density-based methods, because it approximates the overall density function to perform the clustering. The main idea is the following: For each data point, the algorithm determines an *influence function* which describes its impact in the neighbourhood. The overall density function is calculated from these influence functions. The local maxima of the density function are defined as *density attractors*, which form the cluster cores. The attractors can be used as cluster centroids like in representative-based methods or one can construct arbitrary-shaped clusters by combining several attractors. All other points are assigned to the same cluster as their attractors, if their density function value is sufficiently high. Otherwise, they are considered as outliers. The DENCLUE algorithm has several advantages: it can detect arbitrary-shaped clusters and give them a compact mathematical representation, it is very robust to noise and outliers and quite efficient (compared to other density-based methods). The main problem is that the results depend on several input parameters (including the influence function), which can be difficult to select. In addition, it can handle only numerical data (Berkhin, 2006; Halkidi et al., 2001; Han and Kamber, 2006, Ch.7.6.3).

Wavecluster (Sheikholeslami et al., 1998) is an interesting clustering algorithm which is sometimes classified as a grid-based method. The basic idea is to divide the data space into small hyperrectangular cells, map this compressed data with a wavelet transformation, and then search for dense areas in the transformed space. The underlying idea is that in the transformed space even arbitrary-shaped clusters become better distinguishable. Wavecluster has several attractive features. It can detect clusters of different shapes and sizes, it is robust to noise and outliers, produces good quality results, and is quite efficient. However, the user has to define several parameters, including the wavelet type. These parameters affect, among other things, how many and how well-separated clusters are found. In addition, Wavecluster can handle only numerical data (Halkidi et al., 2001; Han and Kamber, 2006, Ch.7.7.2).

For educational data, density-based methods are quite promising, at least for small-dimensional data. They can detect arbitrary-shaped clusters and are robust to outliers. Some methods (like DBSCAN and its successors) can handle also categorical and mixed data. One drawback is that the nonparametric density-based methods do not offer any easily interpretable representation for the clusters like mixture models. Generally, density-based methods are not recommended to high-dimensional data, because all data tends to become sparse in high dimensions (Jain, 2010). In addition, most empirical comparisons of density-based methods have been made with 2- or 3-dimensional data, and it is not known whether they produce equally good results with higher dimensional data.

Which method to choose?

As we have seen, all clustering methods have their strengths and weaknesses. There is no universally superior clustering method, but the best choice depends always on the context. Even for typical educational data, we cannot give any unique recommendation. However, we can give some suggestions based on our evaluation.

If the data is purely categorical or contains both numerical and categorical variables, the easiest choice is to use methods, which require only pair-wise distances among data points. These include hierarchical methods, DBSCAN-style density-based methods, and spectral clustering. Among hierarchical methods, CHAMELEON has been especially recommended (Han and Kamber, 2006). The classical hierarchical methods have several weaknesses (especially, the stability problem) and we would not recommend them as a primary choice. All above mentioned methods, CHAMELEON, DBSCAN, and spectral clustering, can detect arbitrary-shaped clusters. At least DBSCAN is said to be robust to outliers. In addition, it has an extra advantage that it does not require the number of clusters as a parameter but it may have problems with clusters of different densities. Spectral clustering is potentially a very powerful technique, but one should be ready to experiment with different similarity matrices.

For numerical data, there are more good choices. For arbitrary-shaped clusters, the best candidates are the kernel *k*-means, density-based methods like DENCLUE and Wavecluster, previously mentioned CHAMELEON and spectral clustering, and maybe mixture model methods with certain assumptions. The kernel *k*-means, spectral clustering, and Wavecluster all use the same trick and map the data into a new feature space to detect linearly non-separable clusters. In addition, at least the kernel *k*-means, DENCLUE, and Wavecluster are said to be robust to outliers. Mixture model methods are especially attractive for the educational data because they allow overlapping clusters and produce extra information in the form of probabilities. However, there are some practical difficulties which can restrict their use. The assumption of Gaussian distributions with variable independence is usually unrealistic, but more flexible mixture models cannot be estimated accurately, unless there is sufficiently data with respect to the data dimensionality. Because educational data sets tend to be quite small, the more complex mixture models can be used only for small-dimensional data.

Finally, we recall that the clustering method alone does not guarantee good or even meaningful results. Feature extraction and selection can have a crucial impact on the results. Irrelevant features or just too many features can easily hide the real clusters. On the other hand, good features may be able to represent the clusters so clearly that even poorer methods can find them. For this reason, it is advisable to try different feature extraction and selection schemes and cluster even the same set of features with different methods and parameter settings. Clustering is anyway exploratory by its nature and one cannot know beforehand, where the most useful clusters hide. One should also remember the possibility that there are no clusters at all in the data, even if the clustering algorithm always tries to return some solution.

Future Research Directions

In distance learning, adaptive and intelligent learning systems play a central role. Therefore, a relevant question is what clustering can offer to these tools. We see at least three different ways how clustering could be used to implement the "intelligence" of a learning system.

First, the clustering process could be automated to give regular reports to teachers on students' performance as the course proceeds. As we have seen, this kind of information can be very important for detecting possible problems or special needs in time. For this purpose, one should first analyze an example data set and select appropriate features, clustering methods, and their parameters. After that, the same clustering procedure could be executed automatically. A related research challenge is how clustering could be done dynamically, by updating the previous clustering, when new feature values are recorded.

Second, clustering methods can facilitate or even be a part of predictive models, which are the heart of intelligent systems. Usually, the predictive models are classifiers, which classify the user or situation, so that the system can select an optimal action. Sometimes, the predictive models can also be regression models, which predict a numerical score. For classifier construction, clustering methods are especially useful. Cluster analysis reveals what kind of classes can be separated with the given features and, thus, clustering can be used to select suitable features or class labels. If the clustering produced

sufficiently homogeneous clusters (with respect to class labels), it can be later used as a K-nearest neighbour classifier. Similarly, cluster analysis can reveal reasoning rules for predicting scores or even selecting optimal actions.

Third, clustering can be used to recommend tasks or pieces of learning material. If a student has proved to need more practice in certain kind of tasks or topics, similar items can be selected from the same cluster. On the other hand, if the student already masters one cluster, then the system should recommend material from other clusters.

In addition, clustering can be used to allocate students into homogeneous (intra-cluster) or heterogeneous (inter-cluster) groups. This problem is related to one of the current clustering trends, semi-supervised clustering (Jain, 2010). In semi-supervised clustering, background knowledge is integrated into clustering in the form of constraints. For example, if two students have already collaborated together, one can impose a constraint which either forces them to the same cluster or to different clusters.

Another current trend is research on clustering heterogeneous data, which cannot be represented naturally by a fixed set of variables (Jain, 2010). One interesting problem is how to partition a large graph into cohesive subgraphs, when the nodes and edges are associated by attributes. The problem is actually equivalent to clustering relational data. In the educational domain, this kind of graph partitioning could be used to summarize concept maps (allowing edge labels or different edge types) or to study students' collaboration and communication networks. Another problem is how to cluster a set of graphs, like concept maps. Text clustering is also an important research trend. One tricky problem is how to cluster short pieces of text, which may be only a couple sentences long. A good technique to cluster students' free-formed answers, preferably together with other variables, would help in the analysis of questionnaires.

There are also general research challenges, which are relevant to educational data. One important problem is how to construct a mixture model for mixed data, containing both numerical and categorical variables. Further research on similarity measures for mixed data (like empirical comparisons) would also be welcome. Another important but often neglected problem concerns the validation of clustering results. A lot of different validation indices have been proposed (see e.g. (Gan et al., 2007, Ch.17)), but without any significance testing, they can be used merely for comparing clusterings. Statistical significance testing would require laboursome randomization tests which are very seldom done. In addition, one should select the validity index according to the clustering criterion which is not always obvious. In this area, we would need more efficient algorithms for significance testing, implementations of indices in data mining tools, and guidelines for matching indices and (often implicit) clustering criteria. A related but easier problem is to test the stability of clustering results (i.e. the effect of small perturbations in the data), but implementations are still missing from data mining tools.

Conclusions

In this chapter, we have considered the problem of clustering student data. First, we specified the main purposes of clustering students and the characteristics of typical student data. After that, we described the most important clustering methods and evaluated their suitability to clustering student data.

Based on our evaluation, there is no superior clustering method, which would fulfil all desirable properties. In principle, the mixture model methods could often produce the most informative and attractive clustering models, but there are still practical problems which limit their use. Among strict clustering methods, density-based methods, spectral clustering, kernel *k*-means, and the hierarchical CHAMELEON algorithm look the most promising.

An interesting dilemma is why the *k*-means clustering has been so popular in the educational data mining, even if it is among the least suitable methods for typical educational data. In other fields, its popularity is more understandable, due to its efficiency and, perhaps, more easily detectable clusters.

However, in our field, the efficiency is seldom a bottleneck and we encourage to try more developed methods.

References

Aggarwal, C., Hinneburg, A., and Kleim, D. (2001). On the surprising behavior of distance metrics in high dimensional space. In Proceedings of the 8th International Conference on Database Theory (ICDT 2001), volume 1973 of Lecture Notes in Computer Science, pages 420-434. Springer-Verlag.

Aggarwal, C. and Zhai, C. (2012). A Survey of Text Clustering Algorithms, pages 77-128. Boston: Kluwer Academic Publishers.

Agrawal, R., Gehrke, J., Gunopulos, D., and Raghavan, P. (1998). Automatic subspace clustering of high dimensional data for data mining applications. In Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data (SIGMOD'98), pages 94-105, New York, NY, USA. ACM Press.

Andreopoulos, B., An, A., Wang, X., and Schroeder, M. (2009). A roadmap of clustering algorithms: finding a match for a biomedical application. 10(3):297-314.

Banfield, J. and Raftery, A. (1993). Model-based Gaussian and non-Gaussian clustering. Biometrics, 49:803-821.

Berkhin, P. (2006). Survey of clustering data mining techniques. In Kogan, J., Nicholas, C., and Teboulleeds, M. (Eds.), Grouping Multidimensional Data: Recent Advances in Clustering, pages 25-71. Springer.

Bezdek, J. (1981). Pattern Recognition with Fuzzy Objective Function Algorithms. Norwell, MA, USA: Kluwer Academic Publishers.

Cheng, V., Li, C., Kwok, J., and Li, C.-K. (2004). Dissimilarity learning for nominal data. Pattern Recognition, 37(7):1471-1477.

Dasgupta, A. and Raftery, A. (1998). Detecting features in spatial point processes with clutter via model-based clustering. Journal of the American Statistical Association, 93(441):294-302.

Dhillon, I., Guan, Y., and Kulis, B. (2004). Kernel k-means: spectral clustering and normalized cuts. In Proceedings of the tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 551-556, New York, NY, USA. ACM.

Dhillon, I., Guan, Y., and Kulis, B. (2007). Weighted graph cuts without eigenvectors a multilevel approach. IEEE Transactions on Pattern Analysis and Machine Intelligence, 29(11):1944-1957.

Duda, R., Hart, P., and Stork, D. (2000). Pattern Classification. New York: Wiley Interscience Publication, 2nd edition.

Ester, M., Kriegel, H., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, pages 226-231.

Estivill-Castro, V. (2002). Why so many clustering algorithms? A position paper. SIGKKD Explorations, 4(1):65-75.

Fraley, C. and Raftery, A. (1998). How many clusters? Which clustering method? Answers via model-based cluster analysis. The Computer Journal, 41(8):578-588.

Fraley, C. and Raftery, A. (2000). Model-based clustering, discriminant analysis, and density estimation. Journal of the American Statistical Association, 97:611-631.

Gan, G., Ma, C., and Wu, J. (2007). Data Clustering: Theory, Algorithms, and Applications. ASA-SIAM Series on Statistics and Applied Probability. SIAM.

Ghosh, J. (2004). Scalable clustering methods for data mining. In Ye, N. (Ed.), Hand Book of Data Mining, chapter 10. Lawrence Erlbaum Associates.

Guha, S., Rastogi, R., and Shim, K. (1998). Cure: an efficient clustering algorithm for large databases. In Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data, SIGMOD '98, pages 73-84, New York, USA. ACM.

Halkidi, M., Batistakis, Y., and Vazirgiannis, M. (2001). On clustering validation techniques. Journal of Intelligent Information Systems, 17:107-145.

Hämäläinen, W. and Vinni, M. (2010). Classifying educational data: special problems and guidelines, pages 57-74. Chapman & Hall/CRC Data Mining and Knowledge Discovery Series. CRC Press.

Han, J. and Kamber, M. (2006). Data Mining: Concepts and Techniques. Elsevier/Morgan Kaufmann, second edition.

Hand, D., Mannila, H., and Smyth, P. (2002). Principles of Data Mining. Cambridge, Massachussetts, USA: MIT Press.

Hinneburg, A., Hinneburg, E., and Keim, D. (1998). An efficient approach to clustering in large multimedia databases with noise. In Agrawal, R., Stolorz, P., and Piatetsky-Shapiro, G. (Eds.), Proceedings of the 4th International Conference in Knowledge Discovery and Data Mining (KDD 98), pages 58-65. AAAI Press.

Huang, Z. (1998). Extensions to the k-means algorithm for clustering large data sets with categorical values. Data Mining and Knowledge Discovery, 2(3):283-304.

Huikkola, M., Silius, K., and Pohjolainen, S. (2008). Clustering and achievement of engineering students based on their attitudes, orientations, motivations and intentions. WSEAS Transactions on Advances in Engineering Education, 5(1):342-354.

Ichino, M. and Yaguchi, H. (1994). Generalized Minkowski metrics for mixed feature-type data analysis. IEEE Transactions on Systems, Man, and Cybernetics, 24(4):698-708.

Jain, A. (2010). Data clustering: 50 years beyond k-means. Pattern Recognition Letters, 31(8):651-666.

Jain, A. and Dubes, R. (1988). Algorithms for Clustering Data. Englewood Cliffs, NJ, USA: Prentice Hall.

Jain, A., Duin, P., and Mao, J. (2000). Statistical pattern recognition: a review. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(1):4-37.

Jain, A., Murty, M., and Flynn, P. (1999). Data clustering: a review. ACM Computing Surveys, 31(3):264-323.

Kamvar, S., Klein, D., and Manning, C. (2002). Interpreting and extending classical agglomerative clustering algorithms using a model-based approach. In Proceedings of 19th International Conference on Machine Learning (ICML-2002), pages 283-290.

Kantardzic, M. (2011). Data Mining: Concepts, Models, Methods, and Algorithms. New Jersey: John Wiley & Sons, IEEE Press, 2nd edition.

Karypis, G., Han, E.-H., and Kumar, V. (1999). Chameleon: hierarchical clustering using dynamic modeling. Computer, 32(8):68-75.

Khan, L. and Luo, F. (2005). Hierarchical clustering for complex data. International Journal on Artificial Intelligence Tools, 14(5).

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. Biological Cybernetics, 43:59-69.

Käser, T., Busetto, A., Solenthaler, B., Kohn, J., von Aster, M., and Gross, M. (2013). Cluster-based prediction of mathematical learning patterns. In Lane, H., Yacef, K., Mostow, J., and Pavlik, P. (Eds.), Proceedings of the 16th international conference on Articial Intelligence in Education, volume 7926 of Lecture Notes in Computer Science, pages 389-399, Berlin, Heidelberg: Springer.

Lopez, M., Luna, J., Romero, C., and Ventura, S. (2012). Classification via clustering for predicting final marks based on student participation in forums. In Yacef, K., Zaiane, O., Hershkovitz, H., Yudelson, M., and Stamper, J. (Eds.), Proceedings of the 5th International Conference on the Educational Data Mining, pages 148-151. http://www.educationaldatamining.org/.

Luxburg, U. (2007). A tutorial on spectral clustering. Statistics and Computing, 17(4):395-416.

Murtagh, F. and Raftery, A. (1984). Fitting straight lines to point patterns. Pattern Recognition, 17:479-483.

Nugent, R., Dean, N., and Ayers, E. (2010). Skill set profile clustering: The empty k-means algorithm with automatic specification of starting cluster centers. In R.S.J.d Baker, A. Merceron, R.B. and Pavlik, P. J. (Eds.), Proceedings of the 3rd International Conference on Educational Data Mining, pages 151-160. http://www.educationaldatamining.org/.

Pardos, Z. A., Trivedi, S., Heffernan, N. T., and Srkzy, G. N. (2012). Clustered knowledge tracing. In Cerri, S., Clancey, W., Papadourakis, G., and Panourgia, K. (Eds.), Proceedings of the 11th international conference on Intelligent Tutoring Systems, volume 7315 of Lecture Notes in Computer Science, pages 405-410, Berlin, Heidelberg: Springer.

Parsons, L., Haque, E., and Liu, H. (2004). Subspace clustering for high dimensional data: a review. ACM SIGKDD Explorations Newsletter - Special issue on learning from imbalanced datasets, 6(1):90-105.

Pelleg, D. and Moore, A. (2000). X-means: Extending k-means with efficient estimation of the number of clusters. In Proceedings of the Seventeenth International Conference on Machine Learning, pages 727-734, San Francisco: Morgan Kaufmann.

Perera, D., Kay, J., Koprinska, I., f, K. Y., and Zaiane, O. (2009). Clustering and sequential pattern mining of online collaborative learning data. IEEE Transactions on Knowledge and Data Engineering, 21(6):759-772.

Schmitt, N., Oswald, F., Kim, B., Imus, A., Merritt, S., Friede, A., and Shivpuri, S. (2007). The use of background and ability profiles to predict college student outcomes. Journal of Applied Psychology, 92(1):165.

Schölkopf, B., Smola, A., and Müller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. Neural Computation, 10:1299-1319.

Sheikholeslami, G., Chatterjee, S., and Zhang, A. (1998). Wavecluster: A multi-resolution clustering approach for very large spatial databases. In Gupta, A., Shmueli, O., and Widom, J. (Eds.), Proceedings of the 24rd International Conference on Very Large Data Bases (VLDB'98), pages 428-439. Morgan Kaufmann.

Smyth, P. (2000). Model selection for probabilistic clustering using crossvalidated likelihood. Statistics and Computing, 10(1):63-72.

Stanford, D. and Raftery, A. (2000). Principal curve clustering with noise. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22:601-609.

Trivedi, S., Pardos, Z., and Heffernan, N. (2011a). Clustering students to generate an ensemble to improve standard test score predictions. In Biswas, G., Bull, S., Kay, J., and Mitrovic, A. (Eds.),

Proceedings of the 15th international conference on Artificial Intelligence in Education, volume 6738 of Lecture Notes in Computer Science, pages 377-384, Berlin, Heidelberg: Springer.

Trivedi, S., Pardos, Z., Sarkozy, G., and Heffernan, N. (2011b). Spectral clustering in educational data mining. In Pechenizkiy, M., Calders, T., Conati, C., Ventura, S., Romero, C., and Stamper, J. (Eds.), Proceedings of the 4th International Conference on Educational Data Mining, pages 129-138. http://www.educationaldatamining.org/.

Wang, X. and Hamilton, H. (2003). DBRS: A density-based spatial clustering method with random sampling. In Advances in Knowledge Discovery and Data Mining, Proceedings of the 7th Pacific-Asia Conference PAKDD 2003, volume 2637 of Lecture Notes in Computer Science, pages 563-575, Berlin, Heidelberg: Springer.

Ward, J. (1963). Hierarchical grouping to optimize an objective function. Journal of the American Statistical Association, 58:236-244.

Wilson, D. and Martinez, T. (1997). Improved heterogeneous distance functions. Journal of Artificial Intelligence Research, 6:1-34.

Xu, R. and Wunsch, D. (2005). Survey of clustering algorithms. IEEE Transactions on Neural Networks, 16(3):645-678.

Xu, R. and Wunsch, D. (2008). Clustering. IEEE Press Series on Computational Intelligence. John Wiley/IEEE Press.

Yona, G., Dirks, W., and Rahman, S. (2009). Comparing algorithms for clustering of expression data: how to assess gene clusters. Methods in Molecular Biology, 541:479-509.

Zhang, T., Ramakrishnan, R., and Livny, M. (1997). BIRCH: A new data clustering algorithm and its applications. Data Mining and Knowledge Discovery, 1(2):141-182.

Zhong, S. and Ghosh, J. (2003). A unified framework for model-based clustering. Journal of Machine Learning Research, 4:1001-1037.

Additional reading section

Aggarwal, C. and Reddy, C. (2013). Data Clustering: Algorithms and Applications. Chapman & Hall/CRC Data Mining and Knowledge Discovery Series. Boca Raton: Chapman & Hall/CRC.

Aggarwal, C. and Zhai, C. (2012). A survey of text clustering algorithms. In Aggarwal, C. and Zhai, C. (Eds.), Mining Text Data, pages 77-128. Boston: Kluwer Academic Publishers.

Berkhin, P. (2006). Survey of clustering data mining techniques. In Kogan, J., Nicholas, C., and Teboulleeds, M. (Eds.), Grouping Multidimensional Data: Recent Advances in Clustering, pages 25-71. Springer.

Duda, R., Hart, P., and Stork, D. (2000). Pattern Classification. New York: Wiley-Interscience Publication, 2nd edition.

Estivill-Castro, V. (2002). Why so many clustering algorithms? A position paper. SIGKKD Explorations, 4(1):65-75.

Everitt, B. (2011). Cluster Analysis. Wiley Series in Probability and Statistics. John Wiley & Sons Ltd., 5th edition.

Fraley, C. and Raftery, A. (2000). Model-based clustering, discriminant analysis, and density estimation. Journal of the American Statistical Association, 97:611-631.

Gan, G., Ma, C., and Wu, J. (2007). Data Clustering: Theory, Algorithms, and Applications. ASA-SIAM Series on Statistics and Applied Probability. SIAM.

Ghosh, J. (2004). Scalable clustering methods for data mining. In Ye, N. (Ed.), Hand Book of Data Mining, chapter 10. Lawrence Erlbaum Associates.

Halkidi, M., Batistakis, Y., and Vazirgiannis, M. (2001). On clustering validation techniques. Journal of Intelligent Information Systems, 17:107-145.

Han, J. and Kamber, M. (2006). Data Mining: Concepts and Techniques. Elsevier/Morgan Kaufmann, second edition.

Hand, D., Mannila, H., and Smyth, P. (2002). Principles of Data Mining. MIT Press, Cambridge, Massachussetts, USA.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer Series in Statistics. Springer-Verlag, 2nd edition.

Huikkola, M., Silius, K., and Pohjolainen, S. (2008). Clustering and achievement of engineering students based on their attitudes, orientations, motivations and intentions. WSEAS Transactions on Advances in Engineering Education, 5(1):342-354.

Jain, A. (2010). Data clustering: 50 years beyond k-means. Pattern Recognition Letters, 31(8):651-666.

Jain, A. and Dubes, R. (1988). Algorithms for Clustering Data. Englewood Cliffs, NJ, USA: Prentice Hall.

Jain, A., Murty, M., and Flynn, P. (1999). Data clustering: a review. ACM Computing Surveys, 31(3):264-323.

Kantardzic, M. (2011). Data Mining: Concepts, Models, Methods, and Algorithms. John Wiley & Sons, IEEE Press, New Jersey, 2nd edition.

Kaufman, L. and Rousseeuw, P. (1990). Finding Groups in Data: An Introduction to Cluster Analysis. Wiley Series in Probability and Statistics. Wiley-Interscience.

Kolatch, E. (2001). Clustering algorithms for spatial databases: A survey. Technical report, University of Maryland, Department of Computer Science.

Lopez, M., Luna, J., Romero, C., and Ventura, S. (2012). Classification via clustering for predicting final marks based on student participation in forums. In Yacef, K., Zaiane, O., Hershkovitz, H., Yudelson, M., and Stamper, J. (Ed.), Proceedings of the 5th International Conference on the Educational Data Mining, pages 148-151. http://www.educationaldatamining.org/.

Luxburg, U. (2007). A tutorial on spectral clustering. Statistics and Computing, 17(4):395-416.

Mirkin, B. (2012). Clustering: A Data Recovery Approach. Chapman & Hall/CRC Computer Science and Data Analysis Series. Chapman and Hall, Boca Raton, 2nd edition.

Nugent, R., Dean, N., and Ayers, E. (2010). Skill set profile clustering: The empty k-means algorithm with automatic specification of starting cluster centers. In R.S.J.d Baker, A. Merceron, R. B. and Pavlik, P. J. (Eds.), Proceedings of the 3rd International Conference on Educational Data Mining, pages 151-160. http://www.educationaldatamining.org/.

Parsons, L., Haque, E., and Liu, H. (2004). Subspace clustering for high dimensional data: a review. *ACM SIGKDD Explorations Newsletter - Special issue on learning from imbalanced datasets*, 6(1):90-105.

Trivedi, S., Pardos, Z., and Heffernan, N. (2011a). Clustering students to generate an ensemble to improve standard test score predictions. In Biswas, G., Bull, S., Kay, J., and Mitrovic, A. (Eds.), Proceedings of the 15th international conference on Artificial Intelligence in Education, volume 6738 of Lecture Notes in Computer Science, pages 377-384, Berlin, Heidelberg. Springer.

Trivedi, S., Pardos, Z., Srkzy, G., and Heffernan, N. (2011b). Spectral clustering in educational data mining. In Pechenizkiy, M., Calders, T., Conati, C., Ventura, S., Romero, C., and Stamper, J. (Eds.), *Proceedings of the 4th International Conference on Educational Data Mining*, pages 129-138. www.educationaldatamining.org.

Xu, R. and Wunsch, D. (2005). Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3):645-678.

Xu, R. and Wunsch, D. (2008). Clustering. *IEEE Press Series on Computational Intelligence*. John Wiley/IEEE Press.

Zhong, S. and Ghosh, J. (2003). A unified framework for model-based clustering. *Journal of Machine Learning Research*, 4:1001-1037.

Key terms and definitions

Centroid: A representative point of a cluster in its center.

Clustering: A grouping of data points, where points in one group are similar or close to each other but different or distant from points in the other groups.

Dendrogram: A tree diagram for representing a hierarchy of clusters.

Density function: A function that specifies a continuous probability distribution.

Kernel: A function that returns an inner product between the images of data points in some space.

Mixture model: A probabilistic model which is a combination of simpler models.

Outlier: A data point which does not belong to any cluster.