Novel Trends in Clustering

Claudia Plant, Christian Böhm
Technische Universität München, Munich Germany,
Ludwig Maximilians Universität München, Munich, Germany

ABSTRACT

Clustering or finding a natural grouping of a data set is essential for knowledge discovery in many applications. This chapter provides an overview on emerging trends within the vital research area of clustering including subspace and projected clustering, correlation clustering, semi-supervised clustering, spectral clustering and parameter-free clustering. To raise the awareness of the reader for the challenges associated with clustering, the chapter first provides a general problem specification and introduces basic clustering paradigms. The requirements from concrete example applications in life sciences and the web provide the motivation for the discussion of novel approaches to clustering. Thus, this chapter is intended to appeal to all those interested in the state-of-the art in clustering including basic researchers as well as practitioners.

KEYWORDS

Data Mining
Data Mining Algorithms
Knowledge Discovery
Clustering
Database

INTRODUCTION

In many applications, for example in medicine, life sciences, physics and market observation, terabytes of data is collected every day. Consider for example metabolite profiling (Baumgartner & Graber 2008). As an evolving branch of life sciences, Metabolomics studies the highly complex metabolism of cells, tissues, organs and organisms. One major focus of research is on identifying subtle changes related to disease onset and progression. Small molecules involved in primary and intermediate metabolism are called metabolites. Metabolite profiling provides techniques to quantify the amount of metabolites in a sample. Due to recent advances of high-throughput technologies such as tandem mass spectrometry (MS/MS) hundreds of metabolites can be detected from a single blood sample. As a second example consider web usage. For each user accessing a page, the corresponding web server logs information including IP address, time of access, file path, browser and amount of transferred data. Huge volumes of web server log data is generated every day and its potential for commercial and non-commercial applications such as designing online shops or providing users with personalized content in digital libraries (Zaiane & al. 1998) is far from being fully exploited.

In both applications scenarios, extraction of information from the massive amounts of data is a non-trivial, highly challenging task. In both scenarios we want to learn unknown regularities and structure in the data with very little previous knowledge. In metabolite profiling, we want to gain novel insights how certain diseases change the pattern of metabolites. Simple statistic tests often applied in biomedicine can provide valuable information. However, only a tiny part of the information potentially available in the data can be accessed but large parts remain unexplored. There may be several sub-types of the disease each associated with a unique pattern of altered metabolism. Also in the healthy controls there may be different types of normal yet unexplored metabolic patterns. Similarly, in the second scenario we want to find groups of users with similar behavior to provide them personalized content.

As an important area within data mining, clustering aims at partitioning the data into groups such that the data objects assigned to a common group called cluster are as similar as possible and the objects assigned to different clusters differ as much as possible. With the term ‘data objects’ we denote the instances subjected to a cluster analysis. Often, data objects can be represented as a feature vectors. In the scenario of metabolite profiling, the data objects are the subjects. Each subject is represented by a vector composed of the amounts of the measured metabolites. The dimensionality of the resulting feature space equals the number of metabolites. Alternatively, it could also be interesting to cluster the metabolites in the space defined by the subjects with the objective to identify groups of metabolites having similar prevalence across subjects.

Figure 1 displays examples of different types of clusters in vector data. The simplest type of a cluster is a spherical Gaussian. An example in two-dimensional space is depicted in Figure 1(a). Both coordinates follow a Gaussian distribution and are statistically independent from each other. As we will see in the next section, basic clustering algorithms can reliably detect such clusters. More complicated are correlation clusters with orthogonal major directions, as depicted in Figure 1(b). The objects of this cluster follow a line in one dimensional space which is characterized by a strong linear dependency between the coordinates. In addition, the major directions of the cluster are orthogonal and can be detected by Principal Component Analysis. Figure 1(c) displays a non-linear correlation cluster. There exists a distinct dependency between the two coordinates but this dependency cannot be captured by a linear model. Linear and non-linear correlation clusters frequently occur in high dimensional vector data, such as metabolic data. Typically, such clusters exist in a subset of the dimensions only. Figure 1(d) displays a non-Gaussian correlation cluster with non-orthogonal major directions. Besides the cluster, there are some outliers. Noise points and outliers are common in real world data.

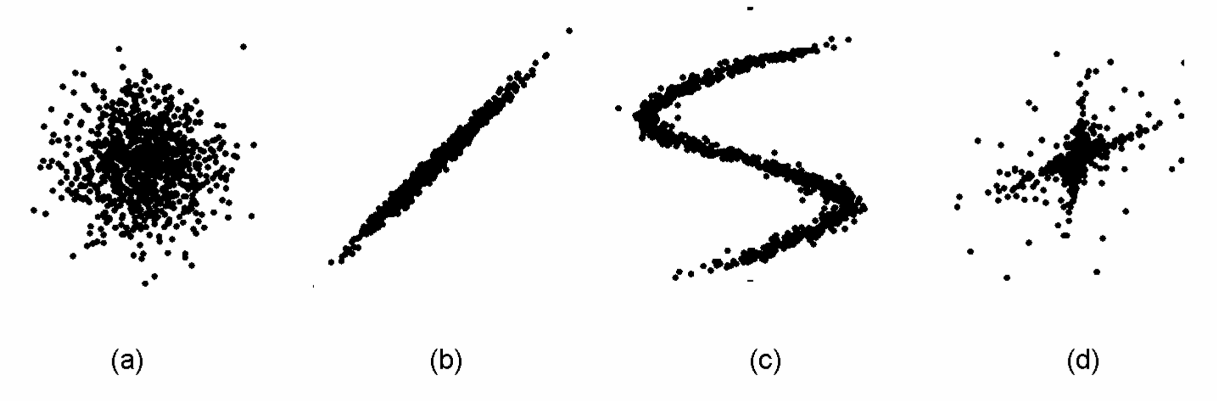

Figure 1. Different types of clusters in vector data: (a) spherical Gaussian; (b) correlation

cluster; (c) non-linear correlation cluster; (d) non-Gaussian cluster with non-orthogonal major directions.

This chapter provides a survey on novel trends in clustering. We will especially focus on highlighting the conceptual similarities and differences among approaches. In addition, a special focus will be on the applicability to real-world problems. Thus, we hope to provide conceptual survey which is valuable and maybe even inspiring for different groups of readers: scientists and students but also for practitioners looking for solutions in a concrete application. We can only provide a very incomplete snapshot focusing on some current and emerging vital trends and already apologize for all the important approaches which are missing.

We will discuss novel algorithms which are suitable to detect clusters in high dimensional feature space, including linear and non-linear correlation clusters with orthogonal and non-orthogonal major directions in noisy real world data, as depicted in Figure 1(b)-(d). We will focus not only on vector data but also introduce solutions for clustering other types of data, for example graphs or data streams. For example in the internet scenario, it is interesting to cluster the users according to their behavior. The file path allows tracking the path of users within a website and can be represented as a graph. For online shop design, it would be interesting to cluster the users based on their file paths. The result could help to improve the structure of a website for specific groups of customers.

Most approaches to clustering require defining a suitable representation of the data objects, for example as feature vectors or graphs together with a notion of object similarity. In most applications, this is challenging and a mathematical similarity measure which fully agrees with the needs of the application may not even exist. In information retrieval, this problem is commonly referred to as the *semantic gap*. External side information which is often available in the form of expert knowledge can be very helpful to cope with this problem. For example it is known from literature that some metabolites are similar since they fulfill a common function in the organism. Semi-supervised clustering, an emerging research area which has recently attracted much attention focuses on integrating such side information into clustering. We will discuss some interesting solutions. Besides a suitable representation of the data objects and a notion of similarity, most clustering algorithms require input parameters which are often difficult to estimate, for example the number of desired clusters. We will introduce some recent approaches to parameter-free clustering which avoid crucial parameters by the application of information-theoretic concepts.

BACKGROUND

To make this chapter self-contained, and to illustrate some of the challenges associated with clustering, we will briefly discuss two fundamental clustering paradigms: iterative partitioning clustering and hierarchical density-based clustering. These two paradigms introduce two very different cluster notions which have been taken up and further elaborated by many other approaches.

For illustration and comparison, we introduce iterative partitioning clustering on the algorithm K-means (Duda & Hart 1973). K-means requires a metric distance function in vector space. In addition, the user has to specify the number of desired clusters K as an input parameter. Usually K-means starts with an arbitrary partitioning of the objects into K clusters. After this initialization, the algorithm iteratively performs the following two steps until convergence: (1) Update centers: For each cluster, compute the mean vector of its assigned objects. (2). Re-assign objects: Assign each object to its closest center. The algorithm converges as soon as no object changes its cluster assignment during two subsequent iterations. In most cases, fast convergence can be observed. The optimization function of K-means is well defined. The algorithm minimizes the sum of squared distances of the objects to their cluster centers. This optimization goal coincides well with our definition of the clustering problem provided in the beginning: Objects assigned to a common cluster should be as similar as possible. The second aspect of the definition that objects in different clusters should differ as much as possible is implicitly addressed at the same time. However, finding a global minimum of the objective function is a NP-hard problem (see for example Meila 2008). The objective function is non-linear and non-convex, which implies that no efficient algorithm can be provided to detect the global minimum exactly. K-means converges to a local minimum of the objective function in a very acceptable time frame. In many cases, the result is close to optimal and K-means is thus among the most wide-spread clustering algorithms. In practice it is useful to try different random initializations and keep the best result. There are many algorithms following the K-means paradigm, perhaps most importantly the expectation maximization (EM) algorithm (Dempster et al. 1997) to detect Gaussian mixture models with fuzzy cluster assignment. As K-means, the EM algorithm consists of two steps which are iterated until convergence: (1) Update centers to maximize the log-likelihood of the data (this step is often called M-step where M stands for maximization) and (2) assign objects proportionally to their likelihood to all centers (this step is called E-step where E stands for expectation, since the expected value of the log-likelihood is computed). Another important branch of iterative partitioning clustering consists of K-medoid methods such as PAM (Partitioning around Medoids) (Kaufmann & Rousseeuw 1990) or CLARANS (Clustering Large Applications based on Randomized Search) (Ng & Han 1994). Instead of the mean, these methods select objects from the data set as cluster representatives. Therefore, also non-vector metric data can be clustered. However, selecting a suitable K is major problem with all these algorithms.

To avoid the problems with parameterization and local minima, hierarchical or density-based clustering can be an attractive alternative. Single Link (Jain & Dubes 1988) is the basic algorithm for hierarchical clustering. As result, this algorithm produces a tree-style visualization of the hierarchical cluster structure which is often called dendrogram. At the lowest level of the hierarchy all objects are represented as singleton clusters. In each step the closest pairs of objects are merged to form a cluster at the next higher level. Besides the distance function, no parameterization is required. There is no objective function to be minimized and the result is determinate. The runtime of Single Link is quadratic in the number of objects, which is acceptable in most applications. However, the result is often hard to interpret, especially for large data sets, since the visualization is the only output of the algorithm. If a partitioning into distinct clusters is desired, it is difficult to find a suitable level for horizontally cutting the hierarchy. In the presence of noise objects, the so-called Single Link effect can occur: Clusters may get connected by a chain of noise objects. Popular variants of Single Link which are somewhat less prone to the Single Link effect are Average Link and Complete Link. These algorithms introduce distance functions between sets of objects for clustering. In Average Link the average distance between two sets of objects is applied, in Complete Link the maximum distance.

Strongly related to the cluster notion of Single Link is the idea of density-based clustering. In density-based clustering, clusters are regarded areas of high object density which are separated by areas of lower object density. The algorithm DBSCAN (Density-Based Spatial Clustering of Applications with Noise) proposed in (Ester et al. 1996) formalizes this idea by two parameters: *MinPts* specifying a number of objects and ε specifying a volume. An object is called *core object* if it has at least *MinPts* objects within its ε neighborhood. If one object *P* is in the ε-neighborhood of a core-object *Q*, then *P* is said to be *directly density reachable* from *Q*. The *density-connectivity* is the symmetric, transitive closure of the *direct density reachability*, and a *density-based* (*ε, MinPts*)*-cluster* is defined as a maximal set of density-connected objects. It can be proven that an (*ε*, *MinPts*)*-cluster* can be detected by collecting all density reachable objects starting from an arbitrary core object which is implemented in DBSCAN. See Figure 2 for an illustration of the definitions of DBSCAN. As K-means, DBSCAN determines a non-hierarchical, disjoint partitioning of the data set into clusters. However, the number of clusters does not need be specified in advance and the algorithm is robust against noise objects. Nevertheless, the number of clusters detected by DBSCAN depends on the choice of the parameters ε and *MinPts*. For some data sets even no suitable parameterization exists, for example in the case of various object densities in different areas of the data space or in the case of a hierarchical cluster structure. To cope with these problems, the algorithm OPTICS (Ordering Points to Identify the Clustering Structure) (Ankerst et al. 1999) has been proposed which is a hierarchical extension of DBSCAN but also related to Single Link. The main idea of OPTICS is to compute all possible clusterings for varying ε simultaneously during a single traversal of the data set. The output of OPTICS is a linear order of the data objects according to their hierarchical cluster structure which is visualized in the so-called reachability-plot. OPTICS is equivalent to Single-Link if the *MinPts*-Parameter of OPTICS is set to one. OPTICS avoids the Single Link effect if *MinPts* is set to larger values.

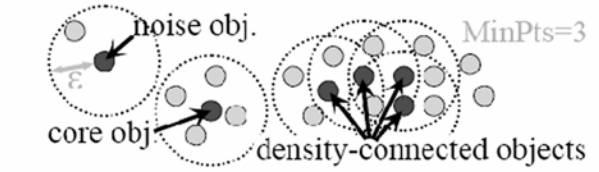

Figure 2. Definitions of DBSCAN.

We introduced K-means and Single Link as representatives of two different clustering paradigms. It is important to point out that clustering algorithms differ in their cluster notion. K-means provides a well defined objective function which intuitively coincides with our idea of clustering. But this specific objective function also narrows the type of clusters which can be detected: Only spherically shaped Gaussian clusters are captured by this definition. Single Link has a more general cluster notion: Arbitrarily shaped dense areas of the feature space are regarded as clusters. As discussed, the algorithms further differ in the need for parameter settings and the type of the result. When introducing recent clustering paradigms, we will always re-visit these points since they are essential for the choice of a suitable clustering algorithm for a certain application.

Figure 3 (a) displays a simple two dimensional data set consisting of three Gaussian clusters. For comparison, Figure 3(b) displays the result of the iterative partitioning EM algorithm and Figure 3(c) the result of the hierarchical algorithm Single Link on this data set. Correctly parameterized with K=3 the result of EM clustering is a Gaussian mixture model with a good fit to the data. This result includes location and variance of each cluster which is often important for interpretation. The result of Single Link is a hierarchical visualization of the cluster structure. The three clusters are clearly visible in the dendrogram. However, the dendrogram always implies a hierarchical structure, even if there is no distinct cluster hierarchy in the data as in this example. In contrast to the result of EM, the dendrogram does not provide a model on the data. But note that Single Link does not require any input parameters whereas the result of EM strongly depends on a suitable parameterization. In summary, let us point out that both algorithms are suitable to detect the clusters in this simple toy example. In addition, the choice of a specific algorithm always comes along with characteristic benefits and drawbacks.

Detecting clusters in real world data is usually much more difficult. The data are often represented in a high dimensional sparse feature space. Commonly, the data contains noise objects and features which are not relevant for clustering. Or the data has a complex non-vector structure. In the following, we will see that basic algorithms like K-means, EM, Single Link and DBSCAN are not sufficient to cope with the challenges emerging from modern applications, such as our example scenarios in life sciences and web usage. We will discuss some of the most vital research topics in clustering addressing the major challenges of clustering real world data.

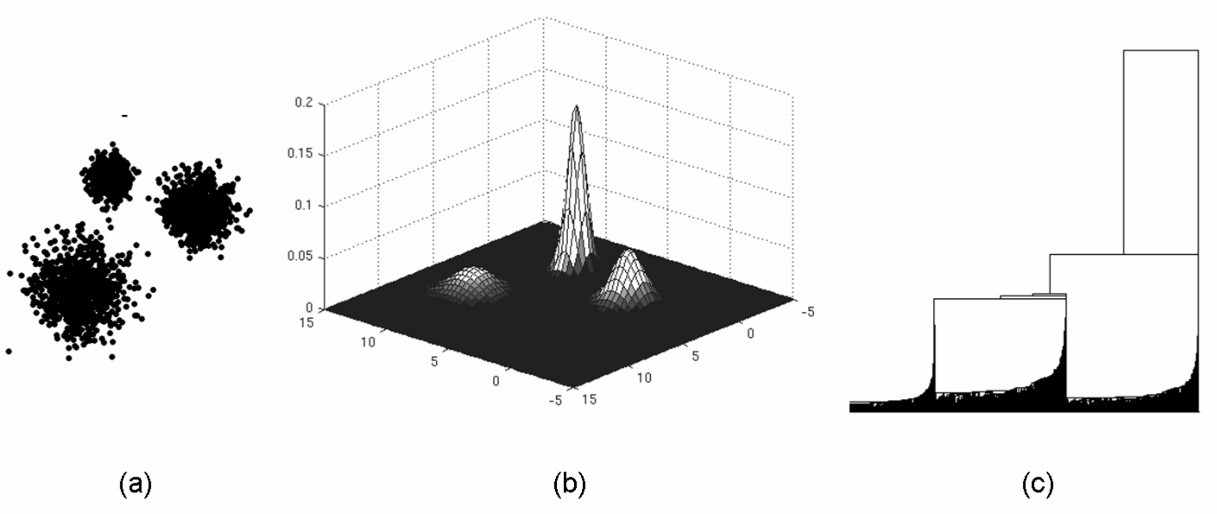

Figure 3. Clustering basics: (a) example data set with three spherical Gaussian clusters; (b) Gaussian mixture model obtained by EM clustering; (c) dendrogram obtained by Single Link.

SUBSPACE CLUSTERING AND PROJECTED CLUSTERING

In metabolite profiling, hundreds of quantified metabolites can be detected from the serum samples of thousands of subjects. In this scenario, the data are represented in a very high dimensional vector space, regardless if we intend to cluster the subjects or the metabolites. When clustering such data with the basic algorithms as K-means or Single Link we face problems related to the so-called *curse of dimensionality*. The curse of dimensionality subsumes all the strange effects in high dimensional vector spaces which make clustering, indexing and classification difficult. With increasing dimensionality, more and more objects are located at the boundaries of the feature space and the distances between objects assimilate, for details see for example (Weber et al. 1998). The curse of dimensionality can already be visualized at the transition from one to two dimensional spaces, as Figure 4(a) demonstrates. In one dimensional space there are two distinct clusters, denoted by C1 and C2. Adding a single noise dimension significantly worsens the cluster structure. When adding more dimensions, we often see no cluster structure in the dendrogram. The result of K-means and EM is very unstable over different runs with different random initializations, since no distinct minimum of the objective function exists. The curse of dimensionality can significantly affect the result of basic clustering algorithms starting already at a moderate dimensionality of about ten. Global dimensionality reduction like Principal Component Analysis (PCA) can be applied to transform the data to a lower dimensional space. However, at least for clustering, PCA is not the best choice to cure the curse of dimensionality. PCA preserves the overall variance in the data. This implies, if the data has high variance without cluster structure in the full dimensional space, the clustering can not be much better in a subspace selected by PCA.

Subspace clustering is a better solution. Many high dimensional data sets exhibit rich cluster structures in axis-parallel subspaces which are spanned by subsets of the features. In our metabolite scenario, clusters of patients can only be identified in a subspace spanned by a subset of the metabolites. In addition, patients can be clustered differently in different subspaces. See Figure 3 (b) and (c) for an illustration of this effect. The subspace clustering problem is highly complex since a vector space of dimensionality d has 2^d axis-parallel subspaces. Thus, an exhaustive search for the best subspaces is infeasible in higher dimensions. The algorithm CLIQUE (Clustering in Quest) (Agrawal et al. 1998), the fundamental approach to subspace clustering, introduces a monotonic criterion for object density which allows effective pruning in combination with bottom-up search. To define the density criterion, the data space is partitioned by an axis-parallel grid into equal-sized units of width ε. Only units whose densities exceed a threshold τ are retained. A cluster is defined as a maximal set of connected dense units. This cluster notion allows effective pruning of the search space using the upwards monotonicity of the density criterion: Only subspaces containing dense units may be part of a higher dimensional subspace cluster. Successive modifications of CLIQUE with similar algorithmic paradigm include ENCLUS (Entropy-based Subspace Clustering) (Cheng et al. 1998) and MAFIA (Merging Adaptive Finite Intervals) (Nagesh et al. 2000). A drawback of these methods is the use of grids. In general, the efficiency and the accuracy of these approaches heavily depend on the positioning and resolution of the grids. Objects that naturally belong to a cluster may be missed or objects that are naturally noise may be assigned to a cluster due to an unfavorable grid position. The algorithm SUBCLU (Density-connected Subspace Clustering) (Kailing et al. 2004) avoids this problem by defining a monotonic density criterion relying on concepts of density-based clustering without grids. More precisely, the core object property of DBSCAN is applied in the definition of the density criterion. As a grid cell, the core object property is upwards monotonic.

Recently proposed algorithms like SURFING (Subspaces Relevant for Clustering) (Baumgartner et al. 2004), SCHISM (Support and Chernoff-Hoeffding Bound-based Interesting Subspace Miner) (Sequeira & Zaki 2004) and DUSK (Dimensionality-unbiased Subspace Clustering) (Assent et al. 2007) refrain from a cluster notion based on a monotonic density criterion, since this leads to major problems. With increasing dimensionality, the object density naturally decreases. Therefore, a fixed density criterion implicitly specifies the dimensionality of subspace clusters which can be detected by the algorithm. Parameterization of such algorithms is difficult. Subspace clusters of various dimensionalities cannot be detected in a single run of the algorithm. However, with the fixed density criterion, also the monotonicity property is dropped and heuristic search strategies are required. SURFING and DUSK propose criteria to rate the interestingness of subspaces for clustering based on statistics of the data distribution which allow detecting subspace clusters of various dimensionalities. SCHISM also employs a variable density criterion and heuristic search but uses a grid-based data quantization as CLIQUE. All described algorithms apply bottom-up search, which implies that at least parts of the subspace clusters must be visible in the one dimensional subspaces.

An alternative solution to the curse of dimensionality in clustering is projected clustering. In contrast to subspace clustering, algorithms for projected clustering assign each data object to only one distinct cluster and determine the best subspace, or the best projection, for this cluster. Instead of performing a bottom-up search, most algorithms for projected clustering start in the full dimensional space and, similar to K-means, iteratively optimize some objective function. The first approach to projected clustering is the algorithm PROCLUS (Projected Clustering) (Aggarwal et al. 1999). Provided with the input parameters K and *l*, PROCLUS returns a partitioning of the data into K clusters having an average dimensionality *l*. Objects not fitting well to any of the clusters are assigned to noise. To achieve comparability of distances among objects assigned to subspace clusters of different subspace dimensionality, the Manhattan distance is normalized by the subspace dimensionality. In the initialization stage, PROCLUS selects initial medoids as cluster representatives from the data objects. In the iterative phase, the set of medoids and their associated subspaces are refined using a greedy hill-climbing technique. After this iterative search, an additional pass over the data is performed for refinement of clusters, medoids and associated subspaces.

Procopiuc et al. (2002) introduce an alternative notion of projected clusters, called optimal projective clusters, together with an algorithm called DOC (Density-based Optimal Projective Clustering) to find such clusters. An optimal projective cluster is defined using two parameters: α specifying a fraction of the data objects and ω specifying the width of a hypercube. An optimal projected cluster is defined as a set of points C associated with a subspace of dimensions D such that C contains more than α% points of the objects and the projection of C onto the subspace spanned by D is contained in a hyper-cube of width ω. Based on Monte-Carlo sampling, the algorithm DOC returns approximations of the optimal projected clusters. DOC does not require the user to specify K, but the parameterization remains difficult for the same reasons as discussed for subspace clustering with a fixed density criterion. In addition, the cluster notion of DOC does not care about the data distribution inside the hypercube representing a cluster. Not always the hypercube contains exactly one cluster, it may contain several clusters, noise points and empty space as well. The algorithm PreDeCon (Subspace Preference Weighted Density- connected Clustering) (Böhm et al. 2004) partially avoids the problems of difficult parameterization and undesired behavior in the presence of noise by extending the basic concepts of density-based clustering to projected clustering. In this approach, the notion of subspace preference clusters is introduced which are defined as density-based clusters in subspaces. The dimensions of the subspace of a cluster are determined by selecting directions of low variance within the ε neighborhood of core objects.

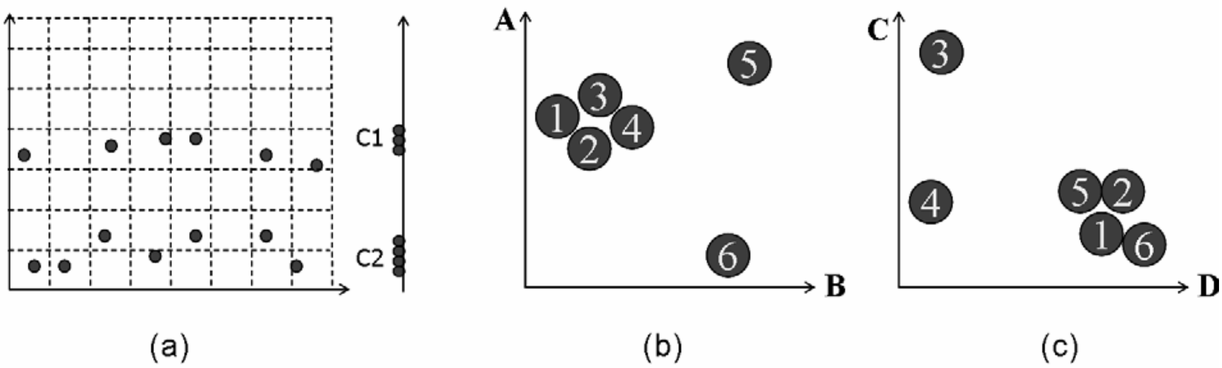

Figure 4. Subspace and Projected Clustering. (a) The curse of dimensionality: In one dimensional space there are two distinct clusters C1 and C2. In two dimensional space the cluster structure is blurred due to one noise dimension; (b) and (c): Objects can be differently clustered in different subspaces. For example, in the subspace spanned by the features A and B, object 1 is clustered together with the objects 2, 3, and 4. In the subspace spanned by C and D, object 1 clustered together with the objects 2, 5, and 6. This information is preserved in subspace clustering but not in projected clustering.

CORRELATION CLUSTERING

The detection of correlations between different features in a given data set is a very important data mining task. High correlation of features may result in a high degree of co-linearity or even a perfect one, corresponding to approximate linear dependencies between two or more attributes. These dependencies can be arbitrarily complex, one or more features might depend on a combination of several other features. In the data space, dependencies of features are manifested as lines, planes, or, generally speaking, hyper-planes exhibiting a relatively high density of data objects compared to the surrounding space. See Figure 1(b) for an example of a correlation cluster in two-dimensional space. Knowing of correlations is traditionally used to reduce the dimensionality of the data set by eliminating redundant features. However, detection of correlated features may also help to reveal hidden causalities which are of great interest to the domain expert. Recently, correlation clustering has been introduced as a novel concept of knowledge discovery in databases to detect dependencies among features and to cluster data objects sharing a common pattern of dependencies. It corresponds to the marriage of two widespread ideas: First, correlation analysis usually performed by Principle Component Analysis (PCA) and, second, clustering which aims at identifying local subgroups of data objects sharing high similarity. Correlation clustering groups the data set into subsets called correlation clusters such that the objects in the same correlation cluster are all associated to a common hyper-plane of arbitrary dimensionality. In addition, many algorithms for correlation cluster analysis also require the objects of a cluster to exhibit a certain density, i.e. feature similarity. When comparing correlation clustering with subspace and projected clustering, we can observe, that in correlation clustering, the clusters exist in an arbitrarily oriented subspace rather than in an axis-parallel one. Therefore, correlation clustering is sometimes also referred to as *generalized subspace clustering*.

Correlation clustering has been successfully applied to several application domains. For example, customer recommendation systems are important tools for target marketing. For the purpose of data analysis for recommendation systems, it is important to find homogeneous groups of users with similar ratings in subsets of the attributes. In addition, it is interesting to find groups of users with correlated affinities. This knowledge can help companies to predict customer behaviour and thus develop future marketing plans. In molecular biology, correlation clustering is an important method for the analysis of several types of data. In metabolic screening, the collected data usually contain the concentrations of certain metabolites in the blood of thousands of patients. In such data sets, it is important to find homogeneous groups of patients with correlated metabolite concentrations indicating a common metabolic disease. Thus, several metabolites can be linearly dependent on several other metabolites. Uncovering these patterns and extracting the dependencies of these clusters is a key step towards understanding metabolic or genetic disorders and designing individual drugs. A second example where correlation clustering is a sound methodology for data analysis in molecular biology is DNA micro-array data analysis. Micro-array data comprise the expression levels of thousands of genes in different samples such as experimental conditions, cells or organisms. Roughly speaking, the expression level of a gene indicates how active this gene is. The recovering of dependencies among different genes in certain conditions is an important step towards a more comprehensive understanding of the functionality of organisms which is a prominent aspect of systems biology. When the samples represent some patients, it is important to detect homogeneous groups of persons exhibiting a common linear dependency among a subset of genes in order to determine potential pathological subtypes of diseases and to develop individual treatments.

One of the first approaches to correlation clustering (there called generalized projected clustering) was ORCLUS (for Oriented Clustering), proposed in (Aggarwal & Yu, 2000). As a variant of K-means, this algorithm requires the user to specify in advance the number of clusters K, as well as the subspace dimensionality of the clusters, l. Like K-means, the algorithm performs a loop until convergence, in which points are assigned to clusters and cluster representatives are re-determined. In ORCLUS, the cluster representatives correspond to the mean vectors, complemented by a linear subspace which is determined according to the Eigenvectors of the associated points. A number l_c of these Eigenvectors is selected where l_c decreases in each iteration step from full dimensionality to the user-specified final dimensionality l. In addition, the number of clusters is also decreased from step to step by a user-defined parameter α. The general idea of ORLCUS is that each cluster is associated to its own individual subspace which is arbitrarily oriented. This set of subspaces is searched by first partitioning the data set into spherical clusters (of full dimensionality) and then successively stripping off noisy dimensions. In the case of clearly separated correlation clusters this method performs well. However, when correlation clusters touch each other, the initial search in full dimensionality may be misleading, and the algorithm may easily fail to separate neighbouring clusters with different subspace orientation.

An approach particularly addressing this problem is 4C (Computing Clusters of Correlation-connected Objects), as proposed in (Böhm et al. 2004). 4C is founded on the paradigms of density-based partitioning clustering (for DBSCAN, cf. the background section) and Principal Component Analysis (PCA). A point P is considered as a core object if its ε-neighbourhood not only contains a sufficiently high number (parameter *MinPts*) of neighbouring points but also has a dimensionality of at most l, as determined by the PCA of the neighbouring points. To allow imperfect correlations, a dimension is considered flat if the corresponding Eigenvalue is below a specified threshold d. Each core object is then associated to an ellipsoid which exactly reflects the extension of the neighbouring points: It is extended (with radius ε) in l dimensions and flat (with radius ε/κ where κ is usually set to $1/d$) in the remaining dimensions. Two objects are directly density reachable from each other if both objects are mutually located in their ε-neighbourhood with respect to their associated ellipsoids. The remaining definitions of correlation connection and correlation reachability are then defined like the density connection and density reachability, respectively in the DBSCAN algorithm (symmetric and transitive closure of the direct correlation reachability). A correlation cluster is a maximal set of correlation density connected objects. Like the DBSCAN algorithm, 4C starts by selecting an arbitrary starting object, determining whether it is a core object and if so, putting its neighbours into a queue called seed list. The cluster is extended by iteratively taking neighbours out of the seed list and performing them in the same way, until the seed list is empty. Then a new, unprocessed object is selected as the next starting object, until all objects have been considered. The algorithm is well able to handle difficult situations where points of different clusters are close together. Likely, the points located at the boundary, will not be assigned to any of the correlation clusters (because their neighbourhood is not flat in a sufficient number of dimensions) but the remaining database points are not touched by this problem. The difficulty with this method is the selection of the radius ε: If ε is chosen too large, then too many points which actually belong to different correlation clusters are located in the neighbourhood, and an incorrect subspace dimension may be determined. In contrast, if ε is chosen too small, no core objects may be found. A hierarchical extension of 4C called HiCO which is based on the hierarchical density-based clustering notion of OPTICS (cf. the background section) and PCA has also been proposed (Achtert et al. 2006). A further extension, DiSH (Achtert et al. 2007) considers more general, network-based relationships between correlation clusters.

In (Tung et al. 2006) the authors propose CURLER, a method for finding and visualizing clusters even with a nonlinear correlation. CURLER uses the concept of micro-clusters that are generated using a variant of EM clustering. The micro-clusters are merged to discover correlation clusters. The merging of micro-clusters is based on the concept of co-sharing, i.e. the co-sharing level of two micro-clusters corresponds to the total amount of points which are associated to both clusters simultaneously. From the set of all micro-clusters, an arbitrary starting object is selected, and, like in OPTICS or Single Link, in each step that cluster is selected as the next micro-cluster in the micro-cluster order which has the highest co-sharing level to the starting object or any of the previously placed micro-clusters. CURLER also defines a visualization technique similar to the reachability-plot of OPTICS. CURLER improves over ORCLUS and 4C as the correlations underlying the clusters are not necessarily linear. Thus, also clusters like the example in Figure

1(c) can be detected. Furthermore, as a fuzzy approach, CURLER assumes each data object to belong to all clusters simultaneously, but with different probabilities for each cluster assigned. By merging several clusters according to their co-sharing level, the algorithm on the one hand becomes less sensitive to the predefined number K of clusters, thus also overcoming a severe limitation of any K-means related approach. On the other hand, the user cannot directly derive a model describing the correlations, since the original K models are no longer present in the resulting clustering. Therefore, one can say that the algorithm can handle arbitrary non-linear correlations but is not able to identify the type of correlation.

The most important problem of all approaches previously presented in this section is the missing robustness with respect to noise objects and correlation clusters which are close to each other. All the applied techniques, iterative methods, density-based methods and micro-clusters are to some degree sensitive to the situation when the correlation cannot be determined by looking at the full-dimensional local neighbourhood of an object. The algorithm CASH (Clustering in Arbitrary Subspaces based on the Hough transform) presented in (Achtert et al. 2008) tackles this problem by a sophisticated parameter space transformation: Instead of clustering the points directly, the idea is first to replace each point by the (infinite) set of all possible, λ-dimensional planes in which the point is contained. For low (two or three) dimensional applications from image processing domains, this idea is already well-known, and denoted by the term Hough transform. In these low dimensional domains, the infinite set of all lines or planes can easily be discretely represented by a finite selection of some possible sample planes (for example sampled by discrete angles). Matching planes can easily be determined by counting of accumulators, where each accumulator corresponds to one possible plane. In higher dimensional spaces, a sufficient discretization of all possible planes is not possible due to the curse of dimensionality. But the set of all possible planes passing through a point can also be represented by a function. If the planes are represented by Euclidean coordinates, these functions are linear functions. If the planes are represented by spherical coordinates (which is advantageous because orthographic planes cannot be represented in Euclidean coordinates) then the functions are trigonometric (combinations of sine and cosine functions). Points sharing a common plane are determined by those trigonometric functions which have a common intersection at an arbitrary point of the parameter space. See Figure 5 for an illustration of the transformation to parameter space. The algorithm CASH decomposes the trigonometric functions recursively until a point in the parameter space is found in which many trigonometric functions coincide. That means, even if different planes intersect each other or are hidden in a noisy environment, the points in the parameter space representing the correlation clusters exists and can be found. With increasing dimensionality the number of required decompositions of the trigonometric functions increases. Therefore, also the processing time increases, but is still of cubic order in the number of dimensions. Of all algorithms presented in this section, CASH reveals least sensitivity with respect to noise objects and intersecting clusters.

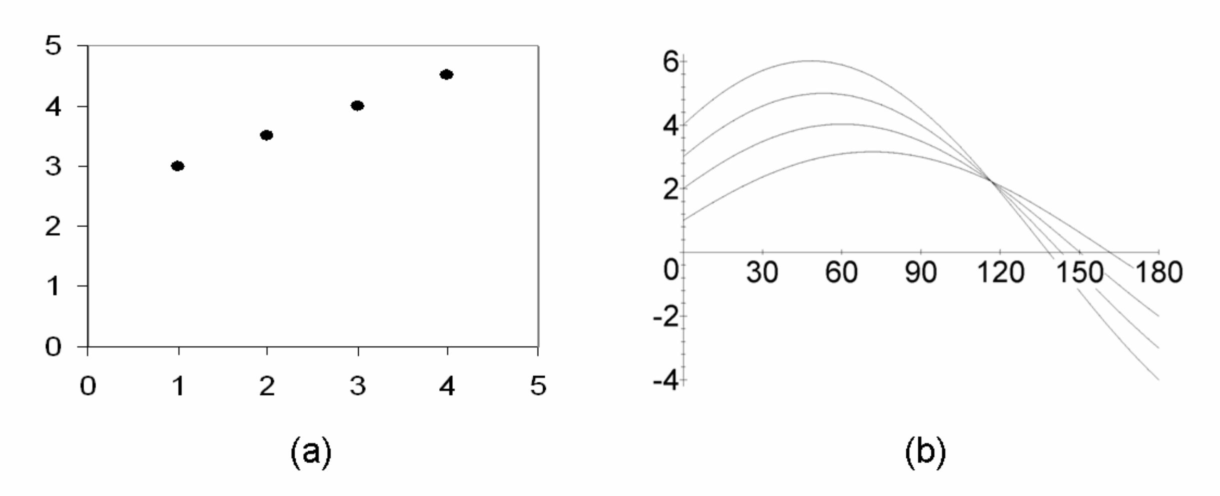

Figure 5. Transformation of the objects into parameter space for correlation clustering as performed by the algorithm CASH: (a) correlation cluster in original space; (b) representation of the objects as trigonometric functions in parameter space.

SEMI-SUPERVISED CLUSTERING

Semi-supervised clustering is an emerging area which evolved from an important need of numerous applications: integrating side-information or supervision into clustering. Semi-supervised clustering may be beneficial for bridging the so-called semantic gap in human-computer interaction. Often, purely feature-based similarity measures can not satisfactorily represent the complex human notion of similarity. Consider for example clustering web pages, which may contain images, text and hyperlinks. It is difficult to define an appropriate similarity measure based on the content of the web pages. As a valuable source of side information, ratings of users on the similarity of some selected pages can be integrated into clustering. As an example from biomedicine, consider clustering of gene expression data. The expression levels of thousands of genes can be measured simultaneously using the micro-array technology. Clustering is often applied to detect functionally related genes. There are two major challenges associated with clustering gene expression data: First, the curse of dimensionality, which can be addressed by subspace or projected clustering. Secondly, there exists a huge amount of side information on functionally related genes, where most of this knowledge is contained in publications available in biomedical literature databases. Similar to the webpage example, the side-information is often incomplete, i.e. there are genes for which we have expression measurements but no literature information. As demonstrated in (Zeng et al. 2007), semi-supervised clustering exploiting both sources of knowledge, the feature information of gene expression as measured in the micro-array experiment and the literature information outperforms conventional clustering.

Most algorithms for semi-supervised clustering model potentially incomplete side-information by a set of constraints. Must-link constraints are established between pairs of objects which should be clustered together, for example two genes which are known to be functionally related from literature, and cannot-link-constraints are established between objects which should be assigned to different clusters. This set of constraints is then incorporated into a modified clustering algorithm. Following the classification of clustering algorithms introduced in the background section, we can assign most algorithms for semi-supervised clustering into the classes of iterative partitioning and hierarchical density-based algorithms.

The first iterative partitioning algorithm for semi-supervised clustering is COP-K-means introduced in (Wagstaff et al 2001). COP-K-means extends K-means with must-link and cannot-link constraints formulated as described above. The cluster assignment step of the basic K-means algorithm is modified as follows: Objects with cannot-link constraints must be assigned to different clusters. If this is not possible, i.e. if no cluster exists to host an object, the algorithm aborts. Note that supervision is only used for the cluster assignment of the objects for which side information is available, also often called the *labeled* objects, and not for the other objects, called the *un-labeled* objects. Shental et al. (2003) propose a constrained version of the EM algorithm. Must-link constraints can be respected by a modification of the E- step: For two objects with a must-link constraint, only the proportion of probability respecting the constraint is considered. The incorporation of cannot-link-constraints is more complex since they are not transitive. To incorporate both types of constraints into EM, a Markov network is applied. Because of the probabilistic nature of the EM algorithm, supervision not only affects the clustering of the labeled objects, but implicitly also the clustering of the un-labeled objects. The MPCK-means (Metric Pairwise Constraint K-means) algorithm proposed in (Bilenko et al. 2004) integrates constraints and metric learning into K-means and thereby explicitly extends the influence of supervision to the un-labeled objects. For each cluster, Euclidean distance is parameterized with a weight matrix which is updated in each iteration of the algorithm. The update rule for the weight matrix considers the constraint violations inside a cluster proportionally to their severity. If for example the cluster contains two cannot-link objects which are very close together, the metric needs to be altered more drastically as if these points are already far away from each other. In the second case it is likely that the objects are assigned to different clusters in the next iteration even without any metric change. Although each cluster has its own associated weight matrix, in the update step global metric learning is performed by the linear transformation best representing the metric changes in all clusters. In (Basu et al. 2004) this idea is theoretically liked to Hidden Markov Random Fields (HMRF) and extended to the Bregman divergences, a wide range of distance functions including cosine similarity for text data. The foundation on HMRF allows defining a kernel for semi-supervised graph clustering (Kulis et al. 2005).

Due to the different cluster notion, hierarchical and density-based algorithms to semi-supervised clustering integrate supervision information in different ways. The first algorithm in this category is the CCL (Constrained Complete Link) algorithm proposed by (Klein et al. 2002) which integrates constraints into Complete Link clustering. Before clustering, the constraints are used to modify the distance matrix between objects: the distance between must-link objects is set to zero. To preserve the metric property of the data space, distances between other objects are adjusted by a shortest-path algorithm. The distance between cannot-link objects is set to a value larger than the maximal distance occurring in the data set. Cannot-link constraints are implicitly propagated during clustering. The result of the CCL algorithm is of other type (dendrogram) but besides this very similar to the result of COP-K-means. Both algorithms perform clustering without violating any of the constraints. This rigid preservation of constraints can lead to very unnatural clustering results if the constraints do not agree well with the data distribution. Consider for example the data set displayed in Figure 6. The labeled objects are visualized by larger symbols and the clustering results are marked by different colors and are additionally annotated. The result of COP-K-means in Figure 6(a) only respects the constraints and not the data distribution. The recently proposed hierarchical density-based algorithm HISSCLU (Böhm & Plant 2008) alleviates this problem by proposing a different way of incorporating supervision into density-based clustering. Instead of for formulating pair-wise must-link and cannot-link constraints, the labeled objects are utilized as seeds for cluster expansion. Density-based clusters are expanded starting at each labeled object simultaneously. During this cluster expansion process, labels are propagated to the un-labeled objects. In areas of the feature space where conflicts caused by differently labeled objects exist, a local distance weighting is applied. Similar to MPCK-means, HISSCLU achieves more natural results in the case of inconsistent information provided by the data and the supervision (for comparison Figure 6(b) displays the result of HISSCLU and Figure 6(c) the result of HISSCLU). Experiments demonstrate that, especially in the case of some wrongly labeled objects (for example originating from erroneous user ratings), the local metric adaptation applied in HISSCLU outperforms the global metric learning scheme of MPCK-means. In addition, HISSCLU provides a visualization of the hierarchical cluster structure which displays how consistent both sources of information actually are.

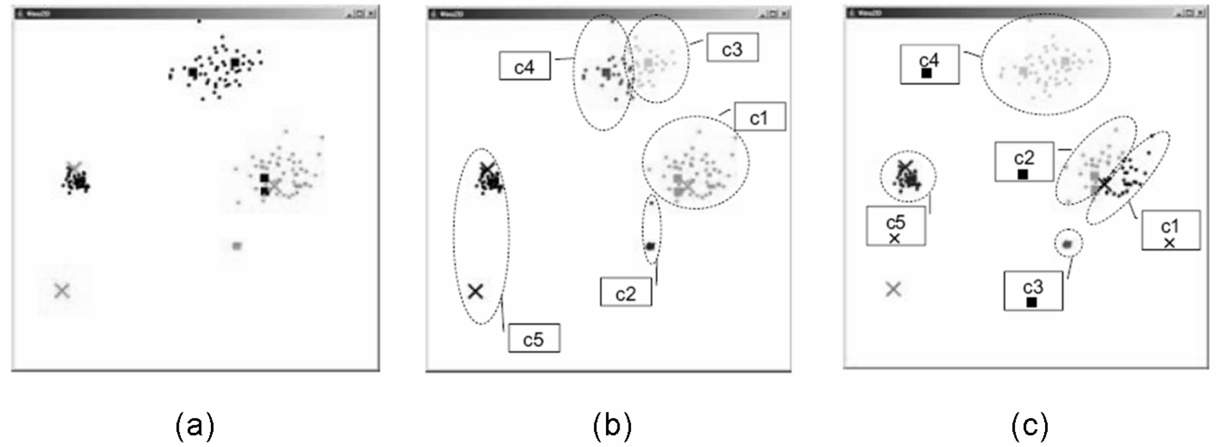

Figure 6. Results of Semi-supervised Clustering. The example consists of three clusters containing six labeled objects of two different classes. (a) COP-K-means: unnatural clustering respecting the constraints; (b) MPC-K-means: Although it contains uniformly labeled objects, the cluster on top is split into two parts (c3 and c4) due to the limitations of K-means; (c) HISSCLU: Most natural result by density-based clustering and local distance weighting (Figure from Böhm & Plant 2008).

In this section, our major focus is on algorithms closely interrelating supervision and clustering. It should be mentioned that there are several recent approaches focusing on methods for global (Xing et al. 2003, Bar-Hillel et al. 2003) or local (Chang & Yeung 2004) metric learning from supervision information which can be used prior to an arbitrary clustering algorithm. Yip et al. (2005) propose a semi-supervised algorithm for projected clustering which is an interesting option to cope with the curse of dimensionality. Their algorithm SSPC (Semi-supervised Projected Clustering) considers not only supervision for objects in the form of class labels but also supervision for dimensions. This type of supervision is modeled by specifying dimensions which are relevant for certain classes. The algorithmic paradigm of SSPC is similar to the algorithms PROCLUS and ORCLUS with some modifications. Most importantly, labeled objects and labeled dimensions are used for initialization and the parameter *l* specifying the average cluster dimensionality is replaced by a more intuitive parameter specifying the maximum variance for relevant dimensions.

SPECTRAL CLUSTERING

Spectral clustering considers the clustering problem from the perspective of graph theory. The data is provided by a similarity matrix from which a weighted graph is constructed. For many types of data the graph representation is very natural, for example for social networks where the nodes are different people and the edges represent friendship. The clustering problem is to find a partitioning of the graph such that the edges between different clusters have a very low weight. Different objective functions for a good partitioning have been proposed. If for example a partitioning into two clusters is desired, the Minimum Cut objective function (minCut) just removes the edge having the lowest weight. The minCut problem can easily be solved (Stoer & Wagner 1997) but leads to an undesired partitioning in many cases for example by simply removing a single outlying vertex from the graph. One option to circumvent this problem is to request that the clusters should be reasonably large. This is implemented in the objective function Ratio Cut (rCut) first introduced by (Hagen & Kahng 1992) which considers the ratio between the weight of the cut edges and the size of the resulting clusters. As an alternative, the objective function Normalized Cut (nCut) (Shi & Malik 2000) considers the connectivity between clusters, expressed by the weight of the cut edges as in rCut, but in relation to the within cluster density which is expressed by the sum of weights of the edges within the clusters. Several further objective functions for balanced graph partitioning have been proposed, for example minMaxCut (Ding et al. 2001) which sums up the weights within each cluster separately and thus strikes to obtain individual clusters of high object density. Introducing balancing constraints however makes the graph partitioning problem NP-hard (Wagner & Wagner 1993).

Spectral clustering proposes algorithms to solve relaxed forms of the balanced graph partitioning problem. Most algorithms for spectral clustering such as (Shi & Malik 2000, Ng et al. 2002) follow a similar paradigm. After creating a weighted similarity graph from data, the Laplacian of this graph is constructed. The similarity graph is represented by a symmetric adjacency matrix A. The unnormalized Laplacian of a graph is obtained by subtracting the adjacency matrix from the degree matrix, i.e. $L = D-A$. The spectrum of the Laplacian obtained by Eigenvalue decomposition has interesting properties for clustering, for example the number of constant eigenvectors coincides with the number of connected components of a graph. Usually, we have a fully connected graph in clustering. In this case, the clusters can be detected by mapping the data objects to the space spanned by the first K eigenvectors and performing standard K-means. It can be proven that this procedure yields an approximation of balanced graph partitioning. The algorithms differ in the ways if and how the Laplacian is normalized. Thereby, different objective functions can be optimized. Recall that the objectives of clustering are two-fold: First, the objects in different clusters should be as dissimilar as possible and secondly the objects within one cluster should be as similar as possible. The simplest case of no normalization of the Laplacian addresses only the first goal. Performing K-means in the Eigenvector space approximates the rCut objective function which only considers balance in the number of objects which is achieved by K-means. To explicitly require within cluster similarity, normalization with the inverted degree matrix is required. The algorithm proposed in (Shi & Malik 2000) approximates the optimal nCut by decomposing a normalized Laplacian defined as $D^{-1/2} L$. It can be proven that minMaxCut can be approximated by the same generalized Eigenproblem. Ng et al. (2002)

propose symmetric normalization of the Laplacian, i.e. $D^{-1/2}$ L $D^{-1/2}$. This paper also provides derivations of spectral clustering from the perspectives of perturbation theory and random walks on graphs. Figure 7 provides an example of non-Gaussian vector data clustered with this algorithm. It becomes evident that the transformation of the clustering problem to the graph-cut perspective allows detecting arbitrarily shaped clusters.

Dhillon (2001) and Zha et al. (2001) propose algorithms for simultaneously clustering documents and words of a word-document co-occurrence matrix based on the spectral clustering idea. Documents and words are arranged in a bipartite graph and objective functions similar to nCut for bipartite graph partitioning are introduced. A more general framework for spectral clustering of multi-type relational data is introduced in (Long et al. 2006). This method allows simultaneously clustering objects of multiple types which are related to each other, for example web pages, queries and web users. Technically this is achieved by collective factorization of related matrices.

Unlike many algorithms for partitioning clustering, spectral clustering requires no assumptions on the data distribution and the algorithms can be easily implemented using standard linear algebra packages. However, the result strongly depends on the construction of the similarity matrix and a suitable choice of the number of clusters K. Bach and Jordan (2003) propose a technique for metric learning to construct the similarity matrix together with a novel algorithm approximating nCut by weighted K-means in Eigenvector space. Zelnik-Manor and Perona (2004) propose guidelines for parameter settings. Another limitation of spectral methods is that they require decomposing an *n x n* matrix for a data set of *n* points and are therefore not suitable for very large data sets. Fowlkes et al. (2004) propose a sampling-based method to approximate spectral clustering of large data sets. As an alternative, Dhillon et al. (2004) propose a weighted kernel-K-means algorithm to minimize Ncut without matrix decomposition.

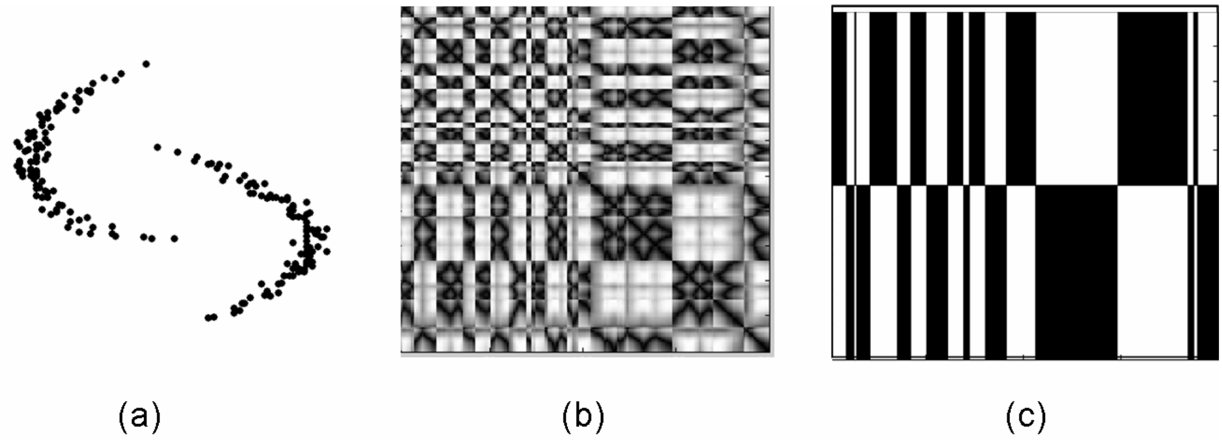

Figure 7. Spectral Clustering following Ng et al. (2002): (a) Data set with two non-linear correlation clusters; (b) Color coded visualization of the normalized Laplacian; (c) Second largest Eigenvector provides cluster indicators which can be trivially separated by bisecting K-means.

PARAMETER-FREE CLUSTERING

Most approaches to clustering introduced so far suffer from a common problem: To obtain a good result, the user needs to select suitable values for parameters such as the number of clusters K in K-means and related approaches, density thresholds or neighborhood sizes in density-based clustering, or the subspace dimensionality in projected clustering. In practice, the best way to cope with this problem often is to run the algorithm several times with different parameter settings. Thereby, suitable values for the parameters can be learned in a trial and error fashion. However, this process is very time consuming. Confronted with a large data set to be clustered only a few trials are feasible, or suitable parameter settings need to be estimated from a small sample. Anyhow, it cannot be guaranteed that at least useful values for the parameters are obtained by this procedure. The large number of parameters required in many algorithms even conflicts with the unsupervised nature of the clustering problem as introduced in the beginning. The goal of clustering is to find a natural grouping of the data without requiring any background knowledge. But without background knowledge, it is often very difficult to specify appropriate parameter settings.

Recently, parameter-free clustering therefore has attracted increasing attention. Most approaches, such as the algorithms X-Means (Pelleg & Moore 2001), G-Means (Hammerly & Elkan 2003) and OCI (Böhm et al. 2008) are founded on information theory and closely related concepts. The basic idea is to relate clustering to data compression. Assume that data consisting of feature vectors should be transferred via a communication channel from a sender to a receiver. Without clustering, each coordinate needs to be fully coded by transforming the numerical value into a bit string. If the data exhibits regularities, clustering can drastically reduce the communication costs. For example the EM algorithm can be applied to determine a model for the data. With this model, the data can be compressed very effectively since only the deviations from the model need to be encoded which requires much less bits than the full coordinates. In addition, the model itself needs to be encoded and transferred. The model can be regarded as a codebook which allows the receiver to de-compress the data again. This basic idea, often referred as the Minimum Description Length Principle (MDL) (Grünwald 2005) allows comparing different clusterings: Assume we have two different clusterings A and B of the same data set. We can state that A is better than B if it allows compressing the data set more effectively than B. Note that we consider the overall communication cost comprising data and model here and not only the code length spent for the data. Thereby we achieve a natural balance between the complexity of the model and its fit to the data. Closely related ideas developed by different communities include the Bayesian Information Criterion (BIC), the Aikake Information Criterion (AIC) and the Information Bottleneck method (IB) (Tishby et al. 2000).

A first line of papers are based on the Information Bottleneck method. The fundamental idea behind IB is compressing only the relevant characteristics of the data which leads to a lossy compression. To judge relevance, besides the data a second source of information is required which is often called the *auxiliary variable*. Therefore, these approaches are also related to semi-supervised clustering but are discussed here because of their information theoretic foundation. Relevance is defined as the amount of information that the data provide about the auxiliary variable. The clustering problem is considered as finding a lossy compression of the data preserving as much information on the auxiliary variable as possible. IB is particularly useful for co-occurance data such as words and documents. In this context it is interesting to discover clusters of words which contain relevant information on the documents (Slonim & Tishby 2000, Dhillon et al. 2003). This principle has been extended by Slonim et al. (2001) to the multivariate information bottleneck technique which allows extracting different meaningful partitions of the data simultaneously. As demonstrated in (Tishby & Slonim 2000), the IB principle can also be applied for clustering general metric data. This algorithm first transforms the similarity matrix of the data into a Markov process by assigning a state of a Markov chain to each data object and the transition probability between states is defined as an exponential function of the pair-wise distances. A transition matrix is defined whose entries specify the transition probabilities between the different states. Using this transition matrix, a random walk with an infinite number of steps would provide no information on the starting point. For clustering, it is interesting to consider the information loss on the starting point for a random walk of some fixed number of steps *t*. The rate of information loss is slow if the random walk is stabilized by structures in the data, for example if it remains within one cluster. IB is applied to find a partitioning of the data into clusters which best predicts the information loss after *t* steps of random walk (which is the auxiliary variable in this case). However, this approach is not completely parameter-free since a suitable number of steps *t* needs to be selected.

The second category of approaches directly focus on parameter-free partitioning clustering and are based on MDL and related ideas such as BIC. For these methods, the data itself is the only source of knowledge. Information-theoretic arguments are applied for model selection during clustering and, in contrast to the approaches based on IB, these approaches involve a lossless compression of the data. The work of Still and Bialek (2004) provides important theoretical background by using information-theoretic arguments to relate the maximal number of clusters that can be detected by partitioning clustering with the size of the data set. The algorithm XMeans (Pelleg & Moore 2000) combines the K-means paradigm with the Bayesian Information Criterion for parameter-free clustering. XMeans involves an efficient top-down splitting algorithm where intermediate results are obtained by bisecting K-means and are evaluated with BIC. However, due to the properties of K-means, only spherically Gaussian clusters can be detected. The algorithm G-means (Gaussian means) introduced in (Hamerly & Elkan 2003) has been designed for parameter-free correlation clustering. G-means follows a similar algorithmic paradigm as XMeans with top-down splitting and the application of bisecting K-means upon each split. However, the criterion to decide whether a cluster should be split up into two is based on a statistical test for Gaussianity. Splitting continues until the clusters are Gaussian, which implies of course, that non-Gaussian clusters can not be detected. The algorithm PG-means (Projected Gaussian means) (Feng & Hamerly 2006) is similar to G-means but learns models with increasing K with the EM algorithm. In each iteration, various one dimensional projections of the data and the model are tested for Gaussianity. Experiments demonstrate that PG-means is less prone to over fitting than G-means. Figueiredo and Jain (2002) propose a parameter-free EM algorithm based on the MDL principle. In contrast to XMeans which applies BIC to evaluate intermediate results, an MDL-based model selection criterion is directly integrated into EM. Due to the properties of EM Gaussian data is assumed, but the algorithm can be supplied with a covariance matrix and thus supports the same cluster notion as G-means and PG-means.

It turns out that the underlying clustering algorithm and the choice of the similarity measure are already some kind of parameterization which implicitly comes with specific assumptions. The commonly used Euclidean distance for example assumes Gaussian data. In addition, the algorithms discussed so far are very sensitive with respect to noise objects or outliers. These problems are addressed by the recently proposed algorithm RIC (Robust Information-theoretic Clustering) (Böhm et al. 2006). This algorithm can be applied for post-processing an arbitrary imperfect initial clustering. This approach is based on MDL and introduces a coding scheme especially suitable for clustering together with algorithms for purifying the initial clusters from noise. The coding scheme for a cluster is illustrated in Figure 8. Each coordinate of each cluster is associated with a probability density function (PDF). Best compression can be achieved if the value of a coordinate is encoded with a bit string of length anti-proportional to its likelihood. In a first step, RIC removes noise objects from the initial clusters, and then merges clusters if this allows for more effective data compression. The algorithm can operate with arbitrary data distributions which can be described by PDFs. However, a fixed set of PDFs needs to be selected in advance. The algorithm OCI (Outlier-robust Clustering using Independent Components) (Böhm et al. 2008) provides parameter-free clustering of noisy data and allows detecting non-Gaussian clusters with non-orthogonal major directions as the example in Figure 1(d). Technically this is achieved by defining a very general cluster notion based on the Exponential Power Distribution (EPD) and by integrating Independent Component Analysis (ICA) into clustering. The EPD includes a wide range of symmetric distribution functions, for example Gaussian, Laplacian and uniform distributions and an infinite number of hybrid types in between. Beyond correlations detected by PCA which correspond to correlation clusters with orthogonal major directions, ICA allows to detect general statistical dependencies in data.

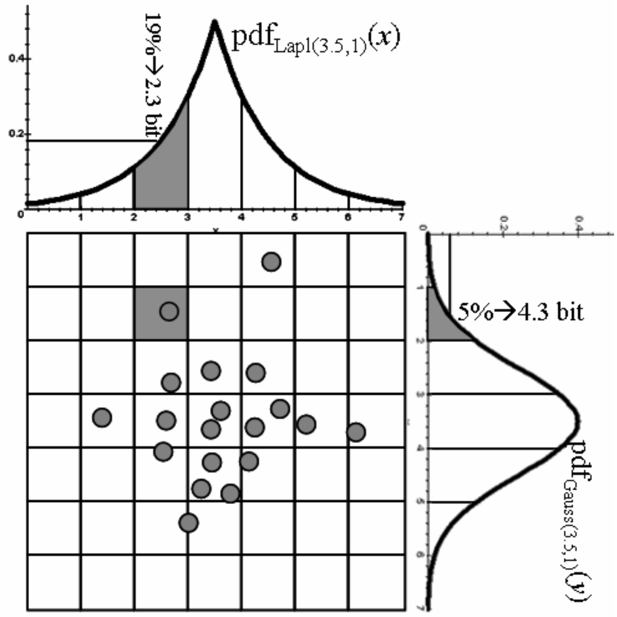
19%→2.3 bit
$\mathrm{pdf}_{\mathrm{Lapl}(3.5,1)}(x)$
0.4
0.2
0
1
2
3
4
5
6
7
x
0
0.2
0.4
5%→4.3 bit
1
2
3
4
5
6
7
$\mathrm{pdf}_{\mathrm{Gauss}(3.5,1)}(y)$

Figure 8.Parameter-free clustering. Coding scheme for cluster objects of the RIC algorithm. In addition to the data, type and parameters of the PDF need to be coded for each cluster. (Figure from Böhm et al. 2006).

FUTURE TRENDS

In this section, we point out some further trends from which we believe that they will attract even more attention in the future; one is clustering of uncertain data. Uncertainty is a natural element in many applications, for example due to the limited resolution and accuracy of data acquisition techniques or due to the application of aggregated features. Sometimes uncertainty is even willingly introduced, for example by adding small perturbations to the data to mask sensitive features in privacy-preserving data mining (Aggarwal 2007). Some recent papers focus on clustering uncertain data, such as (Hamdan & Govaert 2005, Kriegel & Pfeifle 2005, Ngai et al. 2006). These approaches model uncertain objects by probability density functions and propose adapted versions of partitioning clustering such as K-means (Ngai et al. 2006), EM (Hamdan & Govert 2005) and DBSCAN (Kriegel & Pfeifle 2005). A special case of uncertainty is clustering moving objects, for example for location-based services. In this case, specialized algorithms can be applied, for example (Li et al. 2004) since the location of the data objects is usually described in two-dimensional space but with an additional temporal dimension.

The temporal dimension is even more emphasized in clustering data streams. In a streaming environment the data objects are coming into the system at a very high rate, consider for example sensor measurements or the click streams of users on a website. The goal of clustering is thus to provide at any time a good clustering of the sequence of objects seen so far. Algorithms for clustering data streams have to be very efficient to keep pace with high-throughput streams. Usually, complete re-clustering is not feasible and efficient incremental algorithms are needed. Designing algorithms with limited memory usage is an additional challenge arising of the special properties of massive streams which do not allow storing all objects. Clustering of streaming data has recently attracted much attention with a lot of research papers, for example (O’Callaghan et al. 2003, Aggarwal et. al. 2003, Nasaroi et al. 2003), to mention a few. To cope with the special requirements of data streams, for example (Aggarwal et. al. 2003) introduces CluStream, a general framework. The fundamental idea of CluStream is to divide the clustering process into two steps. The micro-clustering phase involves efficient online collecting of statistical data from the stream. Initial micro-clusters are created by applying K-means to the first objects of the stream. Whenever a new object of the stream arrives, the micro-clusters are updated by either assigning the new object to one of the micro-clusters or creating a new cluster, if the object does not fit to any of the existing clusters. Each micro-cluster is associated with statistical information, including for example its radius. In the macro-clustering phase the final clustering result is determined with a variant of K-means. As input parameters, the user needs to specify the timeframe of interest *h* and the desired number of clusters K. Multiple runs of this algorithm allow a flexible exploration of stream clusters over various time frames. A very different approach to clustering noisy multidimensional streaming data has been proposed by (Nasraoui et al. 2003). Inspired by the immune system of living organisms, this paper introduces a model for an artificial immune system suitable for clustering. Data objects of the stream are regarded as antigens which are presented to the lymphocytes. In the immune system of living organisms, antigens are roughly speaking substances which are responsible to initiate an immune response by generating antibodies. If the amount of antibodies exceeds a certain threshold, the lymphocytes become active and kill infected cells. The artificial immune system model allows for automatically detecting clusters and outliers. Two interesting recent approaches for streaming data even addressing several challenges discussed in this chapter simultaneously are the work of Aggarwal et al. (2004) for projected clustering of high-dimensional data streams and the approach of Aggarwal and Yu (2008) for clustering uncertain streaming data. We are convinced that the special requirements of the streaming environment will continue to promote creative research in clustering.

There is a huge variety of approaches for clustering other special types of data. One interesting direction which we want mention is clustering graphs. Unlike spectral clustering focusing on detecting clusters within one graph, the objective here is to find clusters in a database of graph structured objects. Especially in biology and chemistry large amounts of graph structured objects are collected, representing for example chemical compounds or molecular structures. However, only relatively few papers, such as (Tsuda & Kudo 2006) focus on clustering such type of data.

Not only the amount of data collected in modern applications is rapidly increasing but also the structure of data becomes more and more rich, diverse and complex. Therefore, integrative clustering of information from different sources will continue to attract much attention. Related to semi-supervised clustering is the task of clustering of multi-represented objects. As discussed, algorithms for semi-supervised clustering typically consider only relatively simple types of side information such as constraints or labels. The goal of multi-represented clustering is integrative clustering of several equally complex sources. First approaches have been proposed for different underlying clustering paradigms, for example spectral clustering (De Sa 2005), the EM algorithm (Bickel & Scheffer 2004) and density-based clustering (Achtert et al. 2006).

The presentation of highly specialized methods for the needs emerging from the application side may lead to the impression that the research community working on clustering (which is anyhow split up into different sub-communities originating from data mining, databases, machine learning, statistics and physics) is continuously diversifying. But there is also a lot of effort on integration. Theoretical work on similarities or even equivalence of at first glance completely different clustering paradigms not only leads to interesting insights but can also result in substantial gains in effectiveness and efficiency. As mentioned, (Dhillon et al 2004) demonstrate the equivalence of the normalized cut objective function in spectral clustering with weighted kernel K-means. This allows more efficient spectral clustering without matrix decomposition. Song et al. (2007) provide a unified view of many clustering algorithms including K-means, spectral and hierarchical clustering, regarding the clustering problem as maximization of dependence between the data objects and their cluster labels. A formulation of this idea using the Hilbert-Schmidt Independence Criterion and kernel methods is elaborated. In addition, the authors provide guidelines for practical application. The trend towards a unified view is not restricted to clustering paradigms only, but also on integrating clustering and closely related techniques from mathematics and statistics, especially techniques for matrix factorization and dimensionality reduction. Ding and He (2004) explore the relationship between K-means clustering and Principle Component Analysis (PCA). Principle Components actually are the continuous solutions of the cluster membership indicators obtained by K-means. This result allows providing lower bounds on the optimality of K-means. In addition, K-means can significantly profit from PCA: PCA provides a good initialization for K-means, and there is a theoretical justification to apply PCA as a preprocessing for dimensionality reduction before K-means (at least for data of moderate to medium dimensionality). These examples demonstrate that the integrative view of different clustering paradigms and related techniques not only has a theoretical value but also has an impact on the application of clustering algorithms in practice. We believe that this research direction has great potential in the future.

But not only within clustering there is a trend towards unification. Clustering also fruitfully integrates into other related research areas. Within data mining and machine learning there are close relationships to the areas of classification and outlier detection. The evolving research area of semi-supervised learning is crossing the borders between traditional unsupervised clustering without external knowledge and classification, which is the classical task within supervised learning. The goal of outlier detection is to find the exceptional objects of a data set. To specify what exceptional or outstanding means in the context of the given data set, it is necessary to have an idea about what is normal or common. Therefore, outlier detection is closely related to clustering and we expect further interactions between these areas in the future. Going beyond data mining, we expect that there will be even more interaction of clustering with other research areas, for example information retrieval, indexing, parallel and distributed computing.

CONCLUSION

At first glance, the problem specification of clustering as introduced in the introduction seems to be very simple: Find a natural partitioning of the data into groups or clusters such that the objects assigned to a common cluster are as similar as possible and objects assigned to different clusters differ as much as possible. This very general problem specification is highly relevant in a large variety of applications, wherever an overview on huge amounts of data is desired. With the technological progress, larger amounts of data can be acquired and stored at decreasing costs. Thus, the practical relevance of clustering is constantly increasing. We have seen that clustering is indeed not a trivial task at all. Finding a natural grouping of a small set of objects may be easy for humans because of our advanced cognitive abilities, most importantly our ability to focus on relevant information and our ability to intuitively select a suitable level of abstraction. However, the problem size in real applications exceeds our processing capability by orders of magnitude. Thus, we need efficient and effective algorithms for automatically clustering large complex data sets. Recent developments in clustering are exactly addressing the following questions:

1. How can we automatically find out which part of the information potentially contained in the data actually is relevant for clustering?
2. How can we exploit the cognitive abilities of humans or other types of expert knowledge to improve the clustering result?
3. How can we automatically select a suitable level of abstraction in clustering?

Automatically selecting that information from data which is relevant for clustering is very challenging. If the data is represented in a high dimensional vector space, approaches to subspace and projected clustering provide solutions to this problem. Subspace clustering aims at automatically detecting interesting dimensions for clustering and preserves the information that objects can be clustered differently in different subspaces. Projected clustering detects clusters which are associated with a specific subspace where each object is exclusively assigned to one cluster. Clusters in real-world data are not restricted to axis-parallel subspaces, but can be associated with arbitrary linear or non-linear hyper-planes and subspaces. Correlation clustering focuses on detecting such clusters, which are characterized by specific patterns of linear or non-linear feature dependencies. Especially the result of subspace, projected and correlation clustering provides interesting insights on why objects are clustered together which is very important for interpretation. For example we can learn from correlation clustering of metabolic data that a specific pattern of linear dependency of metabolites is characteristic for certain disorder. For general metric data represented by a similarity matrix, spectral clustering algorithms are very suitable. Selecting relevant information for clustering in this context means learning a suitable similarity measure. Recent approaches propose techniques for automatically adjusting the similarity measure by metric learning to improve the cluster structure.

Semi-supervised approaches to clustering address the second question. These approaches demonstrate that the clustering result can be substantially improved by external side information. This side information is usually obtained by human experts or other sources of knowledge, such as literature databases. Most algorithms require side information only for very few data objects to obtain good results. Therefore, semi-supervised clustering provides the potential to combine the cognitive abilities of humans with the computing power of machines for clustering.

The third question is mainly addressed by approaches to parameter-free clustering. To be most informative to the user, the clustering result must have a suitable level of abstraction. The clustering should concisely summarize the important characteristics of the data without over fitting. For most clustering algorithms the resolution of the result depends on input parameters which are difficult to estimate. Approaches to parameter-free clustering automatically select a suitable level of abstraction by introducing ideas from information theory into clustering.

There are many challenges for clustering in the future, which cannot all be mentioned here. Definitely, there is a strong need for highly scalable techniques and for techniques which can combine data originating from different sources. The development of novel techniques will be promoted by the needs of novel applications. Clustering is an important step on the path from data to knowledge and will therefore continue attracting the attention of generations of researchers to come.

REFERENCES

Achtert, E., Böhm, C., David, J., Kröger, P. & Zimek, A. (2008). Robust Clustering in Arbitrarily Oriented Subspaces. In: *Proc. of SIAM International Conference on Data Mining (SDM)* (pp. 763-774).

Achtert, E, Böhm, C., Kriegel, H.-P., Kröger, P., Müller-Gorman, I. & Zimek, A. (2006). Finding Hierarchies of Subspace Clusters. In: *Proc. of European* Conference *on Principles and Practice of Knowledge Discovery (PKDD)* (pp. 446-453).

Achtert, E., Böhm, C., Kriegel, H.-P., Kröger, P., Müller-Gorman, I. & Zimek, A. (2007). Detection and Visualization of Subspace Cluster Hierarchies. In: *Proc. of International* Conference *on Database Systems for Advanced Applications (DASFAA*) (pp. 152-163).

Achtert, E., Kriegel, H.-P., Pryakhin, A. & Schubert, M. (2006). Clustering Multi-represented Objects Using Combination Trees. In: *Proc. of Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)* (pp. 174-178).

Aggarwal, C. C. (2007). On Density Based Transforms for Uncertain Data Mining. In*: Proc. of* IEEE *International Conference on Data Engineering (ICDE)* (pp. 866-875).

Aggarwal, C. C., Han, J., Wang, J. & Yu, P. S. (2003). A Framework for Clustering Evolving Data Streams. In: *Proc. of International Conference on Very Large Data Bases (VLDB)* (pp.81-92).

Aggarwal, C. C., Han, J., Wang, J. & Yu, P. S. (2004). A Framework for Projected Clustering of High Dimensional Data Streams. In: *Proc. of International Conference on Very Large Data Bases (VLDB)* (pp.852-863).

Aggarwal, C. C. & Yu, P. S. (2000). Finding Generalized Projected Clusters in High Dimensional Space. In: *Proc. of ACM SIGMOD International Conference on Management of Data (SIGMOD)* (pp. 70-81).

Aggarwal, C. C. & Yu, P. S. (2008). A Framework for Clustering Uncertain Data Streams. In: *Proc. of* IEEE *International Conference on Data Engineering (ICDE)* (pp. 150-159).

Aggarwal, C. C., Wolf, J. L., Yu, P .S., Procopiuc, C. & Park, J. S. (1999). Fast algorithms for projected clustering. In: *Proc. of ACM SIGMOD International Conference on Management of Data (SIGMOD)* (pp. 61-72).

Agrawal, R., Gehrke J., Gunopulos, D. & Raghavan, P. (1998). Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications. In: *Proc. of ACM SIGMOD International Conference on Management of Data (SIGMOD)* (pp. 94-105).

Assent, I., Krieger, R., Müller, E. & Seidl, T. (2007). DUSC: Dimensionality Unbiased Subspace Clustering. In: *Proc. of IEEE International Conference on Data Mining (ICDM)* (pp. 409-414).

Bach, F. & Jordan, M. (2003). Learning Spectral Clustering. In: *Proc. of 16th Annual Conference on Advances in Neural Information Processing Systems (NIPS).*

Baumgartner, C. & Graber, A. (2008). Data Mining and Knowledge Discovery in Metabolomics. In: Masseglia, F., Poncelet, P. Teisseire, M. (Eds.) *Successes and New Directions in Data Mining.* (pp. 141-166). PA: IBI Global.

Baumgartner C., Kailing K., Kriegel H.-P., Kröger P. & Plant C. (2004). Subspace Selection for Clustering High-Dimensional Data. In: *Proc. of IEEE International Conference on Data Mining (ICDM)* (pp.11-18).

Bar-Hillel, A., Hertz, T., Shental, N. & Weinshall, D. (2003). Learning Distance Functions using Equivalence Relations. In: *Proc. of International Conference on Machine Learning (ICML)* (pp. 11-18).

Basu, S., Bilenko, M. & Mooney, R. J. (2004). A Probabilistic Framework for Semi-supervised Clustering. In: *Proc. of ACM SIGKDD International* Conference *on Knowledge Discovery and Data Mining (KDD)* (pp. 59-68).

Bickel, S. & Scheffer, T. (2004). Multi-View Clustering. In: *Proc. of IEEE International Conference on Data Mining (ICDM)* (pp. 19-26).

Bilenko, M., Basu, S. & Mooney, R. J. (2004). Integrating Constraints and Metric Learning in Semi-supervised Clustering. In: *Proc. of International Conference on Machine Learning (ICML)* (pp. 81-88).

Böhm, C., Faloutsos, C. & Plant, C. (2008). Outlier-robust Clustering using Independent Components. In: *Proc. of ACM SIGMOD International Conference on Management of Data (SIGMOD)* (pp. 185-198).

Böhm C., Kailing K., Kröger P. & Zimek A. (2004). Computing Clusters of Correlation Connected Objects. In: *Proc. of ACM SIGMOD International Conference on Management of Data (SIGMOD)* (pp. 455-466).

Böhm, C. & Plant C. (2008). HISSCLU: A Hierarchical Density-based Method for Semi-supervised Clustering. In: *Proc. of International Conference on Extending Database Technology (EDBT)* (pp. 440-451).

Chang, H. & Yeung D.-Y. (2004). Locally Linear Metric Adaptation for Semi-supervised Clustering. In: *Proc. of International Conference on Machine Learning (ICML)* (pp. 153-160).

Cheng, C.-H., Fu, A.-C. & Zhang, Y. (1999). Entropy-Based Subspace Clustering for Mining Numerical Data. In: *Proc. of ACM SIGKDD International* Conference *on Knowledge Discovery and Data Mining (KDD)* (pp. 84-93).

De Sa, V. R. (2005). Spectral Clustering with two Views. In: *Proc. of International Conference on Machine Learning (ICML). Workshop on Learning with Multiple Views.*

Dempster, A. P., . Laird, N. M. & Rubin, D. B. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. In: *Journal of the Royal Statistical Society, number 39* (pp. 1-31).

Dhillon, I. S. (2001). Co-clustering Documents and Words using Bipartite Spectral Graph Partitioning. In: *Proc. of ACM SIGKDD International* Conference *on Knowledge Discovery and Data Mining (KDD)* (pp.269–274).

Dhillon, I. S., Guan, Y. & Kulis, B. (2004). Kernel K-means: Spectral Clustering and Normalized Cuts. In: *Proc. of ACM SIGKDD International* Conference *on Knowledge Discovery and Data Mining (KDD)* (pp. 551-556).

Ding, C. H. Q. & He, X. (2004). K-means Clustering via Principal Component Analysis. In: *Proc. of International Conference on Machine Learning (ICML)* (pp. 225-232).

Ding, C. H. Q., He, X., Zha, H., Gu, M. & Simon, H. (2001). A Min-max Cut Algorithm for Graph Partitioning and Data Clustering. In: *Proc. of IEEE International Conference on Data Mining (ICDM)* (pp.107–114).

Duda, R. O. & Hart, P. E. (1973). *Pattern Classification and Scene Analysis*. PA: John Wiley & Sons.

Ester, M., Kriegel, H.-P., Sander, J. & Xu, X. (1996). A Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In: *Proc. of ACM SIGKDD International* Conference *on Knowledge Discovery and Data Mining (KDD)* (pp.226–231).

Feng, Y. & Hamerly, G. (2006). PG-means: Learning the Number of Clusters in Data. In: *Proc. of 19th Annual Conference on Advances in Neural Information Processing Systems (NIPS)* (pp. 393-400).

Figueiredo, M. A. T. & Jain, A. K. (2002). Unsupervised Learning of Finite Mixture Models. In: *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)* 24(3) (pp. 381-396).

Fowlkes, C., Belongie, S., Chung, F. R. K. & Malik J. (2004). Spectral Grouping Using the Nyström Method. In: *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)* 26(2) (pp.214-225).

Grünwald, P. (2005). A Tutorial Introduction to the Minimum Description Length Principle. In: Grünwald, P., Mynung, I. J., Pitt, M. (Eds.) *Advances in Minimum Description Length: Theory and Applications*, PA: MIT Press.

Hamdan, H. & Govaert, G. (2005). Mixture Model Clustering of Uncertain Data. In: *IEEE International Conference on Fuzzy Systems* (pp. 879–884).

Hagen, L. & Kahng, A. (1992). New Spectral Methods for Ratio Cut Partitioning and Clustering. In*: IEEE Trans. Computer-Aided Design, 11(9)* (pp. 1074-1085).

Hamerly, G. & Elkan, C. (2003). Learning the K in K-means. In: *Proc. of 16th Annual Conference on Advances in Neural Information Processing Systems (NIPS).*

Jain, A. K. & Dubes, R. C. (1988). *Algorithms for Clustering Data*. PA: Prentice-Hall.

Kailing, K. & Kriegel, H.-P. & Kröger, P. (2004). Density-connected Subspace Clustering for High Dimensional Data. In: *Proc. of SIAM International Conference on Data Mining (SDM)* (pp. 246-257).

Kaufman, L. & Rousseeuw, P. J. (1990). *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley series in probability and mathematical statistics. PA: John Wiley and Sons.

Klein, D., Kamvar, D. & Manning, C. (2002). From Instance-Level Constraints to Space-Level Constraints: Making Most of Prior Knowledge in Data Clustering. In: *Proc. of International Conference on Machine Learning (ICML)* (pp. 307-314).

Kulis, B., Basu, S., Dhillon, I. S. & Mooney R. J. (2005). Semi-supervised Graph Clustering: a Kernel Approach. In: *Proc. of International Conference on Machine Learning (ICML)* (pp. 457-464).

Kriegel, H.-P. & Pfeifle, M. (2005). Density-based Clustering of Uncertain Data. In: *Proc. of ACM SIGKDD International* Conference *on Knowledge Discovery and Data Mining (KDD)* (pp.672–677).

Li, Y., Han, J. & Yang J. (2004). Clustering moving objects. In: *Proc. of ACM SIGKDD International* Conference *on Knowledge Discovery and Data Mining (KDD)* (pp.617–622).

Long, B., Zhang, Z., Wu, X. & Yu, P. S. (2006). Spectral clustering for Multi-type Relational Data. In: *Proc. of International Conference on Machine Learning (ICML)* (pp.585-592).

Meila, M. (2008). The Uniqueness of a Good Optimum for K-means. In: *Proc. of International Conference on Machine Learning (ICML)* (pp. 625-632).

Nasraoui, O., Uribe, C.C., Coronel, C.R. & Gonzales, F. (2003). TECNO-STREAMS: Tracking Evolving Clusters in Noisy Data Streams with a Scalable Immune System Learning Model. In: *Proc. of IEEE International Conference on Data Mining (ICDM)* (pp. 235-242).

Ng, R. T. & and Han, J. (1994). Efficient and Effective Clustering Methods for Spatial Data Mining. In: *Proc. of International Conference on Very Large Data Bases (VLDB)* (pp. 144-155).

Ng, A., Jordan, M. & Weiss, Y. (2002). On Spectral Clustering: Analysis and an Algorithm. In: Dietterich, T., Becker, S., Ghahramani Z. (Eds.) *Advances in Neural Information Processing Systems 14.* MIT Press (pp. 849-856).

Ngai, W., Kao, B., Chui, C., Cheng, R., Chau, M. & Yip, K. Y. (2006). Efficient Clustering of Uncertain Data. In: *Proc. of IEEE International Conference on Data Mining (ICDM)* (pp. 436-445).

Nagesh, H., Goil, S. & Choudhary A. (2000). A Scalable Parallel Subspace Clustering Algorithm for Massive Data Sets. In: *Proc. of IEEE International Conference on Parallel Processing* (pp.477-).

O'Callaghan, L., Meyerson, A., Motwani, M., Mishra, N. & Guha, S. (2002). Streaming-Data Algorithms for High-Quality Clustering. In: *IEEE* In*ternational Conference on Data Engineering (ICDE)* (pp.685-).

Pelleg, D. & Moore, A. (2000) X-means: Extending K-means with Efficient Estimation of the Number of Clusters. In: *Proc. of International Conference on Machine Learning (ICML)* (pp. 727–734).

Procopiuc, C. M., Jonesý, M., Pankaj, K., Agarwal, M. & Muraliý M. (2002). A Monte Carlo Algorithm for Fast Projective Clustering. In: *Proc. of ACM SIGMOD International Conference on Management of Data (SIGMOD)* (pp. 418–427).

Shi, J. & Malik, J. (2000). Normalized Cuts and Image Segmentation. In*: IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(8)* (pp. 888-905).

Slonim, N. & Tishby, N. (2000). Document clustering using Word Clusters via the Information Bottleneck Method. In*: Proc. of International ACM/*SIGIR Conference *on Research and Development in Information Retrieval* (pp.208-215).

Song, L., Smola, A. J., Gretton, A. & Borgwardt, K. M. (2007). A Dependence Maximization View of Clustering. In: *Proc. of International Conference on Machine Learning (ICML)* (pp.815-822).

Stoer, M. & Wagner, F. (1997). A Simple Min-cut Algorithm. In: *Journal of the ACM* 44(4), (pp. 585–591).

Tishby, N., Pereira, F. C. & Bialek, W. (2000). The Information Bottleneck Method. In: *CoRR physics/0004057.*

Tishby, N. & Slonim, N. (2000). Data Clustering by Markovian Relaxation and the Information Bottleneck Method. *Proc. of 13th Annual Conference on Advances in Neural Information Processing Systems (NIPS)* (pp. 640-646).

Tsuda, K. & Kudo T. (2006). Clustering Graphs by Weighted Substructure Mining. In: *Proc. of International Conference on Machine Learning (ICML)* (pp. 953-960).

Wagner, D. & Wagner, F. (1993). Between Min-cut and Graph Bisection. In: *Proc. of International Symposium on Mathematical Foundations of Computer Science (MFCS)* (pp. 744–750).

Wagstaff, K., Cardie, C., Rogers, S. & Schroedel, S. (2001). Constrained K-means Clustering with Background Knowledge. In: *Proc. of International Conference on Machine Learning (ICML)* (pp.577-584).

Weber, R., Schek, H.-J. & Blott, S. (1998). A Quantative Analysis and Performance Study for Similarity-search Methods in High-dimensional Spaces. In: *Proc. of International Conference on Very Large Data Bases (VLDB)* (pp. 194-205).

Xing, E., Ng, A., Jordan, M., & Russell, S. (2003). Distance Metric Learning, with Application to Clustering with Side-information. In: *Proc. of 15th Annual Conference on Advances in Neural Information Processing Systems (NIPS)* (pp. 505-512).

Yip, K. Y., Cheung, D. W. & Ng, M. K. (2005). On Discovery of Extremely Low-Dimensional Clusters using Semi-Supervised Projected Clustering. In: *IEEE* Int*ernational Conference on Data Engineering (ICDE)* (pp. 329-340).

Zaiane, O. R., Man, X., & Han J. (1998). Discovering Web Access Patterns and Trends by Applying OLAP and Data Mining Technology on Web Logs. In: *IEEE Forum on Research and Technology Advances in Digital Libraries (ADL)* (pp. 19-29).

Zelnik-Manor L. & Perona P. (2004). Self-Tuning Spectral Clustering. In: *Proc. of 17th Annual Conference on Advances in Neural Information Processing Systems (NIPS)* (pp. 1601-1608).

Zeng, E., Chengyong Y., Tao L. & Narasimhan, G. (2007). On the Effectiveness of Constraints Sets in Clustering Genes. In: *IEEE International Conference on Bioinformatics and Bioengineering (BIBE)* (pp. 79-86).