Chapter 1
Cluster Analysis and K-means Clustering: An Introduction

1.1 The Emergence of Data Mining

The phrase "data mining" was termed in the late eighties of the last century, which describes the activity that attempts to extract *interesting patterns* from data. Since then, data mining and knowledge discovery has become one of the hottest topics in both academia and industry. It provides valuable business and scientific intelligence hidden in a large amount of historical data.

From a research perspective, the scope of data mining has gone far beyond the database area. A great many of researchers from various fields, e.g. computer science, management science, statistics, biology, and geography, have made great contributions to the prosperity of data mining research. Some top annual academic conferences held specifically for data mining, such as KDD (ACM SIGKDD International Conference on Knowledge Discovery and Data Mining),[1] ICDM (IEEE International Conference on Data Mining),[2] and SDM (SIAM International Conference on Data Mining),[3] have become the main forums and prestigious brands that lead the trend of data mining research, and have a farreaching influence on big sharks such as Google, Microsoft, and Facebook in industry. Many top conferences in different research fields are now open for the submission of data mining papers. Some A^+ journals in management field, such as Management Science, Information Systems Research, and MIS Quarterly, have also published business intelligence papers based on data mining techniques. These facts clearly illustrate that data mining as a young discipline is fast penetrating into other well-established disciplines. Indeed, data mining is such a hot topic that it has even become an "obscured" buzzword misused in many related fields to show the advanced characteristic of the research in those fields.

[1] http://www.kdd.org/.

[2] http://www.cs.uvm.edu/~icdm/.

[3] http://www.informatik.uni-trier.de/~ley/db/conf/sdm/.

J. Wu, *Advances in K-means Clustering*, Springer Theses,
DOI: 10.1007/978-3-642-29807-3_1, © Springer-Verlag Berlin Heidelberg 2012

From an application perspective, data mining has become a powerful tool for extracting useful information from tons of commercial and engineering data. The driving force behind this trend is the explosive growth of data from various application domains, plus the much more enhanced storing and computing capacities of IT infrastructures at lower prices. As an obvious inter-discipline, data mining discriminates itself from machine learning and statistics in placing ever more emphasis on data characteristics and being more solution-oriented. For instance, data mining has been widely used in business area for a number of applications, such as customer segmentation and profiling, shelf layout arrangement, financial-asset price prediction, and credit-card fraud detection, which greatly boost the concept of business intelligence. In the Internet world, data mining enables a series of interesting innovations, such as web document clustering, click-through data analysis, opinion mining, social network analysis, online product/service/information recommendation, and location-based mobile recommendation, some of which even show appealing commercial prospects. There are still many applicative cases of data mining in diverse domains, which will not be covered any more. An interesting phenomenon is, to gain the first-mover advantage in the potentially huge business intelligence market, many database and statistical software companies have integrated the data mining module into their products, e.g. SAS Enterprise Miner, SPSS Modeler, Oracle Data Mining, and SAP Business Object. This also helps to build complete product lines for these companies, and makes the whole decision process based on these products transparent to the high-end users.

1.2 Cluster Analysis: A Brief Overview

As a young but huge discipline, data mining cannot be fully covered by the limited pages in a monograph. This book focuses on one of the core topics of data mining: cluster analysis. Cluster analysis provides insight into the data by dividing the objects into groups (clusters) of objects, such that objects in a cluster are more similar to each other than to objects in other clusters [48]. As it does not use external information such as class labels, cluster analysis is also called unsupervised learning in some traditional fields such as machine learning [70] and pattern recognition [33].

In general, there are two purposes for using cluster analysis: understanding and utility [87]. Clustering for understanding is to employ cluster analysis for automatically finding conceptually meaningful groups of objects that share common characteristics. It plays an important role in helping people to analyze, describe and utilize the valuable information hidden in the groups. Clustering for utility attempts to abstract the prototypes or the representative objects from individual objects in the same clusters. These prototypes/objects then serve as the basis of a number of data processing techniques such as summarization, compression, and nearest-neighbor finding.

Cluster analysis has long played an important role in a wide variety of application domains such as business intelligence, psychology and social science, information

retrieval, pattern classification, and bioinformatics. Some interesting examples are as follows:

- **Market research.** Cluster analysis has become the "killer application" in one of the core business tasks: marketing. It has been widely used for large-scale customer segmentation and profiling, which help to locate targeted customers, design the 4P (product, price, place, promotion) strategies, and implement the effective customer relationship management (CRM) [12, 13].
- **Web browsing.** As the world we live has entered the Web 2.0 era, information overload has become a top challenge that prevents people from acquiring useful information in a fast and accurate way. Cluster analysis can help to automatically categorize web documents into a concept hierarchy, and therefore provide better browsing experience to web users [43].
- **Image indexing.** In the online environment, images pose problems of access and retrieval more complicated than those of text documents. As a promising method, cluster analysis can help to group images featured by the bag-of-features (BOF) model, and therefore becomes a choice for large-scale image indexing [97].
- **Recommender systems.** Recent year have witnessed an increasing interest in developing recommender systems for online product recommendation or location-based services. As one of the most successful approaches to build recommender systems, collaborative filtering (CF) technique uses the known preferences of a group of users to make recommendations or predictions of the unknown preferences for other users [86]. One of the fundamental tools of CF, is right the clustering technique.
- **Community Detection.** Detecting clusters or communities in real-world graphs such as large social networks, web graphs, and biological networks, is a problem of considerable interests that has received a great deal of attention [58]. A range of detection methods have been proposed in the literature, most of which are borrowed from the broader cluster analysis field.

The above applications clearly illustrate that clustering techniques are playing a vital role in various exciting fields. Indeed, cluster analysis is always valuable for the exploration of unknown data emerging from real-life applications. That is the fundamental reason why cluster analysis is invariably so important.

1.2.1 Clustering Algorithms

The earliest research on cluster analysis can be traced back to 1894, when Karl Pearson used the moment matching method to determine the mixture parameters of two single-variable components [78]. Since then, tremendous research efforts have been devoted to designing new clustering algorithms for cluster analysis. It has been pointed out by Milligan [68] that the difficulties of cluster analysis lie in the following three aspects: (1) Clustering is essentially an inexhaustible combinatorial problem; (2) There exist no widely accepted theories for clustering; (3) The definition of a cluster seems to be a bit "arbitrary", which is determined by the data characteristics

and the understandings of users. These three points well illustrate why there are so many clustering algorithms proposed in the literature, and why it is valuable to formulate the clustering problems as optimization problems which can be solved by some heuristics.

In what follows, we categorize the clustering algorithms into various types, and introduce some examples to illustrate the distinct properties of algorithms in different categories. This part has been most heavily influenced by the books written by Tan et al. [87] and Jain and Dubes [48]. Note that we have no intention of making this part as a comprehensive overview of clustering algorithms. Readers with this interest can refer to the review papers written by Jain et al. [49], Berkhin [11], and Xu and Wunsch [96]. Some books that may also be of interest include those written by Anderberg [3], Kaufman and Rousseeuw [53], Mirkin [69], etc. A paper by Kleinberg [55] provides some in-depth discussions on the clustering theories.

Prototype-Based Algorithms. This kind of algorithms learns a prototype for each cluster, and forms clusters by data objects around the prototypes. For some algorithms such as the well-known K-means [63] and Fuzzy c-Means (FCM) [14], the prototype of a cluster is a centroid, and the clusters tend to be globular. Self-Organizing Map (SOM) [56], a variant of artificial neural networks, is another representative prototype-based algorithm. It uses a neighborhood function to preserve the topological properties of data objects, and the weights of the whole network will then be trained via a competitive process. Being different from the above algorithms, Mixture Model (MM) [65] uses a probability distribution function to characterize the prototype, the unknown parameters of which are usually estimated by the Maximum Likelihood Estimation (MLE) method [15].

Density-Based Algorithms. This kind of algorithms takes a cluster as a dense region of data objects that is surrounded by regions of low densities. They are often employed when the clusters are irregular or intertwined, or when noise and outliers are present. DBSCAN [34] and DENCLUE [46] are two representative density-based algorithms. DBSCAN divides data objects into core points, border points and noise, respectively, based on the Euclidean density [87], and then finds the clusters naturally. DENCLUE defines a probability density function based on the kernel function of each data object, and then finds the clusters by detecting the variance of densities. When it comes to data in high dimensionality, the density notion is valid only in subspaces of features, which motivates the subspace clustering. For instance, CLIQUE [1], a grid-based algorithm, separates the feature space into grid units, and finds dense regions in subspaces. A good review of subspace clustering can be found in [77].

Graph-Based Algorithms. If we regard data objects as nodes, and the distance between two objects as the weight of the edge connecting the two nodes, the data can be represented as a graph, and a cluster can be defined as a connected subgraph. The well-known agglomerative hierarchical clustering algorithms (AHC) [87], which merge the nearest two nodes/groups in one round until all nodes are connected, can be regarded as a graph-based algorithm to some extent. The Jarvis-Patrick algorithm (JP) [50] is a typical graph-based algorithm that defines the shared nearest-neighbors for each data object, and then sparsifies the graph to obtain the clusters. In recent

years, spectral clustering becomes an important topic in this area, in which data can be represented by various types of graphs, and linear algebra is then used to solve the optimization problems defined on the graphs. Many spectral clustering algorithms have been proposed in the literature, such as Normalized Cuts [82] and MinMaxCut [31]. Readers with interests can refer to [74] and [62] for more details.

Hybrid Algorithms. Hybrid algorithms, which use two or more clustering algorithms in combination, are proposed in order to overcome the shortcomings of single clustering algorithms. Chameleon [51] is a typical hybrid algorithm, which firstly uses a graph-based algorithm to separate data into many small components, and then employs a special AHC to get the final clusters. In this way, bizarre clusters can be discovered. FPHGP [16, 43] is another interesting hybrid algorithm, which uses association analysis to find frequent patterns [2] and builds a data graph upon the patterns, and then applies a hypergraph partitioning algorithm [52] to partition the graph into clusters. Experimental results show that FPHGP performs excellently for web document data.

Algorithm-Independent Methods. Consensus clustering [72, 84], also called clustering aggregation or cluster ensemble, runs on the clustering results of basic clustering algorithms rather than the original data. Given a set of basic partitionings of data, consensus clustering aims to find a single partitioning that matches every basic partitioning as closely as possible. It has been recognized that consensus clustering has merits in generating better clusterings, finding bizarre clusters, handling noise and outliers, and integrating partitionings of distributed or even inconsistent data [75]. Typical consensus clustering algorithms include the graph-based algorithms such as CPSA, HGPA and MCLA[84], the co-association matrix-based methods [36], and the prototype-based clustering methods [89, 90]. Some methods that employ meta-heuristics also show competitive results but at much higher computational costs [60].

1.2.2 Cluster Validity

Cluster validity, or clustering evaluation, is a necessary but challenging task in cluster analysis. It is formally defined as giving *objective* evaluations to clustering results in a *quantitative* way [48]. A key motivation of cluster validity is that almost every clustering algorithm will find clusters in a data set that even has no natural cluster structure. In this situation, a validation measure is in great need to tell us how well the clustering is. Indeed, cluster validity has become the core task of cluster analysis, for which a great number of validation measures have been proposed and carefully studied in the literature.

These validation measures are traditionally classified into the following two types: external indices and internal indices (including the relative indices) [41]. External indices measure the extent to which the clustering structure discovered by a clustering algorithm matches some given external structure, e.g. the structure defined by the class labels. In contrast, internal indices measure the goodness of a clustering structure without respect to external information. As internal measures often make latent assumptions on the formation of cluster structures, and usually have much higher

computational complexity, more research in recent years prefers to use external measures for cluster validity, when the purpose is only to assess clustering algorithms and the class labels are available.

Considering that we have no intention of making this part as an extensive review of all validation measures, and only some external measures have been employed for cluster validity in the following chapters, we will focus on introducing some popular external measures here. Readers with a broader interest can refer to the review papers written by Halkidi et al. [41, 42], although discussions on how to properly use the measures are not presented adequately. The classic book written by Jain and Dubes [48] covers fewer measures, but some discussions are very interesting.

According to the different sources, we can further divide the external measures into three categories as follows:

Statistics-Based Measures. This type of measures, such as Rand index (R), Jaccard coefficient (J), Folks and Mallows index (FM), and Γ statistic (Γ) [48], originated from the statistical area quite a long time ago. They focus on examining the group membership of each object pair, which can be quantified by comparing two matrices: the Ideal Cluster Similarity Matrix (ICuSM) and the Ideal Class Similarity Matrix (ICaSM) [87]. ICuSM has a 1 in the ij-th entry if two objects i and j are clustered into a same cluster and a 0, otherwise. ICaSM is defined with respect to class labels, which has a 1 in the ij-th entry if objects i and j belong to a same class, and a 0 otherwise. Consider the entries in the upper triangular matrices (UTM) of ICuSM and ICaSM. Let f_{00} (f_{11}) denote the number of entry pairs that have 0 (1) in the corresponding positions of the two UTMs, and let f_{01} and f_{10} denote the numbers of entry pairs that have different values in the corresponding positions of the two UTMs. R, J, and FM can then be defined as: $R = \frac{f_{00}+f_{11}}{f_{00}+f_{10}+f_{01}+f_{11}}$, $J = \frac{f_{11}}{f_{10}+f_{01}+f_{11}}$, and $FM = \frac{f_{11}}{\sqrt{(f_{11}+f_{10})(f_{11}+f_{01})}}$. The definition of Γ is more straightforward by computing the correlation coefficient of the two UTMs. More details about these measures can be found in [48].

Information-Theoretic Measures. This type of measures is typically designed based on the concepts of information theory. For instance, the widely used Entropy measure (E) [98] assumes that the clustering quality is higher if the entropy of data objects in each cluster is smaller. Let $E_j = \sum_i p_{ij} \log p_{ij}$, where p_{ij} is the proportion of objects in cluster j that are from class i, n_j is the number of objects in cluster j, and $n = \sum_j n_j$. We then have $E = \sum_j \frac{n_j}{n} E_j$. The Mutual Information measure (MI) [85] and the Variation of Information measure (VI) [66, 67] are another two representative measures that evaluate the clustering results by comparing the information contained in class labels and cluster labels, respectively. As these measures have special advantages including clear concepts and simple computations, they become very popular in recent studies, even more popular than the long-standing statistics-based measures.

Classification-Based Measures. This type of measures evaluates clustering results from a classification perspective. The F-measure (F) is such an example, which was originally designed for validating the results of hierarchical clustering [57], but also used for partitional clustering in recent studies [83, 95]. Let p_{ij} denote

the proportion of data objects in cluster j that are from class i (namely the precision of cluster j for objects of class i), and q_{ij} the proportion of data objects from class i that are assigned to cluster j (namely the recall of class i in cluster j) [79]. We then have the F-measure of class i as: $F_i = \max_j \frac{2p_{ij}q_{ij}}{p_{ij}+q_{ij}}$, and the overall F-measure of clustering results as: $F = \sum_i \frac{n_i}{n} F_i$, where n_i is the number of objects of class i, and $n = \sum_i n_i$. Another representative measure is the Classification Error (ε), which tries to map each class to a different cluster so as to minimize the total misclassification rate. Details of ε can be found in [21].

Sometimes we may want to compare the clustering results of different data sets. In this case, we should normalize the validation measures into a value range of about [0,1] or [-1,+1] before using them. However, it is surprising that only a few research has addressed the issue of measure normalization in the literature, including [48] for Rand index, [66] for Variation of Information, and [30] for Mutual Information. Among these studies, two methods are often used for measure normalization, i.e. the expected-value method [48] and the extreme-value method [61], which are both based on the assumption of the multivariate hypergeometric distribution (MHD) [22] of clustering results. The difficulty lies in the computation of the expected values or the min/max values of the measures, subjecting to MHD. A thorough study of measure normalization has been provided in Chap. 5, and we therefore will not go into the details here.

1.3 K-means Clustering: An Ageless Algorithm

In this book, we focus on K-means clustering, one of the oldest and most widely used clustering algorithms. The research on K-means can be traced back to the middle of the last century, conducted by numerous researchers across different disciplines, most notably Lloyd (1957, 1982) [59], Forgey (1965) [35], Friedman and Rubin (1967) [37], and MacQueen (1967) [63]. Jain and Dubes (1988) provides a detailed history of K-means along with descriptions of several variations [48]. Gray and Neuhoff (1998) put K-means in the larger context of hill-climbing algorithms [40].

In a nutshell, K-means is a prototype-based, simple partitional clustering algorithm that attempts to find K non-overlapping clusters. These clusters are represented by their centroids (a cluster centroid is typically the mean of the points in that cluster). The clustering process of K-means is as follows. First, K initial centroids are selected, where K is specified by the user and indicates the desired number of clusters. Every point in the data is then assigned to the closest centroid, and each collection of points assigned to a centroid forms a cluster. The centroid of each cluster is then updated based on the points assigned to that cluster. This process is repeated until no point changes clusters.

It is beneficial to delve into the mathematics behind K-means. Suppose $\mathcal{D} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n\}$ is the data set to be clustered. K-means can be expressed by an objective function that depends on the proximities of the data points to the cluster centroids as follows:

$$\min_{\{m_k\},1\leq k\leq K}\sum_{k=1}^{K}\sum_{\boldsymbol{x}\in C_k}\pi_{\boldsymbol{x}}\text{dist}(\boldsymbol{x},\boldsymbol{m}_k), \tag{1.1}$$

where $\pi_{\boldsymbol{x}}$ is the weight of $\boldsymbol{x}$, n_k is the number of data objects assigned to cluster C_k, $\boldsymbol{m}_k = \sum_{\boldsymbol{x}\in C_k}\frac{\pi_{\boldsymbol{x}}\boldsymbol{x}}{n_k}$ is the centroid of cluster C_k, K is the number of clusters set by the user, and the function "dist" computes the distance between object $\boldsymbol{x}$ and centroid $\boldsymbol{m}_k$, $1 \leq k \leq K$. While the selection of the distance function is optional, the squared Euclidean distance, i.e. $\| \boldsymbol{x} - \boldsymbol{m} \|^2$, has been most widely used in both research and practice. The iteration process introduced in the previous paragraph is indeed a gradient-descent alternating optimization method that helps to solve Eq. (1.1), although often converges to a local minima or a saddle point.

Considering that there are numerous clustering algorithms proposed in the literature, it may be argued that why this book is focused on the "old" K-means clustering. Let us understand this from the following two perspectives. First, K-means has some distinct advantages compared with other clustering algorithms. That is, K-means is very simple and robust, highly efficient, and can be used for a wide variety of data types. Indeed, it has been ranked the second among the top-10 data mining algorithms in [93], and has become the defacto benchmark method for newly proposed methods. Moreover, K-means as an optimization problem still has some theoretical challenges, e.g. the distance generalization problem studied in Chap. 3. The emerging data with complicated properties, such as large-scale, high-dimensionality, and class imbalance, also require to adapt the classic K-means to different challenging scenarios, which in turn rejuvenates K-means. Some disadvantages of K-means, such as performing poorly for non-globular clusters, and being sensitive to outliers, are often dominated by the advantages, and partially corrected by the proposed new variants.

In what follows, we review some recent research on K-means from both the theoretical perspective and the data-driven perspective. Note that we here do not expect to coverage all the works of K-means, but would rather introduce some works that relate to the main themes of this book.

1.3.1 Theoretical Research on K-means

In general, the theoretical progress on K-means clustering lies in the following three aspects:

Model Generalization. The Expectation-Maximization (EM) [26] algorithm-based Mixture Model (MM) has long been regarded as the generalized form of K-means for taking the similar alternating optimization heuristic [65]. Mitchell (1997) gave the details of how to derive squared Euclidean distance-based K-means from the Gaussian distribution-based MM, which unveil the relationship between K-means and MM [70]. Banerjee et al. (2005) studied the von Mises-Fisher distribution-based MM, and demonstrated that under some assumptions this model could reduce to K-means with cosine similarity, i.e. the spherical K-means [4]. Zhong and Ghosh (2004) proposed the Model-Based Clustering (MBC) algorithm [99], which unifies

MM and K-means via the introduction of the deterministic annealing technique. That is, MBC reduces to MM when the temperature $T = 1$, and to K-means when $T = 0$; As T decreases from 1 to 0, MBC gradually changes from allowing soft assignment to only allowing hard assignment of data objects.

Search Optimization. One weakness of K-means is that the iteration process may probably converge to a local minimum or even a saddle point. The traditional search strategies, i.e. the batch mode and the local mode, cannot avoid this problem, although some research has pointed out that using the local search immediately after the batch search may improve the clustering quality of K-means. The "kmeans" function included in MATLAB v7.1 [64] implemented this hybrid strategy. Dhillon et al. (2002) proposed a "first variation" search strategy for spherical K-means, which shares some common grounds with the hybrid strategy [27]. Steinbach et al. (2000) proposed a simple bisecting scheme for K-means clustering, which selects and divides a cluster into two sub-clusters in each iteration [83]. Empirical results demonstrate the effectiveness of bisecting K-means in improving the clustering quality of spherical K-means, and solving the random initialization problem. Some meta-heuristics, such as deterministic annealing [80, 81] and variable neighborhood search [45, 71], can also help to find better local minima for K-means.

Distance Design. The distance function is one of the key factors that influence the performance of K-means. Dhillon el al. (2003) proposed an information-theoretic co-clustering algorithm based on the distance of Kullback-Leibler divergence (or KL-divergence for short) [30] originated from the information theory [23]. Empirical results demonstrate that the co-clustering algorithm improves the clustering efficiency of K-means using KL-divergence (or Info-Kmeans for short), and has higher clustering quality than traditional Info-Kmeans on some text data. Banerjee et al. (2005) studied the generalization issue of K-means clustering by using the Bregman divergence [19], which is actually a family of distances including the well-known squared Euclidean distance, KL-divergence, Itakura-Saito distance [5], and so on. To find clearer boundaries between different clusters, kernel methods have also been introduced to K-means clustering [28], and the concept of distance has therefore been greatly expanded by the kernel functions.

1.3.2 Data-Driven Research on K-means

As the emergence of big data in various research and industrial domains in recent years, the traditional K-means algorithm faces great challenges stemming from the diverse and complicated data factors, such as the high dimensionality, the data streaming, the existence of noise and outliers, and so on. In what follows, we focus on some data-driven advances in K-means clustering.

K-means Clustering for High-Dimensional Data. With the prosperity of information retrieval and bioinformatics, high-dimensional text data and micro-array data have become the challenges to clustering. Numerous studies have pointed out that K-means with the squared Euclidean distance is not suitable for high-dimensional data clustering because of the "curse of dimensionality" [8].

One way to solve this problem is to use alternative distance functions. Steinbach et al. (2000) used the cosine similarity as the distance function to compare the performance of K-means, bisecting K-means, and UPGMA on high-dimensional text data [83]. Experimental results evaluated by the entropy measure demonstrate that while K-means is superior to UPGMA, bisecting K-means has the best performance. Zhao and Karypis (2004) compared the performance of K-means using different types of objective functions on text data, where the cosine similarity was again employed for the distance computation [98]. Zhong and Ghosh (2005) compared the performance of the mixture model using different probability distributions [100]. Experimental results demonstrate the advantage of the von Mises-Fisher distribution. As this distribution corresponds to the cosine similarity in K-means [4], these results further justify the superiority of the cosine similarity for K-means clustering of high-dimensional data.

Another way to tackle this problem is to employ dimension reduction for high-dimensional data. Apart from the traditional methods such as the Principal Component Analysis, Multidimensional Scaling, and Singular Value Decomposition [54], some new methods particularly suitable for text data have been proposed in the literature, e.g. Term-Frequency-Inverse-Document-Frequency (TFIDF), Latent Semantic Indexing (LSI), Random Projection (RP), and Independent Component Analysis (ICA). A comparative study of these methods was given in [88], which revealed the following ranking: ICA $\succ$ LSI $\succ$ TFIDF $\succ$ RP. In particular, ICA and LSI show significant advantages on improving the performance of K-means clustering. Dhillon et al. (2003) used Info-Kmeans to cluster term features for dimension reduction [29]. Experimental results show that their method can improve the classification accuracy of the Naïve Bayes (NB) classifier [44] and the Support Vector Machines (SVMs) [24, 91].

K-means Clustering on Data Stream. Data stream clustering is a very challenging task because of the distinct properties of stream data: rapid and continuous arrival online, need for rapid response, potential boundless volume, etc. [38]. Being very simple and highly efficient, K-means naturally becomes the first choice for data stream clustering. We here highlight some representative works. Domingos and Hulten (2001) employed the Hoeffding inequality [9] for the modification of K-means clustering, and obtained approximate cluster centroids in data streams, with a probability-guaranteed error bound [32]. Ordonez (2003) proposed three algorithms: online K-means, scalable K-means, and incremental K-means, for binary data stream clustering [76]. These algorithms use several sufficient statistics and carefully manipulate the computation of the sparse matrix to improve the clustering quality. Experimental results indicate that the incremental K-means algorithm performs the best. Beringer and Hullermeier (2005) studied the clustering of multiple data streams [10]. The sliding-window technique and the discrete Fourier transformation technique were employed to extract the signals in data streams, which were then clustered by K-means algorithm using the squared Euclidean distance.

Semi-Supervised K-means Clustering. In recent years, more and more researchers recognize that clustering quality can be effectively improved by using *partially available* external information, e.g. the class labels or the pair-wise con-

straints of data. Semi-supervised K-means clustering has therefore become the focus of a great deal of research. For instance, Wagstaff et al. (2001) proposed the COP-KMeans algorithm for semi-supervised clustering of data with two types of pair-wise constraints: must-link and cannot-link [92]. The problem of COP-KMeans is that it cannot handle inconsistent constraints. Basu et al. (2002) proposed two algorithms, i.e. SEEDED-KMeans and CONSTRAINED-KMeans, for semi-supervised clustering using partial label information [6]. Both the two algorithms employ seed clustering for initial centroids, but only CONSTRAINED-KMeans reassigns the data objects outside the seed set during the iteration process. Experimental results demonstrate the superiority of the two methods to COP-KMeans, and SEEDED-KMeans shows good robustness to noise. Basu et al. (2004) further proposed the HMRF-KMeans algorithm that based on the hidden Markov random fields for pair-wise constraints [7], and the experimental results show that HMRF-Kmeans is significantly better than K-means. Davidson and Ravi (2005) proved that to satisfy all the pair-wise constraints in K-means is NP-complete, and thus only satisfied partial constraints to speed up the constrained K-means clustering [25]. They also proposed δ- and ε-constraints in the cluster level to improve the clustering quality.

K-means Clustering on Data with Other Characteristics. Other data factors that may impact the performance of K-means including the scale of data, the existence of noise and outliers, and so on. For instance, Bradley et al. (1998) considered how to adapt K-means to the situation that the data could not be entirely loaded into the memory [17]. They also studied how to improve the scalability of the EM-based mixture model [18]. Some good reviews about the scalability of clustering algorithms can be found in [73] and [39]. Noise removal, often conducted before clustering, is very important for the success of K-means. Some new methods for noise removal include the well-known LOF [20] and the pattern-based HCleaner [94], and a good review of the traditional methods can be found in [47].

1.3.3 Discussions

In general, K-means has been widely studied in a great deal of research from both the optimization and the data perspectives. However, there still some important problems remain unsolve as follows.

First, few research has realized the impact of skewed data distribution (i.e. the imbalance of true cluster sizes) on K-means clustering. This is considered dangerous, because data imbalance is a universal situation in practice, and cluster validation measures may not have the ability to capture its impact to K-means. So we have the following problems:

Problem 1.1 How can skewed data distributions make impact on the performance of K-means clustering? What are the cluster validation measures that can identify this impact?

The answer to the above questions can provide a guidance for the proper use of K-means. This indeed motivates our studies on the uniform effect of K-means in Chap. 2, and the selection of validation measures for K-means in Chap. 5.

Second, although there have been some distance functions widely used for K-means clustering, their common grounds remain unclear. Therefore, it will be a theoretical contribution to provide a general framework for distance functions that are suitable for K-means clustering. So we have the following problems:

Problem 1.2 Is there a unified expression for all the distance functions that fit K-means clustering? What are the common grounds of these distance functions?

The answer to the above questions can establish a general framework for K-means clustering, and help to understand the essence of K-means. Indeed, these questions motivate our study on the generalization of distance functions in Chap. 3, and the answers help to derive a new variant of K-means in Chap. 4.

Finally, it is interesting to know the potential of K-means as a utility to improve the performance of other learning schemes. Recall that K-means has some distinct merits such as simplicity and high efficiency, which make it a good booster for this task. So we have the following problem:

Problem 1.3 Can we use K-means clustering to improve other learning tasks such as the supervised classification and the unsupervised ensemble clustering?

The answer to the above question can help to extend the applicability of K-means, and drive this ageless algorithm to new research frontiers. This indeed motivates our studies on rare class analysis in Chap. 6 and consensus clustering in Chap. 7.

1.4 Concluding Remarks

In this chapter, we present the motivations of this book. Specifically, we first highlight the exciting development of data mining and knowledge discovery in both academia and industry in recent years. We then focus on introducing the basic preliminaries and some interesting applications of cluster analysis, a core topic in data mining. Recent advances in K-means clustering, a most widely used clustering algorithm, are also introduced from a theoretical and a data-driven perspectives, respectively. Finally, we put forward three important problems remained unsolved in the research of K-means clustering, which indeed motivate the main themes of this book.

References

1. Agrawal, R., Gehrke, J., Gunopulos, D., Raghavan, P.: Automatic subspace clustering of high dimensional data for data mining applications. In: Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data, pp. 94–105 (1998)

2. Agrawal, R., Imielinski, T., Swami, A.: Mining association rules between sets of items in large databases. In: Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, pp. 207–216 (1993)
3. Anderberg, M.: Cluster Analysis for Applications. Academic Press, New York (1973)
4. Banerjee, A., Dhillon, I., Ghosh, J., Sra, S.: Clustering on the unit hypersphere using von mises-fisher distributions. J. Mach. Learn. Res. **6**, 1345–1382 (2005)
5. Banerjee, A., Merugu, S., Dhillon, I., Ghosh, J.: Clustering with bregman divergences. J. Mach. Learn. Res. **6**, 1705–1749 (2005)
6. Basu, S., Banerjee, A., Mooney, R.: Semi-supervised clustering by seeding. In: Proceedings of the Nineteenth International Conference on, Machine Learning, pp. 19–26 (2002)
7. Basu, S., Bilenko, M., Mooney, R.: A probabilistic framework for semi-supervised clustering. In: Proceedings of 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 59–68 (2004)
8. Bellman, R.E., Corporation, R.: Dynamic Programming. Princeton University Press, New Jersey (1957)
9. Bentkus, V.: On hoeffding's inequalities. Ann. Probab. **32**(2), 1650–1673 (2004)
10. Beringer, J., Hullermeier, E.: Online clustering of parallel data streams. Data Knowl. Eng. **58**(2), 180–204 (2005)
11. Berkhin, P.: Survey of clustering data mining techniques. Technical Report, Accrue Software, San Jose (2002)
12. Berry, M., Linoff, G.: Data Mining Techniques: For Marketing, Sales, and Customer Support. Wiley, New York (1997)
13. Berry, M., Linoff, G.: Matering Data Mining: The Art and Science of Customer Relationship Management. Wiley, New York (1999)
14. Bezdek, J.: Pattern Recognition with Fuzzy Objective Function Algoritms. Plenum Press, New York (1981)
15. Bilmes, J.: A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. Technical Report, ICSITR-97-021, International Computer Science Institute and U.C. Berkeley (1997)
16. Boley, D., Gini, M., Gross, R., Han, E., Hastings, K., Karypis, G., Kumar, V., Mobasher, B., Moore, J.: Partitioning-based clustering for web document categorization. Decis. Support Syst. **27**(3), 329–341 (1999)
17. Bradley, P., Fayyad, U., Reina, C.: Scaling clustering algorithms to large databases. In: Proceedings of the 4th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 9–15 (1998)
18. Bradley, P., Fayyad, U., Reina, C.: Scaling em (expectation maximization) clustering to large databases. Technical Report, MSR-TR-98-35, Microsoft Research (1999)
19. Bregman, L.: The relaxation method of finding the common points of convex sets and its application to the solution of problems in convex programming. USSR Comput. Math. Math. Phys. **7**, 200–217 (1967)
20. Breunig, M., Kriegel, H., Ng, R., Sander, J.: Lof: identifying density-based local outliers. In: Proceedings of 2000 ACM SIGMOD International Conference on Management of Data, pp. 93–104 (2000)
21. Brun, M., Sima, C., Hua, J., Lowey, J., Carroll, B., Suh, E., Dougherty, E.: Model-based evaluation of clustering validation measures. Pattern Recognit. **40**, 807–824 (2007)
22. Childs, A., Balakrishnan, N.: Some approximations to the multivariate hypergeometric distribution with applications to hypothesis testing. Comput. Stat. Data Anal. **35**(2), 137–154 (2000)
23. Cover, T., Thomas, J.: Elements of Information Theory, 2nd edn. Wiley-Interscience, Hoboken (2006)
24. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods. Cambridge University Press, Cambridge (2000)
25. Davidson, I., Ravi, S.: Clustering under constraints: feasibility results and the k-means algorithm. In: Proceedings of the 2005 SIAM International Conference on Data Mining (2005)

26. Dempster, A., Laird, N., Rubin, D.: Maximum-likelihood from incomplete data via the em algorithm. J. Royal Stat. Soc. Ser. B **39**(1), 1–38 (1977)
27. Dhillon, I., Guan, Y., Kogan, J.: Iterative clustering of high dimensional text data augmented by local search. In: Proceedings of the 2002 IEEE International Conference on Data Mining, pp. 131–138 (2002)
28. Dhillon, I., Guan, Y., Kulis, B.: Kernel k-means: Spectral clustering and normalized cuts. In: Proceedings of the tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 551–556. New York (2004)
29. Dhillon, I., Mallela, S., Kumar, R.: A divisive information-theoretic feature clustering algorithm for text classification. J. Mach. Learn. Res. **3**, 1265–1287 (2003)
30. Dhillon, I., Mallela, S., Modha, D.: Information-theoretic co-clustering. In: Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 89–98 (2003)
31. Ding, C., He, X., Zha, H., Gu, M., Simon, H.: A min-max cut for graph partitioning and data clustering. In: Proceedings of the 1st IEEE International Conference on Data Mining, pp. 107–114 (2001)
32. Domingos, P., Hulten, G.: A general method for scaling up machine learning algorithms and its application to clustering. In: Proceedings of the 18th International Conference on, Machine Learning, pp. 106–113 (2001)
33. Duda, R., Hart, P., Stork, D.: Pattern Classification, 2nd edn. Wiley-Interscience, New York (2000)
34. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the 2nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 226–231 (1996)
35. Forgy, E.: Cluster analysis of multivariate data: efficiency versus interpretability of classifications. Biometrics **21**(3), 768–769 (1965)
36. Fred, A., Jain, A.: Combining multiple clusterings using evidence accumulation. IEEE Trans. Pattern Anal. Mach. Intell. **27**(6), 835–850 (2005)
37. Friedman, H., Rubin, J.: On some invariant criteria for grouping data. J. Am. Stat. Assoc. **62**, 1159–1178 (1967)
38. Gaber, M.M., Zaslavsky, A., Krishnaswamy, S.: Mining data streams: a review. SIGMOD Rec. **34**(2), 18–26 (2005)
39. Ghosh, J.: Scalable clustering methods for data mining. In: Ye, N. (ed.) Handbook of Data Mining, pp. 247–277. Lawrence Ealbaum (2003)
40. Gray, R., Neuhoff, D.: Quantization. IEEE Trans. Info. Theory **44**(6), 2325–2384 (1998)
41. Halkidi, M., Batistakis, Y., Vazirgiannis, M.: Cluster validity methods: Part I. SIGMOD Rec. **31**(2), 40–45 (2002)
42. Halkidi, M., Batistakis, Y., Vazirgiannis, M.: Clustering validity checking methods: Part II. SIGMOD Rec. **31**(3), 19–27 (2002)
43. Han, E.H., Boley, D., Gini, M., Gross, R., Hastings, K., Karypis, G., Kumar, V., Mobasher, B., Moore, J.: Webace: a web agent for document categorization and exploration. In: Proceedings of the 2nd International Conference on Autonomous Agents, pp. 408–415 (1998)
44. Hand, D., Yu, K.: Idiot's bayes—not so stupid after all? Int. Stat. Rev. **69**(3), 385–399 (2001)
45. Hansen, P., Mladenovic, N.: Variable neighborhood search: principles and applications. Euro. J. Oper. Res. **130**, 449–467 (2001)
46. Hinneburg, A., Keim, D.: An efficient approach to clustering in large multimedia databases with noise. In: Proceedings of the 4th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 58–65. AAAI Press, New York (1998)
47. Hodge, V., Austin, J.: A survey of outlier detection methodologies. Artif. Intell. Rev. **22**, 85–126 (2004)
48. Jain, A., Dubes, R.: Algorithms for Clustering Data. Prentice Hall, Englewood Cliffs (1988)
49. Jain, A., Murty, M., Flynn, P.: Data clustering: A review. ACM Comput. Surv. **31**(3), 264–323 (1999)

50. Jarvis, R., Patrick, E.: Clusering using a similarity measure based on shared nearest neighbors. IEEE Trans. Comput. **C-22(11)**, 1025–1034 (1973)
51. Karypis, G., Han, E.H., Kumar, V.: Chameleon: a hierarchical clustering algorithm using dynamic modeling. IEEE Comput. **32**(8), 68–75 (1999)
52. Karypis, G., Kumar, V.: A fast and highly quality multilevel scheme for partitioning irregular graphs. SIAM J. Sc. Comput. **20**(1), 359–392 (1998)
53. Kaufman, L., Rousseeuw, P.: Finding Groups in Data: An Introduction to Cluster Analysis. Wiley Series in Probability and Statistics. Wiley, New York (1990)
54. Kent, J., Bibby, J., Mardia, K.: Multivariate Analysis (Probability and Mathematical Statistics). Elsevier Limited, New York (2006)
55. Kleinberg, J.: An impossibility theorem for clustering. In: Proceedings of the 16th Annual Conference on Neural Information Processing Systems, pp. 9–14 (2002)
56. Kohonen, T., Huang, T., Schroeder, M.: Self-Organizing Maps. Springer,Heidelberg (2000)
57. Larsen, B., Aone, C.: Fast and effective text mining using linear-time document clustering. In: Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 16–22 (1999)
58. Leskovec, J., Lang, K.J., Mahoney, M.: Empirical comparison of algorithms for network community detection. In: Proceedings of the 19th International Conference on, World Wide Web, pp. 631–640 (2010)
59. Lloyd, S.: Least squares quantization in pcm. IEEE Trans. Info. Theory **28**(2), 129–137 (1982)
60. Lu, Z., Peng, Y., Xiao, J.: From comparing clusterings to combining clusterings. In: Fox, D., Gomes, C. (eds.) Proceedings of the 23rd AAAI Conference on Artificial Intelligence, pp. 361–370. AAAI Press, Chicago (2008)
61. Luo, P., Xiong, H., Zhan, G., Wu, J., Shi, Z.: Information-theoretic distance measures for clustering validation: Generalization and normalization. IEEE Trans. Knowl. Data Eng. **21**(9), 1249–1262 (2009)
62. Luxburg, U.: A tutorial on spectral clustering. Stat. Comput. **17**(4), 395–416 (2007)
63. MacQueen, J.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, pp. 281–297 (1967)
64. MathWorks: K-means clustering in statistics toolbox. http://www.mathworks.com
65. McLachlan, G., Basford, K.: Mixture Models. Marcel Dekker, New York (2000)
66. Meila, M.: Comparing clusterings by the variation of information. In: Proceedings of the 16th Annual Conference on Computational Learning Theory, pp. 173–187 (2003)
67. Meila, M.: Comparing clusterings—an axiomatic view. In: Proceedings of the 22nd International Conference on, Machine learning, pp. 577–584 (2005)
68. Milligan, G.: Clustering validation: Results and implications for applied analyses. In: Arabie, P., Hubert, L., Soete, G. (eds.) Clustering and Classification, pp. 345–375. World Scientific, Singapore (1996)
69. Mirkin, B.: Mathematical Classification and Clustering. Kluwer Academic Press, Dordrecht (1996)
70. Mitchell, T.: Machine Learning. McGraw-Hill, Boston (1997)
71. Mladenovic, N., Hansen, P.: Variable neighborhood search. Comput. Oper. Res. **24**(11), 1097–1100 (1997)
72. Monti, S., Tamayo, P., Mesirov, J., Golub, T.: Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data. Mach. Learn. **52**(1–2), 91–118 (2003)
73. Murtagh, F.: Clustering massive data sets. In: Abello, J., Pardalos, P.M., Resende, M.G. (eds.) Handbook of Massive Data Sets, pp. 501–543. Kluwer Academic Publishers, Norwell (2002)
74. Ng, A.Y., Jordan, M.I., Weiss, Y.: On spectral clustering: analysis and an algorithm. In: Advances in Neural Information Processing Systems, pp. 849–856. MIT Press (2001)
75. Nguyen, N., Caruana, R.: Consensus clusterings. In: Proceedings of the 7th IEEE International Conference on Data Mining, pp. 607–612. Washington (2007)

76. Ordonez, C.: Clustering binary data streams with k-means. In: Proceedings of the SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (2003)
77. Parsons, L., Haque, E., Liu, H.: Subspace clustering for high dimensional data: a review. SIGKDD Explor. **6**(1), 90–105 (2004)
78. Pearson, K.: Contributions to the mathematical theory of evolution. Philos. Trans. Royal Soc. Lond. **185**, 71–110 (1894)
79. Rijsbergen, C.: Information Retrieval, 2nd edn. Butterworths, London (1979)
80. Rose, K.: Deterministic annealing for clustering, compression, classification, regression and related optimization problems. Proc. IEEE **86**, 2210–2239 (1998)
81. Rose, K., Gurewitz, E., Fox, G.: A deterministic annealing approach to clustering. Pattern Recognit. Lett. **11**, 589–594 (1990)
82. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **22**(8), 888–905 (2000)
83. Steinbach, M., Karypis, G., Kumar, V.: A comparison of document clustering techniques. In: Proceedings of the KDD Workshop on Text Mining (2000)
84. Strehl, A., Ghosh, J.: Cluster ensembles—a knowledge reuse framework for combining partitions. J. Mach. Learn. Res. **3**, 583–617 (2002)
85. Strehl, A., Ghosh, J., Mooney, R.: Impact of similarity measures on web-page clustering. In: Proceedings of the AAAI Workshop on AI for Web Search (2000)
86. Su, X., Khoshgoftaar, T.M.: A survey of collaborative filtering techniques. Advances in Artificial Intelligence 2009, Article ID 421,425, 19 pp (2009)
87. Tan, P.N., Steinbach, M., Kumar, V.: Introduction to Data Mining. Addison-Wesley, Reading (2005)
88. Tang, B., Shepherd, M., Heywood, M., Luo, X.: Comparing dimension reduction techniques for document clustering. In: Proceedings of the Canadian Conference on, Artificial Intelligence, pp. 292–296 (2005)
89. Topchy, A., Jain, A., Punch, W.: Combining multiple weak clusterings. In: Proceedings of the 3rd IEEE International Conference on Data Mining, pp. 331–338. Melbourne (2003)
90. Topchy, A., Jain, A., Punch, W.: A mixture model for clustering ensembles. In: Proceedings of the 4th SIAM International Conference on Data Mining. Florida (2004)
91. Vapnik, V.: The Nature of Statistical Learning. Springer, New York (1995)
92. Wagstaff, K., Cardie, C., Rogers, S., Schroedl, S.: Constrained k-means clustering with background knowledge. In: Proceedings of the Eighteenth International Conference on, Machine Learning, pp. 577–584 (2001)
93. Wu, X., Kumar, V., Quinlan, J.R., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G.J., Ng, A., Liu, B., Yu, P.S., Zhou, Z.H., Steinbach, M., Hand, D.J., Steinberg, D.: Top 10 algorithms in data mining. Knowl. Inf. Syst. **14**(1), 1–37 (2008)
94. Xiong, H., Pandey, G., Steinbach, M., Kumar, V.: Enhancing data analysis with noise removal. IEEE Trans. Knowl. Data Eng. **18**(3), 304–319 (2006)
95. Xiong, H., Wu, J., Chen, J.: K-means clustering versus validation measures: a data-distribution perspective. IEEE Trans. Syst. Man Cybern. Part B Cybern. **39**(2), 318–331 (2009)
96. Xu, R., Wunsch, D.: Survey of clustering algorithms. IEEE Trans. Neural Netw. **16**(3), 645–678 (2005)
97. Yang, J., Yuz, K., Gongz, Y., Huang, T.: Linear spatial pyramid matching using sparse coding. In: Proceedings of the 2009 IEEE Conference on Computer Vision and, Pattern Recognition, pp. 1794–1801 (2009)
98. Zhao, Y., Karypis, G.: Criterion functions for document clustering: experiments and analysis. Mach. Learn. **55**(3), 311–331 (2004)
99. Zhong, S., Ghosh, J.: A unified framework for model-based clustering. J. Mach. Learn. Res. **4**(6), 1001–1037 (2004)
100. Zhong, S., Ghosh, J.: Generative model-based document clustering: a comparative study. Knowl. Inf. Syst. **8**(3), 374–384 (2005)