Data Clustering: 50 Years Beyond K-Means[1]

Anil K. Jain
Department of Computer Science & Engineering
Michigan State University
East Lansing, Michigan 48824 USA
jain@cse.msu.edu

Abstract: Organizing data into sensible groupings is one of the most fundamental modes of understanding and learning. As an example, a common scheme of scientific classification puts organisms into taxonomic ranks: domain, kingdom, phylum, class, etc.). Cluster analysis is the formal study of algorithms and methods for grouping, or clustering, objects according to measured or perceived intrinsic characteristics or similarity. Cluster analysis does not use category labels that tag objects with prior identifiers, i.e., class labels. The absence of category information distinguishes data clustering (unsupervised learning) from classification or discriminant analysis (supervised learning). The aim of clustering is exploratory in nature to find structure in data. Clustering has a long and rich history in a variety of scientific fields. One of the most popular and simple clustering algorithms, K-means, was first published in 1955. In spite of the fact that K-means was proposed over 50 years ago and thousands of clustering algorithms have been published since then, K-means is still widely used. This speaks to the difficulty of designing a general purpose clustering algorithm and the ill-posed problem of clustering. We provide a brief overview of clustering, summarize well known clustering methods, discuss the major challenges and key issues in designing clustering algorithms, and point out some of the emerging and useful research directions, including semi-supervised clustering, ensemble clustering, simultaneous feature selection, and data clustering and large scale data clustering.

1. Introduction

Advances in sensing and storage technology and dramatic growth in applications such as Internet search, digital imaging, and video surveillance have created many high-volume, high-dimensional data sets. It is estimated that the digital universe was approximately 281 exabytes in 2007, and it is projected to be 10 times the size by 2011. (One exabyte is ~10^{18} bytes or 1,000,000 terabytes) [Gantz, 2008]. Most of this data is stored digitally in electronic media, thus providing huge potential for the development of automatic data analysis, classification, and retrieval techniques. In addition to the growth in the amount of data, the variety of available data (text, image, and video) has also increased. Inexpensive digital and video cameras have made available huge archives of images and

[1] This paper is based on the King-Sun Fu Prize lecture delivered at the 19th International Conference on Pattern Recognition (ICPR), Tampa, FL, December 8, 2008.

videos. The prevalence of RFID tags or transponders due to their low cost and small size has resulted in the deployment of millions of sensors that transmit data regularly. E-mails, blogs, transaction data, and billions of Web pages create terabytes of new data every day. Many of these data streams are unstructured, adding to the difficulty in analyzing them.

This increase in both the volume and the variety of data requires advances in methodology to automatically understand, process, and summarize the data. Data analysis techniques can be broadly classified into two major types [Tukey, 1977]: (i) *exploratory* or descriptive, meaning that the investigator does not have pre-specified models or hypotheses but wants to understand the general characteristics or structure of the high-dimensional data, and (ii) *confirmatory* or inferential, meaning that the investigator wants to confirm the validity of a hypothesis or model or a set of assumptions given the available data. Many statistical techniques have been proposed to analyze the data, such as analysis of variance, linear regression, discriminant analysis, canonical correlation analysis, multidimensional scaling, factor analysis, principal component analysis, and cluster analysis to name a few. A useful overview is given in [Sungur, 2008].

In pattern recognition, data analysis is concerned with predictive modeling: given some training data, we want to predict the behavior of the unseen test data. This task is also referred to as *learning*. Often, a clear distinction is made between learning problems that are (i) supervised (classification) or (ii) unsupervised (clustering), the first involving only *labeled data* (training patterns with known category labels) while the latter involving only *unlabeled data* [Duda *et al.* , 2001]. Clustering is a more difficult and challenging problem than classification. There is a growing interest in a hybrid setting, called *semi-supervised learning* [Chapelle *et al.* , 2006]; in semi-supervised classification, the labels of only a small portion of the training data set are available. The unlabeled data, instead of being discarded, are also used in the learning process. In semi-supervised clustering, instead of specifying the class labels, pair-wise constraints are specified, which is a *weake*r way of encoding the prior knowledge A pair-wise *must-link* constraint corresponds to the requirement that two objects should be assigned the same cluster label, whereas the cluster labels of two objects participating in a *cannot-link* constraint should be different. Constraints can be particularly beneficial in data clustering [Lange *et al.* , 2005, Basu *et al.* , 2008], where precise definitions of underlying clusters are absent. In the search for good models, one would like to include all the available information, no matter whether it is unlabeled data, data with constraints, or labeled data. Figure 1 illustrates this spectrum of different types of learning problems of interest in pattern recognition and machine learning.

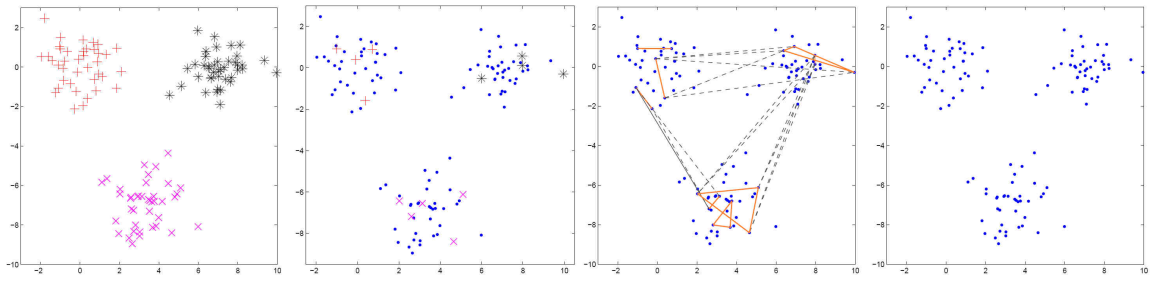
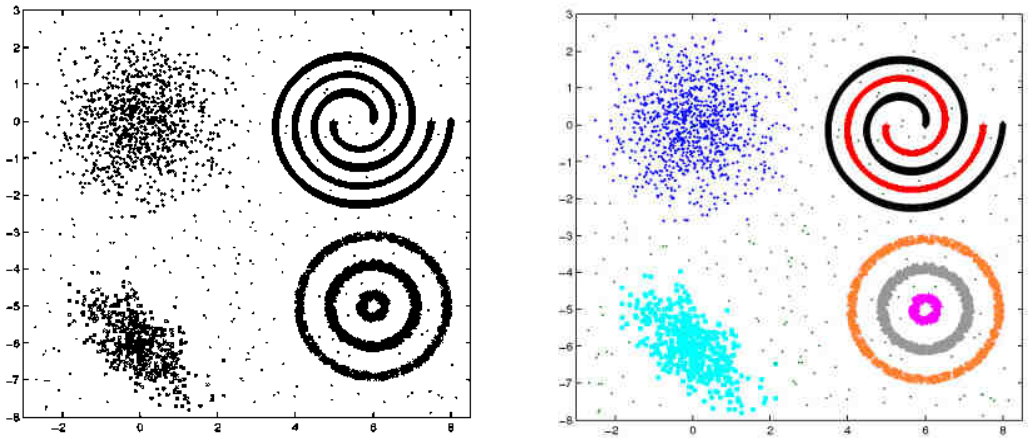


Figure 1 Learning problems: dots correspond to points without any labels. Points with labels are denoted by plus signs, asterisks, and crosses. In (c), the must-link and cannot-link constraints are denoted by solid and dashed lines, respectively (figure taken from [**Lange *et al.* , 2005**]).

2. Data clustering

The goal of data clustering, also known as cluster analysis, is to discover the *natural* grouping(s) of a set of patterns, points, or objects. Webster [Merriam-Webster Online Dictionary, 2008] defines cluster analysis as "a statistical classification technique for discovering whether the individuals of a population fall into different groups by making quantitative comparisons of multiple characteristics." An example of clustering is shown in Figure 2. The objective is to develop an automatic algorithm that will discover the natural groupings (Figure 2 (b)) in the unlabeled data (Figure 2 (a)).

An operational definition of clustering can be stated as follows: Given a *representation* of n objects, find K groups based on a measure of *similarity* such that objects within the same group are *alike* but the objects in different groups are not alike. But, what is the notion of similarity? What is the definition of a cluster? Figure 2 shows that clusters can differ in terms of their *shape*, *size*, and *density*. The presence of noise in the data makes the detection of the clusters even more difficult. An ideal cluster can be defined as a set of points that is *compact* and *isolated*. In reality, a cluster is a subjective entity that is in the eye of the beholder and whose significance and interpretation requires domain knowledge. But, while humans are excellent cluster seekers in two and possibly three dimensions, we need automatic algorithms for high dimensional data. It is this challenge along with the unknown number of clusters in the given data that has resulted in thousands of clustering algorithms that have been published and that continue to appear



(a) Input data (b) Desired clustering

Figure 2 Diversity of clusters**.** The seven clusters in (a) (denoted by seven different colors in 1(b)) differ in shape, size, and density. Although these clusters are apparent to a data analyst, none of the available clustering algorithms can detect all these clusters**.**

2.1 Why clustering?

Cluster analysis is prevalent in any discipline that involves analysis of multivariate data. A search via Google Scholar [gsc, 2009] found 1,660 entries with the words *data clustering* that appeared in 2007 alone. This vast literature speaks to the importance of clustering in data analysis. It is difficult to exhaustively list the numerous scientific fields and applications that have utilized clustering techniques. Image segmentation, an important problem in computer vision, can be formulated as a clustering problem [Frigui & Krishnapuram, 1999, Jain & Flynn, 1996, Shi & Malik, 2000]. Documents can be clustered [Iwayama & Tokunaga, 1995] to generate topical hierarchies for efficient information access [Sahami, 1998] or retrieval [Bhatia & Deogun, 1998]. Clustering is also used to group customers into different types for efficient marketing [Arabie & Hubert, 1994], to group services delivery engagements for workforce management and planning [Hu *et al.* , 2007] as well as to study genome data [Baldi & Hatfield, 2002] in biology.

Data clustering has been used for the following three main purposes.

- *Underlying structure*: to gain insight into data, generate hypotheses, detect anomalies, and identify salient features.
- *Natural classification*: to identify the degree of similarity among forms or organisms (phylogenetic relationship).

- *Compression*: as a method for organizing the data and summarizing it through cluster prototypes.

An example of class discovery is shown in Figure 3. Here, clustering was used to discover subclasses in an online handwritten character recognition application [Connell & Jain, 2002]. Different users write the same digits in different ways, thereby increasing the within-class variance. Clustering the training pattern from a class can discover new subclasses, called the lexemes in handwritten characters. Instead of using a single model for each character, multiple models based on the number of subclasses are used to improve the recognition accuracy.

Given the large number of Web pages on the Internet, most search queries typically result in an extremely large number of hits. This creates the need for search results to be organized. Search engines like Clusty (www.clusty.org) cluster the search results and present them in a more organized way to the user.

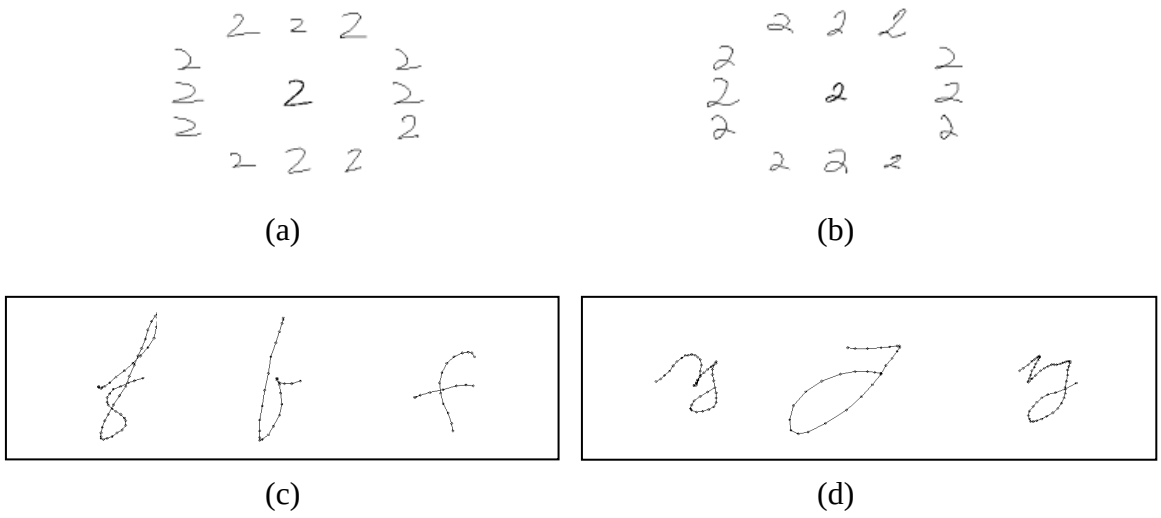


Figure 3 Finding subclasses using data clustering. (a) and (b) show two different ways of writing the digit 2; (c) three different subclasses for the character 'f'; (d) three different subclasses for the letter 'y'.

2.2 Historical developments

The development of clustering methodology has been a truly interdisciplinary endeavor. Taxonomists, social scientists, psychologists, biologists, statisticians, mathematicians, engineers, computer scientists, medical researchers, and others who collect and process real data have all contributed to clustering methodology. According to JSTOR [jst, n.d.], *data clustering* first appeared in the title of a 1954 article dealing with anthropological data. Data clustering is also known as Q-analysis, typology, clumping, and taxonomy [Jain & Dubes, 1988] depending on the field where it is applied. There are several books

published on data clustering; classic ones are by Sokal and Sneath [Sokal & Sneath, 1963], Anderberg [Anderberg, 1973], Hartigan [Hartigan, 1975], Jain and Dubes [Jain & Dubes, 1988] and Duda et al. [Duda *et al.* , 2001]. Clustering algorithms have also been extensively studied in data mining (see books by Han and Kamber [Han & Kamber, 2000] and Tan et al. [Tan *et al.* , 2005]) and machine learning [Bishop, 2006].

Clustering algorithms can be broadly divided into two groups: hierarchical and *partitional*. Hierarchical clustering algorithms recursively find nested clusters either in agglomerative mode (starting with each data point in its own cluster, merge the most similar pair of clusters successively to form a cluster hierarchy) or in divisive (top-down) mode (starting with all the data points in one cluster, recursively divide the cluster into smaller clusters). Partitional clustering algorithms find all the clusters simultaneously as a partition of the data and do not impose a hierarchical structure. Input to a hierarchical algorithm is an *n* x *n* similarity matrix, where *n* is the number of objects to be clustered. On the other hand, a partitional algorithm can use either an *n* x *d* pattern matrix (K-means), where *n* points are embedded in a *d*-dimensional feature space, or an *n* x *n* similarity matrix (Spectral clustering). Note that a similarity matrix can be easily derived from a pattern matrix, but ordination methods such as multi-dimensional scaling (MDS) are needed to derive a pattern matrix from a similarity matrix.

The most well-known hierarchical algorithms are single-link and complete-link; the most popular and the simplest partitional algorithm is K-means. Since partitional algorithms are preferred in pattern recognition due to the nature of available data, our coverage here is focused on these algorithms. K-means has a rich and diverse history as it was independently discovered in different scientific fields by Steinhaus (1955) [Steinhaus, 1956], Lloyd (1957) [Lloyd, 1982], Ball & Hall (1965) [Ball & Hall, 1965] and McQueen (1967) [MacQueen, 1967]. Even though K-means was first proposed over 50 years ago, it is still one of the most widely used algorithms for clustering. Ease of implementation, simplicity, efficiency, and empirical success are the main reasons for its popularity. Details of K-means are summarized below.

2.3 K-Means algorithm

.

Let $X = \{x_i\}, i = 1,...,n$ be the set of *n* d-dimensional points to be clustered into a set of K clusters, $C = \{c_k, k = 1,...,K\}$,. K-means algorithm finds a partition such that the squared error between the empirical mean of a cluster and the points in the cluster is minimized. Let μ_k be the mean of cluster c_k. The squared error between μ_k and the points in cluster c_k is defined as

$$J(c_k) = \sum_{x_i \in c_k} \| x_i - \mu_k \|^2 .$$

The goal of K-means is to minimize the sum of the squared error over all the K clusters,

$$J(C) = \sum_{k=1}^{K} \sum_{x_i \in c_k} \| x_i - \mu_k \|^2 .$$

Minimizing this objective function is known to be an NP-hard problem (even for K = 2) [Drineas *et al.* , 1999]. Thus K-means, which is a greedy algorithm, can only be expected to converge to a local minimum. K-means starts with an initial partition with K clusters and assign patterns to clusters so as to reduce the squared error. Since the squared error tends to decrease with an increase in the number of clusters *K* (with *J*(*C*) = *0* when *K* = *n*), it can be minimized only for a fixed number of clusters. The main steps of K-means algorithm are as follows [Jain & Dubes, 1988].

1. Select an initial partition with K clusters; repeat steps 2 and 3 until cluster membership stabilizes.
2. Generate a new partition by assigning each pattern to its closest cluster center.
3. Compute new cluster centers.

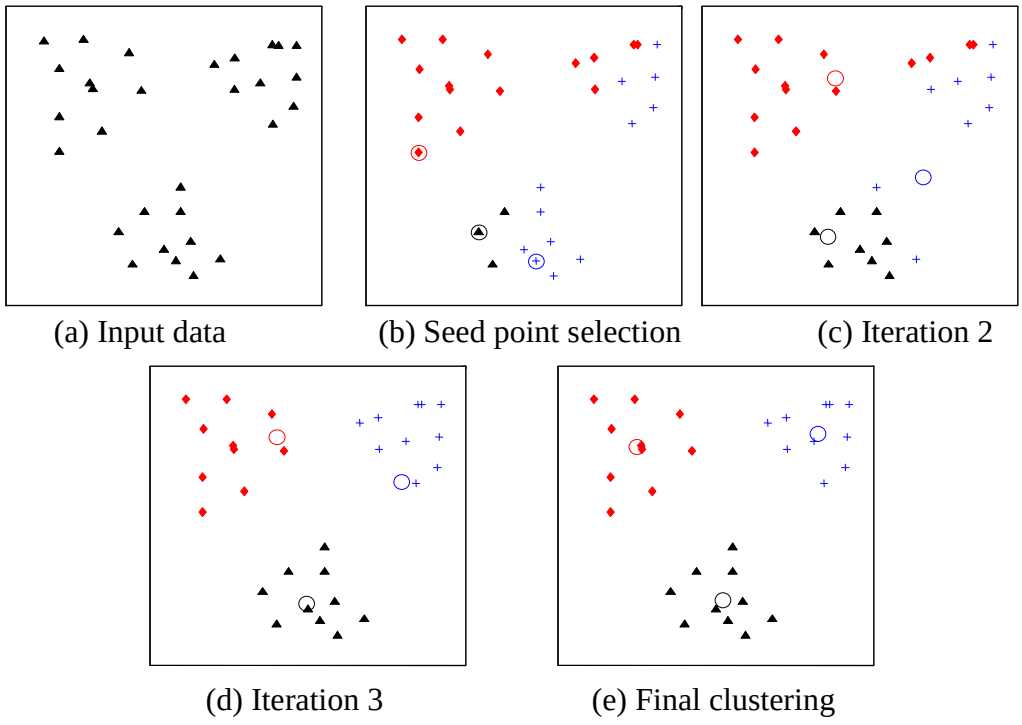


Figure 4 Illustration of K-means algorithm. (a) Two-dimensional input data with three clusters; (b) three seed points selected as cluster centers and initial assignment of the data points to clusters; (c) & (d) intermediate iterations updating cluster labels and their centers; (e) final clustering obtained by K-means algorithm at convergence.

Figure 4 shows an illustration of K-means algorithm on a 2-dimensional dataset with three clusters.

Parameters of K-means

K-means algorithm requires three user-specified parameters: number of clusters *K*, cluster initialization, and distance metric. The most critical choice is K. While no mathematical criterion exists, a number of heuristics are available for choosing K. Typically, K-means is run independently for different values of K and the partition that appears the most meaningful to the domain expert is selected. Different initializations can lead to different final clustering because K-means only converges to local minima. One way to overcome the local minima is to run the K-means algorithm, for a given K, with several different initial partitions and choose the partition with the smallest value of the squared error.

K-means is typically used with the Euclidean metric for computing the distance between points and cluster centers. As a result, K-means finds spherical or ball-shaped clusters in data. K-means with Mahalanobis distance metric has been used to detect hyper-ellipsoidal clusters [Mao & Jain, 1996], but this comes at the expense of higher computation cost. A variant of K-means using the Itakura-Saito distance has been used for vector quantization in speech processing [Linde *et al.* , 1980] and K-means with L1 distance was proposed in [Kashima *et al.* , 2008]. Banerjee et al. [Banerjee *et al.* , 2004] showed that K-means algorithm will have some salient properties (e.g., convergence to local minima, linear separation of classes, linear in number of data points per iteration, etc.) if and only if a Bregman divergence is used as a distance measure.

Extensions of K-means

The basic K-means algorithm has been extended in many different ways. Some of these extensions deal with additional heuristics involving the minimum cluster size and merging and splitting clusters. Two well-known variants of K-means in pattern recognition literature are Isodata [Ball & Hall, 1965] and Forgy [Forgy, 1965]. In K-means, each data point is assigned to a single cluster (called *hard assignment*). *Fuzzy c-means*, proposed by Dunn [Dunn, 1973] and later improved by Bezdek [Bezdek, 1981], is an extension of K-means where each data point can be a member of multiple clusters with a membership value (*soft assignment*). A good overview of fuzzy set based clustering is available in Backer (1978) [Backer, 1978]. Some of the other significant modifications are summarized below. Steinbach et al. [Steinbach *et al.* , 2000] proposed a hierarchical divisive version of K-means, called *bisecting K-means*, that recursively partitions the data into two clusters at each step. A significant speed up of the *K-means pass* [Jain & Dubes, 1988] in the algorithm (computation of distances from all the points to all the cluster centers) involves representing the data using a *kd*-tree and updating cluster means with groups of points instead of a single point [Pelleg & Moore, 1999]. Bradley et al. [Bradley *et al.* , 1998] presented a fast scalable and single-pass version of K-means that does not require all the data to be fit in the memory at the same time. *X-means* [Pelleg & Moore, 2000] automatically finds *K* by optimizing a criterion such as Akaike Information Criterion (AIC) or Bayesian Information Criterion (BIC). In *K-medoid* [Kaufman & Rousseeuw, 2005], clusters are represented using the median of the data instead of the mean. *Kernel K-means* [Scholkopf *et al.* , 1998] was proposed to detect arbitrary shaped clusters, with an appropriate choice of the kernel similarity

function. Note that all these extensions introduce some additional algorithmic parameters that must be specified by the user.

2.4 Major approaches to clustering

As mentioned before, thousands of clustering algorithms have been proposed in the literature in many different scientific disciplines. This makes it extremely difficult to review all the published approaches. Nevertheless, clustering methods differ on the choice of the objective function, probabilistic generative models, and heuristics. We will briefly review some of the major approaches.

Clusters can be defined as high density regions in the feature space separated by low density regions. Algorithms following this notion of clusters directly search for connected dense regions in the feature space. Different algorithms use different definitions of connectedness. The Jarvis-Patrick algorithm defines the similarity between a pair of points as the number of common neighbors they share, where neighbors are the points present in a region of pre-specified radius around the point [Frank & Todeschini, 1994]. Ester et al. [Ester *et al.* , 1996] proposed the DBSCAN clustering algorithm, which is similar to the Jarvis-Patrick algorithm. It directly searches for connected dense regions in the feature space by estimating the density using the Parzen window method. Performance of Jarvis Patrick algorithm and DBSCAN depend on two parameters: neighborhood size in terms of distance, and the minimum number of points in a neighborhood for its inclusion in a cluster. While non-parametric density based methods are attractive because of their inherent ability to deal with arbitrary shaped clusters, they have limitations in handling high-dimensional data. In addition, a number of probabilistic models have been developed for data clustering. The most well known ones are Probabilistic Latent Semantic Analysis (PLSA) [Hofmann & Puzicha, 1998] and Latent Dirichlet Allocation (LDA) [Blei *et al.* , 2003].

When the data is high-dimensional, the feature space is usually sparse, making it difficult to distinguish high-density regions from low-density regions. Subspace clustering algorithms overcome this limitation by finding clusters embedded in low-dimensional subspaces of the given high-dimensional data. CLIQUE [Agrawal *et al.* , 1998] is a scalable clustering algorithm designed to find subspaces in the data with high-density clusters. Although CLIQUE is a non-parametric clustering algorithm, it does not suffer from the dimensionality problem as it estimates the density only in a low dimensional subspace.

Graph theoretic clustering represents the data points as nodes in a weighted graph. The edges connecting the nodes are weighted by their pair-wise similarity. The central idea is to partition the nodes into two subsets A and B such that the cut size, i.e., the sum of the weights assigned to the edges connecting between nodes in A and B, is minimized. Initial algorithms solved this problem using the minimum cut algorithm. But, the minimum cut often results in degenerate clusters, as removing an outlier point also minimizes the cut size. A cluster size (number of data points in a cluster) constraint was later adopted by the Ratio cut algorithm [Hagen & Kahng, 1992]). An efficient approximate graph-cut based

clustering algorithm with cluster size (volume of the cluster, or sum of edge weights within a cluster) constraint, called Normalized Cut, was proposed by Shi and Malik [Shi & Malik, 2000]. A multiclass spectral clustering algorithm was proposed by Yu and Shi [Yu & Shi, 2003]. Meila and Shi [Meila & Shi, 2001] presented a Markov Random Walk approach to spectral clustering and proposed the Modified Normalized Cut (MNCut) algorithm that can handle an arbitrary number of clusters. Another variant of spectral clustering algorithm is proposed by Ng et al. [Ng *et al.* , 2001], where normalized eigenvectors of a kernel matrix are used as a *new* data representation, that are clustered using an algorithm like K-means. Dimensionality reduction from high-dimensional representation of the data to a low dimension using the Laplacian Eigenmap [Belkin & Niyogi, 2002] is another approach to induce clustering in the data.

Several clustering algorithms have information theoretic formulation. For example, the minimum entropy methods approach presented in [Roberts *et al.* , 2001] assumes that the data is generated using a mixture model and each cluster is modeled using a semi-parametric probability density. The parameters are estimated by maximizing the KL-divergence between the unconditional density and the conditional density of a data points conditioned over the cluster. This minimizes the overlap between the conditional and unconditional densities, thereby separating the clusters from each other. In other words, this formulation results in an approach that minimizes the expected entropy of the partitions over the observed data. Information bottleneck method [Tishby *et al.* , 1999] was proposed as a generalization to the rate-distortion theory and adopts a lossy data compression view. In simple words, given a joint distribution over two random variables, Information Bottleneck compresses one of the variables while retaining the maximum amount of mutual information with respect to the other variable. An application of this to document clustering is shown in [Slonim & Tishby, 2000] where the two random variables are words and documents. The words are clustered first, such that the mutual information with respect to documents is maximally retained, and using the clustered words, the documents are clustered such that the mutual information between clustered words and clustered documents is maximally retained.

3. User's dilemma

In spite of the prevalence of such a large number of clustering algorithms, and its success in a number of different application domains, clustering remains a difficult problem. This can be attributed to the inherent vagueness in the definition of a cluster, and the difficulty in defining an appropriate similarity measure and objective function.

The following fundamental challenges associated with clustering were highlighted in [Jain & Dubes, 1988], which are relevant even to this date.

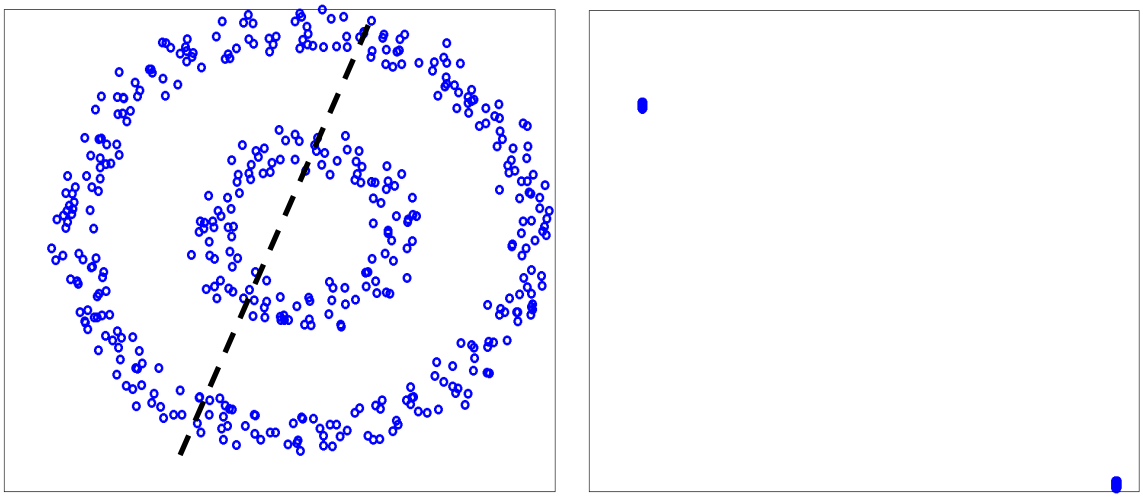
(a) What is a cluster?
(b) What features should be used?
(c) Should the data be normalized?
(d) Does the data contain any outliers?

(e) How do we define the pair-wise similarity?
(f) How many clusters are present in the data?
(g) Which clustering method should be used?
(h) Does the data have any clustering tendency?
(i) Are the discovered clusters and partition valid?

We will highlight and illustrate some of these challenges below.

3.1 Data representation

Data representation is one of the most important factors that influence the performance of the clustering algorithm. If the representation (choice of features) is good, the clusters are likely to be compact and isolated and even a simple clustering algorithm such as K-means will find them. Unfortunately, there is no universally good representation; the choice of representation must be guided by the domain knowledge. Figure 5(a) shows a dataset where K-means fails to partition it into the two "natural" clusters. The partition obtained by K-means is shown by a dotted line separating in Figure 5(a). However, when the same data points in (a) are represented using the top two eigenvectors of the RBF similarity matrix computed from the data in Figure 5(b), they become well separated, making it trivial for K-means to cluster the data [Ng *et al.* , 2001].



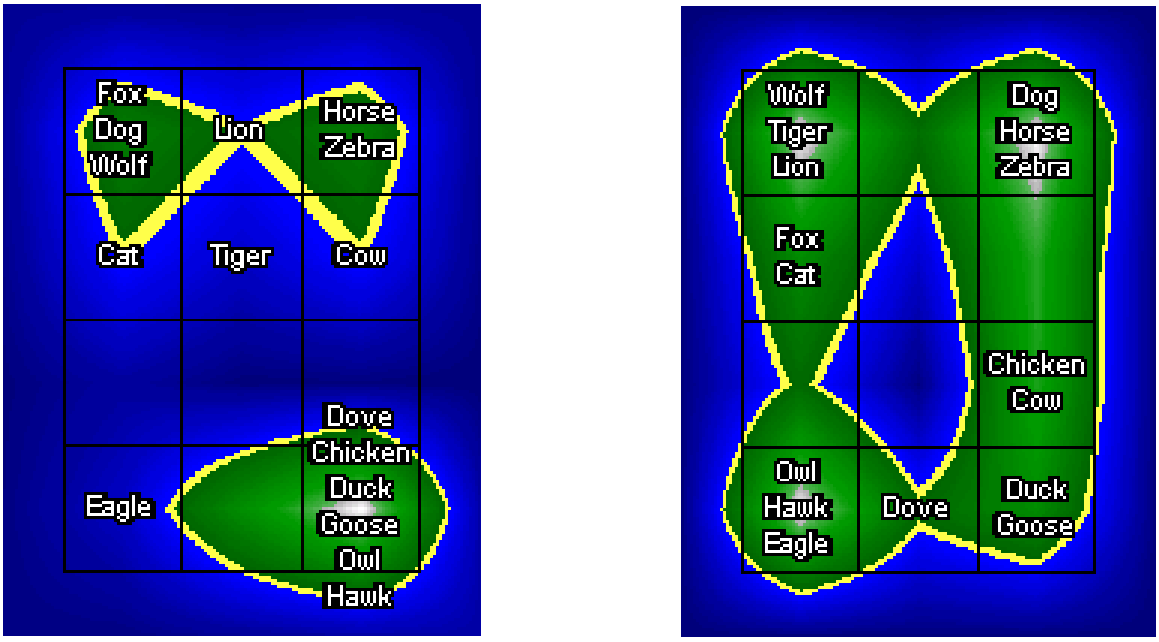
(a) (b)

Figure 5 Importance of a good representation. (a) "Two rings" dataset where K-means fails to find the two "natural" clusters; the dashed line shows the linear cluster separation boundary obtained by running K-means with K = 2. (b) a new representation of the data in (a) based on the top 2 eigenvectors of the graph Laplacian of the data, computed using an RBF kernel; K-means now can easily detect the two clusters

3.2 Purpose of grouping

The representation of the data is closely tied with the purpose of grouping. The representation must go hand in hand with the end goal of the user. An example dataset of 16 animals represented using 13 Boolean features was used in [Pampalk *et al.* , 2003] to

demonstrate how the representation affects the grouping. The animals are represented using 13 Boolean features related to their appearance and activity. When a large weight is placed on the appearance features compared to the activity features, the animals were clustered into *mammals vs. birds*. On the other hand, a large weight on the activity features clustered the dataset into *predators vs. non-predators*. Both these partitioning shown in Figure 6 are equally valid, and uncover meaningful structures in the data. It is up to the user to carefully choose his representation to obtain a desired clustering.



(a) **(b)**

Figure 6 Different weights on features result in different partitioning of the data. Sixteen animals are represented based on 13 Boolean features related to appearance and activity. (a) partitioning with large weights assigned to appearance based features; (b) a partitioning with large weights assigned to the activity features (figure reproduced from [**Pampalk *et al.* , 2003**]).

3.3 Number of Clusters

Automatically determining the number of clusters has been one of the most difficult problems in data clustering. Usually, clustering algorithms are run with different values of *K*; the best value of *K* is then chosen based on a criterion function. Figueiredo and Jain [Figueiredo & Jain, 2002] used the minimum message length (MML) criteria in conjunction with the Gaussian mixture model (GMM) to estimate *K*. Their approach starts with a large number of clusters, and gradually merges the clusters if this leads to a decrease in the MML criterion. Gap statistics [Tibshirani *et al.* , 2001] is another commonly used approach for deciding the number of clusters. The key assumption is that when dividing data into an optimal number of clusters, the resulting partition is most resilient to the random perturbation. Dirichlet Process (DP) [Rasmussen, 2000, Ferguson, 1973] introduces a non-parametric prior for the number of clusters. It is often used by

probabilistic models to derive a posterior distribution for the number of clusters, from which the most likely number of clusters can be computed.. Its key idea is to introduce a non-parametric Bayesian prior for the number of clusters. In spite of these objective criteria, it is not easy to decide which value of *K* leads to more meaningful clusters. 7(a) shows a 2-dimensional synthetic dataset generated from a mixture of six Gaussian components. The true labels of the points are shown in 7(e). When a mixture of Gaussians is fit to the data with 2, 5 and 6 components, shown in 7(b)-(d), respectively, each one of them seems to be a reasonable fit. This data is 2-dimensional, so we can easily visualize and assess how many clusters are good. But, this cannot be done when the data is high dimensional.

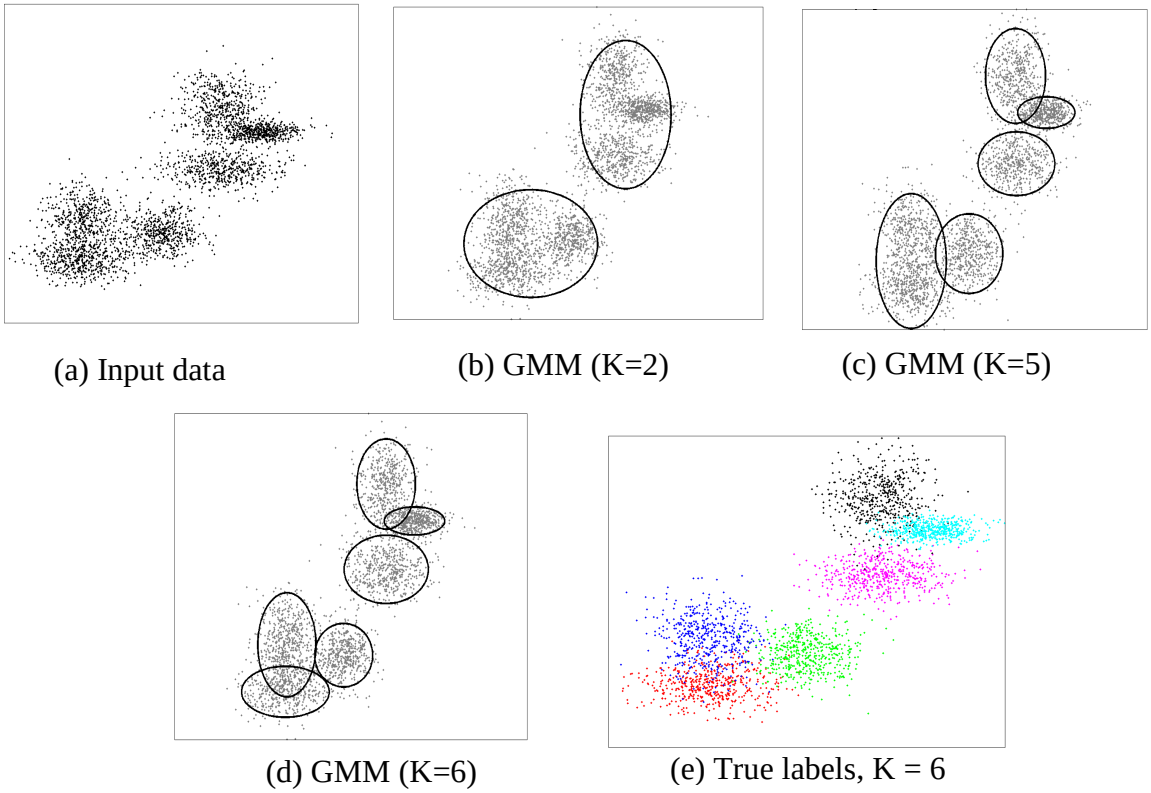
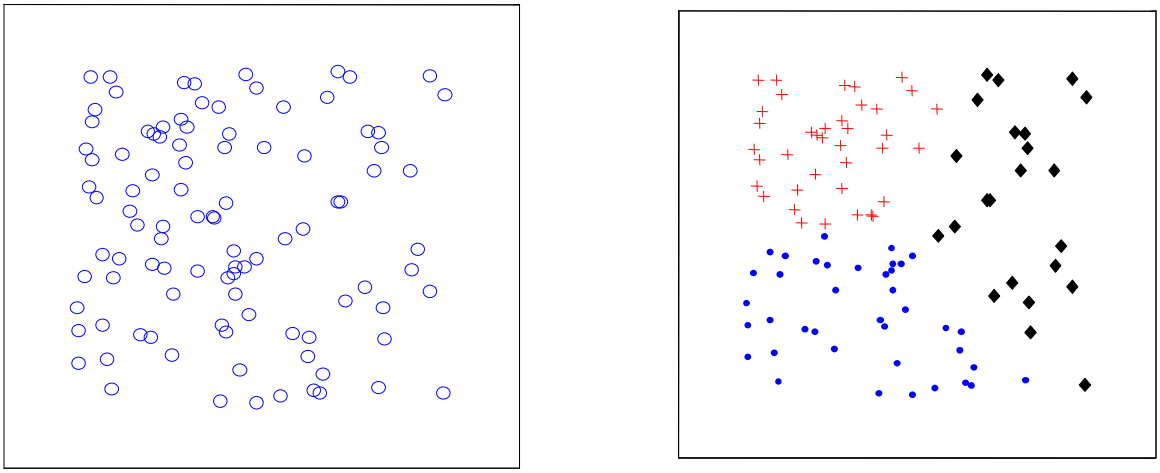


Figure 7 Automatic selection of number of clusters, K. (a) Input data generated from a mixture of six Gaussian distributions; (b)-(d) Gaussian mixture model (GMM) fit to the data with 2, 5 and 6 components, respectively; (e) true labels of the data.

3.4 Cluster validity

Clustering algorithms tend to find clusters in the data irrespective of whether or not any clusters are present. Figure 8(a) shows a dataset with no *natural* clustering; the points here were generated uniformly in a unit square. However, when the K-means algorithm is run on this data with $K = 3$, three clusters are identified as shown in Figure 8(b)! *Cluster validity* refers to formal procedures that evaluate the results of cluster analysis in a

quantitative and objective fashion [Jain & Dubes, 1988]. In fact, even before a clustering algorithm is applied to the data, the user should determine if the data even has a *clustering tendency* [Smith & Jain, 1984].



(a) **(b)**

Figure 8 Cluster validity. (a) A dataset with no "natural" clustering; (b) K-means partition with $K = 3$.

Cluster validity indices can be defined based on three different criteria: *internal*, *relative*, and *external* [Jain & Dubes, 1988]. Indices based on *internal criteria* assess the fit between the structure imposed by the clustering algorithm (clustering) and the data using the data alone. Indices based on *relative criteria* compare multiple structures (generated by different algorithms, for example) and decide which of them is better in some sense. *External indices* measure the performance by matching cluster structure to the a priori information, namely the "true" class labels (often referred to as ground truth). Typically, clustering results are evaluated using the external criterion, but if the true labels are available, why even bother with clustering? The notion of *cluster stability* [Lange *et al.* , 2004] is appealing as an internal stability measure. Cluster stability is measured as the amount of variation in the clustering solution over different sub-samples drawn from the input data. Different measures of variation can be used to obtain different stability measures. In [Lange *et al.* , 2004], a supervised classifier is trained on one of the subsamples of the data, by using the cluster labels obtained by clustering the subsample, as the *true* labels. The performance of this classifier on the testing subset(s) indicates the stability of the clustering algorithm. In model based algorithms (e.g., centroid based representation of clusters in K-means, or Gaussian Mixture Models), the distance between the models found for different subsamples can be used to measure the stability [von Luxburg & David, 2005]. Shamir and Tishby [Shamir & Tishby, 2008] define stability as the generalization ability of a clustering algorithm (in PAC-Bayesian sense). They argue that since many algorithms can be shown to be asymptotically stable, the *rate* at which the asymptotic stability is reached with respect to the number of samples is a more useful measure of cluster stability.

3.5 Comparing clustering algorithms

Different clustering algorithms often result in entirely different partitions even on the same data; see Figure 9. Seven different algorithms were applied to cluster the 15 two-dimensional points in 9(a). FORGY, ISODATA, CLUSTER, and WISH are partitional algorithms that minimize the squared error criterion (they are variants of the basic K-means algorithm). Of the remaining three algorithms, MST (minimum spanning tree) can be viewed as a single-link hierarchical algorithm, and JP is a nearest neighbor clustering algorithm. Note that a hierarchical algorithm can be used to generate a partition by specifying a threshold on the similarity. It is evident that none of the clustering is superior to the other, but some are similar to the other.

An interesting question is to identify algorithms that generate similar partitions irrespective of the data. In other words, can we cluster the clustering algorithms? Jain et al. [Jain *et al.* , 2004] clustered 35 different clustering algorithms into 5 groups based on their partitions on 12 different datasets. The similarity between the clustering algorithms is measured as the averaged similarity between the partitions obtained on the 12 datasets. The similarity between a pair of partitions is measured using the Adjusted Rand Index (ARI). A hierarchical clustering of the 35 clustering algorithms is shown in Figure 10(a). It is not surprising to see that the related algorithms are clustered together. For a visualization of the similarity between the algorithms, the 35 algorithms are also embedded in a two-dimensional space; this is achieved by applying the Sammon's projection algorithm [J. W. Sammon, 1969] to the 35x35 similarity matrix. Figure 10(b) shows that all the CHAMELEON variations (6, 8-10) are clustered into a single cluster. This plot suggests that the clustering algorithms following the same clustering strategy result in similar clustering in spite of minor variations in the parameters or objective functions involved.

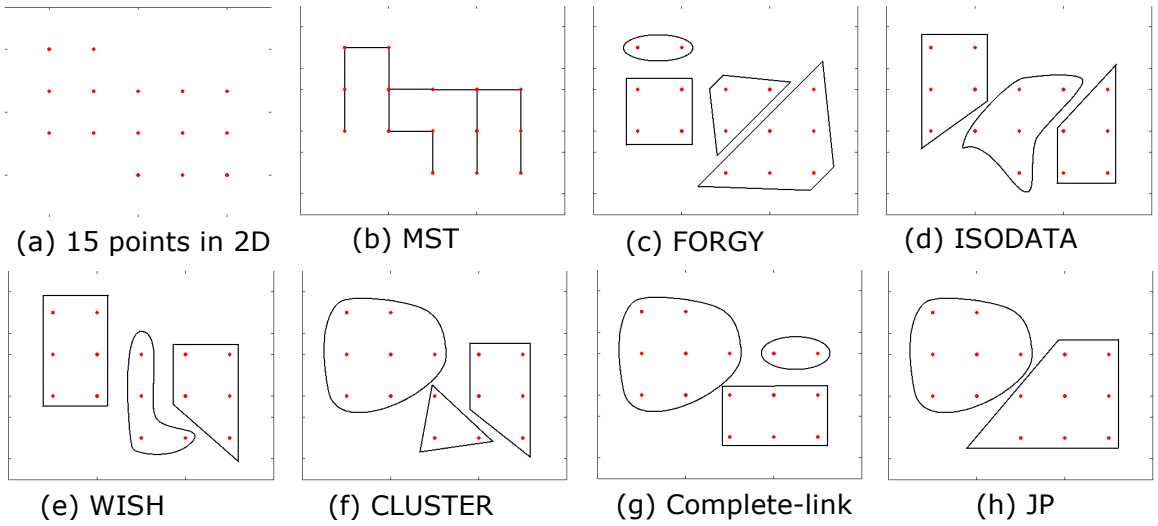


Figure 9 Several clustering of fifteen patterns in two dimensions: (a) fifteen patterns; (b) minimum spanning tree of the fifteen patterns; (c) clusters from FORGY; (d) clusters from ISODATA; (e) clusters from WISH; (f) clusters from CLUSTER; (g) clusters from complete-link hierarchical clustering; and (h) clusters from Jarvis-Patrick clustering algorithm. (Figure reproduced from [**Dubes & Jain, 1976**]).

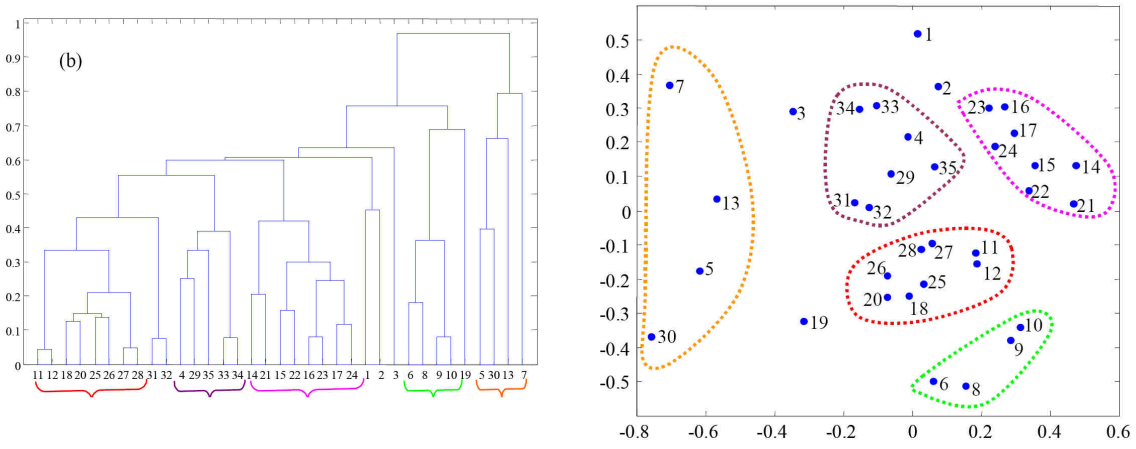


Figure 10 Clustering of clustering algorithms**.** (a) Hierarchical clustering of 35 different algorithms; (b) Sammon's mapping of the 35 algorithms into a two-dimensional space, with the clusters highlighted for visualization. The algorithms in the group (4, 29, 31-35) correspond to K-means, spectral clustering, Gaussian mixture models, and Ward's linkage. The algorithms in group (6, 8-10) correspond to CHAMELEON algorithm with different objective functions.

Clustering algorithms can also be compared at the theoretical level based on their objective functions. In order to perform such a comparison, a distinction should be made between a *clustering method* and a *clustering algorithm* [Jain & Dubes, 1988]. A clustering method is a general strategy employed to solve a clustering problem. A

clustering algorithm, on the other hand, is simply an instance of a method. For instance, minimizing the squared error is a clustering method, and there are many different clustering algorithms, including K-means, that implement the minimum squared error method. Some equivalence relationships even between different clustering methods have been shown. For example, Dhillon et al. [Dhillon *et al.* , 2004] show that spectral methods and kernel K-means are equivalent; for a choice of kernel in spectral clustering, there exists a kernel for which the objective functions of Kernel K-means and spectral clustering are the same. The equivalence between non-negative matrix factorization for clustering and kernel K-means algorithm is shown in [Ding *et al.* , 2005]. All these methods are directly related to the analysis of eigenvectors of the similarity matrix.

The above discussion underscores one of the important facts about clustering; *there is no best clustering algorithm*. Each clustering algorithm imposes a structure on the data either explicitly or implicitly. When there is a good match between the model and the data, good partitions are obtained. Since the structure of the data is not known a priori, one needs to try competing and diverse approaches to determine an appropriate algorithm for the clustering task at hand. This idea of no best clustering algorithm is partially captured by the impossibility theorem [Kleinberg, 2002], which states that no single clustering algorithm simultaneously satisfies the three basic axioms of data clustering, i.e., scale invariance, consistency, and richness.

3.6 Admissibility analysis of clustering algorithms

Fisher and Van Ness [Fisher & vanNess, 1971] formally analyzed clustering algorithms with the objective of comparing them and providing guidance in choosing a clustering procedure. They defined a set of *admissibility criteria* for clustering algorithms that test the sensitivity of clustering algorithms with respect to the changes that do not alter the essential structure of the data. A clustering is called *A-admissible* if it satisfies criterion *A*. Example criteria include *convex, point and cluster proportion, cluster omission*, and *monotone*. They are briefly described below.

- *Convex:* A clustering algorithm is *convex-admissible* if it results in a clustering where the convex hulls of clusters do not intersect.
- *Cluster proportion:* A clustering algorithm is *cluster-proportion admissible* if the cluster boundaries do not alter even if some of the clusters are duplicated an arbitrary number of times.
- *Cluster omission:* A clustering algorithm is *omission-admissible* if by removing one of the clusters from the data and re-running the algorithm, the clustering on the remaining *K-1* clusters is identical to the one obtained on them with *K* clusters.
- *Monotone:* A clustering algorithm is *monotone-admissible* if the clustering results do not change when a monotone transformation is applied to the elements of the similarity matrix.

Fisher and Van Ness proved that one cannot construct algorithms that satisfy certain admissibility criteria. For example, if an algorithm is monotone-admissible, it cannot be a hierarchical clustering algorithm.

Kleinberg [Kleinberg, 2002] addressed a similar problem, where he defined three criteria:

- *Scale invariance:* An arbitrary scaling of the similarity metric must not change the clustering results.
- *Richness:* The clustering algorithm must be able to achieve all possible partitions on the data.
- *Consistency:* By shrinking within-cluster distances and stretching between-cluster distances, the clustering results must not change.

Kleinberg also provides results similar to that of [Fisher & vanNess, 1971], showing that it is impossible to construct an algorithm that satisfies all these properties hence the title of his paper "An Impossibility Theorem for Clustering". Further discussions in [Kleinberg, 2002] reveal that a clustering algorithm can indeed be designed by relaxing the definition of *satisfying* a criterion to *nearly-satisfying* the criterion.

4 Trends in data clustering

Information explosion is not only creating large amounts of data but also a diverse set of data, both *structured* and *unstructured*. *Unstructured data* is a collection of objects that do not follow a specific format. For example, images, text, audio, video etc. On the other hand, in *structured data*, there are semantic relationships within each object that are important. Most clustering approaches ignore the structure in the objects to be clustered and use a feature vector based representation for both structured and unstructured data. The traditional view of data partitioning based on vector-based feature representation does not always serve as an adequate framework. Examples include objects represented using sets of points [Lowe, 2004], consumer purchase records [Guha *et al.* , 2000], data collected from questionnaires and rankings [Critchlow, 1985], social networks [Wasserman & Faust, 1994], and data streams [Guha *et al.* , 2003b]. Models and algorithms are being developed to process huge volumes of heterogeneous data. A brief summary of some of the recent trends in data clustering is presented below.

4.1 Clustering ensembles

The success of ensemble methods for supervised learning has motivated the development of ensemble methods for unsupervised learning [Fred & Jain, 2002]. The basic idea is that by taking *multiple looks* at the same data, one can generate multiple partitions (*clustering ensemble*) of the same data. By combining the resulting partitions, it is possible to obtain a good data partitioning even when the clusters are not compact and well separated. Fred and Jain used this approach by taking an ensemble of partitions obtained by K-means; the ensemble was obtained by changing the value of K and using random cluster initializations. These partitions were then combined using a co-occurrence matrix that resulted in a good separation of the clusters. An example of a clustering

ensemble is shown in Figure 11 where a "two-spiral" dataset is used to demonstrate its effectiveness. K-means is run multiple, say N, times with varying values of the number of clusters *K*. The new similarity between a pair of points is defined as the number of times the two points co-occur in the same cluster in N runs of K-means. The final clustering is obtained by clustering the data based on the new pair-wise similarity. Strehl and Ghosh [Strehl & Ghosh, 2003] proposed several strategies for integrating multiple partitions.

There are many different ways of generating a clustering ensemble and then combining the partitions. For example, multiple data partitions can be generated by: (i) applying different clustering algorithms, (ii) applying the same clustering algorithm with different values of parameters or initializations, and (iii) combining of different data representations (feature spaces) and clustering algorithms. The evidence accumulation step that combines the information provided by the different partitions can be viewed as learning the similarity measure among the data points.

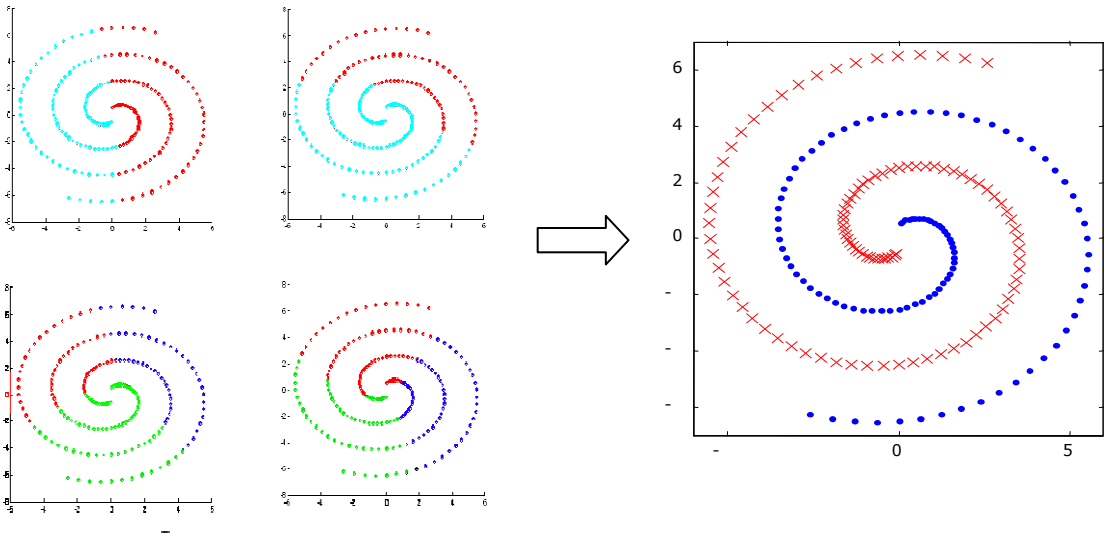


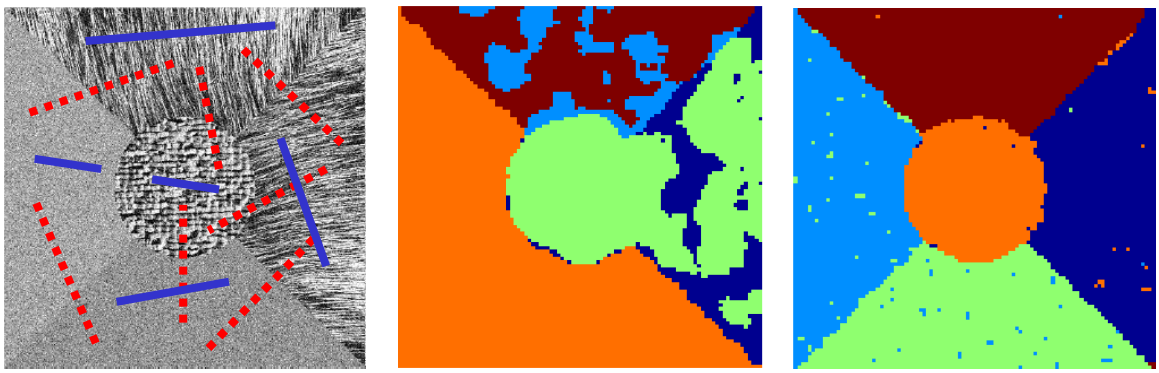
Figure 11 Clustering ensembles. Multiple runs of K-means are used to learn the pair-wise similarity using the "co-occurrence" of points in clusters. This similarity can be used to detect arbitrary shaped clusters.

4.2 Semi-supervised clustering

Clustering is inherently an ill-posed problem where the goal is to partition the data into some unknown number of clusters based on intrinsic information alone. Data driven nature of clustering makes it very difficult to design clustering algorithms that will correctly find clusters in the given data. Any external or *side information* available along with the n x d pattern matrix or the n x n similarity matrix can be extremely useful in finding a good partition of data. Clustering algorithms that utilize such side information are said to be operating in a *semi-supervised mode* [Chapelle *et al.* , 2006]. There are two open questions: (i) how should the side information be specified? and (ii) how is it obtained in practice? One of the most common methods of specifying the side information is in the form of pair-wise constraints. A *must-link constraint* specifies that

the point pair connected by the constraint belong to the same cluster. On the other hand, a *cannot-link constraint* specifies that the point pair connected by the constraint do not belong to the same cluster. It is generally assumed that the constraints are provided by the domain expert. There is limited work on automatically deriving constraints from the data. Some attempts to derive constraints from domain ontology and other external sources into clustering algorithms include the usage of WordNet ontology, gene ontology, Wikipedia, etc. to guide clustering solutions. However, these are mostly feature constraints and not constraints on the instances [Hotho *et al.* , 2003, Banerjee *et al.* , 2007b, Liu *et al.* , 2004]. Other approaches for including side information include (i) "seeding", where some labeled data is used along with large amount of unlabeled data for better clustering [Basu *et al.* , 2002] and (ii) methods that allow encouraging or discouraging some links [Law *et al.* , 2005, Figueiredo *et al.* , 2006].

Figure 12 illustrates the semi-supervised learning in an image segmentation application [6]. The textured image to be segmented (clustered) is shown in Figure 12 (a). In addition to the image, a set of user-specified pair-wise constraints on the pixel labels are also provided. Figure 12 (b) shows the clustering obtained when no constraints are used, while Figure 12 (c) shows improved clustering with the use of constraints. In both the cases, the number of clusters was assumed to be known (K = 5).



(a) Input image and constraints (b) No constraints (c) 10% pixels in constraints

Figure 12 Semi-supervised learning. (a) Input image with must-link (solid blue lines) and must not link (broken red lines) constraints. (b) Clustering (segmentation) without constraints. (c) Improved clustering with 10% of the data points included in the pair-wise constraints [6].

Most approaches [Basu *et al.* , 2004, Lu & Leen, 2007, Chapelle *et al.* , 2006, A. Bar-Hillel & Weinshall, 2003, Hofmann & Buhmann, 1997] to semi-supervised clustering modify the objective function of existing clustering algorithms to incorporate the pair-wise constraints. It is desirable to have an approach to semi-supervised clustering that can improve the performance of an already existing clustering algorithm without modifying it. *BoostCluster* [Liu *et al.* , 2007] adopts this philosophy and follows a boosting framework to improve the performance of any given clustering algorithm using pair-wise constraints. It iteratively modifies the input to the clustering algorithm by generating new data

representations (transforming the n x n similarity matrix) such that the pair-wise constraints are satisfied while also maintaining the integrity of the clustering output. Figure 13 shows the performance of BoostCluster evaluated on handwritten digit database in the UCI repository [Blake & Merz, 1998] with 4,000 points in 256-dimensional feature space. BoostCluster is able to improve the performance of all the three commonly used clustering algorithms, K-means, single-link, and Spectral clustering as pair-wise constraints are added to the data. Only must-link constraints are specified here and the number of true clusters is assumed to be known (K=10).

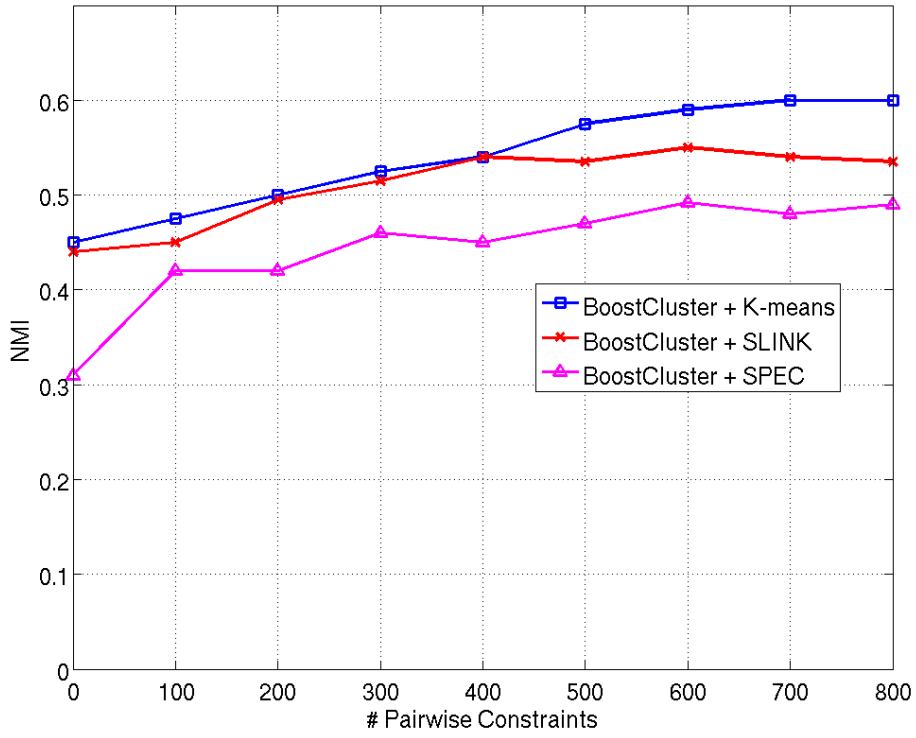


Figure 13 Performance of BoostCluster (measured using Normalized Mutual Information (NMI)) as the number of pair-wise constraints are increased. The three plots correspond to boosted performance of K-means, Single-Link (SLINK), and Spectral clustering (SPEC).

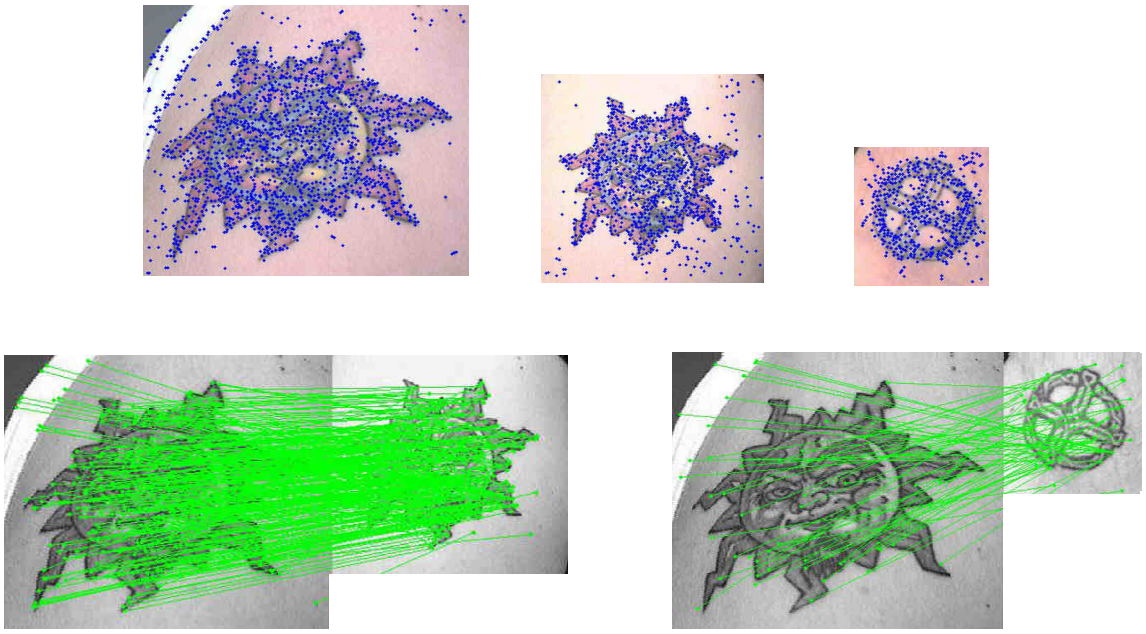
4.3 Large scale clustering

Large-scale data clustering addresses the challenge of clustering millions of data points that are represented in thousands of features. Table 1 shows a few examples of real-world applications for large-scale data clustering. Below, we review the application of large-scale data clustering to content-based image retrieval.

Application	Description	# Objects	# Features

document clustering	group documents of similar topics	10^6	10^4
gene clustering	group genes with similar expression levels	10^5	10^2
content-based image retrieval	quantize low-level image features	10^9	10^2
clustering of earth science data	derive climate indices	10^5	10^2

Table 1: Example applications of large-scale data clustering



(a) 370 (b) 64

Figure 14 Three tattoo images represented using SIFT key points. (a) A pair with similar images has 370 matching key points; (b) pair with different images has 64 matching key points. The green lines show the matching key-points between the images [**Lee *et al.* , 2008**].

The goal of Content Based Image Retrieval (CBIR) is to retrieve visually similar images to a given query image. Although the topic has been studied for the past 15 years or so, there has been only a limited success. Most early work on CBIR was based on computing *color*, *shape*, and *texture* based features and using them to define a similarity between the images. A 2008 survey on CBIR highlights the different approaches used for CBIR

through time [Datta *et al.* , 2008]. Recent approaches for CBIR use point based features. For example, SIFT [Lowe, 2004] descriptors can be used to represent the images (see Figure 14). However, once the size of the image database increases (~10 million), and assuming 10 milliseconds to compute the matching score between an image pair, a linear search would take approximately 30 hours to answer one query. This clearly is unacceptable.

On the other hand, text retrieval applications are much faster. It takes about one-tenth of a second to search 10 billion documents in Google. A novel approach for image retrieval is to convert the problem into a text retrieval problem. The key points from all the images are first clustered into a large number of clusters (which is still much less than the number of key points themselves). These are called as the *visual words.* An image is then represented using the number of key-points from the image that are in each word or each cluster. Now that a vector based representation (of size equal to the number of clusters) is obtained, searching becomes very efficient. One of the major challenges in quantizing key points is the number of objects to be clustered. For a collection of 1,000 images with an average of 1,000 key points and the target number of 5,000 visual words, it requires clustering 10^6 objects into 5,000 clusters.

A large number of clustering algorithms have been developed to efficiently handle large-size data sets. Most of these studies can be classified into four categories:

- Efficient Nearest Neighbor (NN) Search. One of the basic operations in any data clustering algorithm is to decide the cluster membership of each data point, which requires NN search. Algorithms for efficient NN search are either tree-based (e.g. kd-tree [Moore, 1998, Muja & Lowe, 2009]) or random projection based (e.g., Locality Sensitive Hash [Buhler, 2001]).
- Data Summarization. Approaches in this category improve the clustering efficiency by first summarizing a large data set into a relatively small subset, and then applying the clustering algorithms to the summarized data set. Example algorithms include BIRCH [Zhang *et al.* , 1996], divide-and-conquer [Steinbach *et al.* , 2000], and coreset K-means [Har-peled & Mazumdar, 2004].
- Distributed Computing. Approaches in this category [Dhillon & Modha, 1999] divide each step of a data clustering algorithm into a number of procedures that can be computed independently. These independent computational procedures will then be carried out in parallel by different processors to reduce the overall computation time.
- Incremental Clustering. Approaches in this category [Bradley *et al.* , 1998] are designed to operate in a single pass over data points to improve the efficiency of data clustering. This is in contrast to most clustering algorithms that require multiple passes over data points before identifying the cluster centers.
- Sampling-based methods. Approaches in this category are algorithms like CURE [Guha *et al.* , 1998, Kollios *et al.* , 2003] that subsample a large dataset selectively, and perform clustering over the smaller set, which is later transferred to the larger dataset.

4.4 Multi-way clustering

Objects or entities to be clustered are often formed by a combination of *related* heterogeneous components. For example, a document is made of words, title, authors, citations, etc. While objects can be converted into a pooled feature vectors of its components prior to clustering, it is not a natural representation of the objects and may result in poor clustering performance.

Co-clustering [Hartigan, 1972, Mirkin, 1996] aims to cluster both features and instances of the data (or both rows and columns of the n x d pattern matrix) simultaneously to identify the subset of features where the resulting clusters are meaningful according to certain evaluation criterion. This problem was first studied under the name *direct clustering* by Hartigan [Hartigan, 1972]. It is also called *bi-dimensional clustering* [Cheng & Church, 2000], *double clustering*, *coupled clustering* or *bimodal clustering*. This notion is also related to subspace clustering where all the clusters are identified in a common subspace. Co-clustering is most popular in the field of bioinformatics, especially in gene clustering, and has also been successfully applied to document clustering [Slonim & Tishby, 2000, Dhillon *et al.* , 2003].

The co-clustering framework was extended to *multi-way clustering* in [Bekkerman *et al.* , 2005] to cluster a set of objects by simultaneously clustering their heterogeneous components. Indeed, the problem is much more challenging because different pairs of components may participate in different types of similarity relationships. In addition, some relations may involve more than two components. Banerjee et al. [Banerjee *et al.* , 2007a] present a family of multi-way clustering schemes that is applicable to a class of loss functions known as Bregman divergences. Sindhwani et al.[Sindhwani *et al.* , 2008] apply semi-supervised learning in the co-clustering framework.

4.5 Heterogeneous data

In traditional pattern recognition settings, a feature vector consists of measurements of different properties of an object. This representation of objects is not a natural representation for several types of data. *Heterogeneous data* refers to the data where the objects may not be *naturally* represented using a fixed length feature vector.

Rank Data: Consider a dataset generated by ranking of a set of n movies by different people; only some of the n objects are ranked. The task is to cluster the users whose rankings are similar and also to identify the ‘representative rankings’ of each group [Mallows, 1957, Critchlow, 1985, Busse *et al.* , 2007].

Dynamic Data: Dynamic data, as opposed to static data, can change over the course of time e.g., blogs, Web pages, etc. As the data gets modified, clustering must be updated accordingly. A *data stream* is a kind of dynamic data that is transient in nature, and

cannot be stored on a disk. Examples include network packets received by a router and stock market, retail chain, or credit card transaction streams. Characteristics of the data streams include their high volume and potentially unbounded size, sequential access and dynamically evolving nature. This imposes additional requirements to traditional clustering algorithms to rapidly process and summarize the massive amount of continuously arriving data. It also requires the ability to adapt to changes in the data distribution, the ability to detect emerging clusters and distinguish them from outliers in the data, and the ability to merge old clusters or discard expired ones. All of these requirements make data stream clustering a significant challenge since they are expected to be single-pass algorithms [Guha *et al.* , 2003b]. Because of the high-speed processing requirements, many of the data stream clustering methods [Guha *et al.* , 2003a, Aggarwal *et al.* , 2003, Cao *et al.* , 2006] are extensions of simple algorithms such as K-means, K-medoid or density-based clustering, modified to work in a data stream environment setting.

Graph Data: Several objects, such as chemical compounds, protein structures, etc. can be represented most naturally as graphs. Many of the initial efforts in graph clustering have focused on extracting graph features to allow existing clustering algorithms to be applied to the graph feature vectors [Tsuda & Kudo, 2006]. The features can be extracted based on patterns such as frequent subgraphs, shortest paths, cycles, and tree-based patterns. With the emergence of kernel learning, there have been growing efforts to develop kernel functions that are more suited for graph-based data [Kashima *et al.* , 2003]. One way to determine the similarity between graphs is by aligning their corresponding adjacency matrix representations [Umeyama, 1988].

Relational Data: Another area that has attracted considerable interest is clustering relational (network) data. Unlike the clustering of graph data, where the objective is to partition a collection of graphs into disjoint groups, the task here is to partition a large graph (i.e., network) into cohesive subgraphs based on their link structure and node attributes. The problem becomes even more complicated when the links (which represent relations between objects) are allowed to have diverse types. One of the key issues is to define an appropriate clustering criterion for relational data. Newman's modularity function [Newman, 2006, White & Smyth, 2005] is a widely-used criterion for finding community structures in networks, but the measure considers only the link structure and ignores attribute similarities. Since real networks are often dynamic, another issue is to model the evolutionary behavior of networks, taking into account changes in the group membership and other characteristic features [L.Backstrom *et al.* , 2006].

5. Summary

Organizing data into sensible groupings arises naturally in many scientific fields. It is, therefore, not surprising to see the continued popularity of data clustering. It is important to remember that cluster analysis is an exploratory tool; the output of clustering algorithms only suggest hypotheses. While thousands of clustering algorithms have been published and new ones continue to appear, there is no best algorithm. Most algorithms, including the popular K-means, are admissible algorithms. Indeed, the search for a best

clustering algorithm is fruitless and contrary to the exploratory nature of clustering. The challenge in data clustering is to (i) incorporate domain knowledge in the algorithm, (ii) find appropriate representation and measure of similarity, (iii) validate clustering, (iv) devise a rational basis for comparing methods, (v) combine 'multiple looks" of the same data, and (vi) develop efficient algorithms for clustering large datasets (billions of points in thousands of dimensions).

Acknowledgement

I would like to acknowledge the National Science Foundation and the Office of Naval research for supporting my research in data clustering, dimensionality reduction, classification, and semi-supervised learning. I am grateful to Rong Jin, Pang-Ning Tan and Pavan Mallapragada for helping me prepare the King-Sun Fu lecture as well as this manuscript. I have learned much from and enjoyed my fruitful collaborations in data clustering with Eric Backer, Joachim Buhmann, (late) Richard Dubes, Mario Figueiredo, Patrick Flynn, Ana Fred, Martin Law, J. C. Mao, M. Narasimha Murty, Steve Smith, and Alexander Topchy.

References

[jst, n.d.] . *Jstor*. http://www.jstor.org.

[gsc, 2009] . 2009 (February). *Google scholar*. http://scholar.google.com.

[A. Bar-Hillel & Weinshall, 2003] A. BAR-HILLEL, T. HERTZ, N. SHENTAL, & WEINSHALL, D. 2003. Learning distance functions using equivalence relations. *Pages 11–18 of: In proceedings of 20th international conference on machine learning*.

[Aggarwal *et al.* , 2003] AGGARWAL, CHARU C., HAN, JIAWEI, WANG, JIANYONG, & YU, PHILIP S. 2003. A framework for clustering evolving data streams. *Pages 81–92 of: Vldb '2003: Proceedings of the 29th international conference on very large data bases*. VLDB Endowment.

[Agrawal *et al.* , 1998] AGRAWAL, RAKESH, GEHRKE, JOHANNES, GUNOPULOS, DIMITRIOS, & RAGHAVAN, PRABHAKAR. 1998. Automatic subspace clustering of high dimensional data for data mining applications. *Pages 94–105 of: Proceedings of the acm sigmod*.

[Anderberg, 1973] ANDERBERG, M. R. 1973. *Cluster analysis for applications*. Academic Press.

[Arabie & Hubert, 1994] ARABIE, P., & HUBERT, L. 1994. *Advanced methods in marketing research*. Oxford: Blackwell. Chap. Cluster Analysis in Marketing Research, pages 160–189.

[Backer, 1978]BACKER, ERIC. 1978. *Cluster analysis by optimal decomposition of induced fuzzy sets*. Delft University Press.

[Baldi & Hatfield, 2002] BALDI, P., & HATFIELD, G. 2002. *Dna microarrays and gene expression*. Cambridge University Press.

[Ball & Hall, 1965] BALL, G., & HALL, D. 1965. *ISODATA, a novel method of data anlysis and pattern classification*. Tech. rept. NTIS AD 699616. Stanford Research Institute, Stanford, CA.
[Banerjee *et al.* , 2004] BANERJEE, ARINDAM, MERUGU, SRUJANA, DHILLON, INDERJIT, & GHOSH, JOYDEEP. 2004. Clustering with bregman divergences. *Pages 234–245 of: Journal of machine learning research*.
[Banerjee *et al.* , 2007a] BANERJEE, ARINDAM, BASU, SUGATO, & MERUGU, SRUJANA. 2007a. Multi-way clustering on relation graphs. *In: Proceedings of the 7th siam international conference on data mining*.
[Banerjee *et al.* , 2007b] BANERJEE, S., RAMANATHAN, K., & GUPTA, A. 2007b. Clustering short texts using Wikipedia. *In: Proceedings of the sigir*.
[Basu *et al.* , 2002] BASU, SUGATO, BANERJEE, ARINDAM, & MOONEY, RAYMOND. 2002. Semi-supervised clustering by seeding. *In: In proceedings of 19th international conference on machine learning (icml-2002*.
[Basu *et al.* , 2004] BASU, SUGATO, BILENKO, MIKHAIL, & MOONEY, RAYMOND J. 2004. A probabilistic framework for semi-supervised clustering. *Pages 59–68 of: Kdd '04: Proceedings of the tenth acm sigkdd international conference on knowledge discovery and data mining*. New York, NY, USA: ACM.
[Basu *et al.* , 2008] BASU, SUGATO, DAVIDSON, IAN, & WAGSTAFF, KIRI (eds). 2008. *Constrained clustering: Advances in algorithms, theory and applications*. Data Mining and Knowledge Discovery, vol. 3. Chapman & Hall/CRC.
[Bekkerman *et al.* , 2005] BEKKERMAN, RON, EL-YANIV, RAN, & MCCALLUM, ANDREW. 2005. Multi-way distributional clustering via pairwise interactions. *Pages 41–48 of: Icml '05: Proceedings of the 22nd international conference on machine learning*. New York, NY, USA: ACM.
[Belkin & Niyogi, 2002] BELKIN, MIKHAIL, & NIYOGI, PARTHA. 2002. Laplacian eigenmaps and spectral techniques for embedding and clustering. *Pages 585–591 of: Advances in neural information processing systems 14*. MIT Press.
[Bezdek, 1981] BEZDEK, J. C. 1981. *Pattern recognition with fuzzy objective function algorithms*. Plenum Press.
[Bhatia & Deogun, 1998] BHATIA, S., & DEOGUN, J. 1998. Conceptual clustering in information retrieval. *Ieee transactions on systems, man and cybernetics*, **28**(B), 427–436.
[Bishop, 2006] BISHOP, CHRISTOPHER M. 2006. *Pattern recognition and machine learning*. Springer.
[Blake & Merz, 1998] BLAKE, & MERZ, C. J. 1998. *UCI repository of machine learning databases*.
[Blei *et al.* , 2003] BLEI, D. M., NG, A. Y., & JORDAN, M. I. 2003. Latent dirichlet allocation. *Journal of machine learning research*, **3**, 993–1022.
[Bradley *et al.* , 1998] BRADLEY, P. S., FAYYAD, U., & REINA, C. 1998. Scaling clustering algorithms to large databases. *In: Fourth international conference on knowledge discovery and data mining*.
[Buhler, 2001] BUHLER, J. 2001. Efficient large-scale sequence comparison by locality-sensitive hashing. *Bioinformatics*, **17**(5), 419–428.
[Busse *et al.* , 2007] BUSSE, LUDWIG M., ORBANZ, PETER, & BUHMANN, JOACHIM M. 2007. Cluster analysis of heterogeneous rank data. *Pages 113–120 of: Icml '07:*

Proceedings of the 24th international conference on machine learning. New York, NY, USA: ACM.
[Cao *et al.* , 2006] CAO, F., ESTER, M., QIAN, W., & ZHOU, A. 2006. Density-based clustering over an evolving data stream with noise. *In: Proc. siam conf. data mining*.
[Chapelle *et al.* , 2006] CHAPELLE, O., SCHÖLKOPF, B., & ZIEN, A. (eds). 2006. *Semi-supervised learning*. Cambridge, MA: MIT Press.
[Cheng & Church, 2000] CHENG, YIZONG, & CHURCH, GEORGE M. 2000. Biclustering of expression data. *Pages 93–103 of: Proceedings of the eighth international conference on intelligent systems for molecular biology*. AAAI Press.
[Connell & Jain, 2002] CONNELL, S.D., & JAIN, A.K. 2002. Writer adaptation for online handwriting recognition. *Pattern analysis and machine intelligence, ieee transactions on*, **24**(3), 329–346.
[Critchlow, 1985] CRITCHLOW, D. 1985. *Metric methods for analyzing partially ranked data*. Springer.
[Datta *et al.* , 2008] DATTA, RITENDRA, JOSHI, DHIRAJ, LI, JIA, & WANG, JAMES Z. 2008. Image retrieval: Ideas, influences, and trends of the new age. *Acm computing surveys*, **40**(2), article 5.
[Dhillon & Modha, 1999] DHILLON, I., & MODHA, D. 1999. A data-clustering algorithm on distributed memory multiprocessors. *Pages 245–260 of: In proceedings of the kdd'99 workshop on high performance knowledge discovery*.
[Dhillon *et al.* , 2003] DHILLON, INDERJIT S., MALLELA, SUBRAMANYAM, GUYON, ISABELLE, & ELISSEEFF, ANDRÉ. 2003. A divisive information-theoretic feature clustering algorithm for text classification. *Journal of machine learning research*, **3**, 2003.
[Dhillon *et al.* , 2004] DHILLON, INDERJIT S., GUAN, YUQIANG, & KULIS, BRIAN. 2004. Kernel k-means: spectral clustering and normalized cuts. *Pages 551–556 of: Kdd '04: Proceedings of the tenth acm sigkdd international conference on knowledge discovery and data mining*. New York, NY, USA: ACM Press.
[Ding *et al.* , 2005] DING, CHRIS, HE, XIAOFENG, & SIMON, HORST D. 2005. On the equivalence of nonnegative matrix factorization and spectral clustering. *Pages 606–610 of: Proc. siam international conference on data mining*.
[Drineas *et al.* , 1999] DRINEAS, P., FRIEZE, A., KANNAN, R., VEMPALA, S., & VINAY, V. 1999. Clustering large graphs via the singular value decomposition. *Machine learning*, **56**(1-3), 9–33.
[Dubes & Jain, 1976] DUBES, & JAIN. 1976. Clustering techniques: User's dilemma. *Pattern recognition*, 247–260.
[Duda *et al.* , 2001] DUDA, R., HART, P., & STORK, D. 2001. *Pattern classification*. 2 edn. New York: John Wiley & Sons.
[Dunn, 1973] DUNN, J. C. 1973. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Journal of cybernetics*, **3**, 32–57.
[Ester *et al.* , 1996] ESTER, MARTIN, PETER KRIEGEL, HANS, S, JÖRG, & XU, XIAOWEI. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. *Pages 226–231 of: In proceedings of the 2nd international conference on kdd*. AAAI Press.
[Ferguson, 1973] FERGUSON, THOMAS S. 1973. A bayesian analysis of some nonparametric problems. *Annals of statistics*, **1**, 209–230.

[Figueiredo *et al.* , 2006] FIGUEIREDO, M. A. T., CHANG, D. S., & MURINO, V. 2006. Clustering under prior knowledge with application to image segmentation. *Pages 401–408 of: Nips 19*.

[Figueiredo & Jain, 2002] FIGUEIREDO, MARIO, & JAIN, ANIL K. 2002. Unsupervised learning of finite mixture models. *Ieee transactions on pattern analysis and machine intelligence*, **24**(3), 381–396.

[Fisher & vanNess, 1971] FISHER, L., & VANNESS, J. 1971. Admissible clustering procedures. *Biometrika*.

[Forgy, 1965] FORGY, E. W. 1965. Cluster analysis of multivariate data: efficiency vs interpretability of classifications. *Biometrics*, **21**, 768–769.

[Frank & Todeschini, 1994] FRANK, ILDIKO E., & TODESCHINI, ROBERTO. 1994. *Data analysis handbook*. Elsevier Science Inc. Pages 227–228.

[Fred & Jain, 2002] FRED, A., & JAIN, A.K. 2002. Data clustering using evidence accumulation. *In: Proceedings of the international conference on pattern recognition (icpr)*.

[Frigui & Krishnapuram, 1999] FRIGUI, H., & KRISHNAPURAM, R. 1999. A robust competitive clustering algorithm with applications in computer vision. *Ieee transactions on pattern analysis and machine intelligence*, **21**, 450–465.

[Gantz, 2008] GANTZ, JOHN F. 2008 (March). *The diverse and exploding digital universe*. Available online at : http://www.emc.com/collateral/analyst-reports/diverse-exploding-digital-universe.pdf.

[Guha *et al.* , 2003a] GUHA, S., MEYERSON, A., MISHRA, N., MOTWANI, R., & L. O'CALLAGHAN ", TKDE (2003). 2003a. Clustering data streams: Theory and practice. *Transactions on knowledge discovery and engieering*.

[Guha *et al.* , 1998] GUHA, SUDIPTO, RASTOGI, RAJEEV, & SHIM, KYUSEOK. 1998 (June). CURE: an efficient clustering algorithm for large databases. *Pages 73–84 of: Acm sigmod international conference on management of data*.

[Guha *et al.* , 2000] GUHA, SUDIPTO, RASTOGI, RAJEEV, & SHIM, KYUSEOK. 2000. Rock: A robust clustering algorithm for categorical attributes. *Information systems*, **25**(5), 345 – 366.

[Guha *et al.* , 2003b] GUHA, SUDIPTO, MISHRA, NINA, & MOTWANI, RAJEEV. 2003b. Clustering data streams. *Ieee transactions on knowledge and data engineering*, **15**(3), 515–528.

[Hagen & Kahng, 1992] HAGEN, L., & KAHNG, A.B. 1992. New spectral methods for ratio cut partitioning and clustering. *Computer-aided design of integrated circuits and systems, ieee transactions on*, **11**(9), 1074–1085.

[Han & Kamber, 2000] HAN, JIAWEI, & KAMBER, MICHELINE. 2000. *Data mining: Concepts and techniques*. Morgan Kaufmann.

[Har-peled & Mazumdar, 2004] HAR-PELED, SARIEL, & MAZUMDAR, SOHAM. 2004. Coresets for k-means and k-median clustering and their applications. *Pages 291–300 of: In proc. 36th annu. acm sympos. theory comput*.

[Hartigan, 1972] HARTIGAN, J. A. 1972. Direct clustering of a data matrix. *Journal of the american statistical association*, **67**(337), 123–132.

[Hartigan, 1975] HARTIGAN, J.A. 1975. *Clustering algorithms*. John Wiley & Sons.

[Hofmann & Buhmann, 1997] HOFMANN, T., & BUHMANN, J.M. 1997. Pairwise data clustering by deterministic annealing. *Ieee transactions on pattern analysis and machine intelligence*, **19**(1), 1–14.
[Hofmann & Puzicha, 1998] HOFMANN, T., & PUZICHA, J. 1998. *Statistical models for co-occurrence data*. Tech. rept. Artificial Intelligence Laboratory, MIT.
[Hotho *et al.* , 2003] HOTHO, A., STAAB, S., & STUMME, G. 2003. Ontologies to improve text document clustering. *In: Proceedings of the icdm*.
[Hu *et al.* , 2007] HU, J., RAY, B. K., & SINGH, M. 2007. Statistical methods for automated generation of service engagement staffing plans. *Ibm j. res. dev.*, **51**(3), 281–293.
[Iwayama & Tokunaga, 1995] IWAYAMA, M., & TOKUNAGA, T. 1995. Cluster-based text categorization: a comparison of category search strategies. *Pages 273–281 of: Proceedings of the 18th acm international conference on research and development in information retrieval*.
[J. W. Sammon, 1969] J. W. SAMMON, JR. 1969. A nonlinear mapping for data structure analysis. *Ieee transactions on computers*, **18**, 401–409.
[Jain *et al.* , 2004] JAIN, A. K., TOPCHY, A., LAW, M. H. C., & BUHMANN, J. M. 2004. Landscape of clustering algorithms. *Pages 260–263 of: Proceedings of the international conference on pattern recognition*, vol. 1.
[Jain & Dubes, 1988] JAIN, ANIL K., & DUBES, RICHARD C. 1988. *Algorithms for clustering data*. Prentice Hall.
[Jain & Flynn, 1996] JAIN, ANIL K., & FLYNN, P. 1996. *Advances in image understanding*. IEEE Computer Society Press. Chap. Image segmentation using clustering, pages 65–83.
[Kashima *et al.* , 2003] KASHIMA, H., TSUDA, K., & INOKUCHI, A. 2003. Marginalized kernels between labeled graphs. *Pages 321–328 of: Proceedings of the twentieth international conference on machine learning*.
[Kashima *et al.* , 2008] KASHIMA, H., HU, J., RAY, B., & SINGH, M. 2008 (Dec.). K-means clustering of proportional data using L1 distance.
[Kaufman & Rousseeuw, 2005] KAUFMAN, LEONARD, & ROUSSEEUW, PETER J. 2005. *Finding groups in data : An introduction to cluster analysis*. Wiley series in Probability and Statistics.
[Kleinberg, 2002] KLEINBERG, JON. 2002. An impossibility theorem for clustering. *Pages 446–453 of: Nips 15*. MIT Press.
[Kollios *et al.* , 2003] KOLLIOS, G., GUNOPULOS, D., KOUDAS, N., & BERCHTOLD, S. 2003. Efficient biased sampling for approximate clustering and outlier detection in large data sets. *Knowledge and data engineering, ieee transactions on*, **15**(5), 1170–1187.
[Lange *et al.* , 2005] LANGE, T., LAW, M. H., JAIN, A. K., & BUHMANN, J. 2005 (June). Learning with constrained and unlabelled data. *Pages 730–737 of: Ieee computer society conference on computer vision and pattern recognition*, vol. 1.
[Lange *et al.* , 2004] LANGE, TILMAN, ROTH, VOLKER, BRAUN, MIKIO L., & BUHMANN, JOACHIM M. 2004. Stability-based validation of clustering solutions. *Neural computation*, **16**(6), 1299–1323.
[Law *et al.* , 2005] LAW, MARTIN, TOPCHY, ALEXANDER, & JAIN, A. K. 2005. Model-based clustering with probabilistic constraints. *Pages 641–645 of: Proc. siam conference on data mining*.

[L.Backstrom *et al.* , 2006] L.BACKSTROM, HUTTENLOCHER, D., KLEINBERG, J., & LAN, X. 2006. Group formation in large social networks: Membership, growth, and evolution. *In: Proceedings of the kdd*.
[Lee *et al.* , 2008] LEE, JUNG-EUN, JAIN, ANIL K., & JIN, RONG. 2008. Scars, marks and tattoos (SMT): Soft biometric for suspect and victim identification. *In: Proceedings of the biometric symposium*.
[Linde *et al.* , 1980] LINDE, Y., BUZO, A., & GRAY, R. 1980. An algorithm for vector quantizer design. *Ieee transactions on communications*, **28**, 84–94.
[Liu *et al.* , 2004] LIU, J., WANG, W., & YANG, J. 2004. A framework for ontology-driven subspace clustering. *In: Proceedings of the kdd*.
[Liu *et al.* , 2007] LIU, YI, JIN, RONG, & JAIN, A. K. 2007. Boostcluster: Boosting clustering by pairwise constraints. *Pages 450–459 of: In proceedings of the 13th acm sigkdd international conference on knowledge discovery and data mining (kdd)*.
[Lloyd, 1982] LLOYD, S. 1982. Least squares quantization in PCM. *Ieee transactions on information theory*, **28**, 129–137. Originally as an unpublished Bell laboratories Technical Note (1957).
[Lowe, 2004] LOWE, DAVID G. 2004. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, **60**(2), 91–110.
[Lu & Leen, 2007] LU, ZHENGDONG, & LEEN, TODD K. 2007. Penalized probabilistic clustering. *Neural comput.*, **19**(6), 1528–1567.
[MacQueen, 1967] MACQUEEN, J. 1967. Some methods for classification and analysis of multivariate observations. *Pages 281–297 of: Fifth berkeley symposium on mathematics, statistics and probability*. University of California Press.
[Mallows, 1957] MALLOWS, C. L. 1957. Non-null ranking models. *Biometricka*, **44**, 114–130.
[Mao & Jain, 1996] MAO, J., & JAIN, A.K. 1996. A self-organizing network for hyper-ellipsoidal clustering (hec). *Ieee transactions on neural networks*, **7**(Jan), 16–29.
[Meila & Shi, 2001] MEILA, MARINA, & SHI, JIANBO. 2001. A random walks view of spectral segmentation. *In: Proceedings of the aistatas*.
[Merriam-Webster Online Dictionary, 2008] MERRIAM-WEBSTER ONLINE DICTIONARY. 2008 (Feb). *"cluster analysis"*. http://www.merriam-webster-online.com.
[Mirkin, 1996] MIRKIN, BORIS. 1996. *Mathematical classification and clustering*. Kluwer Academic Publishers.
[Moore, 1998] MOORE, ANDREW W. 1998. Very fast EM-based mixture model clustering using multiresolution kd-trees. *Pages 543–549 of: Nips*.
[Muja & Lowe, 2009] MUJA, M., & LOWE, D. G. 2009. Fast approximate nearest neighbors with automatic algorithm configuration. *In: Proccedings of the international conference on computer vision theory and applications (visapp'09)*.
[Newman, 2006] NEWMAN, M. E. J. 2006. Modularity and community structure in networks. *In: Proc. national academy of sciences, usa*.
[Ng *et al.* , 2001] NG, ANDREW Y., JORDAN, MICHAEL I., & WEISS, YAIR. 2001. On spectral clustering: Analysis and an algorithm. *Pages 849–856 of: Advances in neural information processing systems 14*. MIT Press.
[Pampalk *et al.* , 2003] PAMPALK, ELIAS, DIXON, SIMON, & WIDMER, GERHARD. 2003. On the evaluation of perceptual similarity measures for music. *Pages 7–12 of:*

Proceedings of the sixth international conference on digital audio effects (DAFx-03). London, UK: Queen Mary University of London.
[Pelleg & Moore, 1999] PELLEG, DAN, & MOORE, ANDREW. 1999. Accelerating exact k-means algorithms with geometric reasoning. *Pages 277–281 of:* CHAUDHURI, SURAJIT, & MADIGAN, DAVID (eds), *Proceedings of the fifth international conference on knowledge discovery in databases*. AAAI Press.
[Pelleg & Moore, 2000] PELLEG, DAN, & MOORE, ANDREW. 2000. X-means: Extending k-means with efficient estimation of the number of clusters. *Pages 727–734 of: Proceedings of the seventeenth international conference on machine learning*. San Francisco: Morgan Kaufmann.
[Rasmussen, 2000] RASMUSSEN, CARL. 2000. The infinite gaussian mixture model. *Advances in neural information processing systems (nips)*, **12**, 554–560.
[Roberts *et al.* , 2001] ROBERTS, STEPHEN J., HOLMES, CHRISTOPHER, & DENISON, DAVE. 2001. Minimum-entropy data clustering using reversible jump markov chain monte carlo. *Pages 103–110 of: Icann '01: Proceedings of the international conference on artificial neural networks*. London, UK: Springer-Verlag.
[Sahami, 1998] SAHAMI, MEHRAN. 1998. *Using machine learning to improve information access*. Ph.D. thesis, Computer Science Department, Stanford University.
[Scholkopf *et al.* , 1998] SCHOLKOPF, BERNHARD, SMOLA, ALEXANDER, & MULLER, KLAUS-ROBERT. 1998. Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, **10**(5), 1299–1319.
[Shamir & Tishby, 2008] SHAMIR, OHAD, & TISHBY, NAFTALI. 2008. Cluster stability for finite samples. *Pages 1297–1304 of:* PLATT, J.C., KOLLER, D., SINGER, Y., & ROWEIS, S. (eds), *Advances in neural information processing systems 20*. Cambridge, MA: MIT Press.
[Shi & Malik, 2000] SHI, JIANBO, & MALIK, JITENDRA. 2000. Normalized cuts and image segmentation. *Ieee transactions on pattern analysis and machine intelligence*, **22**, 888–905.
[Sindhwani *et al.* , 2008] SINDHWANI, V., HU, J., & MOJSILOVIC, A. 2008. Regularized co-clustering with dual supervision. *In: Nips*.
[Slonim & Tishby, 2000] SLONIM, NOAM, & TISHBY, NAFTALI. 2000. Document clustering using word clusters via the information bottleneck method. *Pages 208–215 of: In acm sigir 2000*. ACM press.
[Smith & Jain, 1984] SMITH, STEPHEN P., & JAIN, ANIL K. 1984. Testing for uniformity in multidimensional data. *Ieee transactions on pattern analysis and machine intelligence*, **6**(1), 73–81.
[Sokal & Sneath, 1963] SOKAL, ROBERT R., & SNEATH, PETER H. A. 1963. *Principles of numerical taxonomy*. W. H. Freeman, San Francisco.
[Steinbach *et al.* , 2000] STEINBACH, M., KARYPIS, G., & KUMAR, V. 2000. A comparison of document clustering techniques. *In: Workshop on kdd*.
[Steinhaus, 1956] STEINHAUS, H. 1956. Sur la division des corp materiels en parties. *Bulletin of acad. polon. sci.*, **IV**(C1. III), 801–804.
[Strehl & Ghosh, 2003] STREHL, ALEXANDER, & GHOSH, JOYDEEP. 2003. Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research*, **3**, 583–617.

[Sungur, 2008] SUNGUR, E. 2008. *Overview of multivariate statistical data analysis*. Available online at : http://www.mrs.umn.edu/ sungurea/multivariatestatistics/overview.html.

[Tan *et al.* , 2005] TAN, PANG-NING, STEINBACH, MICHAEL, & KUMAR, VIPIN. 2005. *Introduction to data mining*. First edn. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.

[Tibshirani *et al.* , 2001] TIBSHIRANI, R., WALTHER, G., , & HASTIE, T. 2001. Estimating the number of clusters in a data set via the gap statistic. *Journal of the royal statistical society*, **B**, 411–423.

[Tishby *et al.* , 1999] TISHBY, NAFTALI, PEREIRA, FERNANDO C., & BIALEK, WILLIAM. 1999. The information bottleneck method. *Pages 368–377 of: Proceedings of the 37th allerton conference on communication, control and computing*.

[Tsuda & Kudo, 2006] TSUDA, KOJI, & KUDO, TAKU. 2006. Clustering graphs by weighted substructure mining. *Pages 953–960 of: Icml '06: Proceedings of the 23rd international conference on machine learning*. New York, NY, USA: ACM.

[Tukey, 1977] TUKEY, JOHN WILDER. 1977. *Exploratory data analysis*. Addison-Wesley.

[Umeyama, 1988] UMEYAMA, S. 1988. An eigendecomposition approach to weighted graph matching problems. *Ieee transactions on pattern analysis and machine intelligence*, **10**(5), 695–703.

[von Luxburg & David, 2005] VON LUXBURG, U., & DAVID, BEN S. 2005. Towards a statistical theory of clustering. *In: Pascal workshop on statistics and optimization of clustering*.

[Wasserman & Faust, 1994] WASSERMAN, S., & FAUST, K. 1994. *Social network analysis: methods and applications*. Cambridge University Press.

[White & Smyth, 2005] WHITE, SCOTT, & SMYTH, PADHRAIC. 2005. A spectral clustering approach to finding communities in graph. *In: Proceedings of the siam data mining*.

[Yu & Shi, 2003] YU, STELLA X., & SHI, JIANBO. 2003. Multiclass spectral clustering. *Pages 313–319 of: Iccv*.

[Zhang *et al.* , 1996] ZHANG, TIAN, RAMAKRISHNAN, RAGHU, & LIVNY, MIRON. 1996. Birch: an efficient data clustering method for very large databases. *Sigmod rec.*, **25**(2), 103–114.