A Survey of Data Clustering for Cancer Subtyping

Yan Yan*, Frederick C Harris, Jr.*
University of Nevada, Reno,
Reno, Nevada, USA. *

Abstract

Cancer subtyping remains a challenging task in microarray data analysis. The major goals of a successful cancer subtyping system are accuracy and reliability. Cluster analysis techniques have proven to be effective in this area. To facilitate further development in cancer subtyping based on microarray data, we provide a comprehensive review of the major cluster analysis algorithms from the clinical and computational domains that have been applied on microarray mRNA expression data and miRNA expression data for cancer subtyping, as well as other clustering algorithms with potential application in cancer subtyping.

Key Words: Algorithms, cancer subtype detection, data clustering, microarrays

1 Introduction

Clustering is an interdisciplinary research topic and is also known by researchers in different fields as unsupervised learning, exploratory data analysis, grouping, clumping, taxonomy, typology, and Q-analysis [138]. Cluster analysis is defined as 'a statistical classification technique for discovering whether the individuals of a population fall into different groups by making quantitative comparisons of multiple characteristics' and its first known use was in 1948 (Merriam-Webster Online Dictionary, 2013). The clustering algorithm was first developed by biologists in numerical taxonomy study in 1963 before being utilized by statisticians [134]. Clustering is used for class discovery, *i.e.* exploration or discovery of the underlying patterns of a dataset by separating the dataset into groups, with little or no prior knowledge [86, 136, 254, 255]. Clustering is also used for natural classification, *i.e.* identifying the degree of similarity among organisms, and compression, *i.e.* organizing and summarizing data using cluster prototypes [138]. Clustering has become increasingly popular as society increasingly generates an overwhelming amount of data, and it is often used as the first step in data analysis or as a preparation step for experimental work [163, 256].

There is no universally agreed upon definition of clusters [86]. A cluster is a set of objects that are compact (or similar to each other) and isolated (or dissimilar) from other clusters. In reality, cluster definition is subjective, and its significance and interpretation requires related domain knowledge [138]. Similarity measure is used by clustering methods to calculate the similarity between two objects. Different similarity measures will have different clustering results, as some objects may be similar to one another using one measure but dissimilar using another. Similarity between two objects can be measured in different ways, and the three dominant methods are distance measures, correlation measures, and association measures [134]. Common similarity measures include Euclidean distance, Manhattan distance, Maximum norm, Mahalanobis distance, Pearson coefficient, Spearman's rank correlation coefficient, angle between two vectors, and the Hamming distance.

Since the process of clustering is subjective, judging the relative efficacy of clustering methods is difficult [20, 139]. Cluster validity is used to assess clustering results and can be classified into three categories: a) Internal validities formulate quality as a function of the given data set [130]. Examples include Dunn's Validity Index, Silhouette Value, Hubert Gamma Statistic, Entropy, Xie-Beni, Normalized Mutual Information. b) External validities assess quality by additional external information such as category labels [130]. Examples include Jaccard Index, Rand Index, Adjusted Rand Index, Variation of Information, Kappa Statistic, CA. c) Relative validities evaluate a clustering result by comparing it to results from other clustering methods.

The procedure of cluster analysis includes four steps [254]: Step one is feature selection or extraction. Feature selection selects a subset of all features, and feature extraction generates novel features from the original ones by using some transformations [31, 135, 139, 254]. Step two is clustering algorithm design or selection. Since clustering algorithms group objects based on some proximity measure, this step usually includes choosing an appropriate proximity measure and construction of a clustering criterion function, creating an optimization problem that has been well studied in the literature. Step three is cluster validation. This step calculates a confidence level for the clustering results. Step four is results interpretation. This step provides meaningful insights from the data.

There is no single clustering algorithm that performs best across all problems or data sets [152, 254]. Therefore, it is important to study the characteristics of the problem and use an appropriate clustering strategy [254].

Properties to be considered in choosing a clustering algorithm include [28]: a) feature type (numeric and non-numeric), b) scalability (large datasets), c) handling high dimensional data, d) finding clusters of irregular shape, e) handling outliers, f) time complexity of the algorithm, g) data order dependency, h) assignment type (hard or strict vs. soft or fuzzy), i) prior

*Department of Computer Science and Engineering. Email: fred.harris@cse.unr.edu

knowledge and user defined parameters dependency, and j) interpretability and visulization of results.

Despite many examples of successful applications of cluster analysis, there still remain many challenges due to the existence of many inherent uncertain factors [254]. The following fundamental challenges in clustering [136, 138] are relevant even today [138]: a) definition of a cluster, b) selection of features, c) normalization of the data, d) outlier detection, e) definition of pair-wise similarity, f) number of clusters, g) selection of clustering method, h) existence of clustering tendency, and i) validity of the clusters.

Some recent trends in clustering include [138]: semi-supervised clustering utilizing external or side information; interactive clustering, where a user can specify or change program parameters based on domain knowledge or results from previous clustering iterations; clustering ensembles, where the partitions resulting from different algorithms (or the same algorithm with different parameters) are combined; multi-objective clustering, where the clustering algorithm optimizes multiple specific objectives; large-scale clustering, which handles very large databases; multi-way clustering, which extends the bi-clustering framework and simultaneously clusters heterogeneous components of the data objects [26]; and heterogeneous data clustering for data comprising multiple types, such as rank data, dynamic data, graph data, and relational data [134].

Clustering techniques can be organized into categories. Different criteria may result in different categories of clustering algorithms [254]. Furthermore, categorization of clustering algorithms is not straightforward or canonical, and categories can overlap [28]. For convenience, in this review we use the following taxonomy, which is also widely used in the literature: hierarchical clustering (Section 2), partitioning clustering (Section 3), graph-based clustering (Section 4), distribution-based clustering (Section 5), density-based clustering (Section 6), grid-based clustering (Section 7), clustering big data (Section 8), clustering high dimensional data (Section 9), and other clustering techniques (Section 10).

2 Hierarchical Clustering

Hierarchical clustering algorithms organize a data set into a hierarchical structure according to a similarity measure [254]. It is based on the belief that nearby objects are more related than objects that are farther away [183]. These algorithms connect objects based on their similarity to form clusters, which is usually represented using a dendrogram. Hierarchical clustering algorithms differ in the choice of similarity measures, the linkage criterion (distance between clusters), and whether the process is agglomerative (bottom-up) or divisive (top-down). Agglomerative hierarchical clustering starts with singleton clusters and then recursively merges appropriate clusters, and divisive hierarchical clustering starts with one cluster containing all objects and recursively splits appropriate clusters [28].

Divisive clustering is very expensive in computation [86] and is not commonly used in practice [254]. We focus on the agglomerative clustering first and then mention two divisive clustering algorithms named MONA and DIANA [146, 254].

There are many agglomerative hierarchical clustering algorithms based on different linkage criterion. The single linkage method or nearest neighbor method [110, 136, 215, 220, 221, 254] uses the distance between two closest objects in different clusters, and the shortest distance determines the merge of two clusters. The complete linkage method or farthest neighbor method [67, 149, 223, 254] uses the distance between two farthest objects in different clusters, and the shortest distance determines the merge of two clusters. These two methods are the simplest and most popular [254]. Average linkage methods include UPGMA (Unweighted Pair-Group Method using Arithmetic averages), WPGMA (Weighted Pair-Group Method using Arithmetic averages), UPGMC (Unweighted Pair Group Method using Centroids), and WPGMC (Weighted Pair Group Method using Centroids). UPGMA and UPGMC use a simple average, while WPGMA and WPGMC use a weighted average where the weight is the inverse of cluster size. UPGMA [63, 87, 136, 220, 222] uses average distance between two objects in different clusters, and the shortest average distance determines the merge of two clusters. WPGMA or weighted average linkage method [182] uses weighted average distance between two objects in different clusters, and the shortest average distance determines the merge of two clusters. UPGMC or centroid linkage method [220] uses Euclidean distance between unweighted centroids (calculated by arithmetic mean) of different clusters, and the shortest distance determines the merge of two clusters. WPGMC or median linkage method [220] uses Euclidean distance between weighted centroids of different clusters, and the shortest distance determines the merge of two clusters. Minimum-variance method or Ward's method [245] considers the relationship of all objects in a cluster. Its objective is to form clusters such that the increase of variance within each group is minimized [247]. Further readings about these methods include [86, 254, 259].

What follows are examples of divisive hierarchical clustering algorithms. DIANA [146] (DIvisive ANAlysis Clustering) selects in each dividing step the cluster with the largest diameter and divides it into two new clusters. MONA [146] (MONothetic Analysis Clustering of Binary Variables) divides clusters based on a single well-chosen variable (or feature), whereas most other hierarchical methods use all variables (or features).

Advantages of hierarchical clustering are a) Good visualization with dendrogram representation [136, 231, 254, 256], b) Very informative descriptions with dendrogram representation [136, 231, 254, 256], and c) Flexibility regarding the number of clusters, since the clustering results can be obtained by cutting the dendrogram at different levels.

Disadvantages of hierarchical clustering are [254, 256]: a) Lacking of robustness and sensitivity to noise and outliers. b) High computational complexity, which limit their application on large scale data. c) Tendency to form clusters with

spherical shapes instead of natural shapes. d) Prone to reversal phenomenon [189].

BIRCH [269] (Balanced Iterative Reducing and Clustering using Hierarchies) clusters incoming data objects incrementally and dynamically. It first builds a CF (Clustering Feature) tree dynamically as new data objects are inserted and then applies an agglomerative hierarchical clustering algorithm to the nodes represented by their CF vectors. After obtaining a centroid for each cluster, it assigns each data object to its nearest centroid. CURE [112] (Clustering Using REpresentatives) uses a number of representative data points in a cluster to evaluate the distance between clusters. Closest cluster pair are merged at each step of its hierarchical clustering process. ROCK [113] (RObust Clustering using linKs) uses links and not distances when merging clusters for boolean and categorical data. DISMEA [224] uses the k-means algorithm to divide a cluster into two clusters. The Edwards and Cavalli-Sforza Method [79] divides all available clusters at each step. Minimum Spanning Tree-based clustering algorithms [80, 190, 266] construct an MST (Minimum Spanning Tree) [156, 185, 200] from a data set and produce a group of clusters by removing selected edges. Figure 1 shows an example of hierarchical clustering.

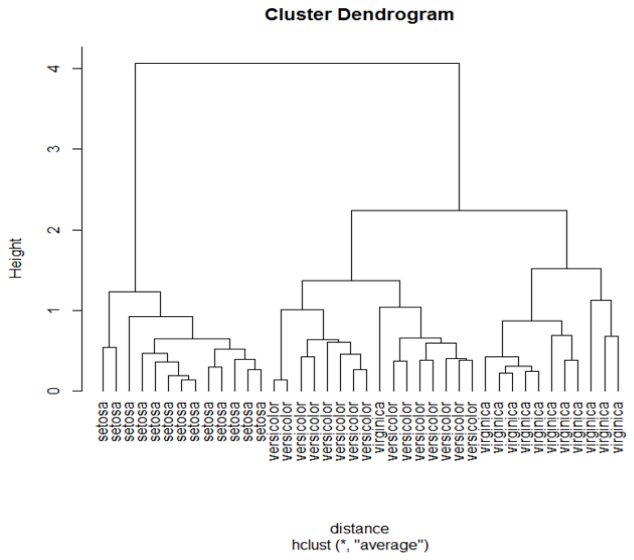

Figure 1: Hierarchical clustering [124]

3 Partitioning Clustering

Partitioning clustering algorithms divide objects into clusters without hierarchical structure. Clusters are represented by a central vector. Given the number of clusters, partitioning clustering assigns the objects to the closest cluster center. Partitioning algorithms can be grouped into k-means methods and k-medoids methods. k-means methods use the centroid of objects within a cluster as center. k-medoids methods use the most appropriate object within a cluster as center.

K-means clustering [28, 93, 120, 121, 167, 226, 254, 256] is very simple, but one of the best known and popular clustering algorithms. There are many variations of the basic k-means clustering. Classic k-means reassigns data objects based on optimization of the objective function. If a reassigning has a positive effect, the data object is reassigned and the cluster centers are updated. ISODATA [19] (Iterative Self-Organizing Data Analysis Technique) splits and merges intermediate clusters based on a user-defined threshold and iterates until the threshold is reached. FORGY [93] reassigns objects to nearest centroids and recomputes centroids. It iterates until a stopping criterion is achieved. Fuzzy c-means [29, 77] assigns fuzzy cluster membership to each data object, and updates cluster centers and membership after each iteration. Methods to speed up k-means and fuzzy C-means such as brFCM (bit reduction by Fuzzy C-Means) [83] replace similar data objects with their centroid before clustering.

Variations of k-medoid [146] methods are as follows. PAM (Partitioning Around Medoids) assigns each data object to the closest medoid and iteratively reassigns objects and updates medoids to optimize the objective function. CLARA (Clustering LARge Applications) [146] applies PAM on multiple subsets or samples of the data set, and selects the best clustering as output. CLARANS (Clustering Large Applications based upon RANdomized Search) [187] searches a graph where each node is a set of medoids. It selects a node randomly in search for a local minimum among its neighbor nodes through iterations and outputs the best node to form clustering results.

Advantages of partitioning clustering are a) simple, straightforward and easy implementation, b) fast execution with computation complexity of $O(n)$, c) very suitable for compact and hyperspherical clusters, d) computational rigor (firm foundation of analysis of variances).

Disadvantages of partitioning clustering are a) they are still subjective processes that are sensitive to assumptions, b) they require the number of clusters to be specified in advance, c) they prefer clusters of approximately similar size, as they will always assign an object to the nearest center, often leading to incorrectly cut borders in between clusters, d) they are subject to easy trapping in local minima and sensitivity to the initial partition (hill-climbing optimization method).

Other developments are as follows. Bisecting k-means [225] recursively partitions a cluster into two. KD-trees k-means [195] uses the KD-Tree data structure to speed up the assignment of data objects to their closest cluster by reducing the number of nearest-neighbor queries in the traditional algorithm. Scaling k-means [37] retains important data objects and summarizes or discards other objects. Centroids of the resulting data set are then used on the whole data set. X-means [196] finds the number of clusters K automatically by optimizing a criterion function such as AIC (Akaike Information Criterion) or BIC (Bayesian Information Criterion). Kernel k-means [209] enhances k-means by using a kernel function that nonlinearly maps the original feature space to

a higher dimensional one, where clusters are more separable. Weighted kernel k-means [71] further extends kernel k-means by assigning a weight for each cluster. The weight is defined as the reciprocal of the number of data objects in the cluster. GA k-means [16] applies a genetic algorithm to improve cluster centers initialization for k-means. Simulated annealing [7, 132, 151, 211] uses simulated annealing optimization to avoid local optima and find the global minimum solution. Soft assignment [267] assigns data objects to different clusters with appropriate weights to improve the optimization process. It uses Harmonic Averages of the distances from the data object to all the centers. Mahalanobis distance [170] is used to detect clusters with hyperellipsoidal shapes. Maximum of intra-cluster variances [109] can be used as the objective function instead of the sum to obtain good clustering results. K-prototypes [131] incorporates categorical data as a generalization approach. Accelerated k-means by triangle inequality [81] avoids unnecessary distance calculations by using the triangle inequality and keeping track of lower and upper bounds for distances between data objects and cluster centers. K-means++ [12] improves the speed and the accuracy of k-means by using a simple randomized seeding technique. Figure 2 shows an example of partitioning clustering.

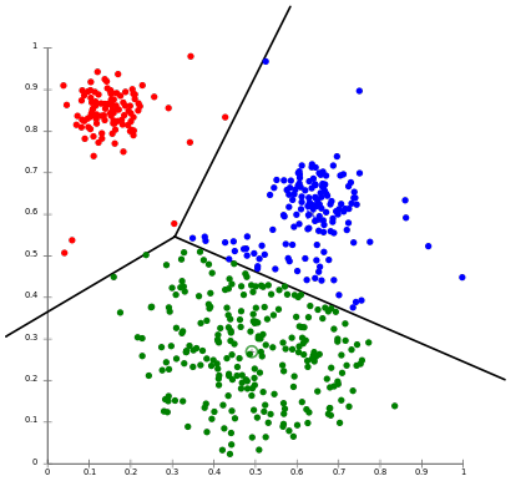

Figure 2: Partitioning clustering [57]

4 Graph-based Clustering

Graph-based clustering algorithms construct a graph/hypergraph from the data and then partition the graph/hypergraph into subgraphs/subhypergraphs or clusters. Each vertex represents a data object, and the edge weight represents the similarity of two vertices [50]. The edges in the same subgraph/subhypergraph should have high weights, and the edges between different subgraphs/subhypergraph should have low weights [50]. It is also called spectral clustering [138].

Representative algorithms are as follows. Chameleon [143] uses a connectivity graph and graph partitioning to build small clusters, followed by the agglomerative hierarchical clustering process. Its key feature is that it considers both interconnectivity and closeness when merging clusters. CACTUS (Clustering Categorical Data Using Summaries) [100] detects candidate clusters based on the summary of the data set and determines the actual clusters through a validation process against the candidate clusters. It uses a similarity graph to represent the inter-attribute and intra-attribute summaries [98]. A Dynamic System-based Approach or STIRR (Sieving Through Iterated Relational Reinforcement) [106] represents each attribute value as a weighted vertex in a graph. It iteratively assigns and propagates weights until a fixed point is reached. Different weight groups correspond to different clusters on the attribute. ROCK (Robust Clustering algorithm for Categorical Data) [113] repeatedly merges two clusters until the specified number of clusters is reached, and it uses data sampling to improve complexity. It uses a connectivity graph to calculate the similarities between data objects [98].

The advantages of graph-based clustering are [50]: a) A graph is an elegant data structure that can model many real applications. b) It is based on solid mathematical foundations, including spectral theory and Markov stochastic process. c) It produces optimal clustering (optimizing a quality measure instead of acting greedily toward the final clustering).

The major disadvantage of graph-based clustering is that it may be slow when working on large scale graphs [50].

Other developments are as follows. The Ratio Cut algorithm [117] adopts a cluster size constraint, which is the number of data points in a cluster. The Normalized Cut (NCut) algorithm [214] is an approximate graph-cut based clustering algorithm with a cluster size constraint, which is the volume of the cluster or sum of edge weights within a cluster. It also has a multiclass version [264]. The MNCut (Modified Normalized Cut) algorithm [174] gives a new interpretation to the NCut algorithm in the framework of a Markov Random Walk. Ng's method [186] derives a new data representation from normalized eigenvectors of a kernel matrix simultaneously and in a particular manner. Laplacian Eigenmap [27] uses the eigenvectors of the graph Laplacian to represent data. Pairwise Data Clustering by Deterministic Annealing [126] uses proximity measures between the data objects to represent data. Dominant Sets Pairwise Clustering [191] relates clusters to maximal dominant sets [180] in pair-wise clustering. Fast approximate spectral clustering [260] applies a distortion-minimizing local transformation to the data to speed up conventional spectral clustering. Active spectral clustering [243] follows the concept of constrained clustering and uses pairwise relations. Its constraints are specified in a incremental manner. Locally-scaled spectral clustering using empty region graphs [60] employs β-skeleton (a subset of empty region graphs) and non-linear diffusion to define a locally adapted affinity matrix which

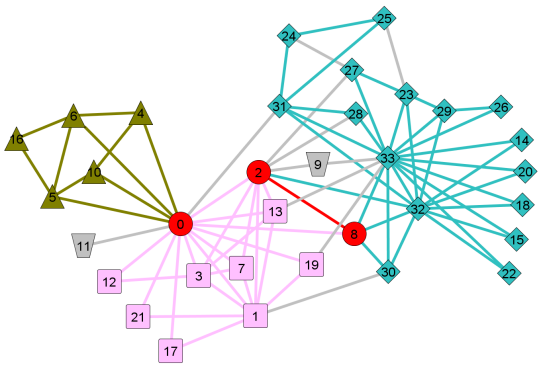

Figure 3: Graph-based clustering [85]

defines the similarity of two data objects. Figure 3 shows an example of graph-based clustering.

5 Distribution-based Clustering

Distribution-based clustering views or assumes that the data are generated by a mixture of probability distributions, each of which represents a different cluster [99, 172]. This way, a cluster can be seen as objects generated by the same distribution. Thus, a particular clustering method can be expected to produce good results when the data conform to the method's distribution model [99]. It is also called model-based clustering. There are usually two approaches to form the model: the classification likelihood approach and the mixture likelihood approach [99].

Distribution-based clustering has a long history. Early works include [30, 65, 210, 249]. A survey of cluster analysis in a probabilistic and inferential framework is presented in [33].

Representative algorithms are as follows. The EM (Expectation-Maximization) clustering algorithm [69] is the most popular method in distribution-based clustering. It tries to fit the data set into the assumed number of Gaussian distributions by moving the means of Gaussian distributions toward the cluster centers. COOLCAT (reducing the entropy, or COOLing of the CATegorical data clusters)[22] uses entropy to cluster categorical data. It consists of data sampling and incremental assignment. STUCCO (Search and Testing for Understandable Consistent Contrasts) [25] uses tree searching and significant contrast-sets to find clusters. GMDD (Gaussian Mixture Density Decomposition) [271] uses a recursive approach and identifies each Gaussian component in the mixture successively. Autoclass [49] is based on the classic distribution-based approach and uses a Bayesian method to determine the optimal clusters. P-AutoClass [198] is a parallel version of Autoclass and can be used on large data sets.

The advantages of distribution-based clustering are as follows [28]: a) It can be modified to handle complex data, b) It has a solid theoretical foundation, c) Its results are easily interpretable, d) It not only provides clusters, but also produce complex models that capture relationships among attributes, e) Results are independent of the timing of consecutive batches of data, f) It is good for online learning since the intermediate mixture model can be used to cluster objects, g) the Mixture model can be naturally generalized to cluster heterogeneous data.

The disadvantage of distribution-based clustering is the difficulty in choosing the appropriate model complexity (since a more complex model will usually be able to explain the data better but may cause an overfitting problem from excessive parameter set).

Other developments are as follows. Latent Dirichlet Allocation (LDA) [32] uses a hierarchical Bayesian model that has three levels. Each data object is modeled as a finite mixture over an underlying set of groups (or clusters) of objects. Each group (or cluster) is modeled as an infinite mixture over a set of group (or cluster) probabilities. Pachinko Allocation Model (PAM) [161] uses a Directed Acyclic Graph (DAG) to model cluster correlations. The leaves of the DAG represent data objects, and the interior nodes represent correlations. Undirected graphical model for data clustering [246] is based on exponential family distributions and the semantics of undirected graphical models. It uses the technique of minimizing contrastive divergence to speed up the process. Robust cluster analysis via mixture models method [173] uses the mixtures of multivariate t distributions approach to the clustering. It also uses the t distribution to cluster high-dimensional data via mixtures of factor analyzers. Online learning for LDA method [125] is an online Variational Bayes (VB) algorithm for LDA. It uses natural gradient step in online stochastic optimization, which converges to a local optimum of the VB objective function. Figure 4 shows an example of distribution-based clustering.

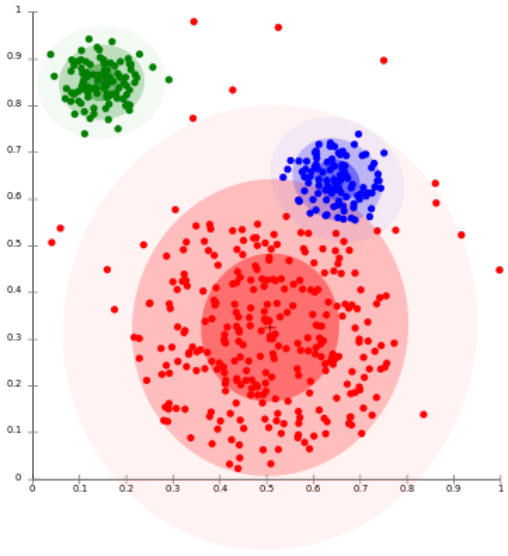

Figure 4: Distribution-based clustering [56]

6 Density-based Clustering

Density-based clustering defines clusters as dense regions of data objects separated by low-density regions. A cluster is a connected dense component and grows in any direction that density leads [99]. Objects in low-density areas which separate clusters are usually considered to be noise and border points. There are two major approaches for density-based clustering [28]: the connectivity approach pins density to a training data point; the density function approach pins density to a point in the attribute space.

Representative algorithms for the connectivity approach are as follows. DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [84] starts by selecting a data object and tries to find all data objects density-reachable from it to form a cluster. If none are found, the algorithm selects a new data point and repeats. GDBSCAN (Generalized DBSCAN) [205] generalizes the concept of neighborhood by permitting the use of any distance function besides Euclidian distance and allows other measures besides simply counting the objects to define the cardinality of that neighborhood. OPTICS (Ordering Points To Identify the Clustering Structure) [11] is like an extended DBSCAN algorithm. It does not assign cluster memberships but stores the order in which the data objects are processed as well as the core-distance and a reachability-distance for each data object. An extended DBSCAN is used to assign cluster memberships. DBCLASD (Distribution Based Clustering of LArge Spatial Databases) [257] uses the notion of clusters based on the distance distribution and incrementally augments an initial cluster by its neighboring points as long as the nearest neighbor distance set of the resulting cluster still fits the expected distance distribution.

Representative algorithms for the density function approach are as follows. DENCLUE (DENsity-based CLUstEring) [122] calculates the impact of each data object within its neighborhood (*i.e.* influence function) and determines clusters mathematically by identifying local maxima of the overall density function (*i.e.* density-attractors).

The advantages of density-based clustering are as follows [28, 99]: a) They can find clusters of arbitrary shapes, in contrast to many other methods. b) Time complexity is low (linear or $O(n)$). c) It is deterministic for core and noise points (but not for border points), therefore there is no need to run it multiple times. d) It can handle noise well. e) The number of clusters is not required, since it finds clusters and the number of clusters automatically. f) Results are independent of data ordering. g) There are no limitations on the dimension or attribute types.

The disadvantages of density-based clustering are as follows: a) It is often difficult to detect cluster boarders when the cluster density decreases continuously (*i.e.* arbitrary borders). b) For a mixtures of Gaussians data set, distribution-based clustering (*e.g.* EM) usually outperforms density-based clustering. c) Limitations in processing high-dimensional data, since it is difficult to distinguish high-density regions from low-density regions when the data is high-dimensional [138]. d) Most density-based clustering algorithms were developed for spatial data [99].

Other developments are as follows. BRIDGE [64] integrates the k-means algorithm and the DBSCAN algorithm. K-means is first performed, and then DBSCAN is used on each partition. Finally, results are improved by removing the noise found by DBSCAN. Jarvis-Patrick algorithm [94] partitions the data set into clusters based on the number of shared nearest neighbors. It first identifies the k nearest neighbors of each data object and then merges two data objects at a time. C-DBSCAN (Constrained-DBSCAN) [204] enhances the DBSCAN algorithm with pairwise constraints. SCAN (Structural Clustering Algorithm for Networks) [258] can detect hubs and outliers, in addition to clusters in networks (or graphs). It uses a structural similarity measure to cluster vertices. Figure 5 shows an example of density-based clustering.

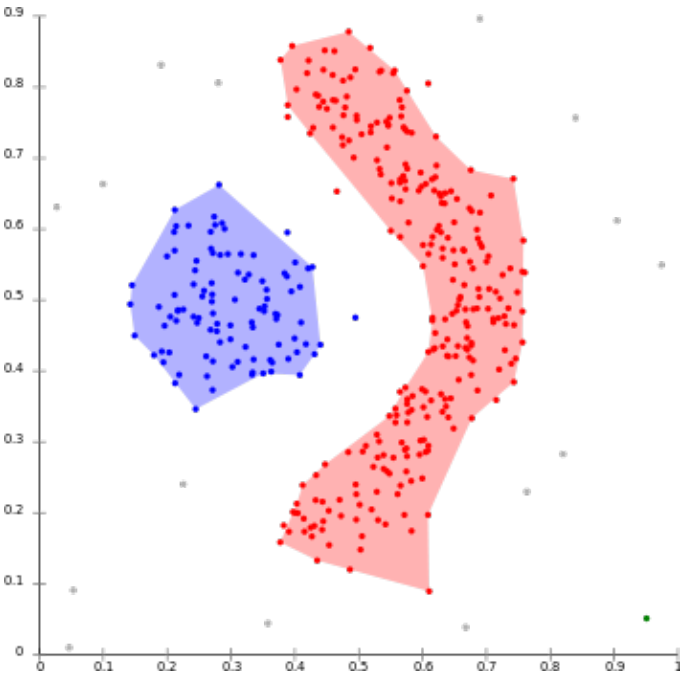

Figure 5: Density-based clustering [55]

7 Grid-based Clustering

Grid-based clustering operates on space partitioning instead of data partitioning to produce clusters [28]. It first creates the grid structure by partitioning the data space into cells (or cubes) and then clusters the cells based on their densities.

Representative algorithms are as follows. BANG-clustering [28, 207] uses a multi-dimensional grid data structure to organize or partition the data. It uses the cell information in the grid and clusters the cells. STING (A STatistical INformation Grid approach) [241] uses a hierarchical structure of grid cells with a top-down approach. It labels a cell to be relevant or not at a specified confidence level. Then, it finds all the regions formed by relevant cells. STING+ [28,

242] uses a similar hierarchical cell structure as STING and introduces an active spatial data mining approach. OptiGrid (Optimal Grid) [123] constructs an optimal grid partitioning of the data by finding the best partitioning hyperplanes for each dimension with projections of the data. GRIDCLUS (GRID-CLUStering) [206] organizes the space surrounding the clusters with a grid data structure. It uses a topological neighbor search to cluster the grid cells. GDILC (Grid-based Density-IsoLine Clustering) [261] is based on the idea that the density-isoline figure reflects the distribution of data. It uses a grid-based approach to calculate the density and finds dense regions. WaveCluster (Wavelet-based clustering) [212] transforms the original feature space by applying wavelet transform and then finds the dense regions in the new space. It yields sets of clusters at different resolutions and scales, which can be chosen based on the user's needs. FC (Fractal Clustering) [21] adds one data object at a time to one cluster in such a way that the fractal dimension changes the least after adding the data object.

The advantages of grid-based clustering are as follows [28, 99]: a) It is fast and works well with large data sets (since speed is independent of the number of objects in the data), b) It handles noise well, c) It is independent of data ordering, d) It can handle attributes of different types, e) It can be used as an intermediate step in many other algorithms such as CLIQUE and MAFIA.

The disadvantages of grid-based clustering are as follows: a) Most algorithms need the user to specify grid size or density thresholds, which can be difficult (fine grid sizes result in high computational time, while coarse grid sizes result in low quality of clusters) [99]. b) Some grid-based clustering algorithms (*e.g.* STING, WaveCluster) are not good at high dimensional data [99].

Other developments are as follows. AMR (Adaptive Mesh Refinement clustering) [162] creates grids at multiple resolutions where higher resolution grids are applied to the localized denser regions. O-Cluster (Orthogonal partitioning CLUSTERing) [175] is a variant of OptiGrid. It creates a hierarchical grid-based structure by making axis-parallel (orthogonal) partitions on the input data. It operates recursively, and the final irregular grid frames the data into clusters. CBF (Cell-Based Filtering) [47] splits each dimension into a set of partitions using a filtering-based index. It then creates cells based on the overlapping regions of the partitions. PGMCLU (Parallel Grid-based CLUstering algorithm for Multi-density datasets) [251] consists of parallel data partitioning, local clustering, and merging local clusters. It introduces a new measure called grid compactness for the degree of tightness between data objects within the grid, and the notion of grid feature for summarizing the information about a grid. Figure 6 shows an example of grid-based clustering.

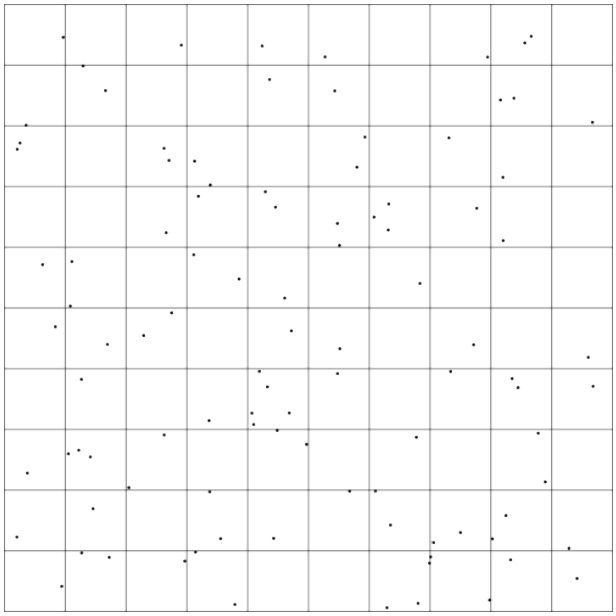

Figure 6: Grid-based clustering [263]

8 Clustering Big Data

Big data clustering refers to clustering on millions of data objects [138]. These algorithms need to have good scalability and process big data within reasonable computing time and memory space [28]. A high computational complexity would dramatically limit an algorithm's application to big data. The strategies used for big data clustering can be categorized into sampling, data summarization, distributed computing, and incremental learning.

8.1 Sampling

Sampling methods select a sample of the original large data set and perform clustering over the sample data. Old-fashioned sampling methods may or may not use rigorous statistical reasoning. Newer sampling methods use special uniform checks to control their adequacy [28]. Advantages are that it is simple to implement and can screen out most outliers. However, small clusters may be missed.

Examples are as follows. CURE (Clustering using REpresentatives) [112] and ROCK (RObust Clustering using linKs) [113] were covered in Section 2. CLARA (Clustering LARge Applications) [145] draws several samples from the data set, runs PAM on each of them, and selects the best result. CLARANS (Clustering Large Applications based on RANdomized Search) [187] starts with a new randomly-selected node (a set of k potential medoids) in the graph in search of the local optimum. It repeats if a local optimum is found.

8.2 Data Summarization

Data summarization methods calculate data summary statistics and perform clustering on the summaries instead of the original data. The advantage is that the requirement for the storage of and frequent operations on the large amount of data are greatly reduced, saving both computational time and storage

space. The disadvantage is reduced cluster quality.

Examples are as follows. BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) was covered in Section 2. BUBBLE [101] instantiates generalized BIRCH for data in a distance space. BUBBLE-FM (BUBBLE-FastMap) [101] improves upon BUBBLE by reducing the computation time using FastMap [88]. EMADS (EM Algorithm for Data Summaries) [141] directly generates a Gaussian mixture model from simplified data summaries. bEMADS (BIRCH's EMADS) [141] uses data summarization procedures in the BIRCH algorithm.

8.3 Distributed Computing

Distributed computing methods divide a large data set into smaller data sets and perform clustering on each smaller data set. The advantage is that clusterings on each smaller data set can be done in parallel to reduce the overall computation time [138]. The disadvantage is the overhead and complexities due to the dividing and combining steps.

Examples are as follows. Parallel k-means [70] is a parallel implementation of the k-means clustering algorithm. DBDC (Density Based Distributed Clustering) [140] clusters distributed data locally and extracts suitable representatives from these local clusters to send to a global site where the complete clusters are restored based on the local representatives. It uses a density-based clustering algorithm for both local and global clustering. Parallel spectral clustering in distributed systems [51] makes the dense similarity matrix sparse by retaining nearest neighbors using a parallel approach.

8.4 Incremental Learning

Incremental learning methods process one data object at a time and may discard it. They require only one single pass over all data objects, in contrast to most clustering methods that require multiple passes over data objects before identifying the cluster centers [138]. Advantages are: improved clustering efficiency in terms of data storage and processing time (they can admit new data objects without learning from scratch [256]); handling outliers well [28]; resumable processing which makes it very suitable for dynamic big data sets [28]. Disadvantages are that results depend on data order and may not be stable [43, 178, 256], and can result in lower quality clusters [28].

Examples are as follows. DIGNET [232, 244] moves cluster centers toward a new data point with each new addition. Hartigan's leader algorithm [120] uses a distance/similarity threshold to decide if a data point should be added to the cluster or used for a new cluster center. ART (Adaptive Resonance Theory) family [42, 256] simulates neural circuits that are believed to trigger fast learning. It includes a large family of neural network variants such as ART1 [43], ART2 [42], Gaussian ART [248], Bayesian ART [236], Ellipsoid ART [9], ART tree [45, 250], ARTMAP [44], Q-learning ART [38], Fuzzy ART [41]. Charikar's incremental clustering [48] maintains a clustering of the data objects so that the maximum cluster diameter is minimized as new data objects are added. GenIC (Generalized Incremental algorithm for Clustering) [114] divides the data stream into chunks or windows, updating each cluster center with each new data object addition and merging clusters at the end of a window of data. Cobweb [91] is an incremental system for hierarchical clustering, which enables bi-directional hill-climbing search through the space of hierarchical schemes.

9 Clustering High Dimensional Data

High Dimensional Data clustering refers to clustering on data objects that represent from a few dozen to thousands or more features. Such high dimensional data are often seen in areas such as medicine (*e.g.* microarray experiments), and text documents (*e.g.* word-frequency vector methods [46]). Clustering high dimensional data is tremendously difficult. One problem is that increased irrelevant features eliminate the likelihood of clustering tendency [28]. Another problem is the 'curse of dimensionality', or lack of data separation, in high dimensional space (the problem becomes severe for dimensions greater than 15) [28]. Performing feature selection before applying clustering can improve the first problem. Principal Component Analysis (PCA) [193] is commonly used. However, the dimension may still be high after feature selection. In this review, we discuss techniques that have been developed to address such situations: projected clustering, subspace clustering, bi-clustering (or co-clustering), tri-clustering, hybrid approaches, and correlation clustering.

9.1 Projected Clustering

Projection techniques map data objects from a high dimensional space to a low dimensional space, while maintaining some of the original data's characteristics [13].

Examples are as follows. PreDeCon [34] finds subsets of feature vectors that have low variance along subsets of attributes. PROCLUS [3] finds the candidate clusters and dimensions by using medoids. For each medoid, the subspace is determined based on attributes with low variance. Random projections for k-means clustering [36] implements a dimensionality reduction technique for k-means clustering based on random projections.

9.2 Subspace Clustering

Subspace clustering algorithms identify clusters in appropriate subspaces of the original data space.

Examples are as follows. CLIQUE (CLustering In QUEst) [5] partitions the data space into units and then finds the maximum sets of connected dense units. SUBCLU (density-connected Subspace Clustering) [155] adopts the notion of density-connectivity introduced in DBSCAN (Section 6) and uses the monotonicity of density-connectivity to prune subspaces. CACTUS (Clustering Categorical Data Using

Summaries) is covered in Section 4. ENCLUS (ENtropy-based CLUStering) [53] finds clusters in subspaces based on entropy values of subspaces. Subspaces with lower entropy values typically have clusters. It then applies CLIQUE or other clustering algorithms to such subspaces. MAFIA (Merging of Adaptive Finite Intervals) [108] uses adaptive grids in each dimension and then merges them to find clusters in higher dimensions. OptiGrid (Optimal Grid) is covered in Section 7. MrCC (Multi-resolution Correlation Cluster detection) [58] constructs a novel data structure based on multi-resolution and detects correlation clusters by identifying initial clusters as axis-parallel hyper-rectangles with high data densities, followed by merging overlapping initial clusters. Figure 7 shows an example of subspace clustering.

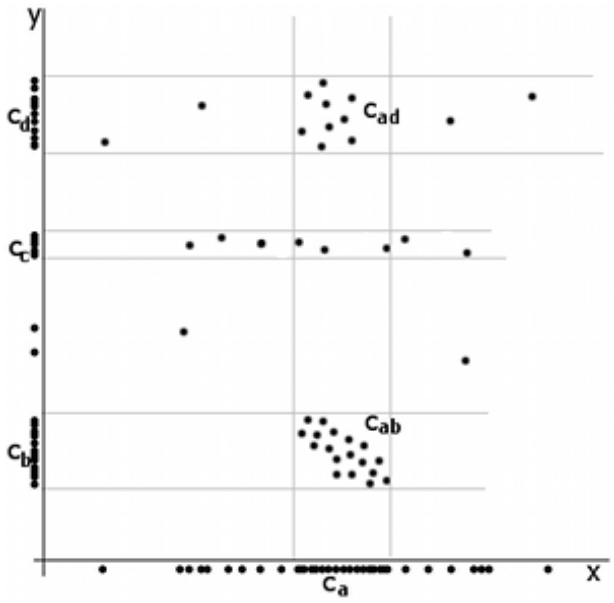

Figure 7: Subspace clustering [216]

9.3 Hybrid Approaches

Hybrid approaches find overlapping clusters. Some of them find only potentially interesting subspaces and use full-dimensional clustering algorithms to obtain the final clusters.

Examples are as follows. DOC (Density-based Optimal projective Clustering) [201] uses a global density threshold to compute an approximation of an optimal projective cluster. FIRES (FIlter REfinement Subspace clustering) [154] first computes one-dimensional clusters and then merges them by applying 'clustering of clusters' based on the number of intersecting points between clusters. P3C (Projected Clustering via Cluster Cores) [176, 177] first computes intervals matching or approximating higher-dimensional subspace clusters on every dimension and then aggregates those intervals into cluster cores. The cluster cores are refined and used to assign data objects.

9.4 Bi-clustering

Bi-clustering is also called bi-dimensional clustering [54], co-clustering, coupled clustering, or bimodal clustering. Bi-clustering is popular in bioinformatics research, especially in gene or sample clustering. For gene expression data, there are experimental conditions in which the activity of genes is uncorrelated. This causes limitations for results obtained by standard clustering methods. So bi-clustering algorithms that can perform simultaneous clustering on the genes and conditions are developed to find subgroups of genes and subgroups of conditions in which the genes exhibit highly correlated activities for every condition [168].

Examples are as follows. CTWC (Coupled Two-Way Clustering) [104] generates submatrices by an iterative process and considers only those submatrices whose rows and columns belong to genes and samples/conditions that were in a stable cluster in a previous iteration. ITWC (Interrelated Two-Way Clustering) [230] clusters the rows and then clusters the columns, based on each row cluster. It keeps the cluster pairs that are most dissimilar. Block Clustering [120] sorts the data by row mean or column mean and splits the rows or columns such that the variance within each 'block' is reduced. It then repeats and splits rows or columns differently. δ-biclusters [54] or CC algorithm (Cheng and Church's) finds biclusters whose rows and conditions show coherent values, using mean-squared residue. SAMBA (Statistical-Algorithmic Method for Bicluster Analysis) [229] uses probabilistic modeling and graph theoretic techniques to find subsets of rows whose values are very different in a subset of columns. Plaid Models [159] allows biclusters to overlap, *i.e.* a gene or a sample/condition can belong to more than one cluster. Information-theoretic co-clustering [72] intertwines the row and column clusterings to increase mutual information.

9.5 Correlation Clustering

Correlation clustering uses the correlations among attributes to guide the clustering process. These correlations may be different and exist in different clusters and cannot be reduced to uncorrelated ones by traditional global decorrelation techniques. Such correlations create clusters with different spatial shapes, and local correlation patterns are used to define the similarity between data objects. Correlation clustering is closely related to biclustering.

Examples are as follows. ORCLUS (ORiented projected CLUSter generation) [4] is similar to k-means but uses a distance function based on an eigensystem, *i.e.* the distance in the projected subspace. The eigensystem is adapted during iterations and close pairs of clusters are merged. 4C (Computing Correlation Connected Clusters) [34] takes a density-based approach and uses a density criterion to grow clusters. The density criterion is the minimal number of data objects within the neighborhood of a data object. The neighborhood is based on distance between two data objects in the eigensystems. HiCO (Hierarchical COrrelation clustering) [2] defines the

similarity between two data objects based on their local correlation dimensionality and subspace orientation. It takes a hierarchical density-based approach to obtain correlation clusters. CASH (Clustering in Arbitrary Subspaces based on the Hough transform) [1] is based on the Hough transform [129], which maps the data space into parameter space. It then uses a grid-based approach to find dense regions in the parameter space and corresponding data subsets in the original data space. It recursively applies itself on such corresponding data subsets.

10 Other Clustering Techniques

10.1 Neural Network-Based Clustering

The neural network approach has been studied intensively by mathematicians, statisticians, physicists, engineers, and computer scientists [157]. A neural network is an interconnected group of artificial neurons and an adaptive system for information processing. Neural-network-based clustering is competitive-learning-based clustering, not statistical model-identification based clustering. For competitive-learning-based clustering, the first phase is learning where the algorithmic parameters are adjusted, and the second phase is generalization [74]. Competitive learning can be implemented using a two-layer neural network: the input layer and the output layer [74].

Examples are as follows. A SOM (Self-Organizing Map) [153] consists of nodes or neurons, each of which is associated with a weight vector and a position in the map space. It creates a mapping from a higher dimensional input space to a lower dimensional output space. SOM clustering computes the distance of the input pattern to each neuron and finds the winning neuron. LVQ (Learning Vector Quantization) or VQ (Vector Quantization) [39, 102] is a classical quantization technique for signal processing. It models the probability density functions by using the distribution of prototype vectors. It divides a set of vectors into groups that have approximately the same number of vectors closest to them. Basic VQ is k-means clustering, and LVQ is a precursor to self-organizing maps (SOM) [102]. Neural gas [171] is inspired by SOM. It is a simple algorithm and finds optimal data representations based on feature vectors. During the adaptation process, the feature vectors distribute themselves dynamically like a gas within the data space. ART model is covered in Section 8.4.

10.2 Evolutionary Clustering

Evolutionary computation has many applications in computer science, bioinformatics, pharmacometrics, engineering, physics, and economics. Evolutionary computation is inspired by the biological mechanisms of evolution, and uses iterative processes such as growth or development followed by selection in a population of candidate solutions. Clustering methods that use local search techniques including hill-climbing approach-based k-means suffer from local minima problems. The recent advancements in evolutionary computational technologies [92] provide an alternate and effective way to find the global or approximately global optimum [256]. PSO (Particle Swarm Optimization) simulates social behavior in nature, such as bird flocking or fish schooling [148]. ACO (Ant Colony Optimization) algorithms model the behaviors of ants in nature [73]. GAs (Genetic Algorithms) [127] mimic natural selection and use evolutional mechanisms such as crossover, mutation and selection to generate solutions.

Examples are as follows. PPO (Particle-Pair Optimizer) [75] is a modification of the Particle Swarm Optimizer. It uses two particle pairs to search for the global optima in parallel and uses k-means for efficient clustering. Niching genetic k-means [213] modifies Deterministic Crowding [169], one of the niching genetic algorithms, and incorporates one step of k-means into its regeneration steps [213]. EvoCluster algorithm [166] encodes cluster structure in a chromosome, in which one gene represents one cluster or the objects belonging to one cluster. Reproduction operators are used between chromosomes. GenClust [103] is a simple algorithm and proceeds in stages. It uses genetic operators and a fitness function to compute partitions in a new stage based on partitions in the previous stage.

10.3 Kernel Clustering

Kernel-based learning such as Support Vector Machines (SVMs) [61, 208, 199] has had successful applications in pattern recognition and machine learning and is becoming increasingly important [199]. Kernel methods [62] perform a nonliner mapping of the low dimensional input data into a high dimensional space, which becomes linearly separable. To improve efficiency, they avoid explicitly defining the nonlinear mapping by using kernel functions, such as polynomial kernels, sigmoid kernels, and Gaussian radial basis function (RBF) kernels. This is the known as the *kernel trick*.

Examples are as follows. SVC (Support Vector Clustering) [239, 265] uses SVM training to find the cluster boundaries and an adjacency matrix to assign a cluster label to each data object [256]. Variations of SVC include Iterative One-Class SVC [40], and rough Set SVC [192]. Kernel k-means [107] uses a kernel method to calculate the distance between items in a data set, instead of using the Euclidean distance as in regular k-means. Variations include Incremental Kernel-k-means [209]. Kernel deterministic annealing clustering [262] uses an adaptively selected Gaussian parameter and a Gaussian kernel to determine the nonlinear mapping. Kernel fuzzy clustering [164, 268, 270] applies kernel techniques to fuzzy clustering algorithms by replacing the original Euclidean distance with a kernel-induced distance. Kernel Self-Organizing Maps [10, 35, 158] perform self-organizing between an input data object and the corresponding prototype in the mapped high dimensional feature space or in the mapped space completely.

10.4 Sequential Data Clustering

Sequential data are sequences of numerical data or non-numerical symbols and can be generated from speech processing, video analysis, text mining, gene sequencing, and medical diagnosis. Time series data or temporal data are a type of sequential data, which, unlike static data, contain feature values that change over time. Since sequential data usually have variable length, dynamic behaviors, and time constraints [116, 228], they cannot be represented as points in the multi-dimensional feature space and thus cannot be analyzed using any of the clustering techniques we have mentioned thus far [256]. Clustering techniques targeting sequential data have been developed, and they commonly use three strategies: proximity-based approaches, feature-based approaches, and model-based approaches.

Proximity-based approaches use proximity information such as the distance or similarity between pairs of sequences. They then use hierarchical or partitional clustering algorithms to group the sequences into clusters [256]. Examples are as follows. The Needleman-Wunsch algorithm [78, 184] uses basic dynamic programming and is a global optimal alignment algorithm. The Smith-Waterman algorithm [78, 217] is based on Needleman-Wunsch algorithm, and also uses dynamic programming. It compares multi-lengthed sequence segments using character-to-character pair-wise comparisons. FASTA (FAST-All) [194] first finds segments of the two sequences that have some degree of similarity and marks these potential matches. It then performs a more time-consuming optimized search approach such as the Smith-Waterman algorithm. BLAST (Basic Local Alignment Search Tool) [8] searches for short alignment matches between two sequences using a heuristic approach, which approximates the Smith-Waterman algorithm. GeneRage [82] automatically clusters sequence datasets by using Smith-Waterman dynamic programming alignment and single-linkage clustering. SEQOPTICS (SEQuence clustering with OPTICS) [52] implements Smith-Waterman algorithms as the distance measurement and uses OPTICS [11] to perform sequence clustering.

Feature-based approaches map sequences onto multi-dimensional data points using feature extraction methods and then use vector-based clustering algorithms on the data points [256]. Examples are as follows. Scalable sequential data clustering [115] uses a k-means based clustering algorithm which has near-linear time complexity to improve the scalability problem. Pattern-oriented hierarchical clustering [179] uses a hierarchical algorithm, which can generate the clusters as well as the clustering models based on sequential patterns found in the database. The wavelet-based anytime algorithm [237] combines a novel k-means based clustering algorithm and the multi-resolution property of wavelets. It repeatedly uses coarse clustering to obtain a clustering at a slightly finer level of approximation.

Model-based approaches assume sequences that belong to one cluster are generated from one probabilistic model [256]. Examples are as follows. Autoregressive moving average (ARMA) models [18, 253] derive an EM algorithm to learn the mixing coefficients and the parameters of the component ARMA models. They use the Bayesian information criterion (BIC) to determine the number of clusters. The Markov chain approach [202, 219] models dynamics as Markov chains and then applies an agglomerative clustering procedure to discover a set of clusters that best capture different dynamics. The Polynomial models approach [17, 97] assumes the underlying model is a mixture of polynomial functions. It uses an EM algorithm to estimate the cluster membership probabilities, using weighted least squares to fit the models. The Hidden Markov Model (HMM) [188, 218] is a probabilistic model-based approach. It uses HMMs, which have shown capabilities in modeling the structure of the generative processes underlying real-world time series data.

10.5 Ensemble Clustering

Clustering ensembles have emerged to improve robustness, stability and accuracy of clustering results [105]. A cluster ensemble combines the results of multiple clustering algorithms to obtain a consensus result [197]. It can produce better average performance and avoid worst case results. Other usages of clustering ensembles include improving scalability by performing clustering on subsets of data in parallel and then combining the results, and data integration when data is distributed across multiple sources [137].

There are two main steps in a clustering ensemble: generation and consensus. In the generation step, several approaches are used [235]: different clustering algorithms, a single algorithm with different parameter initializations, different object representations, different object projections, and different subsets of objects.

In the consensus step, several approaches are used: relabeling and voting, Mutual Information (MI), co-association based functions, finite mixture models, a graph/hypergraph partitioning approach, and others.

The relabeling and voting approach is also called the direct approach. It finds the correspondence of the cluster labels among different clustering results and then uses a voting method to determine the final cluster label for a data object. Examples are as follows. BagClust1 [76] applies a clustering procedure to each bootstrap sample and obtains the final partition by plurality voting so that the majority cluster label for each data object determines the final cluster membership. BagClust2 [76] introduces a new dissimilarity matrix which contains the proportion of time each pair of data objects were clustered together in the bootstrap clusters. It then performs clustering on the dissimilarity matrix to obtain the final partition.

The MI approach uses MI to measure and quantify the statistical information shared between a pair of clusterings. It can automatically select the best clustering method from several algorithms. Examples are as follows. A Genetic Algorithm (GA) clustering ensemble [15] uses a GA to obtain the best partition and the co-association function as the consensus

function. It determines fitness function parameters based on co-association function values. The information theory based GA clustering ensemble [165] uses a GA to find a combined clustering by minimizing an information-theoretical criterion function. The generalized MI clustering ensemble [233] introduces a new consensus function using a generalized mutual information definition. The consensus function is related to the classical intraclass variance criterion.

The co-association based functions approach is also called the pair-wise approach. It uses a co-association matrix in the consensus step. Examples are as follows. Clusterfusion [147] first generates an agreement matrix with each cell containing the number of agreements amongst clustering methods and then uses the matrix to cluster data objects. Voting-k-Means [95] transforms data partitions into a co-association matrix with coherent association mappings. It then extracts underlying clusters from this matrix. Evidence accumulation-based clustering [96] maps data partitions created by each individual clustering into a new similarity matrix, based on voting. It then uses the single link algorithm to extract clusters from this matrix.

Finite mixture model approach assumes that the probability of assigning a label to a data object is based on a finite mixture model or that the labels are 'modeled as random variables drawn from a probability distribution described as a mixture of multivariate component densities' [235]. It obtains the consensus clustering result by solving a maximum likelihood estimation problem. Mixture model clustering ensemble [234] uses a probabilistic model of consensus based on a finite mixture of multinomial distributions in a space of clusterings. It finds a combined partition by solving the corresponding maximum likelihood problem with the EM algorithm.

The graph/hypergraph partitioning approach considers the combination problem as a graph or hypergraph partitioning problem. Methods taking this approach differ in how they build a (hyper)graph from the clusterings, as well as how they define the cuts on the graph to obtain the consensus partition [235]. Examples are as follows. METIS [144] is a multi-level graph partitioning system. It collapses vertices and edges of the graph, partitions the resulting coarsened graph, and then refines the partitions. SPEC (spectral graph partitioning algorithm) [186] tries to optimize the normalized cut criterion. It treats the rows of the largest eigenvalues matrix as multiple dimensional embeddings of the vertices of the graph and then uses k-means to cluster the embedded points. CSPA (Cluster based Similarity Partitioning Algorithm) [227] first creates a graph based on a co-association matrix, and then performs METIS clustering on the graph. HGPA (Hypergraph Partitioning Algorithm) [227] uses a hyperedge in a graph to represent each cluster. It then uses minimal cut algorithms such as HMETIS [142] to find good hypergraph partitions. MCLA (Meta Clustering Algorithm) [227] determines soft cluster membership values for each data object by using hyperedge collapsing operations. HBGF (Hybrid Bipartite Graph Formulation) [90] constructs a bipartite graph where data objects and clusters are both modeled as vertices. It later partitions the bipartite graph with an appropriate graph partitioning method.

Other approaches are as follows. The cumulative voting consensus method [14] solves the cluster label alignment problem by using cumulative voting, where a probabilistic mapping between labels is computed. Bipartite Merger and Metis merger [128] are approaches for merging an ensemble of clustering solutions using sets of cluster centers. They are highly scalable and provide competitive results. Weighted consensus clustering [160] weights each input clustering. It determines weights in a way so that the clusters are better separated. Bayesian Cluster Ensembles [240] takes a Bayesian approach to combine clusterings. It uses a variational approximation based algorithm for learning. This way, it is able to avoid the cluster label correspondence problems. Figure 8 shows an example of ensemble clustering.

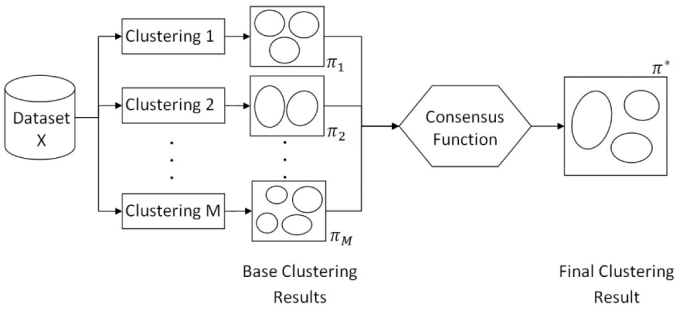

Figure 8: Ensemble clustering [133]

10.6 Multi-objective Clustering

Conventional clustering algorithms use a single clustering objective function only, which may not be appropriate for the diversities of the underlying data structures. Multi-objective clustering uses multiple clustering objective functions simultaneously. Such methods consider clustering as a multi-objective optimization problem [89].

Examples are as follows. FCPSO (Fuzzy Clustering-based Particle Swarm Optimization) [6] uses an external repository to save nondominated particles during the search process and a fuzzy clustering technique to manage the size of the repository. It also uses a fuzzy-based iterative feedback mechanism to determine the compromised solution among conflicting objectives. Evolutionary Multiobjective Clustering [118] and MOCK (MultiObjective Clustering with automatic k-determination) [119] use an evolutionary approach to solve the multi-objective problem in clustering. They are based on a multi-objective evolutionary algorithm named PESA-II (Pareto Envelope-based Selection Algorithm version 2) [59] to optimize two complementary clustering objectives. Multi-objective real coded genetic fuzzy clustering [181] aims to optimize multiple validity measures simultaneously. It encodes the cluster centers in its chromosomes while optimizing the fuzzy compactness within a cluster and fuzzy separation among

clusters. EMO-CC (Evolutionary MultiObjective Conceptual Clustering) [203] combines evolutionary algorithms with multi-objective optimization techniques and relies on the NSGA-II multi-objective genetic algorithm [66]. It can discover less obvious but informative data associations.

10.7 Semi-supervised Clustering

Semi-supervised clustering provides limited supervision to unsupervised clustering. There are many cases when some knowledge about the data is available such as the constraints between data objects or cluster labels for some data objects. Such knowledge can be used to guide the clustering process. There are several approaches for semi-supervised clustering: similarity-adapting methods, search-based methods, and other methods.

Similarity-adapting methods use a similarity measure which is adapted to make the available constraints more easily satisfied [111]. Examples are as follows. Distance metric learning based clustering [252] learns a distance metric based on examples of similar pairs of data objects in the input space using convex optimization. Space-level constraints based clustering [150] exploits space-level implications based on instance-level constraints. It uses an all-pairs-shortest-paths algorithm to adjust the distance metric.

Search-based methods modify the clustering algorithm itself to use the available constraints or labels to guide the search for an appropriate clustering [111]. Examples are as follows. Seeded-K Means and Constrained-K Means [23] generate initial seed clusters based on labeled data. The latter also generates constraints from labeled data and guides the clustering process using those constraints. Semi-Supervised Clustering Using Genetic Algorithms [68] modifies k-means clustering to minimize within-cluster variance and a measure of cluster impurity. Clustering with Instance-level Constraints [238] incorporates hard constraints using a modified version of Cobweb (covered in Section 8.4) which partitions the data.

Other methods include the probabilistic semi-supervised clustering with constraints method [24], which derives an objective function from the joint probability defined over the Hidden Markov Random Field model and performs semi-supervised clustering by minimizing this object function.

11 Conclusions

We have presented a survey of the literature on clustering techniques. For convenience, in this review we used the following taxonomy, which is also widely used in the literature: hierarchical clustering (Section 2), partitioning clustering (Section 3), graph-based clustering (Section 4), distribution-based clustering (Section 5), density-based clustering (Section 6), grid-based clustering (Section 7), clustering big data (Section 8), clustering high dimensional data (Section 9), and other clustering techniques (Section 10).

References

[1] Elke Achtert, Christian Böhm, Jörn David, Peer Kröger, and Arthur Zimek. “Global Correlation Clustering Based on the Hough Transform”. *Statistical Analysis and Data Mining*, 1(3):111–127. 2008.

[2] Elke Achtert, Christian Böhm, Peer Kröger, and Arthur Zimek. “Mining Hierarchies of Correlation Clusters”. In “Scientific and Statistical Database Management, 2006. 18th International Conference on”, IEEE, pp. 119–128. 2006.

[3] Charu C. Aggarwal, Joel L. Wolf, Philip S. Yu, Cecilia Procopiuc, and Jong Soo Park. “Fast Algorithms for Projected Clustering”. In “SIGMOD ’99: Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data”, ACM, New York, NY, USA, pp. 61-72. doi:10.1145/304182.304188. 1999.

[4] Charu C. Aggarwal and Philip S. Yu. “Finding Generalized Projected Clusters in High Dimensional Spaces”. *SIGMOD Rec.*, 29(2):70–81. doi:10.1145/335191.335383. May 2000.

[5] Rakesh Agrawal, Johannes Gehrke, Dimitrios Gunopulos, and Prabhakar Raghavan. “Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications”. *SIGMOD Rec.*, 27(2):94–105. doi:10.1145/276305.276314. Jun. 1998.

[6] Shubham Agrawal, B. K. Panigrahi, and Manoj Kumar Tiwari. “Multiobjective Particle Swarm Algorithm with Fuzzy Clustering for Electrical Power Dispatch.” *IEEE Trans. Evolutionary Computation*, 12(5):529–541. 2008.

[7] Khaled S. Al-Sultan and Shokri Z. Selim. “A Global Algorithm for the Fuzzy Clustering Problem.” *Pattern Recognition*, 26(9):1357–1361. 1993.

[8] Stephen F. Altschul, Warren Gish, Webb Miller, Eugene W. Myers, and David J. Lipman. “Basic Local Alignment Search Tool”. *Journal of Molecular Biology*, 215(3):403 – 410. doi:http://dx.doi.org/10.1016/S0022-2836(05)80360-2. 1990.

[9] G.C. Anagnostopoulos and M. Georgiopoulos. “Ellipsoid ART and ARTMAP for Incremental Clustering and Classification”. In “IJCNN’01. International Joint Conference on Neural Networks. Proceedings (Cat. No.01CH37222)”, pp. 1221–1226 vol.2. doi:10.1109/IJCNN.2001.939535. 2001.

[10] Peter Andras. “Kernel-Kohonen Networks.” *Int. J. Neural Syst.*, 12(2):117–135. 2002.

[11] Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, and Jörg Sander. “OPTICS: Ordering Points to Identify the Clustering Structure”. *SIGMOD Rec.*, 28(2):49–60. doi:10.1145/304181.304187. Jun. 1999.

[12] David Arthur and Sergei Vassilvitskii. “k-means++: The Advantages of Careful Seeding”. In “Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA ’07)”, Society for Industrial

and Applied Mathematics, Philadelphia, PA, USA, pp. 1027–1035. 2007.

[13] Roberto Avogadri and Giorgio Valentini. "Fuzzy Ensemble Clustering Based on Random Projections for DNA Microarray Data Analysis". *Artificial Intelligence in Medicine*, 45(2):173–183. 2009.

[14] Hanan G Ayad and Mohamed S Kamel. "Cumulative Voting Consensus Method for Partitions with Variable Number of Clusters". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(1):160–173. 2008.

[15] J. Azimi, M. Mohammadi, A. Movaghar, and M. Analoui. "Clustering Ensembles Using Genetic Algorithm". In "International Workshop on Computer Architecture for Machine Perception and Sensing, 2006. CAMP 2006.", pp. 119–123. doi: 10.1109/CAMP.2007.4350366. 2007.

[16] G. Phanendra Babu and M. Narasimha Murty. "A Near-Optimal Initial Seed Value Selection in K-means Means Algorithm Using a Genetic Algorithm." *Pattern Recognition Letters*, 14(10):763–769. 1993.

[17] A. Bagnall, G. Janacek, B. Iglesia, and M. Zhang. "Clustering Time Series from Mixture Polynomial Models with Discretized Data". In "Proc. 2nd Australasian Data Mining Workshop", pp. 105–120. 2003.

[18] Anthony J. Bagnall and Gareth J. Janacek. "Clustering Time Series from ARMA Models with Clipped Data." In Won Kim, Ron Kohavi, Johannes Gehrke, and William DuMouchel, editors, "KDD", ACM, pp. 49–58. 2004.

[19] G. H. Ball and D. J. Hall. "ISODATA, an Iterative Method of Multivariate Analysis and Pattern Classification". *IFIPS Congress*. 1965.

[20] A. Baraldi and E. Alpaydin. "Constructive Feedforward ART Clustering Networks – Part I and II". *IEEE Trans. Neural Netw.*, 13(3). May 2002.

[21] D. Barbara and P. Chen. "Using the Fractal Dimension to Cluster Datasets". In "Proc. of the 6th International Conference on Knowledge Discovery and Data Mining", ACM, pp. 260–264. 2000.

[22] Daniel Barbara, Julia Couto, and Yi Li. "COOLCAT: An Entropy-Based Algorithm for Categorical Clustering". In "In Proceedings of the Eleventh International Conference on Information and Knowledge Management", ACM, pp. 582–589. 2002.

[23] S. Basu, A. Banerjee, and R. Mooney. "Semi-Supervised Clustering by Seeding". In "Proceedings of the International Conference on Machine Learning", pp. 27–34. 2002.

[24] Sugato Basu, Mikhail Bilenko, Arindam Banerjee, and Raymond J Mooney. "Probabilistic Semi-Supervised Clustering with Constraints". *Semi-Supervised Learning*, pp. 71–98. 2006.

[25] Stephen D. Bay and Michael J. Pazzani. "Detecting Change in Categorical Data: Mining Contrast Sets". In "Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining", ACM, New York, NY, USA, KDD '99, pp. 302–306. doi: 10.1145/312129.312263. 1999.

[26] Ron Bekkerman, Ran El-Yaniv, and Andrew McCallum. "Multi-way Distributional Clustering via Pairwise Interactions". In "Proceedings of the 22nd International Conference on Machine learning", ACM, pp. 41–48. 2005.

[27] Mikhail Belkin and Partha Niyogi. "Laplacian Eigenmaps for Dimensionality Reduction and Data Representation". *Neural Computation*, 15(6):1373–1396. 2003.

[28] Pavel Berkhin. "Survey Of Clustering Data Mining Techniques". Tech. rep., Accrue Software, San Jose, CA. 2002.

[29] J. Bezdek, C. Coray, R. Gunderson, and J. Watson. "Detection and Characterization of Cluster Substructure I. Linear Structure: Fuzzy c-Lines". *SIAM Journal on Applied Mathematics*, 40(2):339–357. doi:10.1137/0140029. 1981.

[30] D. A. Binder. "Bayesian Cluster Analysis". *Biometrika*, 65(1):31–38. 1978.

[31] C. Bishop. *Neural Networks for Pattern Recognition*. Oxford Univ. Press. 1995.

[32] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. "Latent Dirichlet Allocation". *J. Mach. Learn. Res.*, 3:993–1022. Mar. 2003.

[33] Hans H. Bock. "Probabilistic Models in Cluster Analysis". *Computational Statistics & Data Analysis*, 23(1):5–28. Nov. 1996.

[34] Christian Böhm, Karin Kailing, Hans-Peter Kriegel, and Peer Kröger. "Density Connected Clustering with Local Subspace Preferences." In "ICDM", IEEE Computer Society, pp. 27–34. 2004.

[35] Romain Boulet, Bertrand Jouve, Fabrice Rossi, and Nathalie Villa. "Batch Kernel SOM and Related Laplacian Methods for Social Network Analysis". *Neurocomputing*, 71(7-9):1257–1273. 2008.

[36] Christos Boutsidis, Anastasios Zouzias, and Petros Drineas. "Random Projections for K-Means Clustering." In "Advances in Neural Information Processing Systems", pp. 298–306. 2010.

[37] Paul S. Bradley, Usama M. Fayyad, and Cory Reina. "Scaling Clustering Algorithms to Large Databases". In "Knowledge Discovery and Data Mining", pp. 9–15. 1998.

[38] Nathan Brannon, John Seiffertt, Timothy Draelos, and Donald C. Wunsch II. "Coordinated Machine Learning and Decision Support for Situation Awareness." *Neural Networks*, 22(3):316–325. 2009.

[39] D. Burton, J. Shore, and J. Buck. "A Generalization of Isolated Word Recognition Using Vector Quantization". In "Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '83.", vol. 8, pp. 1021–1024. doi:10.1109/ICASSP.1983.1171915. Apr 1983.

[40] Francesco Camastra and Alessandro Verri. "A Novel Kernel Method for Clustering." *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(5):801–804. 2005.

[41] G. A. Carpenter, S. Grossberg, N. Markuzon, J.H. Reynolds, and D.B. Rosen. "Fuzzy ARTMAP: A Neural Network Architecture for Incremental Supervised Learning of Analog Multidimensional Maps". *IEEE Transactions on Neural Networks*, 3(5):698–713. September 1992.

[42] Gail A. Carpenter and Stephen Grossberg. "ART 2: Self-Organization of Stable Category Recognition Codes for Analog Input Patterns". *Applied Optics*, 26(23):4919–4930. 1987.

[43] Gail A. Carpenter and Stephen Grossberg. "A Massively Parallel Architecture for a Self-Organizing Neural Pattern Recognition Machine". *Computer Vision, Graphics, and Image Processing*, 37(1):54–115. 1987.

[44] Gail A. Carpenter, Stephen Grossberg, and John H. Reynolds. "ARTMAP: Supervised Real-Time Learning and Classification of Nonstationary Data by a Self-Organizing Neural Network". *Neural Networks*, 4(5):565–588. 1991.

[45] Thomas P Caudell, Scott DG Smith, G Craig Johnson, and Donald C Wunsch II. "Application of Neural Networks to Group Technology". In "Proceedings of SPIE 1469, Applications of Artificial Neural Networks II", SPIE–The International Society for Optical Engineering, pp. 612–621. Jan 1991.

[46] Soumen Chakrabarti. *Mining the Web: Discovering Knowledge from Hypertext Data*. Morgan Kaufmann. 2003.

[47] Jae-Woo Chang and Du-Seok Jin. "A New Cell-Based Clustering Method for Large, High-Dimensional Data in Data Mining Applications". In "Proceedings of the 2002 ACM Symposium on Applied Computing", ACM, pp. 503–507. 2002.

[48] Moses Charikar, Chandra Chekuri, Tomás Feder, and Rajeev Motwani. "Incremental Clustering and Dynamic Information Retrieval". In "Proceedings of the Twenty-ninth Annual ACM Symposium on Theory of Computing", ACM, New York, NY, USA, STOC '97, pp. 626–635. doi:10.1145/258533.258657. 1997.

[49] Peter Cheeseman and John Stutz. "Advances in Knowledge Discovery and Data Mining". In Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, and Ramasamy Uthurusamy, editors, "Bayesian Classification (AutoClass): Theory and Results", American Association for Artificial Intelligence, Menlo Park, CA, USA, pp. 153–180. 1996.

[50] Jiun-Rung Chen. *Efficient Biclustering Methods for Microarray Databases*. Ph.D. thesis, National Sun Yat-sen University. 2010.

[51] Wen-Yen Chen, Yangqiu Song, Hongjie Bai, Chih-Jen Lin, and Edward Y Chang. "Parallel Spectral Clustering in Distributed Systems". *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(3):568–586. 2011.

[52] Yonghui Chen, Kevin D Reilly, Alan P Sprague, and Zhijie Guan. "SEQOPTICS: A Protein Sequence Clustering System". *BMC Bioinformatics*, 7(Suppl 4):S10. 2006.

[53] Chun Hung Cheng, Ada Wai-Chee Fu, and Yi Zhang. "Entropy-Based Subspace Clustering for Mining Numerical Data." In Usama M. Fayyad, Surajit Chaudhuri, and David Madigan, editors, "KDD", ACM, pp. 84–93. 1999.

[54] Yizong Cheng and George M. Church. "Biclustering of Expression Data". In "Proc. of the 8th Intelligent Systems for Molecular Biology", AAAI Press, pp. 93–103. 2000.

[55] Chire. "Density-Based Clustering with DBSCAN". URL http://commons.wikimedia.org/wiki/File:DBSCAN-density-data.svg. Accessed Oct. 2015. 2011.

[56] Chire. "Expectation-Maximization (EM) Clustering Examples". URL http://commons.wikimedia.org/wiki/File:EM-Gaussian-data.svg. Accessed Oct. 2015. 2011.

[57] Chire. "k-Means Clustering Examples". URL http://commons.wikimedia.org/wiki/File:KMeans-Gaussian-data.svg. Accessed Oct. 2015. 2011.

[58] Robson Leonardo Ferreira Cordeiro, Agma J. M. Traina, and Christos Faloutsos. "Finding Clusters in Subspaces of Very Large, Multi-Dimensional Datasets". In "IEEE 26th International Conference on Data Engineering (ICDE), 2010", IEEE, pp. 625–636. 2010.

[59] David W. Corne, Nick R. Jerram, Joshua D. Knowles, and Martin J. Oates. "PESA-II: Region-Based Selection in Evolutionary Multiobjective Optimization". In "Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation", Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, GECCO'01, p. 283–290. 2001.

[60] Carlos D Correa and Peter Lindstrom. "Locally-Scaled Spectral Clustering Using Empty Region Graphs". In "Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining", ACM, pp. 1330–1338. 2012.

[61] Corinna Cortes and Vladimir Vapnik. "Support-Vector Networks". *Machine learning*, 20(3):273–297. doi:10.1023/a:1022627411411. Sep. 1995.

[62] Nello Cristianini and John Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 1st edn. Mar. 2000.

[63] R. D'Andrade. "U-Statistic Hierarchical Clustering". *Psychometrika*, 4. 1978.

[64] M. Dash, H. Liu, and Xiaowei Xu. "'1+1 > 2': Merging Distance and Density Based Clustering". In "Database Systems for Advanced Applications, 2001. Proceedings. Seventh International Conference on", IEEE Computer Society, pp. 32 –39. doi:10.1109/DASFAA.2001.916361. april 2001.

[65] N. E. Day. "Estimating the Components of a Mixture of Normal Distributions". *Biometrika*, 56(3):463–474. doi:10.1093/biomet/56.3.463. 1969.

[66] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II". *IEEE Transactions on Evolutionary Computation,*, 6(2):182–197. 2002.

[67] D. Defays. "An Efficient Algorithm for a Complete Link Method". *The Computer Journal (British Computer Society)*, 20(4):364–366. 1977.

[68] Ayhan Demiriz, Kristin Bennett, and Mark J. Embrechts. "Semi-Supervised Clustering Using Genetic Algorithms". In "Artificial Neural Networks in Engineering (ANNIE-99)", ASME Press, pp. 809–814. 1999.

[69] A. P. Dempster, N. M. Laird, and D. B. Rubin. "Maximum Likelihood From Incomplete Data via the EM Algorithm". *Journal of the Royal Statistical Society, Series B*, 39(1):1–38. 1977.

[70] I. Dhillon and D. Modha. "A Data Clustering Algorithm on Distributed Memory Multiprocessors". In "5th ACM SIGKDD, Large-scale Parallel KDD Systems Workshop", pp. 245–260. 1999.

[71] Inderjit S. Dhillon, Yuqiang Guan, and Brian Kulis. "Kernel K-Means: Spectral Clustering and Normalized Cuts". In "Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining", ACM, pp. 551–556. 2004.

[72] Inderjit S Dhillon, Subramanyam Mallela, and Dharmendra S Modha. "Information-Theoretic Co-Clustering". In "Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining", ACM, pp. 89–98. 2003.

[73] Marco Dorigo and Thomas Stützle. "The Ant Colony Optimization Metaheuristic: Algorithms, Applications, and Advances". In "Handbook of Metaheuristics", Springer, pp. 250–285. 2003.

[74] K.-L. Du. "Clustering: A Neural Network Approach." *Neural Networks*, 23(1):89–107. 2010.

[75] Zhihua Du, Yiwei Wang, and Zhen Ji. "PK-means: A New Algorithm for Gene Clustering." *Computational Biology and Chemistry*, 32(4):243–247. 2008.

[76] Sandrine Dudoit and Jane Fridlyand. "Bagging to Improve the Accuracy of a Clustering Procedure". *Bioinformatics*, 19(9):1090–1099. 2003.

[77] Joseph C. Dunn. "A Fuzzy Relative of the ISODATA Process and its use in Detecting Compact Well-Separated Clusters". *Journal of Cybernetics*, 3:32–57. 1973.

[78] Richard Durbin, Sean R. Eddy, Anders Krogh, and Graeme Mitchison. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press. 1998.

[79] A. W. F. Edwards and L. L. Cavalli-Sforza. "A Method for Cluster Analysis". *Biometrics*, 21:362–375. 1965.

[80] C. Eldershaw and M. Hegland. "Cluster Analysis Using Triangulation". *Computational Techniques and Applications*, pp. 201–208. 1997.

[81] Charles Elkan. "Using the Triangle Inequality to Accelerate k-means". In "Proceedings of the Twentieth International Conference on Machine Learning (ICML)", vol. 3, pp. 147–153. 2003.

[82] A. Enright and C. Ouzounis. "GeneRAGE: A Robust Algorithm for Sequence Clustering and Domain Detection". *Bioinformatics*, 16(5):451–457. 2000.

[83] S. Eschrich, Jingwei Ke, L.O. Hall, and D.B. Goldgof. "Fast Accurate Fuzzy Clustering Through Data Reduction". *IEEE Transactions on Fuzzy Systems*, 11(2):262–270. doi:10.1109/TFUZZ.2003.809902. 2003.

[84] Martin Ester, Hans P. Kriegel, Jorg Sander, and Xiaowei Xu. "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise". In Evangelos Simoudis, Jiawei Han, and Usama Fayyad, editors, "Second International Conference on Knowledge Discovery and Data Mining", AAAI Press, Portland, Oregon, pp. 226–231. 1996.

[85] Tim Evans. "Zachary Karate Club Network Clustered Using Clique Graph methods". URL http://netplexity.org/?p=1261. Accessed Oct. 2015. 2014.

[86] B. Everitt, S. Landau, and M. Leese. *Cluster Analysis*. London: Arnold. 2001.

[87] B. S. Everitt. "Cluster Analysis". In "Cluster Analysis, Second Edition", Heineman Educational Books Ltd. 1980.

[88] Christos Faloutsos and King-Ip Lin. "FastMap: A Fast Algorithm for Indexing, Data-mining and Visualization of Traditional and Multimedia Datasets". *SIGMOD Rec.*, 24(2):163–174. doi:10.1145/568271.223812. May 1995.

[89] A. Ferligoj and V. Batagelj. "Direct Multicriterion Clustering Algorithms". *Journal of Classification*, 9:43–61. 1992.

[90] Xiaoli Zhang Fern and Carla E. Brodley. "Random Projection for High Dimensional Data Clustering: A Cluster Ensemble Approach." In Tom Fawcett and Nina

Mishra, editors, "International Conference on Machine Learning", AAAI Press, pp. 186–193. 2003.

[91] Douglas H Fisher. "Knowledge Acquisition via Incremental Conceptual Clustering". *Machine Learning*, 2(2):139–172. 1987.

[92] David B Fogel. *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*. John Wiley & Sons. 2006.

[93] E. W. Forgy. "Cluster Analysis of Multivariate Data: Efficiency Versus Interpretability of Classifications". *Biometrics*, 21:768–769. 1965.

[94] Ildiko E. Frank and Roberto Todeschini. *The Data Analysis Handbook*. Elsevier Science Inc. 1994.

[95] Ana L. N. Fred. "Finding Consistent Clusters in Data Partitions." In Josef Kittler and Fabio Roli, editors, "Multiple Classifier Systems", Springer, vol. 2096 of *Lecture Notes in Computer Science*, pp. 309–318. 2001.

[96] Ana L. N. Fred and Anil K. Jain. "Data Clustering Using Evidence Accumulation". In "Proceedings 16th International Conference on Pattern Recognition", IEEE, vol. 4, pp. 276–280. 2002.

[97] Scott Gaffney and Padhraic Smyth. "Trajectory Clustering with Mixtures of Regression Models". In "Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining", ACM, pp. 63–72. 1999.

[98] Guojun Gan. *Data Clustering in C++: An Object-Oriented Approach*. Chapman & Hall/CRC Data Mining and Knowledge Discovery Series. Taylor & Francis. 2011.

[99] Guojun Gan, Chaoqun Ma, and Jianhong Wu. *Data Clustering - Theory, Algorithms, and Applications*. SIAM. 2007.

[100] Venkatesh Ganti, Johannes Gehrke, and Raghu Ramakrishnan. "CACTUS - Clustering Categorical Data Using Summaries". In "Proceedings of the fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining", ACM, New York, NY, USA, KDD '99, pp. 73–83. doi:10.1145/312129.312201. 1999.

[101] Venkatesh Ganti, Raghu Ramakrishnan, Johannes Gehrke, Allison Powell, and James French. "Clustering Large Datasets in Arbitrary Metric Spaces". In "Data Engineering, 1999. Proceedings., 15th International Conference on", IEEE, pp. 502–511. 1999.

[102] A. Gersho and B. Ramamurthi. "Image Coding Using Vector Quantization". *International Conference on Acoustics, Speech, and Signal Processing*, 1:428–431. April 1982.

[103] Vito Di Gesù, Raffaele Giancarlo, Giosuè Lo Bosco, Alessandra Raimondi, and Davide Scaturro. "GenClust: A Genetic Algorithm for Clustering Gene Expression Data". *BMC Bioinformatics*, 6(1):289–289. 2005.

[104] Gad Getz, Erel Levine, and Eytan Domany. "Coupled Two-Way Clustering Analysis of Gene Microarray Data". *Proceedings of the National Academy of Sciences*, 97(22):12079–12084. doi:10.1073/pnas.210134797. 2000.

[105] Reza Ghaemi, Md. Nasir Sulaiman, Hamidah Ibrahim, and Norwati Mustapha. "A Survey: Clustering Ensembles Techniques". *World Academy of Science, Engineering and Technology*, 50:636–645. 2009.

[106] David Gibson, Jon Kleinberg, and Prabhakar Raghavan. "Clustering Categorical Data: An Approach Based on Dynamical Systems". *The International Journal on Very Large Data Bases*, 8(3-4):222–236. doi:10.1007/s007780050005. Feb. 2000.

[107] M. Girolami. "Mercer Kernel-Based Clustering in Feature Space". *Neural Networks, IEEE Transactions on*, 13(3):780–784. doi:10.1109/tnn.2002.1000150. Aug. 2002.

[108] Sanjay Goil, Harsha Nagesh, and Alok Choudhary. "MAFIA: Efficient and Scalable Subspace Clustering for Very Large Data Sets". In "Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining", pp. 443–452. 1999.

[109] Teofilo F Gonzalez. "Clustering to Minimize the Maximum Intercluster Distance". *Theoretical Computer Science*, 38(2-3):293–306. 1985.

[110] J. C. Gower and G. J. S. Ross. "Minimum Spanning Trees and Single Linkage Cluster Analysis". *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 18(1):54–64. 1969.

[111] Nizar Grira, Michel Crucianu, and Nozha Boujemaa. "Semi-Supervised Fuzzy Clustering with Pairwise-Constrained Competitive Agglomeration". In "Fuzzy Systems, 2005. FUZZ'05. The 14th IEEE International Conference on", IEEE, pp. 867–872. 2005.

[112] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. "CURE: An Efficient Clustering Algorithm for Large Databases". *SIGMOD Record*, 27(2):73–84. doi:10.1145/276305.276312. Jun. 1998.

[113] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. "ROCK: A Robust Clustering Algorithm for Categorical Attributes". *Information Systems*, 25(5):345 – 366. doi:10.1016/S0306-4379(00)00022-3. 2000.

[114] Chetan Gupta and Robert Grossman. "GenIc: A Single Pass Generalized Incremental Algorithm for Clustering". In "Proceedings of the Fourth SIAM International Conference on Data Mining", SIAM, pp. 147–153. 2004.

[115] Valerie Guralnik and George Karypis. "A Scalable Algorithm for Clustering Sequential Data." In Nick Cercone, Tsau Young Lin, and Xindong Wu, editors, "ICDM", IEEE Computer Society, pp. 179–186. 2001.

[116] Dan Gusfield. *Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology*. Cambridge University Press. 1997.

[117] Lars Hagen and Andrew B. Kahng. "New Spectral Methods for Ratio Cut Partitioning and Clustering". *IEEE Tansactions on Computer-aided Design*, 11(9):1074–1085. September 1992.

[118] Julia Handl and Joshua Knowles. "Evolutionary Multiobjective Clustering". In "Parallel Problem Solving from Nature-PPSN VIII", Springer, pp. 1081–1091. 2004.

[119] Julia Handl and Joshua Knowles. "An Evolutionary Approach to Multiobjective Clustering". *Evolutionary Computation, IEEE Transactions on*, 11(1):56–76. 2007.

[120] J. A. Hartigan. "Clustering algorithms". In "Clustering Algorithms", John Wiley & Sons, Inc. 1975.

[121] J. A. Hartigan and M. A. Wong. "Algorithm AS 136: A k-Means Clustering Algorithm". *Journal of the Royal Statistical Society, Series C (Applied Statistics)*, 28(1):100–108. 1979.

[122] Alexander Hinneburg, Er Hinneburg, and Daniel A. Keim. "An Efficient Approach to Clustering in Large Multimedia Databases with Noise". In "SIGKDD Conference on Knowledge Discovery and Data Mining", AAAI Press, vol. 98, pp. 58–65. 1998.

[123] Alexander Hinneburg and Daniel A. Keim. "Optimal Grid-Clustering: Towards Breaking the Curse of Dimensionality in High-Dimensional Clustering." In Malcolm P. Atkinson, Maria E. Orlowska, Patrick Valduriez, Stanley B. Zdonik, and Michael L. Brodie, editors, "Proceedings of the 25th International Conference on Very Large Data Bases", Morgan Kaufmann, pp. 506–517. 1999.

[124] Ricky Ho. "Machine Learning in R: Clustering". URL http://horicky.blogspot.com/2012/04/machine-learning-in-r-clustering.html. Accessed Oct. 2015. 2012.

[125] Matthew D. Hoffman, David M. Blei, and Francis R. Bach. "Online Learning for Latent Dirichlet Allocation". In "Advances in Neural Information Processing Systems", pp. 856–864. 2010.

[126] Thomas Hofmann and Joachim M. Buhmann. "Pairwise Data Clustering by Deterministic Annealing". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:1–14. 1997.

[127] J. Holland. *Adaption in Natural and Artificial Systems*. Ann Arbor, MI: Univ. Michigan Press. 1975.

[128] Prodip Hore, Lawrence O Hall, and Dmitry B Goldgof. "A Scalable Framework for Cluster Ensembles". *Pattern recognition*, 42(5):676–688. 2009.

[129] P. V. C. Hough. "Methods and Means for Recognizing Complex Patterns". US Patent 3069654. December 1962.

[130] Xiaohua Hu and Illhoi Yoo. "Cluster Ensemble and its Applications in Gene Expression Analysis". In "Proceedings of the Second Conference on Asia-Pacific Bioinformatics", Australian Computer Society, Inc., Darlinghurst, Australia, vol. 29 of *APBC '04*, pp. 297–302. 2004.

[131] Z. Huang. "Extensions to the k-Means Algorithms for Clustering Large Data Sets with Categorical Values". *Data Mining and Knowledge Discovery*, 2(3):283–304. 1998.

[132] Christopher L. Huntley and Donald E. Brown. "A Parallel Heuristic for Quadratic Assignment Problems". *Computers & Operations Research*, 18(3):275–289. 1991.

[133] N. Iam-On, T. Boongeon, S. Garrett, and C. Price. "A Link-Based Cluster Ensemble Approach for Categorical Data Clustering". *Knowledge and Data Engineering, IEEE Transactions on*, 24(3):413–425. doi:10.1109/TKDE.2010.268. March 2012.

[134] V. Ilango, R. Subramanian, and V. Vasudevan. "Cluster Analysis Research Design Model, Problems, Issues, Challenges, Trends and Tools". *International Journal on Computer Science and Engineering*, 3(8):2926–2934. 2011.

[135] A. Jain, R. Duin, and J. Mao. "Statistical Pattern Recognition: A Review". *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(1):4–37. 2000.

[136] A. K. Jain and R. C. Dubes. "Algorithms for Clustering Data". In "Prentice-Hall Advanced Reference Series", Prentice-Hall, Inc. 1988.

[137] Anil K. Jain. "Data Clustering: User's Dilemma". In "Proceedings of the 5th International Conference on Machine Learning and Data Mining in Pattern Recognition", Springer-Verlag, Berlin, Heidelberg, MLDM '07, pp. 1–1. doi:10.1007/978-3-540-73499-4_1. 2007.

[138] Anil K Jain. "Data clustering: 50 Years Beyond K-Means". *Pattern Recognition Letters*, 31(8):651–666. 2010.

[139] Anil K. Jain, M. Narasimha Murty, and Patrick J. Flynn. "Data Clustering: A Review". *ACM Computing Surveys (CSUR)*, 31(3):264–323. 1999.

[140] Eshref Januzaj, Hans-Peter Kriegel, and Martin Pfeifle. "DBDC: Density Based Distributed Clustering". In "Advances in Database Technology-EDBT 2004", Springer, pp. 88–105. 2004.

[141] H. D. Jin. *Scalable Model-Based Clustering Algorithms for Large Databases and Their Applications*. Ph.D. thesis, The Chinese University of Hong Kong. Aug. 2002.

[142] G. Karypis, R. Aggarwal, V. Kumar, and Shashi Shekhar. "Multilevel hypergraph partitioning: Applications in VLSI domain". *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 7(1):69–79. doi:10.1109/92.748202. March 1999.

[143] G. Karypis, E. Han, and V. Kumar. "Chameleon: Hierarchical Clustering Using Dynamic Modeling". *IEEE Computer*, 32(8):68–75. Aug. 1999.

[144] George Karypis and Vipin Kumar. *METIS: A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices*. University of Minnesota, Department of Computer Science. September 1998.

[145] L. Kaufman and P. Rousseeuw. *Clustering by Means of Medoids*. North-Holland. 1987.

[146] Leonard Kaufman and Peter J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*, vol. 344. John Wiley & Sons. 2009.

[147] Paul Kellam, Xiaohui Liu, Nigel Martin, Christine Orengo, Stephen Swift, and Allan Tucker. "Comparing, Contrasting and Combining Clusters in Viral Gene Expression Data". In "Proceedings of 6th Workshop on Intelligent Data Analysis in Medicine and Pharmocology", pp. 56–62. 2001.

[148] James F. Kennedy, James Kennedy, and Russell C. Eberhart. *Swarm Intelligence*. Morgan Kaufmann. 2001.

[149] Benjamin King. "Step-Wise Clustering Procedures". *Journal of the American Statistical Association*, 62(317):86–101. 1967.

[150] D. Klein, S. Kamvar, and C. Manning. "From Instance-Level Constraints to Space-Level Constraints: Making the Most of Prior Knowledge in Data Clustering". In "Proceedings of the 19th International Conference on Machine Learning", pp. 307–314. 2002.

[151] R. W. Klein and R. C. Dubes. "Experiments in Projection and Clustering by Simulated Annealing". *Pattern Recognition*, 22(2):213–220. 1989.

[152] J. Kleinberg. "An Impossibility Theorem for Clustering". *Proceeding Conference Advances in Neural Information Processing Systems*, 15:463–470. 2002.

[153] Teuvo Kohonen. "Self-Organized Formation of Topologically Correct Feature Maps". *Biological Cybernetics*, 43(1):59–69. 1982.

[154] Hans-Peter Kriegel, Peer Kröger, Matthias Renz, and Sebastian Wurst. "A Generic Framework for Efficient Subspace Clustering of High-Dimensional Data." In "Data Mining, Fifth IEEE International Conference on", IEEE Computer Society, pp. 250–257. 2005.

[155] Peer Kröger, Hans-Peter Kriegel, and Karin Kailing. "Density-Connected Subspace Clustering for High-Dimensional Data." In Michael W. Berry, Umeshwar Dayal, Chandrika Kamath, and David B. Skillicorn, editors, "SDM", SIAM, vol. 4. 2004.

[156] Joseph B. Kruskal. "On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem". *Proceedings of the American Mathematical society*, 7(1):48–50. 1956.

[157] N. Kumar and R. S. Joshi. "Data Clustering Using Artificial Neural Networks". In "Proceedings of National Conference on Challenges & Opportunities in Information Technology (COIT-2007)", pp. 197–200. 2007.

[158] K. W. Lau, H. Yin, and S. Hubbard. "Kernel Self-Organising Maps for Classification". *Neurocomputing*, 69(16-18):2033 – 2040. doi:http://dx.doi.org/10.1016/j.neucom.2005.10.003. 2006.

[159] Laura Lazzeroni and Art Owen. "Plaid Models for Gene Expression Data". *Statistica Sinica*, 12(1):61–86. 2000.

[160] Tao Li and Chris H. Q. Ding. "Weighted Consensus Clustering". In "SDM'08", pp. 798–809. 2008.

[161] Wei Li and Andrew McCallum. "Pachinko Allocation: DAG-Structured Mixture Models of Topic Correlations". In "Proceedings of the 23rd International Conference on Machine Learning", ACM, New York, NY, USA, ICML '06, pp. 577–584. doi:10.1145/1143844.1143917. 2006.

[162] Wei-keng Liao, Ying Liu, and Alok Choudhary. "A Grid-Based Clustering Algorithm Using Adaptive Mesh Refinement". In "7th Workshop on Mining Scientific and Engineering Datasets of SIAM International Conference on Data Mining", 2004.

[163] Ricardo Linden. "Clustering Techniques". *Revista de Sistemas de Informação da FSMA*, 4:18–36. 2009.

[164] Jingwei Liu and Meizhi Xu. "Kernelized Fuzzy Attribute C-Means Clustering Algorithm." *Fuzzy Sets and Systems*, 159(18):2428–2445. 2008.

[165] Huilan Luo, Furong Jing, and Xiaobing Xie. "Combining Multiple Clusterings using Information Theory based Genetic Algorithm". In "2006 International Conference on Computational Intelligence and Security", vol. 1, pp. 84–89. doi:10.1109/ICCIAS.2006.294095. 2006.

[166] P. C. H. Ma, K. C. C. Chan, Xin Yao, and D. K. Y. Chiu. "An Evolutionary Clustering Algorithm for Gene Expression Microarray Data Analysis". *Evolutionary Computation, IEEE Transactions on*, 10(3):296 – 314. doi:10.1109/TEVC.2005.859371. June 2006.

[167] J. B. MacQueen. "Some Methods for Classification and Analysis of Multivariate Observations". *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability. University of California Press.*, 1(14):281–297. 1967.

[168] Sara C. Madeira and Arlindo L. Oliveira. "Biclustering Algorithms for Biological Data Analysis: A Survey". *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, 1(1):24–45. doi:http://dx.doi.org/10.1109/TCBB.2004.2. 2004.

[169] S. W. Mahfoud. *Niching Methods for Genetic Algorithms*. Ph.D. thesis, University of Illinois at Urbana-Champaign. 1995.

[170] Jianchang Mao and Anil K. Jain. "A Self-Organizing Network for Hyperellipsoidal Clustering (HEC)". *IEEE Transactions on Neural Networks*, 7(1):16–29. 1996.

[171] Thomas M. Martinetz and Klaus J. Schulten. "A "Neural Gas" Network Learns Topologies". In Teuvo Kohonen, Kai Mäkisara, Olli Simula, and Jari Kangas,

editors, “Proceedings of the International Conference on Artificial Neural Networks (Espoo, Finland)”, Amsterdam; New York: North-Holland, pp. 397–402. 1991.

[172] G. J. McLachlan and K. E. Basford. *Mixture Models: Inference and Applications to Clustering*. Marcel Dekker Inc, New York / Basel. 1988.

[173] Geoffrey J. McLachlan, Shu-Kay Ng, and Richard Bean. “Robust Cluster Analysis via Mixture Models”. *Austrian Journal of Statistics*, 35(2):157–174. 2006.

[174] Marina Meila and Jianbo Shi. “Learning Segmentation by Random Walks”. In “In Advances in Neural Information Processing Systems”, MIT Press, pp. 873–879. 2001.

[175] Boriana L. Milenova and Marcos M. Campos. “O-cluster: Scalable Clustering of Large High Dimensional Data Sets”. In “Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on”, IEEE, pp. 290–297. 2002.

[176] Gabriela Moise, Jörg Sander, and Martin Ester. “P3C: A Robust Projected Clustering Algorithm”. In “ICDM”, IEEE Computer Society, pp. 414–425. 2006.

[177] Gabriela Moise, Jörg Sander, and Martin Ester. “Robust Projected Clustering”. *Knowledge and Information Systems*, 14(3):273–298. 2008.

[178] B. Moore. “ART1 and Pattern Clustering”. *Proceedings Connectionist Models Summer School*, pp. 174–185. 1989.

[179] Tadeusz Morzy, Marek Wojciechowski, and Maciej Zakrzewicz. “Pattern-Oriented Hierarchical Clustering”. In “Proceedings of the third East-European Symposium on Advances in Databases and Information Systems - ADBIS-99, Slovenia, LNCS 1691”, pp. 179–190. 1999.

[180] T. S. Motzkin and E. G. Straus. “Maxima for Graphs and a New Proof of a Theorem of Turán”. *Canadian Journal of Mathematics*, 17(4):533–540. 1965.

[181] Anirban Mukhopadhyay and Ujjwal Maulik. “A Multiobjective Approach to MR Brain Image Segmentation.” *Applied Soft Computing*, 11(1):872–880. 2011.

[182] Fionn Murtagh. “A Survey of Recent Advances in Hierarchical Clustering Algorithms”. *The Computer Journal*, 26(4):354–359. 1983.

[183] Megha Nangia. “Partitional Clustering”. *ACM student chapter, SIGKDD Presentation*. February 2012.

[184] S. B. Needleman and C. D. Wunsch. “A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of Two Proteins”. *Journal of Molecular Biology*, 48(3):443–453. Mar. 1970.

[185] Jaroslav Nešetřil, Eva Milková, and Helena Nešetřilová. “Otakar Boruvka on Minimum Spanning Tree Problem Translation of Both the 1926 Papers, Comments, History”. *Discrete Mathematics*, 233(1):3–36. 2001.

[186] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. “On Spectral Clustering: Analysis and an Algorithm”. In “Advances in Neural Information Processing Systems”, MIT Press, pp. 849–856. 2001.

[187] Raymond T. Ng and Jiawei Han. “Efficient and Effective Clustering Methods for Spatial Data Mining”. In “Proceedings of the 20th International Conference on Very Large Data Bases”, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, VLDB ’94, pp. 144–155. 1994.

[188] Tim Oates, Laura Firoiu, and Paul R. Cohen. “Using Dynamic Time Warping to Bootstrap HMM-Based Clustering of Time Series.” In Ron Sun and C. Lee Giles, editors, “Sequence Learning”, Springer, vol. 1828 of *Lecture Notes in Computer Science*, pp. 35–52. 2001.

[189] Atsuyuki Okabe and Kokichi Sugihara. *Spatial Analysis Along Networks: Statistical and Computational Methods*. John Wiley & Sons. 2012.

[190] Niina Päivinen. “Clustering with a Minimum Spanning Tree of Scale-Free-Like Structure”. *Pattern Recognition Letters*, 26(7):921–930. 2005.

[191] Massimiliano Pavan and Marcello Pelillo. “Dominant Sets and Pairwise Clustering”. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(1):167–172. doi:10.1109/TPAMI.2007.10. Jan. 2007.

[192] Z. Pawlak. *Rough Sets: Theoretical Aspects of Reasoning About Data*. Mathematics and Its Applications. Soviet Series. Springer Netherlands. 1991.

[193] Karl Pearson. “LIII. On Lines and Planes of Closest Fit to Systems of Points in Space”. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572. 1901.

[194] W. R. Pearson and D. J. Lipman. “Improved Tools for Biological Sequence Comparison”. *Proceedings of the National Academy of Sciences of the USA*, 85(8):2444–2448. doi:10.1073/pnas.85.8.2444. Apr. 1988.

[195] Dan Pelleg and Andrew Moore. “Accelerating Exact K-Means Algorithms with Geometric Reasoning”. In “Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining”, ACM, pp. 277–281. 1999.

[196] Dan Pelleg and Andrew W. Moore. “X-Means: Extending K-Means with Efficient Estimation of the Number of Clusters”. In “Proceedings of the Seventeenth International Conference on Machine Learning”, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, ICML ’00, p. 727–734. 2000.

[197] Harun Pirim, Dilip Gautam, Tanmay Bhowmik, Andy D. Perkins, and Burak Ekioglu. “Performance of an Ensemble Clustering Algorithm on Biological Data Sets”. *Mathematical and Computational Applications*, 16(1):87–96. 2011.

[198] Clara Pizzuti and Domenico Talia. "P-AutoClass: Scalable Parallel Clustering for Mining Large Data Sets". *Knowledge and Data Engineering, IEEE Transactions on*, 15(3):629–641. doi:10.1109/TKDE.2003.1198395. Mar. 2003.

[199] J. Platt. *Advances in Kernel Methods: Support Vector Learning*, MIT press, Cambridge, MA, chap. Fast training of SVMs using sequential minimal optimization, pp. 185–208. 1999.

[200] R. C. Prim. "Shortest Connection Networks and Some Generalizations". *Bell System Technology Journal*, 36(6):1389–1401. 1957.

[201] Cecilia Magdalena Procopiuc, Michael Jones, Pankaj K. Agarwal, and T. M. Murali. "A Monte Carlo Algorithm for Fast Projective Clustering." In Michael J. Franklin, Bongki Moon, and Anastassia Ailamaki, editors, "SIGMOD Conference", ACM, pp. 418–427. 2002.

[202] Marco Ramoni, Paola Sebastiani, and Paul R. Cohen. "Bayesian Clustering by Dynamics." *Machine Learning*, 47(1):91–121. 2002.

[203] Rocío Romero-Záliz, Cristina Rubio-Escudero, J. P. Cobb, Francisco Herrera, Oscar Cordón, and Igor Zwir. "A Multiobjective Evolutionary Conceptual Clustering Methodology for Gene Annotation Within Structural Databases: A Case of Study on the Gene Ontology Database." *Evolutionary Computation, IEEE Transactions on*, 12(6):679–701. 2008.

[204] Carlos Ruiz, Myra Spiliopoulou, and Ernestina Menasalvas. "C-DBSCAN: Density-Based Clustering with Constraints". In "Rough Sets, Fuzzy Sets, Data Mining and Granular Computing", Springer, pp. 216–223. 2007.

[205] Jörg Sander, Martin Ester, Hans-Peter Kriegel, and Xiaowei Xu. "Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and Its Applications". *Data Mining and Knowledge Discovery*, 2(2):169–194. doi:10.1023/A:1009745219419. Jun. 1998.

[206] E. Schikuta. "Grid-Clustering: An Efficient Hierarchical Clustering Method for Very Large Data Sets". *Pattern Recognition, 1996., Proceedings of the 13th International Conference on*, 2:101–105. doi: 10.1109/ICPR.1996.546732. Aug. 1996.

[207] Erich Schikuta and Martin Erhart. "The BANG-Clustering System: Grid-Based Data Analysis". In "Advances in Intelligent Data Analysis Reasoning about Data", Springer, pp. 513–524. 1997.

[208] Bernhard Schölkopf, Chris Burges, and Vladimir Vapnik. "Incorporating Invariances in Support Vector Learning Machines". In "Artificial Neural Networks ICANN 96", Springer, pp. 47–52. 1996.

[209] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. "Nonlinear Component Analysis as a Kernel Eigenvalue Problem". *Neural Computation*, 10(5):1299–1319. doi:10.1162/089976698300017467. Jul. 1998.

[210] A. J. Scott and Michael J Symons. "Clustering Methods Based on Likelihood Ratio Criteria". *Biometrics*, pp. 387–397. 1971.

[211] Shokri Z. Selim and K. Alsultan. "A Simulated Annealing Algorithm for the Clustering Problem." *Pattern Recognition*, 24(10):1003–1008. 1991.

[212] Gholamhosein Sheikholeslami, Surojit Chatterjee, and Aidong Zhang. "WaveCluster: A Multi-Resolution Clustering Approach for Very Large Spatial Databases". In "Proceedings of the 24rd International Conference on Very Large Data Bases", Morgan Kaufmann Publishers Inc., pp. 428–439. 1998.

[213] Weiguo Sheng, Allan Tucker, and Xiaohui Liu. *Clustering with Niching Genetic K-Means Algorithm*, Springer Berlin / Heidelberg, pp. 162–173. Lecture Notes in Computer Science, Genetic and Evolutionary Computation – GECCO 2004. 2004.

[214] Jianbo Shi and Jitendra Malik. "Normalized Cuts and Image Segmentation". *IEEE. Reprinted from IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905. August 2000.

[215] R. Sibson. "SLINK: An Optimally Efficient Algorithm for the Single-Link Cluster Method". *The Computer Journal (British Computer Society)*, 16(1):30–34. 1973.

[216] Slowmo. "Example 2D Space with Subspace Clusters". URL http://commons.wikimedia.org/wiki/File:SubspaceClustering.png. Accessed Oct. 2015. 2010.

[217] Temple F. Smith and Michael S. Waterman. "New Stratigraphic Correlation Techniques". *The Journal of Geology*, 88(4):451–457. Jul. 1980.

[218] Padhraic Smyth. "Clustering Sequences with Hidden Markov Models". In Michael Mozer, Michael I. Jordan, and Thomas Petsche, editors, "Advances in Neural Information Processing", MIT Press, pp. 648–654. 1996.

[219] Padhraic Smyth. "Probabilistic Model-Based Clustering of Multivariate and Sequential Data". In "Proceedings of Artificial Intelligence and Statistics", Morgan Kaufmann, pp. 299–304. 1999.

[220] P. Sneath and R. Sokal. "Numerical Taxonomy". In "Numerical Taxonomy", W. H. Freeman and Company. 1973.

[221] Peter H. A. Sneath. "The Application of Computers to Taxonomy". *Journal of general microbiology*, 17(1):201–226. 1957.

[222] R. R. Sokal and C. D. Michener. "A Statistical Method for Evaluating Systematic Relationships". *The University of Kansas Scientific Bulletin*, 38:1409–1438. 1958.

[223] T. Sorensen. "A Method of Establishing Groups of Equal Amplitude in Plant Sociology Based on Similarity of

Species Content and its Application to Analyzes of the Vegetation on Danish Commons". *Biologiske Skrifter*, 5:1–34. 1948.

[224] H. Spath. "Cluster Analysis Algorithms". In "Cluster Analysis Algorithms", West Sussex, Ellis Horwood Limited. 1980.

[225] Michael Steinbach, George Karypis, and Vipin Kumar. "Efficient Algorithms for Creating Product Catalogs". Tech. rep., DTIC Document. 2000.

[226] H. Steinhaus. "Sur la Division des corp Materiels en Parties". *Bulletin of Acad. Polon. Sci.*, 4(12):801–804. 1956.

[227] Alexander Strehl and Joydeep Ghosh. "Cluster Ensembles – A Knowledge Reuse Framework for Combining Multiple Partitions". *Journal of Machine Learning Research*, 3:583–617. 2002.

[228] Ron Sun and C. Lee Giles. *Sequence Learning: Paradigms, Algorithms, and Applications*, vol. 1828. Springer. 2001.

[229] Amos Tanay, Roded Sharan, Martin Kupiec, and Ron Shamir. "Revealing Modularity and Organization in the Yeast Molecular Network by Integrated Analysis of Highly Heterogeneous Genomewide Data". *PNAS*, 101(9):2981–2986. 2004.

[230] Chun Tang, Li Zhang, Aidong Zhang, and Murali Ramanathan. "Interrelated Two-way Clustering: An Unsupervised Approach for Gene Expression Data Analysis". In "Bioinformatics and Bioengineering Conference, 2001. Proceedings of the IEEE 2nd International Symposium on", pp. 41–48. 2001.

[231] S. Theodoridis and K. Koutroumbas. *Pattern Recognition, Third Edition*. Academic Press, Inc., Orlando, FL, USA. 2006.

[232] S. C. A. Thomopoulos, D. K. Bougoulias, and Chin-Der Wann. "Dignet: An Unsupervised-Learning Clustering Algorithm for Clustering and Data Fusion". *Aerospace and Electronic Systems, IEEE Transactions on*, 31(1):21–38. doi:10.1109/7.366289. Jan 1995.

[233] Alexander P. Topchy, Anil K. Jain, and William F. Punch. "Combining Multiple Weak Clusterings." In "Proceedings of the IEEE International Conference on Data Mining", IEEE Computer Society, pp. 331–338. 2003.

[234] Alexander P. Topchy, Anil K. Jain, and William F. Punch. "A Mixture Model for Clustering Ensembles." In "Proceedings SIAM International Conference on Data Mining", SIAM. 2004.

[235] Sandro Vega-Pons and José Ruiz-Shulcloper. "A Survey of Clustering Ensemble Algorithms". *International Journal of Pattern Recognition and Artificial Intelligence*, 25(3):337–372. 2011.

[236] Boaz Vigdor and Boaz Lerner. "The Bayesian ARTMAP". *IEEE Transactions on Neural Networks*, 18(6):1628–1644. 2007.

[237] Michail Vlachos, Jessica Lin, Eamonn Keogh, and Dimitrios Gunopulos. "A Wavelet-Based Anytime Algorithm for K-Means Clustering of Time Series". In "In Proceedings Workshop on Clustering High Dimensionality Data and Its Applications", pp. 23–30. 2003.

[238] K. Wagstaff and C. Cardie. "Clustering with Instance-level Constraints". *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000)*, pp. 1103–1110. 2000.

[239] Haiying Wang, Huiru Zheng, and Francisco Azuaje. "Poisson-Based Self-Organizing Feature Maps and Hierarchical Clustering for Serial Analysis of Gene Expression Data". *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 4(2):163–175. doi:http://0-dx.doi.org.innopac.library.unr.edu/10.1109/TCBB.2007.070204. 2007.

[240] Hongjun Wang, Hanhuai Shan, and Arindam Banerjee. "Bayesian Cluster Ensembles". *Statistical Analysis and Data Mining*, 4(1):54–70. doi:10.1002/sam.10098. 2011.

[241] Wei Wang, Jiong Yang, and Richard R. Muntz. "STING: A Statistical Information Grid Approach to Spatial Data Mining". In Matthias Jarke, Michael J. Carey, Klaus R. Dittrich, Frederick H. Lochovsky, Pericles Loucopoulos, and Manfred A. Jeusfeld, editors, "VLDB'97, Proceedings of 23rd International Conference on Very Large Data Bases, August 25-29, 1997, Athens, Greece", Morgan Kaufmann, pp. 186–195. 1997.

[242] Wei Wang, Jiong Yang, and Richard R. Muntz. "STING+: An Approach to Active Spatial Data Mining." In Masaru Kitsuregawa, Michael P. Papazoglou, and Calton Pu, editors, "Data Engineering, 1999. Proceedings., 15th International Conference on", IEEE Computer Society, pp. 116–125. 1999.

[243] Xiang Wang and Ian Davidson. "Active Spectral Clustering". In "Data Mining (ICDM), 2010 IEEE 10th International Conference on", IEEE, pp. 561–568. 2010.

[244] Chin-Der Wann and Stelios C. A. Thomopoulos. "A Comparative Study of Self-organizing Clustering Algorithms Dignet and ART2." *Neural Networks*, 10(4):737–753. 1997.

[245] Joe H. Ward Jr. "Hierarchical Grouping to Optimize an Objective Function". *Journal of the American Statistical Association*, 58(301):236–244. 1963.

[246] Max Welling. "Learning in Markov Random Fields with Contrastive Free Energies". In "Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics", pp. 397–404. 2005.

[247] O. Wildi. *Data Analysis in Vegetation Ecology*. John Wiley & Sons. 2010.

[248] James R. Williamson. "Gaussian ARTMAP: A Neural Network for Fast Incremental Learning of Noisy

Multidimensional Maps." *Neural Networks*, 9(5):881–897. 1996.

[249] John H. Wolfe. "Pattern Clustering by Multivariate Mixture Analysis". *Multivariate Behavioral Research*, 5(3):329–350. 1970.

[250] Donald C. Wunsch, Thomas P. Caudell, C. David Capps, Robert J. Marks, and R. Aaron Falk. "An Optoelectronic Implementation of the Adaptive Resonance Neural Network." *IEEE Transactions on Neural Networks*, 4(4):673–684. 1993.

[251] Chen Xiaoyun, Chen Yi, Qi Xiaoli, Yue Min, and He Yanshan. "PGMCLU: A Novel Parallel Grid-Based Clustering Algorithm for Multi-Density Datasets". In "Web Society, 2009. SWS'09. 1st IEEE Symposium on", IEEE, pp. 166–171. 2009.

[252] Eric P. Xing, Andrew Y. Ng, Michael I. Jordan, and Stuart Russell. "Distance Metric Learning with Application to Clustering with Side-Information". *Advances in Neural Information Processing Systems*, pp. 521–528. 2003.

[253] Yimin Xiong and Dit-Yan Yeung. "Time Series Clustering with ARMA Mixtures". *Pattern Recognition*, 37(8):1675–1689. 2004.

[254] Rui Xu and D. Wunsch. "Survey of Clustering Algorithms". *Neural Networks, IEEE Transactions on*, 16(3):645 –678. doi:10.1109/TNN.2005.845141. may 2005.

[255] Rui Xu and D. Wunsch. *Clustering*. IEEE/Wiley. 2009.

[256] Rui Xu and D. C. Wunsch. "Clustering Algorithms in Biomedical Research: A Review". *Biomedical Engineering, IEEE Reviews in*, 3:120. 2010.

[257] Xiaowei Xu, Martin Ester, Hans-Peter Kriegel, and Jörg Sander. "A Distribution-Based Clustering Algorithm for Mining in Large Spatial Databases". In "Proceedings of the Fourteenth International Conference on Data Engineering", IEEE Computer Society, Washington, DC, USA, ICDE '98, pp. 324–331. 1998.

[258] Xiaowei Xu, Nurcan Yuruk, Zhidan Feng, and Thomas A. J. Schweiger. "SCAN: A Structural Clustering Algorithm for Networks". In "Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining", ACM, pp. 824–833. 2007.

[259] Ronald R. Yager. "Intelligent Control of the Hierarchical Agglomerative Clustering Process". *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 30(6):835–845. 2000.

[260] Donghui Yan, Ling Huang, and Michael I Jordan. "Fast Approximate SpectralCclustering". In "Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining", ACM, pp. 907–916. 2009.

[261] Zhao Yanchang and Song Junde. "GDILC: a Grid-Based Density-Isoline Clustering Algorithm". In "2001 International Conferences on Info-Tech and Info-Net. Proceedings (Cat. No.01EX479)", vol. 3, pp. 140–145. doi:10.1109/ICII.2001.983048. 2001.

[262] Qiang Yang and Xindong Wu. "10 Challenging Problems in Data Mining Research". *International Journal of Information Technology and Decision Making (IJITDM)*, 05(04):597–604. 2006.

[263] James Yolkowski. "The Clustering Illusion". URL http://mathlair.allfunandgames.ca/clustering.php. Accessed Oct. 2015. 2014.

[264] Stella X. Yu and Jianbo Shi. "Multiclass Spectral Clustering". In "Proceedings of the Ninth IEEE International Conference on Computer Vision - Volume 2", IEEE Computer Society, Washington, DC, USA, ICCV '03, pp. 313–319. 2003.

[265] Stefanos Zafeiriou and Nikolaos A. Laskaris. "On the Improvement of Support Vector Techniques for Clustering by Means of Whitening Transform." *Signal Processing Letters, IEEE*, 15:198–201. 2008.

[266] Charles T Zahn. "Graph-Theoretical Methods for Detecting and Describing Gestalt Clusters". *Computers, IEEE Transactions on*, 100(1):68–86. 1971.

[267] B. Zhang. "Generalized K-Harmonic means – Dynamic Weighting of Data in Un-supervised Learning". *Proceedings of the 1st SIAM ICDM, Chicago, IL, USA*, pp. 1–13. 2001.

[268] Dao-Qiang Zhang and Song-Can Chen. "A Novel Kernelized Fuzzy C-means Algorithm with Application in Medical Image Segmentation". *Artificial Intelligence in Medicine*, 32(1):37–50. 2004.

[269] Tian Zhang, Raghu Ramakrishnan, and Miron Livny. "BIRCH: An Efficient Data Clustering Method for Very Large Databases". *SIGMOD Record*, 25(2):103–114. doi:10.1145/235968.233324. Jun. 1996.

[270] Shangming Zhou and John Q. Gan. "An Unsupervised Kernel Based Fuzzy C-means Clustering Algorithm with Kernel Normalisation". *International Journal of Computational Intelligence and Applications*, 04(04):355–373. doi:10.1142/S1469026804001379. 2004.

[271] H. Zhuang, Y. Huang, K. Palaniappan, and Y. Zhao. "Gaussian Mixture Density Modeling, Decomposition, and Applications". *IEEE Trans. Image Processing*, 5(9):1293–1302. Sep. 1996.

Yan Yan received her MS degree in Computer Science and Engineering from the University of Nevada, Reno. After this she worked in industry as a Software Engineer. She then came back to the University of Nevada, Reno and received her Ph.D in Computer Science and Engineering in 2019.

She is currently working as a Software Engineering Consultant. Her research interests are in Cancer Subtyping and Clustering of large data sets.

Frederick C. Harris Jr. received his BS and MS degrees in Mathematics and Educational Administration from Bob Jones University, Greenville, SC, USA in 1986 and 1988 respectively. He then went on and received his MS and Ph.D. degrees in Computer Science from Clemson University, Clemson, SC, USA in 1991 and 1994 respectively.

He is currently a Professor in the Department of Computer Science and Engineering and the Director of the High Performance Computation and Visualization Lab at the University of Nevada, Reno, USA. He is also the Nevada State EPSCoR Director and the Project Director for Nevada NSF EPSCoR. He has published more than 290 peer-reviewed journal and conference papers along with several book chapters. He has had 14 PhD students and 80 MS Thesis students finish under his supervision. His research interests are in parallel computation, simulation, computer graphics, and virtual reality. He is also a Senior Member of the ACM, and a Senior Member of the International Society for Computers and their Applications (ISCA).