A Survey of Constrained Clustering

Derya Dinler and Mustafa Kemal Tural

Abstract Traditional data mining methods for clustering only use unlabeled data objects as input. The aim of such methods is to find a partition of these unlabeled data objects in order to discover the underlying structure of the data. In some cases, there may be some prior knowledge about the data in the form of (a few number of) labels or constraints. Performing traditional clustering methods by ignoring the prior knowledge may result in extracting irrelevant information for the user. Constrained clustering, i.e., clustering with side information or semi-supervised clustering, addresses this problem by incorporating prior knowledge into the clustering process to discover relevant information from the data. In this chapter, a survey of advances in the area of constrained clustering will be presented. Different types of prior knowledge considered in the literature, and clustering approaches that make use of this prior knowledge will be reviewed.

1 Introduction

Data mining, i.e., science of extracting new and useful information from large data sets, has gained a lot of attention recently among scientists and in society as a whole due to the wide availability of huge data sets and the increased power of computers. According to [47], the amount of data produced worldwide during the year 2002 and stored on paper, film, magnetic media, and optical devises was estimated to be between 3,400,000 and 5,600,000 terabytes. These numbers are about twice the size of the data produced during the year 1999 [46]. The amount of data that has exploded year over year introduced new challenges and led to continued innovation in data storage and data mining techniques. A fundamental categorization in data mining techniques is:

1. Supervised learning
2. Unsupervised learning.

D. Dinler (✉) • M.K. Tural
Industrial Engineering Department, Middle East Technical University, 06800 Inonu Bulvari, Ankara, Turkey
e-mail: dinler@metu.edu.tr; tural@metu.edu.tr

M.E. Celebi, K. Aydin (eds.), *Unsupervised Learning Algorithms*,
DOI 10.1007/978-3-319-24211-8_9

In the former one, the data is split into training and test data. Observations in the training data are provided with desired output values (labels) which are used to construct a function that predicts the output values of the observations in the test data. Classification and regression are good examples of supervised learning problems. Typically, supervised learning algorithms require a large training data set, which is not always available or may be very costly to acquire. In the latter one, however, there is no learning from cases (no training data), but instead one tries to find intrinsic "natural" structures in unlabeled data. Clustering is one of the most famous unsupervised learning problems. As unsupervised learning methods are completely unguided, the structure extracted from the data may not always be relevant to the analyst or user.

In some cases, the data analyst may have a priori (domain) knowledge about the underlying structure of the data. With completely unlabeled data or a few number of labeled data, the supervised learning techniques may not be suitable. One way of learning from such data would be to completely ignore the prior knowledge and apply unsupervised learning methods. This may, however, result in extracting irrelevant structure from the data. Another way of learning from such data is by means of semi-supervised learning which incorporates the prior knowledge to the learning process to improve the quality of the result. Note that this type of learning is at the intersection of (completely) supervised and unsupervised learning.

In the last two decades, semi-supervised clustering which is also known as constrained clustering or clustering with side information has attracted several researchers as it has been observed that even with a small amount of prior knowledge, the clustering performance can be improved and running time of the process can be decreased significantly [62]. A survey of advances in the area of constrained clustering will be presented in this chapter. The survey will cover different types of prior knowledge considered in the literature, and clustering approaches that make use of this knowledge. In Sect. 2, we introduce unsupervised clustering and review some of the widely used unsupervised clustering algorithms which form the basis of several constrained clustering algorithms. In Sect. 3, we provide a review of the constrained clustering literature. Finally, we conclude the chapter with Sect. 4.

2 Unsupervised Clustering

Clustering can be defined as partitioning of unlabeled observations into groups (clusters) so that the observations in a group will be similar to each other with respect to some similarity criteria (i.e., high intra-cluster similarity) and different from the observations in other groups (i.e., high inter-cluster dissimilarity). It aims to find meaningful or useful clusters which are defined according to the goals of the data analyst. In some clustering problems, the number of clusters to be formed, k, is given as a parameter. In some others, however, k may be unknown. Clustering has been employed in many disciplines such as statistics, biology, marketing, and medicine. It has mainly been used for three purposes:

- to discover the underlying structure of the data,
- to find natural classification of data objects by identifying similarities among them, and
- to organize and summarize the data [38].

In some cases, clustering can be used as a preprocessing tool for other tasks such as regression, principal component analysis, and association analysis. See, for instance, [23] in which the authors used clustering techniques to reduce complexity in solving automated planning problems.

Typically, clustering problems are considered as optimization problems with various objectives and solved by exact algorithms, approximation methods, or heuristics [3]. Clustering methods can be broadly divided into two categories: hierarchical and partitional methods. Hierarchical clustering algorithms build nested clusters by merging (agglomerative methods) or splitting (divisive methods) clusters successively. The hierarchy of clusters is represented with a tree known as dendrogram. Partitional clustering algorithms construct a one-level clustering of the observations without a nested structure. It should be noted that in some clustering algorithms all of the data is studied at once whereas in others data objects are taken into consideration one by one. Algorithms falling in the former category are known as the batch algorithms, whereas algorithms in the latter category are known as incremental algorithms. For online applications where the input arrives piece by piece, incremental algorithms are better suited.

Several different criteria have been used in cluster analysis. Among them, the minimum sum-of-squares is the most popular one.

2.1 *Minimum Sum-of-Squares Clustering*

Consider a data set consisting of n data objects in the d-dimensional Euclidean space, $X = \{x_i\}$, $i = 1, \ldots, n$, $x_i \in R^d$. The Minimum Sum-of-Squares Clustering (MSSC) problem is the problem of partitioning the objects into a set of a predefined number, say k, of clusters, $C = \{c_j\}, j = 1, \ldots, k$, so as to minimize the sum of squared Euclidean distances between the data objects and the centers of the clusters (representatives) they belong to. The problem can be formulated as follows:

$$\text{minimize} \sum_{j=1}^{k} \sum_{x_i \in c_j} \left\| x_i - \mu_j \right\|^2, \tag{1}$$

where μ_j represents the center of the jth cluster and $\{c_1, \ldots, c_k\}$ a partition of X.

Note that the center of a cluster may not coincide with any of the data objects belonging to that cluster. In the MSSC problem, a hard assignment of the data objects to clusters is made, i.e., every object is assigned to exactly one cluster. On the other hand, a soft assignment would assign for each object and each cluster a probability that the object belongs to the cluster.

The MSSC problem is known to be NP-hard in general [50] and therefore heuristic algorithms have been widely employed. K-means algorithm and its variants are among the most commonly used heuristics proposed for the MSSC problem.

2.1.1 K-Means Algorithm

K-means algorithm proposed by Macqueen in 1967 is the most famous partitional (batch) clustering algorithm [48]. Because of its easy to implement nature, simplicity, efficiency, and empirical success, its framework has also been commonly used in algorithms developed for constrained clustering problems. The steps of the K-means algorithm are based on two simple observations. First, if the centers of the clusters are known, then each data object is assigned to the cluster whose center is closest to the object. Second, if all the data objects belonging to a cluster are known, then the center of the cluster is computed by averaging all data objects in the cluster.

The basic steps of the K-means algorithm are given below.

Algorithm 1: K-means

```
Initialization: Start with initial cluster centers (seed), $\mu_j, j = 1, \ldots, k$.
repeat
    Assignment: Assign each data object $x_i$ to the closest cluster $j^*$, and let $x_i \in c_{j^*}$, where $j^* = \operatorname{argmin}_j \|x_i - \mu_j\|$.
    Update: Update the cluster centers by averaging the data objects assigned to them, i.e., $\mu_j = \frac{\sum_{x_i \in c_j} x_i}{|c_j|}$.
until Convergence is achieved.
return Partition $\{c_1, \ldots, c_k\}$ of $X$.
```

The algorithm finds a local minimum and the convergence can be slow. It may return some empty clusters. The final partition obtained with the K-means algorithm is highly depended on the initialization step. To overcome this problem, the K-means algorithm may be initialized with multiple different sets of initial centers and the solution with the minimum objective function value may be returned to the user. One way to initialize the K-means algorithm is by randomly choosing the initial centers. Several initialization heuristics have also been proposed with the aim of starting with a "good" set of initial centers, see, for example, [14, 15]. The runtime of a single iteration of the K-means algorithm is equal to $O(nkd)$.

Due to the known issues with the standard K-means algorithm, several variants of the algorithm have been proposed over the years [35, 36, 44].

2.2 Agglomerative Hierarchical Clustering

Traditional agglomerative hierarchical clustering algorithms begin with each data object in a separate cluster and in a progressive manner merge two clusters that are the closest to reduce the number of clusters by one until all the data objects are in a single cluster.

The steps of a basic agglomerative hierarchical clustering algorithm are as follows:

Algorithm 2: Agglomerative hierarchical clustering

```
Let $c_i = \{x_i\}$, for every $i = 1, 2, \ldots, n$ and let $C = \{c_1, c_2, \ldots, c_n\}$
for $k = n$ to 1 do
    $Dendrogram(k) = C$
    Let $(u, v) = \mathrm{argmin}_{i \neq j:\ c_i, c_j \in C}\ \rho\left(c_i, c_j\right)$
    $c_u = c_u \bigcup c_v$ and $C = C \setminus c_v$.
end for
```

Here $\rho\left(c_i, c_j\right)$ represents the distance between the clusters c_i and c_j and (u, v) the arguments of the closest two clusters. Distance between two clusters can be measured in several different ways. In single linkage clustering, the distance between two clusters is measured by the minimum distance between the data objects in the two clusters, i.e.,

$$\rho\left(c_i, c_j\right) = \min_{x \in c_i,\ y \in c_j} d(x, y), \tag{2}$$

where $d(x, y)$ represents the distance between two data objects x and y.

In complete linkage clustering, distance between clusters is measured by the two most distant objects rather than the closest ones. Average linkage clustering uses, on the other hand, the average distance between the objects in the two clusters. There are several other ways to measure distances between groups, see, for instance, [66].

The result of a hierarchical clustering algorithm is usually depicted as a tree known as dendrogram. The root node of the dendrogram represents all the data objects as a single cluster and the leaf nodes represent each object as a single cluster. The height of a node in a dendrogram is proportional to the value of the clustering criterion at this node. By cutting the dendrogram with a horizontal line at the appropriate level, any desired number of clusters can be obtained and the height of the line represents the value of the clustering criterion. For more on hierarchical clustering algorithms and complexity issues, we refer the reader to [27, 49, 53].

2.3 COBWEB

COBWEB is an incremental hierarchical clustering algorithm that was proposed by Fisher in 1987 [26]. It is used to obtain a hierarchy (classification tree) of data objects with nominal features.

In this algorithm, clusters are represented by d probability distributions (one distribution for each dimension or feature) instead of cluster centers. When a data object is available to cluster, algorithm performs one of the four operators:

1. Insert operator: insert the new object to an existing cluster (node),
2. Create operator: create a new cluster which will include the new object,
3. Merge operator: merge two of the existing children (clusters) of a node and place the new object in the resulting hierarchy, and
4. Split operator: split the objects of a cluster and place the new object in the resulting hierarchy.

While deciding which one of these operators will be performed, the algorithm uses the concept of category utility proposed by Gluck and Corter in 1985 [31] which takes both intra-cluster similarity and inter-cluster dissimilarity into account. Category utility of a hierarchical partition is defined as follows:

$$CU(c_1,\ldots,c_k)=\frac{1}{k}\left(\left(\sum_{j=1}^{k}\frac{|c_j|}{n}\sum_{d}\sum_{t\in V(d)}P(D_d=t|c_j)^2\right)-\sum_{d}\sum_{t\in V(d)}P(D_d=t)^2\right), \tag{3}$$

where $\{c_1,c_2,\ldots,c_n\}$ are the nodes in the hierarchy, D_d is one of the dimensions (features), and $V(d)$ is the set of discrete values that feature d can take. $P(D_d=t|c_j)$ is the probability that the dth feature of a data object in cluster c_j is equal to t and $P(D_d=t)$ is the probability that the dth feature of a data object is equal to t.

3 Constrained Clustering

In some cases, there may be available some prior information about the underlying cluster structure of the data in the form of constraints or a few labeled data [34]. Such prior knowledge generally arises from expert opinion, user feedback, or the needs of the problem owner [9]. Incorporating such background knowledge into the clustering process and hence allowing the user to guide the process in a manner toward a “better” partitioning of the data is known as constrained clustering and has been the subject of extensive research recently.

Constrained clustering has been used in several domains and applications. In gene clustering based on gene expression data obtained with DNA microarrays, databases of co-occurrence data have been used to generate constraints forcing that certain genes must be in the same cluster (must-link constraints) [25, 56, 64].

In some agricultural areas, each farmer cultivates a large number of small and dispersed land parcels. This has several disadvantages. A solution to this problem would be land consolidation which refers to the process that the farmers surrender their dispersed parcels in order to receive a more continuous equivalent land area. Land consolidation is clearly a clustering problem with several constraints. Firstly, neighboring parcels should be assigned to a farmer. Secondly, the total land area of a farmer should not change much after consolidation. Thirdly, the quality of soil of each farmer's land should also not change by too much. There are other constraints about the geometry of the land a farmer receives, such as it should not be a continuous long but narrow land [11].

In text clustering, the goal is to automatically categorize a large number of text documents into smaller and manageable groups (clusters) based on their content. The user may specify that some documents should be clustered into the same group as they have similar contents or have the same authors (must-link constraints) and/or some documents should be separated from each other due to the differences in their subjects (cannot-link constraints) [37].

For more applications of constrained clustering, we refer the reader to the survey written by Davidson and Basu [18].

Methods proposed for constrained clustering can be divided into three categories; namely search based (also known as constraint based), distance based (also known as similarity based), hybrid (also known as search and distance based) methods [71]. In search based methods, clustering algorithms are modified to incorporate the prior knowledge into the clustering task. In other words, the solution space to be searched is adjusted according to the constraints. Common techniques in search based methods are modifying the objective function by adding penalty terms for unsatisfied constraints, enforcing constraints to be satisfied and using prior knowledge to initialize clusters.

In distance based methods, an existing clustering method is generally used but the distance measure of the method is modified in accordance with the prior knowledge. The distance measure is adjusted in such a way that data objects that should be placed in the same cluster will be closer to each other while data objects that should be placed in different clusters will be farther away from each other. Hybrid methods integrate search based and distance based methods. They benefit from the advantages of both and generally perform better than the individual methods.

In constrained clustering, the prior knowledge can appear in the form of labeled data or constraints. In the former one, labels of some of the data objects are known, but the amount of available information (number of known labels) may not be sufficient to perform classification. In such a case one can perform clustering instead of classification incorporating the known labels in some way to the clustering procedure. Instead of having labeled data, there may be some constraints on the data objects or clusters. For example, there may be constraints on pairs of data objects in the form of pairwise must-link and cannot-link relations. This type of knowledge is generally more practical. Getting the true labels of the data objects may require too much effort or may be costly, while whether pairs of data objects belong to the same cluster or different clusters can be easily specified by an expert.

Constraints can be considered as hard or soft. If they are considered as hard constraints, they should be satisfied in the final partition of the data objects. A problem with using such constraints is that there may not be any partition at all satisfying all the constraints. In the soft constraint case, however, constraint violations are allowed with associated violation costs.

Constraints encountered in clustering problems can also be categorized as instance-level and cluster-level constraints. Instance-level constraints are pairwise must-link and cannot-link constraints on some pairs of data objects. A must-link constraint enforces that two objects must be placed in the same cluster. On the other hand, a cannot-link constraint enforces that two objects cannot be placed in the same cluster. Must-link constraints form an equivalence relation. The resulting equivalence classes are sometimes called as chunklets [58]. Each chunklet consists of data objects that are known to be in the same cluster. Hence the data can initially be partitioned into chunklets in the presence of must-link constraints.

It should be noted that prior information in the form of pairwise constraints is weaker than the prior information in the form of labeled data [40]. While labeled data can easily be transformed into pairwise constraints, the labels of data objects cannot be inferred from pairwise constraints.

In addition to instance-level constraints, there can be several other constraints in a constrained clustering problem. Balancing constraints would force the sizes of the clusters to be comparable. More generally, given data objects with associated positive weights, one may restrict the total weight of the data objects in each cluster (this is known as the capacitated clustering problem). In particular, if all the weights are one, then the size of each cluster is restricted. There could be constraints that put lower or upper bounds on the radii of the clusters. Given an initial clustering of the data, one may want to obtain a clustering that is "different" from the initial one. After this point, we will call all such constraints, i.e., all the constraints different from instance-level constraints, as cluster-level constraints.

3.1 Constrained Clustering with Labeled Data

3.1.1 Search Based Methods

Constrained clustering with limited number of labeled data can be considered as a multi-objective optimization problem with objectives of maximizing intra-cluster similarity, maximizing inter-cluster dissimilarity and minimizing cluster impurity which is a measure of the consistency between the partition and the prior knowledge (labels).

Basu et al. [4] develop two variants of the K-means algorithm for the clustering problem with labeled data; namely, Seeded-K-means and Constrained-K-means. In the Seeded-K-means algorithm, labeled data is used only in the initialization step of the K-means algorithm. The authors assume that at least one labeled data is available for each of the k clusters. By partitioning the labeled data, k groups are obtained and initial cluster centers are computed by taking the mean of only the labeled data objects in each cluster. Then, assignment and update steps are repeated until convergence. In the Seeded-K-means algorithm, labels are considered as soft constraints. This is because as the labels are not taken into account after the initialization, they are subject to change in later steps. In the Constrained-K-means algorithm, labeled data are considered both in the initialization step and in the assignment steps of the K-means algorithm. The initialization step of the Constrained-K-means is the same as that of the Seeded-K-means. In the Constrained-K-means, only unlabeled data objects are reassigned during the assignment step, but the clusters of labeled data objects are never changed after the initialization. So labels are considered as hard constraints here. When the initial labels are noise free, i.e., there are not inconsistencies in prior knowledge, Constrained-K-means can be preferred. If there is noise in initial labels, then Seeded-K-means may be helpful in reducing the effect of noise by allowing the clustering algorithm to change initial labels. Experimental studies show that both of the proposed algorithms perform better than the K-means algorithm. Also, the algorithms outperform COP-K-means proposed in [63] (see Sect. 3.2.1 for COP-K-means). In his dissertation, Basu provides more discussion on both of the algorithms. For more details, see [3].

For the same problem, Demiriz et al. [22] proposed an objective function that is a linear combination of cluster dispersion (intra-cluster similarity and/or inter-cluster dissimilarity) and cluster impurity. As a cluster impurity measure, the authors used the Gini Index (see, for instance, [13]). As a dispersion measure, the authors considered two alternatives:

1. mean square error: this is the objective function used in the MSSC problem divided by the number of data objects,
2. Davies-Bouldin index [21]: this is an index of dispersion taking both intra-cluster similarity and inter-cluster dissimilarity into account.

As the resulting objective functions are highly nonlinear with many local optima, the authors propose a genetic algorithm (GA), a nature-inspired metaheuristic, for the problem. In their algorithm, authors use operators (crossover, mutation, etc.) from GAlib which is a general purpose GA library instead of defining new GA operators. Besides operators used, another main aspect in GAs is the representation of a solution. In this study, authors used a $k \times d$ matrix in which each row contains the values of the d coordinates of the corresponding cluster center. Since this representation does not depend on n, the proposed GA is scalable to large data sets. Based on the experimental studies, it is shown that constrained clustering performs better than unsupervised clustering. When dispersion measures are investigated separately, using Davies-Bouldin index results in better performance than mean square error since it finds more compact and well-separated clusters.

3.1.2 Distance Based Methods

Constrained clustering problem with labeled data can also be handled in two stages instead of considering it as a multi-objective optimization problem. In the first stage, cluster impurity is considered. In this stage, data objects are transformed into a new space using the prior knowledge on hand. This transformation is done in such a way that objects having the same label will be closer to each other and objects with different labels will be farther away from each other in the new space. This first stage is called as distance metric learning. After the metric is learnt, cluster dispersion is considered in the second stage where traditional clustering algorithms are generally employed.

In [70], authors propose a parametric distance learning method for the clustering problem with labeled data. Data objects are transformed into a new Euclidean space in which the Euclidean distance between two data objects shows the dissimilarity between them. This transformation is performed in such a way that the square of the Euclidean distance between data objects provided with labels in the new space will be equal to the dissimilarity of them in the original space which is calculated as

$$\delta_{ij} = \begin{cases} 1 - e^{-\frac{\left\|x_i - x_j\right\|^2}{\beta}} & \text{if labels of } x_i \text{ and } x_j \text{ are the same },\\ 1 + e^{-\frac{\left\|x_i - x_j\right\|^2}{\beta}} & \text{otherwise,} \end{cases} \tag{4}$$

where $\beta > 0$ is a width parameter and δ_{ij} is a measure of dissimilarity between x_i and x_j in the original space. Such a transformation can be achieved by multidimensional scaling method. As this method may be intractable for large data sets, authors find a regression mapping from the original space to the new space using the prior knowledge (labeled data) to be able to map the unlabeled data objects into the new space. In the new space, the authors then used the K-means algorithm for clustering. Experimental studies show that the method they propose outperforms the traditional K-means algorithm.

3.2 Constrained Clustering with Instance-Level Constraints

3.2.1 Search Based Methods

Clustering with constraints can be considered as a multi-objective optimization problem as in the case with clustering in the presence of labeled data. In addition to cluster dispersion, a measure of constraint violation is used as another objective instead of using cluster impurity. If constraints are seen as hard constraints, then the constraint violation is set to zero.

Wagstaff and Cardie [62] study the clustering problem with must-link and cannot-link constraints. They consider the constraints as hard constraints. To enforce the constraints to be satisfied, authors propose a modified version of COBWEB, called as COP-COBWEB. The main difference of their algorithm from COBWEB is that COP-COBWEB is a partitional clustering algorithm that would return a single level of the dendrogram in COBWEB (level after the root node) rather than a hierarchy in the absence of constraints.

The steps of COP-COBWEB can be described as follows. When a new data object, x_i, arrives, first a must-link check is done. If there is a data object, x_j, that is in one of the existing clusters and that has to be in the same cluster with x_i, then x_i

is placed into the cluster where x_j belongs to and the category utility of the resulting partition is recorded (call it as CU_{ML}). Then, the split operator is applied to the cluster that x_i now belongs to and let CU_{S1} be the resulting category utility. If CU_{ML} is greater than CU_{S1}, then do not split and go to the next data object and otherwise do the splitting and go to the next data object.

If x_i is not inserted into a cluster in the must-link check step, then x_i is inserted in every possible cluster after a cannot-link check. The best two resulting clusters (in terms of the category utility) are recorded (call the resulting category utilities as CU_{I1} and CU_{I2}, respectively). These two clusters are merged if possible after a cannot-link check and the category utility of the resulting partition is recorded (let it be CU_M). Also with the create operator a new cluster is formed containing only x_i and the category utility of the resulting partition is recorded (say CU_C). Furthermore try splitting the best partition obtained with the insert operator that had the category utility of CU_{I1} and record the category utility of the resulting partition (say CU_{S2}). Choose the partition having the largest category utility out of CU_{I1}, CU_M, CU_C, and CU_{S2} and go to the next data object.

Authors experimentally show that including even a small number of constraints into the clustering process increases the clustering accuracy and decreases the running time since it reduces the solution space to be searched. Also, authors discover which constraint type is more beneficial for which type of clustering problem. For example, they observe experimentally that cannot-link constraints result in better accuracy for the data sets for which unsupervised clustering leads in less number of clusters than the true number of clusters. On the contrary, for the data sets for which unsupervised clustering leads in too many or true number of clusters, including must-link constraints is better than including cannot-link constraints.

In [63], authors consider the same problem considered in [62]. They modify the K-means algorithm. Authors take the transitive closure over constraints and used the whole derived set of constraints. For example, if x_i must-link to x_j which cannot-link to x_k, then we know that x_i cannot-link to x_k and this is added to the constraint set. Also if x_i must-link to x_j which must-link to x_k, then we know that x_i must-link to x_k as well and again this is added to the constraint set.

The algorithm proposed in this study, called as COP-K-means, uses the same initialization and update steps of the K-means algorithm but differs from it in the assignment step. In the assignment step, the data objects are considered one by one. For each data object x_i, the closest appropriate cluster is found and the object is placed in that cluster. A cluster is appropriate for a data object x_i, if

1. there exists a data object in the cluster that must-link to x_i, and
2. there does not exist any data object in the cluster that cannot-link to x_i.

Note that the assignment step of the algorithm is order-dependent. If the algorithm cannot find an appropriate cluster for any x_i, then the algorithm stops with no partition to return. The running time of a single iteration of the algorithm is $O(ndk + n^2)$ where n is the number of data objects, d is the number of features (dimension) each data object has, and k is the number of clusters. Authors compare performance of COP-K-means with COP-COBWEB developed in [62]. For data sets for which k is known, both of the algorithms result in similar amount of improvement in clustering accuracy over the original K-means. For data set for which k is unknown, COP-K-means is quite better than the original K-means in terms of clustering accuracy. Also, COP-K-means is better than the original K-means in determining the correct value of the number of clusters which is found by solving both algorithms for many times with different random initializations for different k values.

In her dissertation, Wagstaff provides more discussions on COP-COBWEB and COP-K-means [61]. In addition to these two algorithms which incorporate pairwise must-link and cannot-link constraints as hard constraints into the clustering process, Wagstaff proposes another algorithm which handles pairwise constraints as soft constraints. This algorithm is a modified version of the K-means and called as SCOP-K-means which incorporates violation costs into the objective function to be minimized in the assignment step. Initialization and update steps of the algorithm are the same as the K-means. In the assignment step, each data object is assigned to the cluster which contributes to the objective function the least. Experimental studies show that SCOP-K-means outperforms K-means in terms of accuracy.

Basu et al. in [6] propose PC-K-means, a modified version of the K-means, for clustering with pairwise must-link and cannot-link constraints. Similar to [61], authors consider the constraints as soft constraints. In the algorithm, prior knowledge is used in the initialization step and in the assignment step through a modified objective function. The objective function of the PC-K-means includes constraint violation terms in addition to the objective function of the MSSC problem. To initialize cluster centers, chunklets of size ≥ 2 are used. If the number of such chunklets is equal to k, the centers of them are used as initial centers. If the number of such chunklets is greater than k, then the largest k of them (in terms of size) are used in the initialization. If the number of such chunklets is less than k, then using all the chunklets less than k centers are initialized. The remaining centers are initialized randomly (with an exception). In the assignment step, each data object is assigned to the cluster minimizing the contribution to the objective function. This step is highly order-dependent. Authors considered random ordering of the objects in the assignment step in their computational experiments. Update step of the algorithm is the same as that of the original K-means. Authors prove that PC-K-means converges to a local optimum.

The authors also propose a two-phased method for actively selecting informative constraints. Their method uses farthest-first traversal scheme which aims to select a number of data objects that are far from each other. The active learning method proposed assumes that we can ask a given number of queries, say Q, to a noiseless oracle where the input is a pair of data objects and in return the oracle states whether there is a must-link or cannot-link constraint between the objects. In the first phase, called as the explore phase, the algorithm starts with selecting a data object at random and placing it in the first neighborhood. At the beginning of each explore step, we have a certain number of non-empty neighborhoods. At this point, the algorithm finds the data object that is farthest from the data objects so far traversed. Then by querying, the new data object is placed in an existing neighborhood if at any time a must-link constraint is returned by the oracle. Otherwise, i.e., the new object is cannot-linked to all existing neighborhoods, it is placed in a new neighborhood. The explore phase continues until k neighborhoods are formed or all Q queries are used up. In the latter case the number of neighborhoods formed may be less than k. In the former case, however, if there are still left some unused queries, the algorithm moves to the second phase, called as consolidate phase.

At the beginning of the consolidate phase we have k neighborhoods and certain number of queries to be used. The consolidate phase starts with estimating the centers of the neighborhoods. Then a data object that is not in any of the neighborhoods is selected at random and the distance between this new object and the centers of the neighborhoods are computed. Starting with the closest center, queries will be formed by taking a point from the neighborhood. In at most $(k-1)$ queries, the new object is placed in one of the neighborhoods.

Experimental studies show that the learning curve for the clustering process is significantly steeper with the proposed active learning method. In other words, number of constraints to be included in the clustering process to improve accuracy is much smaller with the proposed method in comparison with random selection of the constraints. It is also experimentally shown that the consolidate phase makes the clustering accuracy even better when compared with the active learning method that includes only the explore phase.

Basu [3] uses the concept of hidden Markov Random Field (HMRF) to solve the same problem considered in [6, 61]. An HMRF is a Markov Random Field in which some of the random variables are hidden (unobservable). A Markov Random Field consists of an undirected graph whose nodes represent random variables and a set of potential functions taking maximal cliques on the graph as input and returning non-negative real numbers. Edges between two random variables represent the dependencies between them. Generally, an HMRF model consists of an observable set of random variables, an unobservable set of random variables, an unobservable set of model parameters, and an observable set of constraints. For the clustering problem with constraints, these components are data objects to be clustered, labels of the data objects, cluster centers and pairwise constraints, respectively. Taking advantage of the resemblance between HMRF models and constrained clustering, the objective function used in this study is defined in the framework of HMRF. To solve the problem the author proposes an iterative K-means like algorithm, called as HMRF-K-means. Initialization step of the algorithm is the same with that of the PC-K-means [6]. After the initialization, the assignment step takes place. In the original K-means, assigning each data point to nearest cluster center minimizes the objective function. But in the framework of HMRF, cluster centers are dependent to each other and assigning data points to clusters is computationally intractable. In the assignment step, the author uses a greedy iterated conditional modes (ICM) approach proposed by Zhang et al. in 2001 [69]. Data points are assigned to clusters that minimize the contribution to the objective function in random order. After all objects are reassigned, the process of random reassigning is repeated until no change occurs in assignments for the two successive iterations. In addition to the ICM method, the author tries two global methods for the assignment step; namely, belief propagation and linear programming relaxation. These methods outperform greedy method when the amount of prior knowledge is small but as this amount increases global methods become computationally expensive and quality of greedy and global methods become comparable. So, greedy method is a wise choice for the assignment step in such cases. Based on the data points assigned to clusters, cluster means are updated. It is proved that the algorithm converges to a local optimum. Experimental studies show that the HMRF-K-means outperforms the original K-means.

Yu and Shi [68] consider the inclusion of must-link constraints in image segmentation and propose a graph-theoretic solution method. In the method, each data object (pixel) is considered as a node in a graph. The weight associated with each edge represents the similarity between the nodes connected by the edge. The authors then formulate the image segmentation problem with must-link constraints as a node partitioning problem in which edges within a partition (cluster) should have a high total weight (high similarity) and edges across partitions should have a low total weight (low similarity). Normalized cuts criterion proposed by Shi and Malik in 2000 [59] considers both of these objectives. By using this criterion the authors model the problem as a constrained optimization problem. Relaxing the discrete assignment constraints, authors showed that the problem resulted in a constrained eigenvalue problem. Authors then find the optimal solution of the constrained eigenvalue problem by eigenvalue decomposition from which a near global optimum solution is obtained for the discrete problem. The algorithm proposed works reasonably well with enough number of must-link constraints. On the other hand, when there are only a few such constraints, the final partition may not be as it is desired. In this case, considering the constraints as soft, the authors modify their algorithm that produces a better partitioning of the data.

In [20], Davidson and Ravi consider the clustering problem in the presence of must-link and cannot-link constraints. They propose a modified version of the K-means algorithm by adding constraint violation terms to the objective function to penalize violated constraints. Their algorithm called as the constrained vector quantization error (CVQE) algorithm therefore considers the instance-level constraints as soft constraints. Initialization step of the CVQE algorithm is the same as that of the K-means. In the assignment step, each pair of data objects that form a constraint are assigned to the clusters in such a way that the objective function is minimized. This takes $O(k^2)$ time per constraint. The remaining data objects are assigned as in the K-means. In the update step, the update rule is computed by taking the first derivative of the objective function and setting it to zero. Computational studies show that inclusion of constraints results in better clustering accuracy and faster convergence.

The advantage of algorithms considering constraints as soft over algorithms considering constraints as hard is that the former usually better handles noisy constraints. Algorithms in which the constraints are taken as hard may not find a feasible partition if there is noise with the constraints. While algorithms in [62, 63] may end up with no partition at all, the algorithms in [3, 6, 20, 61, 68] find a partition in every case with some amount of constraint violations.

The authors in [51] consider the use of instance-level constraints in an agglomerative hierarchical clustering algorithm. The authors first form the chunklets by using the must-link constraints. This initial partition of the data is used in the initial step of their algorithm. Rather than starting with clusters consisting of individual data objects, the authors start with the chunklets and hence making sure that the must-link constraints are all satisfied. As the initial clusters are never split, this makes sure that in the final partition, all of the must-link constraints will be satisfied (so they are considered as hard constraints). An important aspect of agglomerative hierarchical clustering algorithms is the way the distance is measured between the clusters. The distance function used by the authors consists of two terms: a cluster dispersion measure and a constraint violation term for the cannot-link constraints. They use two different dispersion measures; namely the centroid method and the Ward method. The centroid method measures the distance between two clusters as the distance between their centers. Given a cluster c_i and its center μ_i, let $E(c_i) = \sum_{x \in c_i} \|x - \mu_i\|^2$. In the Ward method, the distance between two clusters c_i and c_j is computed as $E(c_i \bigcup c_j) - E(c_i) - E(c_j)$. Starting with the chunklets, the algorithm proposed combines two closest clusters until the desired number of clusters is obtained. The authors also compare the results from the centroid method and the Ward method with and without the Gaussian kernel. They show computationally that their methods outperform the COP-K-means algorithm.

3.2.2 Distance Based Methods

Similar to the clustering problem with labeled data, the clustering problem with instance-level constraints can be handled in two stages. In the first stage, a new distance metric which brings must linked data objects closer and pushes cannot linked data objects apart is defined. A measure of dispersion as a function of this newly defined distance metric is then used in the second stage for clustering.

Klein et al. [40] state that just using the provided instance-level constraints in learning a distance metric that will help to reveal the desired partition of the data may result in missing some spatial information. If there is must-link between x_i and x_j then the data objects at the neighborhoods of x_i and x_j should also be put into the same cluster intuitively. The authors propose a distance measure reflecting both spatial information that instance-level constraints imply and the provided instance-level information. With this distance measure, it is aimed that must-linked objects will be close to each other while cannot-linked objects will be far from each other. Also, with the new distance measure, if x_i and x_j are close to each other, a data object close to x_i will be enforced to be close to x_j, and if x_i and x_j are far apart, a data object close to x_i will be imposed to be far from x_j. The authors use a complete-link hierarchical agglomerative clustering which takes a proximity matrix showing the pairwise proximities of data objects and merges two clusters based on their proximities. To incorporate must-link constraints and the spatial information implied by them into the proximity matrix, the authors run all-pairs-shortest-paths algorithm on the original proximity matrix. By this way, a new proximity matrix that is faithful to the original proximity matrix is obtained. Then, to include the cannot-link constraints, the distance between cannot-linked objects is set to the maximum distance between two data objects in the problem plus 1. The algorithm called as constrained complete-link, CCL, starts with all data objects in different clusters and merges clusters in order of proximities until one cluster including all data objects is obtained. Experimental studies show that CCL outperforms the COP-K-means [63] which does not consider spatial information implied by pairwise constraints.

To obtain a relevant partition of the data objects, distance metric used in the clustering process should reflect the relationships between data objects very well. If two data objects are stated as similar (i.e., there is a must-link constraint between them), the distance between these objects should be small while if they are stated as dissimilar (i.e., there is a cannot-link constraint between them), the distance between them should be high. To find a distance metric satisfying these relationships, Xing et al. [65] propose the following convex optimization model.

$$\begin{aligned} &\text{minimize}_A \sum_{(x_i,x_j)\in ML} (x_i - x_j)^T A (x_i - x_j) \\ &\text{subject to} \\ &\qquad \sum_{(x_i,x_j)\in CL} \sqrt{(x_i - x_j)^T A (x_i - x_j)} \geq 1, \end{aligned} \tag{5}$$

where A is a positive semi-definite matrix, ML is the set of must-link constraints, and CL is the set of cannot-link constraints. Finding a positive-semi-definite matrix A by solving the optimization problem above is equivalent to mapping data objects x_i, $i = 1, 2, \dots, n$ to $y_i = A^{1/2} x_i$, $i = 1, 2, \dots, n$ and using the Euclidean distance between y_i's. If A is restricted to be the identity matrix, then the final distance metric is the standard Euclidean norm. If it is restricted to be diagonal, then this means that different weights are given to different dimensions (features). In this case, the above model can be solved by the Newton–Raphson method. If A is a full matrix, authors propose an efficient method using gradient ascent and iterative projections. With the transformed objects in the new space, authors use the COP-K-means [63]. Experimental results show that using such a mapping in the COP-K-means outperforms both traditional K-means and the COP-K-means. Also, as it is expected, increasing the amount of prior knowledge leads to a better distance metric learning and so better clustering accuracy is achieved.

In [39], Kamvar et al. study the clustering problem involving labeled data and/or instance-level constraints. Their problem can be considered as a clustering problem with just instance-level constraints as they transform the labeled data (if there exist any in the problem) into the instance-level constraints. They propose a spectral clustering method for the problem. In spectral clustering methods, the input is an $n \times n$ similarity matrix, where n is the number of data objects to be clustered. Each entity of this matrix is a value between 0 and 1 showing the amount of pairwise similarity between corresponding data objects. By using eigenvalues and eigenvectors of the normalized similarity matrix, blocks corresponding to the clusters are found in the similarity matrix. In this study, the authors incorporate instance-level constraints into the similarity matrix by replacing similarity entity corresponding to the two must-linked data objects with 1 and two cannot-linked data objects with 0. The authors then find the k largest eigenvectors of the normalized similarity matrix (say $e_j, j = 1, 2, \ldots, k$). After the normalization of the rows of the $n \times k$ matrix $Y = [e_1 \ e_2 \ \ldots \ e_k]$, each row of Y is considered as a data object in R^k. To cluster these transformed data objects, the authors use the traditional K-means algorithm. The empirical results show that the proposed spectral clustering method performs better than the traditional K-means algorithm. It should be noted that the main purpose of the paper is classification. After the clusters are found by the proposed spectral learning method, each data object attains a label based on the prior information. Then, class labels are determined according to the labels of majority in the classes.

Bar-Hillel et al. [2] propose a distance learning method for the clustering problem with must-link constraints. The authors use relative component analysis (RCA) proposed by Shental et al. in 2002 [57]. RCA tries to find reasons of the unwanted variability present in the data and decrease that variability by transforming the data objects into a new space in which the desired structure in the data is more apparent. This transformation is done in such a way that high weights are given to the relevant dimensions (features) while low weights are assigned to the irrelevant dimensions. The relevant dimensions are determined by using chunklets. Given the set of must-link constraints (say ML), RCA starts with calculating the center of each chunklet of size ≥ 2 and computing the covariance matrix C of all centered data objects in these chunklets. After the transformation of the data objects into a new space by whitening transformation, $C^{-1/2}$, COP-K-means [63] is performed. Authors compare their method with the traditional K-means, COP-K-means without any distance learning, COP-K-means with distance learning proposed in [65], expectation-maximization (EM) and constrained EM. The proposed algorithm significantly outperforms the K-means. Also, it performs similar to or better than COP-K-means with distance learning method proposed in [65].

Similar to [2], Chang and Yeung [16] address the constrained clustering problem with only must-link constraints. They propose a distance metric learning method transforming the data objects into a different space. For each pair of data objects, (x_r, x_s), in the set ML which is the set of all pairwise must-link constraints, the vector $(x_s - x_r)$ is linearly transformed to $A_r(x_s - x_r) + c_r$ where A_r is a $d \times d$ matrix and c_r is a d dimensional vector (d is the number of features). Then, every data object, x_i, in the neighborhood of x_r is transformed by the following formula:

$$y_i = x_i + (A_r - I)x_i + b_r, \tag{6}$$

where $b_r = (I - A_r)x_r + c_r$. The authors assume that a data object may belong to more than one neighborhood. So, the transformation formula becomes

$$y_i = x_i + \sum_{(x_r, x_s) \in ML} I_{ri}((A_r - I)x_i + b_r), \tag{7}$$

where I_{ri} is an indicator function showing whether x_i belongs to the neighborhood of x_r or not. The problem here is to estimate $|ML|$ different A_r and c_r. In other words, $O(|ML|d^2)$ transformation parameters are required. To decrease this amount to $|ML|d$, authors use Gaussian neighborhood functions. They try to minimize the sum of squared Euclidean distances between must-linked data objects in the new space and penalty for the degree of transformation. To solve this nonlinear objective function of the distance learning process, authors use an iterative procedure. In the initialization step of the procedure, y_i's are set to x_i's. In each iteration, transformation parameters are optimized for given x_i's and y_i's are then updated by using newly found transformation parameters. Procedure continues until convergence is achieved. To solve the optimization problem in each iteration, authors use two methods: gradient method and iterative majorization method. After transforming data objects into the new space, authors apply K-means and compare their method with traditional K-means, COP-K-means [63] and K-means with distance learning method proposed in [2]. In the experimental studies, the effectiveness of the method is verified as it outperforms the other methods for most of the data sets tried.

3.2.3 Search and Distance Based Methods

Search and distance based methods are hybrid methods. They benefit from the advantages of both search based and distance based methods and generally perform better than the individual methods. Lately, noticing this potential, some authors proposed such hybrid methods.

Basu et al. [5] propose an algorithm called MPC-K-means which is the unification of PC-K-means [6] and a distance learning method. There are two components of the objective function of the PC-K-means; namely, constraint violation terms and the objective function of the MSSC problem. The constraint violation terms are directly proportional with the "seriousness" of the violation. For example, putting must-linked data objects close to each other into different clusters is more serious than putting must-linked data objects that are relatively far from each other into different clusters. The distance calculation in the objective function is parameterized with a positive-definite matrix A which corresponds to feature weighting as follows:

$$\sum_{j=1}^{k} \sum_{x_i \in c_j} \left\| x_i - \mu_j \right\|_A^2 = \sum_{j=1}^{k} \sum_{x_i \in c_j} (x_i - \mu_j)^T A (x_i - \mu_j), \tag{8}$$

where μ_j represents the center of the jth cluster. By this way, the distance metric learning is incorporated into the clustering process. In most of the studies using distance metric learning, distance metric is determined before the clustering algorithm by using only the prior knowledge. In this study, data for which there is no prior information also affects the metric learning process. To initialize cluster centers, the method used in [6] is followed. In the assignment step of the algorithm, data objects are assigned to the clusters that result in least contribution to the objective function. Then, as in K-means, cluster centers are updated by taking average of the data objects assigned to them. Moreover, positive-definite matrix used in distance calculation is updated by taking partial derivatives of the objective function and setting them to zero. The assignment and update steps alternate until convergence. In the experimental studies, authors ablate MPC-K-means as M-K-means that only uses metric learning and does not consider violation terms in the objective function and PC-K-means that only considers pairwise constraint violations in the objective function and does not learn a distance metric. They compare performances of MPC-K-means, PC-K-means, M-K-means, and traditional K-means. Results show that supporting search based method with distance metric learning results in the best clustering accuracy. In other words, MPC-K-means outperforms PC-K-means and M-K-means. But, since distance learning task requires considerable amount of prior knowledge, search based methods (like PC-K-means) are better suited when there is a small amount of prior knowledge. However, if there is enough prior knowledge to learn a distance metric, unified approaches become favorable. By ablating MPC-K-means, authors had a chance to compare search based and distance based methods for constrained clustering in the presence of instance-level constraints to discover their relative strengths and weaknesses.

In [10], Bilenko et al. make minor modifications on the MPC-K-means [5]. While penalty terms included in the objective function for constraint violations are the weights associated with the constraints in [5], the penalty terms are defined as functions in this study. Besides that, in experimental studies, authors compare their algorithm with Supervised-K-means in which the assignment step is supervised in the light of the available constraints in addition to comparisons made in [5]. Results show that the updated MPC-K-means outperforms all of the methods including Supervised-K-means.

Basu et al. propose another hybrid method [7]. Different than their previous work which is developed for just Euclidean distance and presented in [5], the method in this study can be used for any Bregman divergence. Also, it can be used for directional similarity measures. As in [3], authors define the objective function of the problem in the framework of HMRF. The potential function used in the objective function includes penalty terms for must-link and cannot-link constraint violations. These penalty terms reflect the aim of learning the underlying distance measure. For example, if two cannot-linked data objects are near to each other with respect to the distance measure in use and are put into the same cluster, the distance measure should be modified to put these objects far from each other. After the modelling, to solve the problem author proposes an iterative K-means like algorithm. The initialization step of the algorithm is the same with that of HMRF-K-means [3]. In the assignment step, each data object is assigned to the cluster minimizing the contribution to the objective function of the algorithm. In the update step, in addition to the update of the cluster centers based on the data points assigned to them, the distance measure is modified to reduce the objective function. It is proved that the algorithm converges to a local optimum. Authors explain the details of the steps of their algorithm for two distance measures; namely cosine similarity and Kullback–Leibler divergence. Moreover, they give details of the steps of their algorithm for the squared Euclidean distance in [8]. Experimental studies show that clustering with constraints results in better clustering accuracy and the proposed algorithm outperforms the original K-means algorithm.

The methods incorporating distance learning generally learn the distance in the light of prior knowledge before the clustering process. The distance metric learnt is then used without any modification during clustering. In [5, 7, 8, 10], authors incorporate distance metric learning into the clustering process that modifies the distance metric in each iteration.

Law et al. [42, 43] also address the problem of clustering with instance-level constraints. Most of the methods mentioned so far consider the constraints as correct and consistent. In this study, authors consider the constraints as random variables between 0 and 1 which reflects the certainty of the constraints. If there is a must-link constraint between data objects x_i and x_j, the constraint that z_i and z_j (which are the cluster labels that x_i and x_j belongs to, respectively) are equal to l, $l \in \{1, 2, \ldots, k\}$ is given a probability. Having values between 0 and 1 for constraints actually means the use of soft constraints. If the probability that z_i and z_j are equal to l is set to 1, the must-link between x_i and x_j is considered as hard. Authors build a graphical model with these random variables. Since there are hidden variables (labels of data objects) in their model, they consider a missing data problem and propose an EM algorithm. In each maximization step, posterior probabilities are re-estimated. That re-estimation corresponds to the distance metric learning in the algorithm. Experimental studies show that the proposed method is beneficial even with small number of constraints. The method outperforms the hard constraint case and is more robust to noisy data as it allows constraint violations. Moreover, the proposed method performs better than the method with no constraints.

3.3 Constrained Clustering with Cluster-Level Constraints

Constrained clustering with cluster-level constraints did not receive too much attention as clustering with instance-level constraints in data mining community. Constraints like forcing the sizes of the groups (clusters) to be comparable and putting lower or upper bounds on the radii or diameter of the groups are generally used in facility location problems. A facility location problem is the problem of finding locations of a pre-determined number of facilities to serve the customers so as to optimize a certain objective like a function of distances between facilities and customers assigned to them. If we consider the facility locations as cluster centers, customers as data objects and the objective to be optimized in the facility location problem as a cluster distortion measure, facility location problem with such constraints and the clustering with cluster-level constraints are very alike. Interested readers are referred to [24] in which Drezner proposes two heuristics and an optimal algorithm for the p-center problem encountered in the facility location literature which can be considered as clustering with upper bounds on the radii of the clusters.

The methods proposed for clustering with cluster-level constraints are search based methods. A distance metric learning in such problems is not applicable as there is no aim of bringing some data objects closer to each other while pulling some others apart. In the paper by Davidson and Ravi [19], instance-level constraints and two types of cluster-level constraints are used in a search based agglomerative hierarchical clustering algorithm. The cluster-level constraints considered are

- δ-constraint that enforces a distance of at least δ between any two data objects that are not in the same cluster.
- ϵ-constraint that requires that for any data object if there is any other object in the same cluster, then there should be at least one object which is at most ϵ away from it.

The proposed algorithm starts with the chunklets formed using the must-link constraints as the initial partition of the data. For every pair of mergeable clusters (taking the constraints into account), distance is computed. Two closest clusters are then merged. Clusters are iteratively merged unless every pair of clusters is non-mergeable due to the constraints. At this point, the algorithm stops. The authors investigate the benefits of using constraints empirically for agglomerative hierarchical clustering. To improve the efficiency of agglomerative clustering, they also introduce a new constraint called as the γ-constraint which enforces that two clusters whose centers are more than γ away cannot be merged.

In addition, the authors explore the feasibility of agglomerative hierarchical clustering under combinations of the instance-level and δ, ϵ constraints. See Sect. 3.4 for the details.

In the constrained clustering literature, there is a body of research focusing on partitioning the data into clusters of approximately equal size. This has applications, for example, in direct marketing campaigns where one may want to partition customers into segments of roughly equal size so that same number of sales representatives can be allocated to each segment [67].

An approach for obtaining balanced clusters would be using an agglomerative hierarchical clustering algorithm and removing those clusters having a certain size from consideration during the remaining merge steps [1]. The authors in [1] state that this approach may significantly reduce the quality of the partition and does not scale well.

In [30], the authors consider clustering of sensor nodes in a distributed sensor network where each cluster has exactly one master node. To be able to evenly distribute the load on all the master nodes, they consider the problem of finding balanced clusters of sensor nodes while minimizing the total distance between the sensor nodes and master nodes. The problem is formulated as a minimum cost flow (MCF) problem and solved to optimality. In some sensor network applications, sensors in a cluster talk to each other. In such applications, the distance between all pairs of sensor nodes in a cluster should be less than a given threshold. The authors also considered this problem, where the maximum diameter of the clusters is minimized and proposed an algorithm for its solution. Here the diameter of a cluster is defined as the maximum distance between pairs of objects in the cluster.

The problem of finding balanced clusters is also considered in [1]. They propose an algorithm that is based on sampling of the data objects. The steps of their algorithm are

- sampling of a small representative subset of the data objects,
- clustering of the sampled data objects,
- populating and refining the clusters while keeping them balanced.

The authors prove the complexity of their framework and show its efficacy on several data sets including high-dimensional ones.

Balancing constraints can be seen as a special case of size constraints which put constraints on the size of each cluster (or on the total weight of the data objects in each cluster in the case with weights on the data objects). This is known as the capacitated clustering problem. Zhu et al. [73] consider the problem of clustering with size constraints where number of data objects in each cluster is known a priori. They propose a heuristic algorithm to transform the problem into a 0–1 integer linear programming problem.

The clustering problem with size constraints is also considered in [28]. Here the authors propose a modified K-means algorithm (initialization and the assignment steps are modified) that takes the size constraint of each cluster into account.

Bradley et al. [12] draw attention to a drawback of the K-means which may end up with poor local optima containing empty clusters or clusters with very few data points especially when it is used with high-dimensional data ($d \geq 10$) and high number of clusters ($k \geq 20$). To overcome this drawback, the K-means is generally initialized with different starting points and the best partition among different initializations is reported to the user. In this study, to prevent empty clusters or clusters with a few data object to appear, authors add a size constraint for each cluster. Their constraints enforce that c_j should have at least τ_j data objects in the final partition. Authors propose an iterative K-means like algorithm to solve the problem. The initialization and update steps of the algorithm are the same with those of the K-means. In the assignment step, algorithm solves the following optimization problem.

$$\begin{aligned}
&\text{minimize } J(C) = \sum_{j=1}^{k} \sum_{x_i \in c_j} T_{ij} \left\| x_i - \mu_j \right\|^2 \\
&\text{subject to} \\
&\sum_{i=1}^{n} T_{ij} \geq \tau_j \quad \forall j \\
&\sum_{j=1}^{k} T_{ij} = 1 \quad \forall i \\
&T_{ij} \in \{0, 1\},
\end{aligned} \tag{9}$$

where T_{ij} takes value 1 if x_i is assigned to c_j or 0 otherwise and μ_j represents the center of c_j. This model can be considered as a MCF problem in which the data objects are supply nodes, cluster centers are demand nodes and distances between data objects and cluster centers are costs associated with arcs. Thanks to the unimodularity of the problem, relaxing the integrality constraints, leads in optimal solution in which $T_{ij} \in \{0, 1\}$. In the assignment step of the algorithm, above LP is solved by assuming that μ_j's are fixed. Experimental studies show that the constrained version of the K-means converges to better local optima than the original K-means for the same starting points. Also, it better summarizes the data. Other solution approaches and applications of the capacitated clustering problem can be found in [17, 29, 52].

In [60], Tung et al. consider a cluster-level constraint called as the existential constraint which can be thought of as a generalization of the constraints considered in [12]. Let X' be a subset of X, the set of all data objects. The elements of X' are called as pivot elements. An existential constraint enforces that each cluster c_j must include at least τ_j pivot elements. When $X' = X$ the existential constraint is the same as the size constraint considered in [12]. The algorithm proposed in the paper is a graph based algorithm having two phases: pivot moving and deadlock resolution. Authors prove that finding an optimal solution for both of the phases are NP-hard. Let's consider a graph whose nodes are all possible k-partitions of X, nodes satisfying existential constraints are called as valid nodes while others are called as invalid nodes. There is an edge between two nodes if and only if these two solutions differ in the assignment of just one pivot element. Algorithm starts with randomly selecting a valid node (feasible solution) and moves to a neighboring solution which has a refined partition. When there is no such neighbor, algorithm performs deadlock resolution phase to escape the local solution. As it is expected, when n is too high, the number of possible k-partitions of X will be too high. This means that the graph to be searched in the algorithm is of very large scale and efficient search may not be possible. Authors also propose a scaling method using micro-clustering methodology to overcome such situations. Micro-clustering reduces the size of the data by pre-clustering some of them into micro-clusters in which the maximum radius is constrained. The proposed algorithm for constrained clustering and the scaling method are both evaluated with a computational study.

Gonzales [32] considers the clustering problem that minimizes the maximum inter-cluster distance. He proposes an approximation algorithm which has $O(kn)$ time complexity where k is the number of clusters and n is the number of data objects. The algorithm consists of an initialization and $(k - 1)$ expanding steps. In the initialization step, all data objects are assigned to the first cluster, c_1, and one of them is selected randomly as head (center) of the cluster. In the hth expanding step, cluster c_{h+1} is constructed. The data object that has a maximal distance to its cluster center becomes the center of c_{h+1}. Then, every data object in previous clusters $\{c_1, \ldots, c_h\}$ for which distance to the cluster center they belong to is more than the distance to the center of c_{h+1} is moved from its cluster to c_{h+1}. Author proves that the algorithm guarantees to find a solution that is at most 2 times worse than the optimal solution. Moreover, it is proven that this upper bound 2 is the best possible approximation bound if $P \neq NP$.

The clustering algorithms discussed so far partition data objects by taking intra-cluster similarities and inter-cluster dissimilarities. Similarities and dissimilarities are based on features only. Contiguity constrained clustering takes the spatial information of the data objects into account in addition to feature information [45]. Contiguity constraints are usually handled in three ways:

- ignoring the contiguity information completely during the clustering and assessing the final partition by investigation,
- embedding the contiguity information into the similarity/dissimilarity matrix (or distances),
- utilizing a contiguity matrix, which must be consulted before merging the clusters.

Murtgah [54] provides a review of contiguity constrained clustering algorithms. He states that contiguity information is generally provided in two ways. In the first way, a binary $n \times n$ matrix where n is the number of data objects is used to describe contiguity. Here the entry of the matrix in the intersection of the ith row and jth column takes the value 1 if and only if x_i and x_j are contiguous. In the second way, a dissimilarity matrix involving continuous contiguity values are used instead of binary values. The main difference between the algorithms for clustering problems with contiguity information is in their clustering criteria. Author discusses algorithms under categorization based on their application areas. See [54] for more on the contiguity constrained clustering.

3.4 Feasibility Issues

In this section, we review the complexity of the clustering feasibility problem under constraints that is the problem of finding a feasible partition of the data satisfying all of the given constraints. Four types of constraints have been considered in the literature [19, 20, 40]; namely must-link, cannot-link, δ, and ϵ constraints. Two types of feasibility problems are considered:

- Given a value of k, does there exist a feasible partition of the data into k clusters satisfying all the given constraints?
- Given the constraints, does there exist a feasible partition of the data (into any number of clusters) satisfying all the constraints?

The complexity results are due to [19, 20, 40]. Table 1 summarizes the complexity of the clustering feasibility problem under different combinations of the constraints and is taken from [19].

In general, the feasibility problem with only must-link constraints can be solved in polynomial time whether k is given or not. The feasibility problem with only cannot-link constraints is polynomial time solvable when k is not given, but is NP-complete in general when k ($\geq$ 3) is specified. The feasibility problem with only δ constraint or only ϵ constraint can be solved in polynomial time in both cases.

Table 1 Complexity of the clustering feasibility problem

Constraint combination	k given	k not given
Cannot-link	NP-complete [20, 40]	P [19]
Must-link and δ	P [20]	P [19]
Must-link and ϵ	NP-complete [20]	P [19]
δ and ϵ	P [20]	P [19]
Must-link, cannot-link, δ and ϵ	NP-complete [20]	NP-complete [19]

Note that, in general, when a combination of ϵ constraint and must-link constraints are given, the feasibility problem becomes NP-complete for a fixed value of k even though the problem is easy if only one of the constraint types is given.

3.5 Related Studies

Using constraints in classification problems as well as clustering problems has gained a lot of interest recently. In some classification problems, data labeling may be expensive although data collection is easy [41]. As in remote sensing, collecting images of Earth surface with high resolution is easy but stating whether there is water or snow in the images may not be so easy. But an expert can readily state that some of the regions are very alike and they should be labeled with the same label or some of the regions should be labeled with different labels since they are too dissimilar. In other words, having must-link or cannot-link constraints is usually easier than getting the labels. In such situations, semi-supervised classification, i.e., constrained classification can be performed.

As mentioned in Sect. 3.2.2, Kamvar et al. [39] use semi-supervised clustering for semi-supervised classification. Like [39], Lange et al. also use semi-supervised clustering for semi-supervised classification in [41].

Interested readers are referred to [33, 55, 72] which are good surveys on semi-supervised classification.

4 Conclusion

The need for reasonable grouping of data objects naturally appears in many domains like social sciences, computer science, business and marketing, and medicine. Therefore, clustering is very popular among scientist and analysts. However, traditional clustering methods may extract irrelevant information from data since they only use unlabeled data objects as input. This natural difficulty of clustering can be handled by incorporating some amount of prior knowledge into the process. In many domains, such a prior knowledge is available in the form of labeled data objects and/or constraints. In this chapter, we have addressed the clustering in the light of prior knowledge, reviewed some of the developments in the area of constrained clustering, and discussed the selected solution approaches from the literature. The studies in this area show that the inclusion of even a small amount of prior knowledge results in improvements of clustering performance and decrease in running time of the algorithms.

References

1. Banerjee, A., Ghosh, J.: Scalable clustering algorithms with balancing constraints. Data Min. Knowl. Disc. **13**(3), 365–395 (2006). doi:10.1007/s10618-006-0040-z
2. Bar-Hillel, A., Hertz, T., Shental, N., Weinshall, D.: Learning distance functions using equivalence relations. In: Proceedings of 20th International Conference on Machine Learning, vol. 3, pp. 11–18 (2003)
3. Basu, S.: Semi-supervised clustering: probabilistic models, algorithms and experiments. Ph.D. thesis, Austin, TX (2005). AAI3187658
4. Basu, S., Banerjee, A., Mooney, R.: Semi-supervised clustering by seeding. In: Proceedings of 19th International Conference on Machine Learning, pp. 19–26. Citeseer (2002)
5. Basu, S., Bilenko, M., Mooney, R.J.: Comparing and unifying search-based and similarity-based approaches to semi-supervised clustering. In: Proceedings of the ICML-2003 Workshop on the Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining Systems, pp. 42–49. Citeseer (2003)
6. Basu, S., Banerjee, A., Mooney, R.J.: Active semi-supervision for pairwise constrained clustering. In: Proceedings of the SIAM International Conference on Data Mining, vol. 4, pp. 333–344. SIAM (2004)
7. Basu, S., Bilenko, M., Mooney, R.J.: A probabilistic framework for semi-supervised clustering. In: Proceedings of 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 59–68. ACM (2004)
8. Basu, S., Bilenko, M., Banerjee, A., Mooney, R.J.: Probabilistic semi-supervised clustering with constraints. In: Chapelle, O., Schölkopf, B., Zien, A. (eds.) Semi-Supervised Learning, pp. 73–102. The MIT Press, Cambridge (2006)
9. Basu, S., Davidson, I., Wagstaff, K. (eds.): Constrained Clustering: Advances in Algorithms, Theory, and Applications. CRC Press, Boca Raton (2009)
10. Bilenko, M., Basu, S., Mooney, R.J.: Integrating constraints and metric learning in semi-supervised clustering. In: Proceedings of 21st International Conference on Machine Learning, pp. 11–18. ACM (2004)
11. Borgwardt, S., Brieden, A., Gritzmann, P.: Geometric clustering for the consolidation of farmland and woodland. Math. Intell. **36**(2), 37–44 (2014). doi:10.1007/s00283-014-9448-2. http://www.dx.doi.org/10.1007/s00283-014-9448-2
12. Bradley, P.S., Bennett, K.P., Demiriz, A.: Constrained k-means clustering. Tech. Rep., Microsoft Corporation (2000). http://www.machinelearning102.pbworks.com/f/ConstrainedKMeanstr-2000-65.pdf
13. Breiman, L., Friedman, J., Stone, C.J., Olshen, R.A.: Classification and Regression Trees. Chapman and Hall, New York (1984)
14. Celebi, M.E., Kingravi, H.A.: Deterministic initialization of the k-means algorithm using hierarchical clustering. Int. J. Pattern Recogn. Artif. Intell. **26**(07), 1250,018 (2012)
15. Celebi, M.E., Kingravi, H.A., Vela, P.A.: A comparative study of efficient initialization methods for the k-means clustering algorithm. Expert Syst. Appl. **40**(1), 200–210 (2013)
16. Chang, H., Yeung, D.Y.: Locally linear metric adaptation for semi-supervised clustering. In: Proceedings of 21st International Conference on Machine Learning, pp. 153–160. ACM (2004)

17. Chou, C.A., Chaovalitwongse, W.A., Berger-Wolf, T.Y., DasGupta, B., Ashley, M.V.: Capacitated clustering problem in computational biology. Comput. Oper. Res. **39**(3), 609–619 (2012)
18. Davidson, I., Basu, S.: A survey of clustering with instance level constraints. ACM Trans. Knowl. Discov. Data **1**, 1–41 (2007)
19. Davidson, I., Ravi, S.: Agglomerative hierarchical clustering with constraints: theoretical and empirical results. In: Jorge, A.M., Torgo, L., Brazdil, P., Camacho, R., Gama, J. (eds.) Knowledge Discovery in Databases: PKDD 2005, pp. 59–70. Springer, Berlin/Heidelberg (2005)
20. Davidson, I., Ravi, S.: Clustering with constraints: feasibility issues and the k-means algorithm. In: Proceedings of 2005 SIAM International Conference on Data Mining, pp. 138–149. SIAM (2005)
21. Davies, D.L., Bouldin, D.W.: A cluster separation measure. IEEE Trans. Pattern Anal. Mach. Intell. **1**(2), 224–227 (1979)
22. Demiriz, A., Bennett, K.P., Embrechts, M.J.: Semi-supervised clustering using genetic algorithms. In: Proceedings of Artificial Neural Networks in Engineering - ANNIE'99, pp. 809–814 (1999)
23. Dicken, L., Levine, J.: Applying clustering techniques to reduce complexity in automated planning domains. In: Proceedings of 11th International Conference on Intelligent Data Engineering and Automated Learning – IDEAL 2010, vol. 6283, pp. 186–193 (2010)
24. Drezner, Z.: The p-centre problem-heuristic and optimal algorithms. J. Oper. Res. Soc. **35**, 741–748 (1984)
25. Eisen, M.B., Spellman, P.T., Brown, P.O., Botstein, D.: Cluster analysis and display of genome-wide expression patterns. In: Proceedings of the National Academy of Sciences of the United States of America, vol. 95, pp. 14,863–14,868. National Academy Sciences (1998)
26. Fisher, D.H.: Knowledge acquisition via incremental conceptual clustering. Mach. Learn. **2**(2), 139–172 (1987)
27. Gan, G., Ma, C., Wu, J.: Data Clustering: Theory, Algorithms, and Applications (ASA-SIAM Series on Statistics and Applied Probability), SIAM, Philadelphia, ASA, Alexandria, VA (2007)
28. Ganganath, N., Cheng, C.T., Chi, K.T.: Data clustering with cluster size constraints using a modified k-means algorithm. In: Proceedings of 2014 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, pp. 158–161. IEEE (2014)
29. Geetha, S., Poonthalir, G., Vanathi, P.: Improved k-means algorithm for capacitated clustering problem. INFOCOMP J. Comput. Sci. **8**(4), 52–59 (2009)
30. Ghiasi, S., Srivastava, A., Yang, X., Sarrafzadeh, M.: Optimal energy aware clustering in sensor networks. Sensors **2**(7), 258–269 (2002)
31. Gluck, M.A., Corter, J.E.: Information uncertainty and the utility of categories. In: Proceedings of 7th Annual Conference of Cognitive Science Society, pp. 283–287. Lawrence Erlbaum (1985)
32. Gonzalez, T.F.: Clustering to minimize the maximum intercluster distance. Theor. Comput. Sci. **38**, 293–306 (1985)
33. Gordon, A.D.: A survey of constrained classification. Comput. Stat. Data Anal. **21**(1), 17–29 (1996)
34. Grira, N., Crucianu, M., Boujemaa, N.: Unsupervised and semi-supervised clustering: a brief survey. In: A Review of Machine Learning Techniques for Processing Multimedia Content, Report of the MUSCLE European Network of Excellence (2004)
35. Hansen, P., Mladenovic, N.: J-means: a new local search heuristic for minimum sum of squares clustering. Pattern Recogn. **34**(2), 405–413 (2001)
36. Howard, R.: Classifying a population into homogeneous groups. In: Lawrence, J.R. (ed.) Operational Research in the Social Sciences. Tavistock Publication, London (1966)
37. Huang, Y., Mitchell, T.M.: Text clustering with extended user feedback. In: Proceedings of 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 413–420. ACM (2006)
38. Jain, A.K.: Data clustering: 50 years beyond k-means. Pattern Recogn. Lett. **31**(8), 651–666 (2010)

39. Kamvar, S.D., Klein, D., Manning, C.D.: Spectral learning. In: Proceedings of 18th International Joint Conference of Artificial Intelligence, pp. 561–566. Stanford InfoLab (2003)
40. Klein, D., Kamvar, S.D., Manning, C.D.: From instance-level constraints to space-level constraints: making the most of prior knowledge in data clustering. In: Proceedings of 19th International Conference on Machine Learning, pp. 307–314. Stanford (2002)
41. Lange, T., Law, M.H.C., Jain, A.K., Buhmann, J.M.: Learning with constrained and unlabeled data. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 731–738. IEEE (2005)
42. Law, M.H.C., Topchy, A., Jain, A.K.: Clustering with soft and group constraints. In: Fred, A., Caeli, T.M., Duin, R.P.W., Campilho, A.C., Ridder, D. (eds.) Structural, Syntactic, and Statistical Pattern Recognition, pp. 662–670. Springer, Berlin/Heidelberg (2004)
43. Law, M.H.C., Topchy, A.P., Jain, A.K.: Model based clustering with probabilistic constraints. In: Proceedings of 2005 SIAM International Conference on Data Mining, pp. 641–645. Citeseer (2005)
44. Likas, A., Vlassis, N., Verbeek, J.J.: The global k-means clustering algorithm. Pattern Recogn. **36**(2), 451–461 (2003)
45. Luo, Z.: Clustering under spatial contiguity constraint: a penalized k-means method. Tech. Rep., Department of Statistics, Penn State University (2001)
46. Lyman, P., Varian, H.R.: How much information 2000? (2000). http://www.groups.ischool.berkeley.edu/archive/how-much-info
47. Lyman, P., Varian, H.R.: How much information 2003? (2003). http://www2.sims.berkeley.edu/research/projects/how-much-info-2003
48. MacQueen, J.: Some methods for classification and analysis of multivariate observations. In: Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, pp. 281–297, Oakland, CA (1967)
49. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, Cambridge, England (2008)
50. Megiddo, N., Supowit, K.J.: On the complexity of some common geometric location problems. SIAM J. Comput. **13**(1), 182–196 (1984)
51. Miyamoto, S., Terami, A.: Constrained agglomerative hierarchical clustering algorithms with penalties. In: Proceedings of 2011 IEEE International Conference on Fuzzy Systems, pp. 422–427. IEEE (2011)
52. Mulvey, J.M., Beck, M.P.: Solving capacitated clustering problems. Eur. J. Oper. Res. **18**(3), 339–348 (1984)
53. Murtagh, F.: A survey of recent advances in hierarchical clustering algorithms. Comput. J. **26**(4), 354–359 (1983)
54. Murtagh, F.: A survey of algorithms for contiguity-constrained clustering and related problems. Comput. J. **28**(1), 82–88 (1985)
55. Pise, N.N., Kulkarni, P.: A survey of semi-supervised learning methods. In: Proceedings of 2008 International Conference on Computational Intelligence and Security, vol. 2, pp. 30–34. IEEE (2008)
56. Segal, E., Shapira, M., Regev, A., Pe'er, D., Botstein, D., Koller, D., Friedman, N.: Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. Nat. Genet. **34**(2), 166–176 (2003)
57. Shental, N., Hertz, T., Weinshall, D., Pavel, M.: Adjustment learning and relevant component analysis. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) Computer Vision—ECCV 2002, pp. 776–790. Springer, Berlin/Heidelberg (2002)
58. Shental, N., Bar-Hillel, A., Hertz, T., Weinshall, D.: Computing gaussian mixture models with em using equivalence constraints. In: Proceedings of the Advances in Neural Information Processing Systems 16, vol. 16, pp. 465–472. MIT Press (2004)
59. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **22**(8), 888–905 (2000)

60. Tung, A.K.H., Han, J., Lakshmanan, L.V.S., Ng, R.T.: Constraint-based clustering in large databases. In: Bussche, J.V., Vianu, V. (eds.) Database Theory, pp. 405–419. Springer, Berlin/Heidelberg (2001)
61. Wagstaff, K.L.: Intelligent clustering with instance-level constraints. Ph.D. thesis (2002)
62. Wagstaff, K., Cardie, C.: Clustering with instance-level constraints. In: Proceedings of 17th International Conference on Machine Learning, pp. 1103–1110. Standford (2000)
63. Wagstaff, K., Cardie, C., Rogers, S., Schroedl, S.: Constrained k-means clustering with background knowledge. In: Proceedings of 18th International Conference on Machine Learning, vol. 1, pp. 577–584. Williams College (2001)
64. Xenarios, I., Fernandez, E., Salwinski, L., Duan, X.J., Thompson, M.J., Marcotte, E.M., Eisenberg, D.: Dip: the database of interacting proteins: 2001 update. Nucleic Acids Res. **29**(1), 239–241 (2001)
65. Xing, E.P., Jordan, M.I., Russell, S., Ng, A.Y.: Distance metric learning with application to clustering with side-information. In: Proceedings of the Advances in Neural Information Processing Systems 15, pp. 505–512. MIT Press (2002)
66. Xu, R., Wunsch, D.: Clustering, vol. 10. Wiley, Hoboken, New Jersey (2008)
67. Yang, Y., Padmanabhan, B.: Segmenting customer transactions using a pattern-based clustering approach. In: Proceedings of 3rd IEEE International Conference on Data Mining, pp. 411–418. IEEE (2003)
68. Yu, S.X., Shi, J.: Segmentation given partial grouping constraints. IEEE Trans. Pattern Anal. Mach. Intell. **26**(2), 173–183 (2004)
69. Zhang, Y., Brady, M., Smith, S.: Segmentation of brain mr images through a hidden Markov random field model and the expectation-maximization algorithm. IEEE Trans. Med. Imaging **20**(1), 45–57 (2001)
70. Zhang, Z., Kwok, J.T., Yeung, D.Y.: Parametric distance metric learning with label information. In: Proceedings of the International Joint Conference on Artificial Intelligence, pp. 1450–1452 (2003)
71. Zhigang, C., Xuan, L., Fan, Y.: Constrained k-means with external information. In: Proceedings of 8th International Conference on Computer Science & Education, pp. 490–493. IEEE (2013)
72. Zhu, X.: Semi-supervised learning literature survey (2008). URL http://www.pages.cs.wisc.edu/~jerryzhu/research/ssl/semireview.html
73. Zhu, S., Wang, D., Li, T.: Data clustering with size constraints. Knowl.-Based Syst. **23**(8), 883–889 (2010)