Constrained Clustering: Current and New Trends

Pierre Gançarski, Thi-Bich-Hanh Dao, Bruno Crémilleux, Germain Forestier, and Thomas Lampert

Abstract Clustering is an unsupervised process which aims to discover regularities and underlying structures in data. Constrained clustering extends clustering in such a way that expert knowledge can be integrated through the use of user constraints. These guide the clustering process towards a more relevant result. Different means of integrating constraints into the clustering process exist. They consist of extending classical clustering algorithms, such as the well-known k-means algorithm; modelling the constrained clustering problem using a declarative framework; and finally, by directly integrating constraints into a collaborative process that involves several clustering algorithms. A common point of these approaches is that they require the user constraints to be given before the process begins. New trends in constrained clustering highlight the need for better interaction between the automatic process and expert supervision.
This chapter is dedicated to constrained clustering. In particular, after a brief overview of constrained clustering and associated issues, it presents the three main approaches in the domain. It also discusses exploratory data mining by presenting models that develop interaction with the user in an incremental and collaborative way. Finally, moving beyond constraints, some aspects of user implicit preferences and their capture are introduced.

P. Gançarski
ICube, University of Strasbourg, France,
e-mail: gancarski@unistra.fr

T.-B.-H. Dao
LIFO, University of Orléans, France,
e-mail: thi-bich-hanh.dao@univ-orleans.f

B. Crémilleux
Normandy University, UNICAEN, ENSICAEN, CNRS - UMR GREYC, France,
e-mail: bruno.cremilleux@unicaen.fr

G. Forestier
IRIMAS, University of Haute-Alsace, France,
e-mail: germain.forestier@uha.fr

T. Lampert
ICube, University of Strasbourg, France,
e-mail: lampert@unistra.fr

1 Introduction

Supervised learning methods are at the center of artificial intelligence solutions and have, for a long time, proved their viability. The phenomenon of "Big Data"—the tremendous increase in the amount of available data—combined with increased computing capacities has led to the return of neuronal methods in the form of deep learning (LeCun et al, 2015). Through their striking results, these approaches have revolutionised supervised learning in, for example, the analysis and understanding of images. These techniques are quickly becoming generalised to domains related to decision aiding (banks, medicine, etc.) and decision making (automobile, avionics, etc.). For such an algorithm to learn to recognise a concept (such as car, cat, etc.), it is trained using several hundreds of thousands of occurrences of this target concept and the labelling of this training data may require many hours of manual interpretation. Once trained, however, the network can almost instantly recognise the concept in new unseen data with success rates rarely achieved so far.

Nevertheless, these approaches suffer from several drawbacks. On the one hand, the "black box" aspect of the learning process and the nature of the model (i.e. a network with learned weights) make it difficult to understand and interpret by an expert. Extracting reusable or transferable knowledge and applying it to other domains or applications remains a challenging problem. On the other hand, these methods, as with all supervised methods, rely upon the hypothesis that learning (and validation) data provided by users and/or domain experts exist and fully represent the underlying concept and data distribution (i.e. they do not fluctuate with time). The creation of such learning sets proves to be very time-consuming, although crowdsourcing methods, for example, make it possible to alleviate this bottleneck. Finally, creating learning sets implies that a problem can be formalised and objects of interest defined, which is often not a realistic hypothesis and means that the algorithms are subject to error.

These issues explain the development of unsupervised approaches, which allow for the discovery of both regularities and structures in the data. Unsupervised discovery of knowledge is at the core of data mining and more specifically clustering, which is the central theme of this chapter. Experiments have shown the ability of clustering methods to extract meaningful clusters from large amounts of heterogeneous data without requiring any additional prior information. Nevertheless, regardless of the efficiency of these algorithms, the lack of formalism of thematic classes and the absence of real reference data makes it difficult to accurately evaluate the quality of the results. Thus, an expert cannot directly validate their own results and cannot directly modify the clusters according to thematic classes. As such, the process of cluster extraction should be rethought and improved to make the results directly useful for domain experts. To this end, it is obvious that the results proposed by these algorithms should model the thematic "intuition" of the expert, that is to say the potential thematic classes.

The clustering process is by definition unsupervised, which means that it only uses the data and no additional knowledge in order to accentuate the principle of serendipity (the probability of finding something useful without specifically searching for it).

Without any supervision, clustering algorithms often produce irrelevant solutions. Recent studies have focused on approaches to allow the guidance of the clustering process using background knowledge or expert knowledge to avoid *apophenia phenomenium* (the risk of seeing patterns or connections in random or meaningless data). The objective is to allow a human expert to embed their domain knowledge into the data mining process and thus to guide it towards better results.

In order to limit expert intervention, which can be highly time consuming, the ideal solution is to make the expert knowledge actionable in order to automate its use in data mining. Depending on the domain, however, the representation and the type of knowledge to model can be very heterogeneous. Thus, three main knowledge representations that are independent of the application domain have been proposed for clustering, under the form of operable constraints (Basu et al, 2008; Dinler and Tural, 2016).

- The first concerns the use of constraints between objects (*comparison constraints*), mainly of resemblance and dissemblance as for example the relations must-link and cannot-link : two objects should (not) be in the same cluster, having same (different) nature according to expert knowledge
- The second consists of using labelled objects (*labeling constraints*), which corresponds directly to domain knowledge.
- The third is on the clusters themselves (number, size, density, etc.), which corresponds to intrinsic qualities of the clusters (*cluster constraints*).

All these approaches only partially address the issue of transferring thematic constraints to actionable constraints. Thus, it is not realistic to ask an expert to define all the constraints when starting from nothing. Indeed, knowledge discovery, and in the case of discovering more relevant clusters, is an iterative process. This is why, in a second phase, the expert has to be able to successively refine and improve the proposed clustering. In particular, to be able to add, refine, or remove constraints, act directly on the clusters (split, merge, deletion) or freeze the clusters that are most likely candidates for thematic classes. The latter should be labelled early in the process and should not be considered by the clustering algorithm any further.

Interactive learning methods that put the expert at the center of the extraction process are one solution to this problem. Surprisingly little attention has been focussed on their development even though since the emergence of data mining in the 1990s (Fayyad et al, 1996) the importance of interactive and semi-automatic processes of knowledge discovery has been well known. Multiple studies (Anand et al, 1995; Kopanas and Avouris, 2002) have shown the importance played by background knowledge and expert knowledge in the process of data mining. Analysts interact (visualise, select, explore) not only with the data but also with the patterns or models supported by the data (Boulicaut et al, 2006). A very strong feature of big data for a data science project is that the data spans multiple domains and therefore understanding and analysing such data requires different but complementary expertises. Expert and algorithm interaction must be flexible in order to better fit the data's perspective. Thus the process of knowledge extraction cannot be fully automatic. This highlights the need to study mechanisms that allow a better interaction between

automatic processes and expert supervision. Consequently, data analysts have rapidly focused on redefining the role of the expert or simply replacing them altogether. Thus, the formalisation of knowledge and its use is a key issue not only in clustering but also in the whole field of data mining. For this reason, this chapter is concluded by discussing new trends in exploratory data analysis.

The remainder of this chapter is organised as follows. In Section 2 the principles of constrained clustering and the main approaches used to implement them are presented. Section 3 presents a review of classic clustering algorithms that have been extended to include user constraints. In Section 4 are described declarative approaches, while Section 5 presents a collaborative approach. Section 6 introduces new trends in constrained clustering, i.e. interactive/incremental approaches, and user preference based methods.

2 Constrained Clustering

Given a set of objects, cluster analysis aims at grouping the objects together into homogeneous groups, such that the objects in the same group are similar and the objects in different groups are different. The groups are called clusters and the set of groups is a clustering. Clustering is an important task in data mining and numerous methods have been developed for it. A complete overview of clustering algorithms can be found in Chapter 12 of this volume. In practice, the expert usually has some intuition or prior knowledge about the underlying clustering. In order to reach a solution relevant to expert knowledge, recent studies have focused on integrating knowledge to allow guidance on the clustering process. This section recalls some main clustering formulations and introduces constrained clustering through several types of user constraints.

2.1 Cluster Analysis

Let $\mathcal{O}$ be a set of instances (data) $\{o_1,\ldots,o_n\}$, let us assume that there exists a dissimilarity (or a similarity) measure $d(o_i,o_j)$ between any two instances o_i and o_j. Partitional clustering involves finding a partition of $\mathcal{O}$ into K non-empty and (generally) disjoint groups called *clusters* $C_1,\ldots,C_K$, such that instances in the same cluster are very similar and instances in different clusters are dissimilar. The homogeneity of the clusters is usually formalised by an optimisation criterion.

Finding the optimal clustering (i.e. the best number of clusters and the best associated clustering) among all the possible clusterings is intractable in general. For N objects and a given number of clusters K, the number of clustering candidates is

$$S_{N,K} = \frac{1}{K!}\sum_{k=0}^{K}(-1)^k(K-k)^N\binom{K}{k} \simeq \frac{K^N}{K!}, \quad \text{when } N \to \infty, \tag{1}$$

and for all clusterings where the number of clusters can vary

$$B_N = \sum_{k=1}^{N} S_{N,k}. \tag{2}$$

For instance, for $N = 25$, there are $4,638,590,332,229,999,353$ possible clusterings requiring $147,000$ years to be generated on a computer producing one million partitions per second.

In *distance-based clustering* the optimisation criterion that defines the homogeneity of the clusters is based on the distance measure. Different optimisation criteria exist, the most popular are (Hansen and Jaumard, 1997):

- minimising the maximal diameter of the clusters, which is defined by the maximal dissimilarity between two objects in the same cluster;
- maximising the minimal split between clusters, which is the smallest dissimilarity between two objects in different clusters;
- minimising the sum of stars of the clusters, which is defined by the minimum sum of dissimilarities between an object to all other objects in the cluster, for each object in the cluster;
- minimising the within-cluster sum of dissimilarities (WCSD), which is the sum of all the dissimilarities between two objects in the same cluster;
- minimising the within-cluster sum of squares (WCSS), in an Euclidean space WCSS is the sum of squared Euclidean distances between each object o_i and the centroid m_k of the cluster that contains o_i.

Finding a partition maximising the minimal split between clusters is polynomial since the partition can be computed from a minimum spanning tree (Delattre and Hansen, 1980). As for the maximal diameter criterion, the problem is polynomial with 2 clusters ($K = 2$), but as soon clusterings with at least 3 clusters ($K \geq 3$) are considered the problem becomes NP-Hard (Hansen and Delattre, 1978). All the other criteria are NP-Hard. The NP-hardness of the WCSS criterion in general dimensions even with $K = 2$ is shown in (Aloise et al, 2009).

Thus, most of the classic clustering algorithms search for a local optimum. For instance, the k-means algorithm finds a local optimum for the WCSS criterion, the k-median algorithm finds a local optimum for the sum of stars criterion and the FPF (Furthest Point First) algorithm (Gonzalez, 1985) for the diameter criterion.

In *similarity-based clustering*, the optimisation criterion that defines the homogeneity of the clusters is based on a similarity measure. The similarity between the instances is usually defined by an undirected graph where the vertices are the objects and the edges have non-negative weights. Spectral clustering aims to find a partition of the graph such that the edges between different groups have a very low weight and the edges within a group have high weight. Given a cluster C_i, a cut measure $\text{cut}(C_i)$ is defined by the sum of the weights of the edges that link an instance in C_i and an instance not in C_i. The two most common optimisation criteria are (von Luxburg, 2007):

- minimising the ratio cut, which is defined by the sum of $\text{cut}(C_i) \,/\, |C_i|$;

- minimising the normalised cut, which is defined by the sum of $\text{cut}(C_i) \,/\, \text{vol}(C_i)$, where $\text{vol}(C_i)$ measures the weight of the edges within C_i.

These criteria are also NP-Hard. Spectral clustering algorithms solve relaxed versions of those problems: relaxing the normalised cut leads to normalised spectral clustering and relaxing the ratio cut leads to unnormalised spectral clustering.

2.2 User Constraints

In practice, a user may have some requirements for, or prior knowledge about, the final solution. For instance, the user can have some information on the label of a subset of objects (Wagstaff and Cardie, 2000). Several studies have demonstrated the importance of domain knowledge in the data mining processes (Anand et al, 1995). Because of the inherent complexity of the optimisation criteria, classic algorithms always find a local optimum. Several optima may exist, some of which may be closer to the user requirement. It is therefore important to integrate prior knowledge into the clustering process. Prior knowledge is expressed by user constraints to be satisfied by the clustering solution. The subject of these user constraints can be the instances or the clusters (Basu et al, 2008). With the presence of user constraints, clustering problems become harder, as for instance the polynomial criterion of maximising the minimal split between clusters becomes NP-Hard under user constraints (Davidson and Ravi, 2007).

Instance-level constraints are the most widely used type of constraint and were first introduced by Wagstaff and Cardie (2000). Two kinds of instance-level constraints exist: *must-link* (ML) and *cannot-link* (CL).

Definition 1. Two instances o_i and o_j that satisfy an ML constraint must be in the same cluster, i.e. $\forall k \in \{1,\ldots,K\}$, $o_i \in C_k \Leftrightarrow o_j \in C_k$.

Definition 2. Two instances o_i and o_j that satisfy an CL constraint must not be in the same cluster, i.e. $\forall k \in \{1,\ldots,K\}$, $\neg(o_i \in C_k \wedge o_j \in C_k)$.

In *semi-supervised clustering*, this information is available to aid the clustering process and can be inferred from class labels: if two objects have the same label then they are linked by an ML constraint, otherwise by a CL constraint. Supervision by instance-level constraints is however more general and more realistic than class labels. Using knowledge, even when class labels may be unknown, a user can specify whether pairs of points belong to the same cluster or not (Wagstaff et al, 2001). Semi-supervised clustering is therefore a transductive operation because its objective is to define the clusters to explain the processed data and possibly to label the objects not initially labelled. During semi-supervised classification, labelled objects and unlabelled objects are used to construct a classification function. Thus, the objective is to use the unlabelled objects to better understand the configuration of the data space. Semi-supervised classification is an inductive operation, the aim of which is

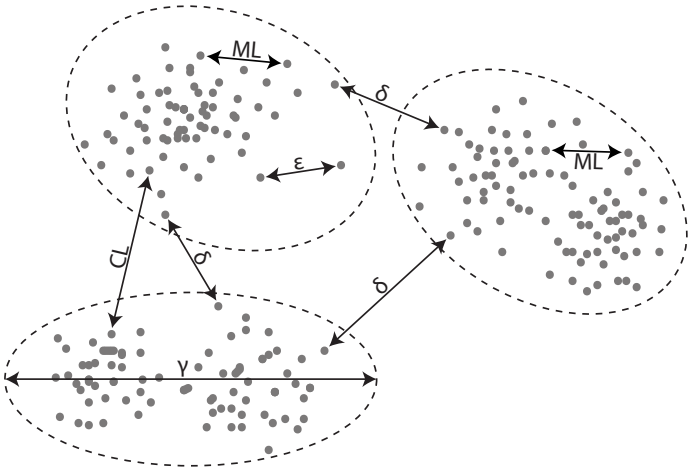

Fig. 1 Examples of ML, CL, δ and ε constraints.

to create a classifier which generalises the model of available data and which can be used subsequently to process other data.

Cluster-level constraints define requirements on the clusters (see Figure 1), for example:

- the number of clusters K;
- their size, a *capacity constraint* expresses a maximal or a minimal limit on the number of objects in each cluster—a minimal capacity constraint states that each cluster must have at least α objects, i.e. $\forall k \in \{1,\ldots,K\}$, $|C_k| \geq \alpha$, and a maximal capacity constraints requires that each cluster must have at most β objects, i.e. $\forall k \in \{1,\ldots,K\}$, $|C_k| \leq \beta$;
- the diameter of the clusters, a *maximum diameter constraint* gives an upper bound γ on the diameter of each cluster, i.e. $\forall k \in \{1,\ldots,K\}$, $\forall o_i, o_j \in C_k$, $d(o_i,o_j) \leq \gamma$;
- the split between clusters, a *minimum split constraint* states that the clusters must be separated by at least δ: $\forall k,k' \in \{1,\ldots,K\}$, $k' \neq k$, $\forall o_i \in C_k$, $\forall o_j \in C_{k'}$, $d(o_i,o_j) \geq \delta$;
- finally, the ε-*constraint*, introduced in (Davidson and Ravi, 2005), demands that each object o_i have in its neighbourhood of radius ε at least one other object in the same cluster: $\forall k \in \{1,\ldots,K\}$, $\forall o_i \in C_k$, $\exists o_j \in C_k, o_j \neq o_i$, $d(o_i,o_j) \leq \varepsilon$, this constraint tries to capture the density notion, used in density based clustering DBSCAN (Ester et al, 1996) and can be generalised to the requirement that each object o_i has in its neighbourhood of radius ε at least m objects in the same cluster.

Note that although the diameter and split constraints state requirements on the clusters, they can be expressed by a conjunction of cannot-link constraints and must-link constraints, respectively (Davidson and Ravi, 2005).

The instances can be described by a set of features that enables the computation of a dissimilarity measure and also by a set of properties from which the definitions of what is actionable/interesting is given. Constraints can therefore also be stated on properties and can be divided into the following categories (Dao et al, 2016):

- *cardinality constraints* place a requirement on the count of the elements in a cluster having a property, they may be as simple as each cluster should contain at least one female to more complex variations such as the number of males must be no greater than two times the number of females;
- *density constraints* relate to cardinality constraints in that they provide requirements on the count of a property except not for an entire cluster but rather a subset of instances in the cluster, e.g. we may require that each person has at least 10 people in his/her cluster that share the same hobby;
- *geometric constraints* place an upper or lower bound on some geometric property of a cluster or cluster combination, e.g. that the maximum diameter of a cluster with respect to the age property is 10 years, this would prevent clusters containing individuals with a wide range of ages;
- *complex logic constraints* express logical combinations of constraints, which can be instance-level or cluster-level, e.g. we may require that any cluster having more than 2 professors should have more than 10 PhD students.

When the instances are described by binary features as in the case of transactional data, constraints can be stated such that each cluster is associated to a definition expressed by a pattern. Four families of constraints can be identified and k-pattern set mining problems can be specified as combinations of them (Guns et al, 2013):

- *individual pattern constraints*, which include among others local pattern mining constraints;
- *redundancy constraints*, which are used to constrain or to minimise the redundancy between different patterns;
- *coverage constraints*, which deal with defining and measuring how well a pattern set covers the data;
- *discriminative constraints*, which are used on labelled data to measure and optimise how well a pattern or pattern set discriminates between positive and negative examples.

3 Extensions of Classic Clustering Algorithms to User Constraints

This section presents a brief review of *partitional* constrained clustering methods and in particular k-Means, metric learning, and spectral graph theory based methods. These all have the following properties in common: (1) they extend a clustering algorithm to integrate user constraints and are therefore specific to the objective function that is optimised by the clustering algorithm, e.g. minimising the sum of squared errors for the k-Means algorithm; (2) they integrate instance-level constraints

or some form of cluster level constraint (e.g. cluster size); (3) they are usually fast and find an approximate solution, and therefore do not guarantee the satisfaction of all the constraints.

3.1 k-Means

In this type of approach, the clustering algorithm or the objective function is modified so that user constraints are used to guide the algorithm towards a more appropriate data partitioning. Most of these works consider instance-level must-link and cannot-link constraints. The extension is done either by enforcing pairwise constraints or by using pairwise constraints to define penalties in the objective function. A survey on partitional and hierarchical clustering with instance level constraints can be found in (Davidson and Basu, 2007).

In the category of enforcing pairwise constraints, the first work proposed a modified version of COBWEB (Fisher, 1987) that tends to satisfy all the pairwise constraints, named COP-COBWEB (Wagstaff and Cardie, 2000). Subsequent work extended the *k*-Means algorithm to instance-level constraints. The *k*-Means algorithm starts with initial assignment seeds and assigns objects to clusters in several iterations. At each iteration, the centroids of the clusters are computed and the objects are reassigned to the closest centroid. The algorithm converges and finds a solution which is a local optimum of the within-cluster sum of squares (WCSS or distortion). To integrate must-link and cannot-link constraints, the COP-KMeans algorithm by Wagstaff et al (2001) extends the *k*-Means algorithm by choosing a reassignment that does not violate any constraints at each iteration. This greedy behavior without backtracking means that COP-KMeans may fail to find a solution that satisfies all the constraints even when such a solution exists. Basu et al (2002) propose two variants of *k*-Means, the Seed-KMeans and Constrained-KMeans algorithms, which allow the use of objects labeled as seeds: the difference between the two being the possibility of changing the class centers or not. In both approaches, it is assumed that there is at least one seed for each cluster and that the number of clusters is known. The seeds are used to overcome the sensitivity of the *k*-Means approaches to the initial parameterisation.

Incorporating must-link and cannot-link constraints makes clustering algorithms sensitive to the assignment order of instances and therefore results in consequent constraint-violation. To address the issue of constraint violation in COP-KMeans, Tan et al (2010) (ICOP-KMeans) and Rutayisire et al (2011) propose a modified version with an assignment order, which is either based on a measure of certainty computed for each instance or a sequenced assignment of cannot-linked instances. MLC-KMeans (Huang et al, 2008) takes an alternative approach by introducing assistant centroids, which are calculated using the points implicated by must-link constraints for each cluster, and which are used to calculate the similarity of instances and clusters.

For high-dimensional sparse data, the SCREEN method (Tang et al, 2007) for constraint-guided feature projection was developed, which can be used with a semi-supervised clustering algorithm. This method considers an objective function to learn the projection matrix, which can project the original high-dimensional dataset into a low-dimensional space such that the distance between any pair of instances involved in the cannot-link constraints are maximised while the distance between any pair of instances involved in the must-link constrains are minimised. A spherical k-Means algorithm is then used to try to avoid violating cannot-link constraints.

Other methods uses penalties as a trade-off between finding the best clustering and satisfying as many constraints as possible. Considering a subset of instances whose label is known, Demiriz et al (1999) modifies the clustering objective function to incorporate a dispersion measure and an impurity measure. The impurity measure is based on Gini Index to measure misplaced known labels. The CVQE (constrained vector quantisation error) method (Davidson and Ravi, 2005) penalises constraint violations using distance. If a must-link constraint is violated then the penalty is the distance between the two centroids of the clusters containing the two instances that should be together. If a cannot-link constraint is violated then the penalty is the distance between the cluster centroid the two instances are assigned to and the distance to the nearest cluster centroid. These two penalty types together with the distortion measure define a new differentiable objective function. An improved version, linear-time CVQE (LCVQE) (Pelleg and Baras, 2007), avoids checking all possible assignments for cannot-link constraints and its penalty calculations takes into account coordinates of the involved instances in the violated constraint. The method PCK-Means (Basu et al, 2004a) formulated the goal of pairwise constrained clustering as minimising a combined objective function, defined as the sum of the total squared distances between the points and their cluster centroids WCSS, and the cost incurred by violating any pairwise constraints. The cost can be uniform but can also take into account the metric of the clusters, as in the MPCK-Means version that integrates both constraints and metric learning. Lagrangian constrained clustering (Ganji et al, 2016) also formulates the objective function as a sum of distortion and the penalty of violating cannot-link constraints (must-link constraints are used to aggregate instances into super-instances so they are all satisfied). This method uses a Lagrangian relaxation strategy of increasing penalties for constraints which remain unsatisfied in subsequent clustering iterations. A local search approach using Tabu search was developed to optimise the objective function, which is the sum of the distortion and the weighted cost incurred by violating pairwise constraints (Hiep et al, 2016). Grira et al (2006) introduced the cost of violating pairwise constraints into the objective function of Fuzzy CMeans algorithm. Li et al (2007) use non-negative matrix factorisation to perform centroid-less constrained k-Means clustering (Zha et al, 2001).

Hybrid approaches integrate both constraint enforcing and metric learning (see Subsection 3.2) into a single framework: MPCK-Means (Bilenko et al, 2004), HMRF-KMeans (Basu et al, 2004b), semi-supervised kernel k-Means (Kulis et al, 2005), and CLWC (Cheng et al, 2008). Bilenko et al (2004) define an uniform framework that integrates both constraint-based and metric-based methods. This

framework represents PCK-Means when considering a constraint-based factor and MPCK-Means when considering both constraint-based and metric-based factors. Semi-supervised HMRF *k*-Means (Basu et al, 2004b) is a probabilistic framework based on Hidden Markov Random Fields, where the semi-supervised clustering objective minimises both the overall distortion measure of the clusters and the number of violated must-link and cannot-link constraints. A *k*-Means like iterative algorithm is used for optimising the objective, where at each step the distortion measure is re-estimated to respect user-constraints. Semi-supervised kernel *k*-Means (Kulis et al, 2005, 2009) is a weighted kernel-based approach, that generalises HMRF *k*-Means. The method can perform semi-supervised clustering on data given either as vectors or as a graph. It can be used on a wide class of graph clustering objectives such as minimising the normalised cut or ratio cut. The framework can be therefore applied on semi-supervised spectral clustering. Constrained locally weighted clustering (CLWC) (Cheng et al, 2008) integrates the local distance metric learning with constrained learning. Each cluster is assigned to its own local weighting vector in a different subspace. The data points in the constraint set are arranged into disjoint groups (chunklets), and the chunklets are assigned entirely in each assignment and weight update step.

Beyond pairwise constraints, Ng (2000) adds suitable constraints into the mathematical program formulation of the *k*-Means algorithm to extend the algorithm to the problem of partitioning objects into clusters where the number of elements in each cluster is fixed. Bradley et al (2000) avoid local solution with empty clusters or clusters having very few points by explicitly adding k minimal capacity constraints to the formulation of the clustering optimisation problem. This work considers that the *k*-Means algorithm and the constraints are enforced during the assignment step at each iteration. Banerjee and Ghosh (2006) proposed a framework to generate balanced clusters, i.e. clusters of comparable sizes. Demiriz et al (2008) integrated a minimal size constraint to *k*-Means algorithm. Considering two types of constraints, the minimum number of objects in a cluster and minimum variance of a cluster, Ge et al (2007) proposed an algorithm that generates clusters satisfying them both. This algorithm is based on a CD-Tree data structure, which organises data points in leaf nodes such that each leaf node approximately satisfies the significance and variance constraint and minimises the sum of squared distances.

3.2 Metric Learning

Metric learning aims to automatically learn a metric measure from training data that best discriminates the comprising samples according to a given criterion. In general, this metric is either a similarity or a distance (Klein et al, 2002). Many machine learning approaches rely on the learned metric; thus metric learning is usually a preprocessing step for such approaches.

In the context of clustering, the metric can be defined as the Mahalanobis distance parameterised by a matrix M, i.e. $\mathbf{d_M}(o_i, o_j) = ||o_i - o_j||_M$ (Bellet et al, 2015).

Unlike the Euclidean distance, which assumes that attributes are independent of one another, the Mahalanobis distance enables the similarity measure to take into account correlations between attributes. Learning the distance $\mathbf{d_M}$ is equivalent to learning the matrix M. For $\mathbf{d_M}$ to satisfy distance proprieties (non-negativity, identity, symmetry, and the triangle inequality) M should be a positive semi-definite real-valued matrix.

To guide the learning process, two sets are constructed from the ML and CL constraints: the set of supposedly similar—must-link—pairs Sim, and the supposedly dissimilar—cannot-link—pairs Dis, such that

- $\text{Sim} = \{(o_i, o_j) \mid o_i \text{ and } o_j \text{ should be as similar as possible}\}$,
- $\text{Dis} = \{(o_i, o_j) \mid o_i \text{ and } o_j \text{ should be as dissimilar as possible}\}$.

It is also possible to introduce unlabeled data along with the constraints to prevent over-fitting.

Several proposals have been made to modify (learn) a distance (or metric) taking into account this principle. We can cite works on the Euclidean distance and shortest path (Klein et al, 2002), Mahanalobis distance (Bar-Hillel et al, 2005, 2003; Xing et al, 2002), Kullback-Leibler divergence (Cohn et al, 2003), string-edit distance (Bilenko and Mooney, 2003), and the Laplacian regulariser metric learning (LRML) method for clustering and imagery (Hoi et al, 2008, 2010).

Yi et al (Yi et al, 2012) describe a metric learning algorithm that avoids the high computational cost implied by the positive semi-definite constraint. Matrix completion is performed on the partially observed constraints and it is observed that the completed similarity matrix has a high probability of being positive semi-definite, thus avoiding the explicit constraint.

3.3 Spectral Graph Theory

Spectral clustering is a non-supervised method that takes as input a pre-calculated similarity matrix (graph) and aims to minimise the ratio cut criterion (von Luxburg, 2007) or the normalised cut criterion (Shi and Malik, 2000). Spectral clustering is often considered superior to classical clustering algorithms, such as k-Means, because it is capable of extracting clusters of arbitrary form (von Luxburg, 2007). It has also been shown that algorithms that build partitions incrementally (like k-Means and EM) are prone to be overly constrained (Davidson and Ravi, 2006). Moreover, spectral clustering has polynomial time complexity. The constraints can be expressed as ML/CL constraints or in the form of labels, these can be taken into account either as "hard" (binary) constraints or "soft" (probabilistic) constraints. The method allows the user to specify a lower bound on constraint satisfaction and all points are assigned to clusters simultaneously, even if the constraints are inconsistent.

Kamvar et al (2003) first integrated ML and CL constraints into spectral clustering. This is achieved by modifying the affinity matrix by setting ML constrained pairs to maximum similarity, 1, and CL constrained pairs to minimum similarity, 0. This has

been extended to out-of-sample points and soft-constraints through regularisation (Alzate and Suykens, 2009). Li et al (2009) point out, however, that a similarity of 0 in the affinity matrix does not mean that the two objects tend to belong to different clusters.

Wang and Davidson (2010a) and Wang et al (2014) introduce a framework for integrating constraints into a spectral clustering. Constraints between N objects are modelled by a matrix Q of size $N \times N$, such that

$$Q_{ij} = Q_{ji} = \begin{cases} +1, & \text{if ML}(i,j), \\ -1, & \text{if CL}(i,j), \\ 0, & \text{otherwise,} \end{cases} \tag{3}$$

upon which a constraint satisfaction measure can be defined. Soft constraints can be taken into account by allowing real values to be assigned to Q or by allowing fuzzy cluster membership values. Subsequently, the authors introduce a method to integrate a user-defined lower-bound on the level of constraint satisfaction (Wang and Davidson, 2010b). Work has also been described that allows for inconsistent constraints (Rangapuram and Hein, 2012).

Based on the Karush-Kuhn-Tucker (Kuhn and Tucker, 1951) conditions, an optimal solution can then be found by first finding the set of solutions satisfying all constraints and then using a brute-force approach to find the optimal solution from this set.

These approaches have been extended to integrate logical combinations of constraints (Zhi et al, 2013), which are translated into linear equations or linear inequations. Furthermore, instead of modifying the affinity matrix using binary values, Anand and Reddy (2011) propose to modify the distances using an all-pairs-shortest-path algorithm such that the new distance metric is similar to the original space.

Lu and Carreira-Perpinán (2008) state that an affinity matrix constructed using constraints is highly informative but only for a small subset of points. To overcome this limitation they propose a method to propagate constraints (in a method that is consistent with the measured similarities) to points that are not directly affected by the original constraint set. These advances are proposed for the two-class problem (multi-class extension is discussed but is computationally inefficient), multi-class alternatives have been proposed (Lu and Ip, 2010; Chen and Feng, 2012; Ding et al, 2013).

Several works (Zhang and Ando, 2006; Hoi et al, 2007; Li et al, 2008, 2009) use the constraints and point similarities to learn a kernel matrix such that points belonging to the same cluster are mapped to be close and points from different clusters are mapped to be well-separated.

Most recently, progress has been made in introducing faster and simpler formulations, while providing a theoretical guarantee of the partitioning quality (Cucuringu et al, 2016).

4 Declarative Approaches for Constrained Clustering

These approaches offer the user a general framework to formalise the problem by choosing an objective function and explicitly stating the constraints. The frameworks are usually developed using a general optimisation tool, such as integer linear programming (ILP), SAT, constraint programming (CP), or mathematical programming. Detailed descriptions of SAT and CP can be found in Chapters 5 and 6 of this volume, respectively. Commonalities between theses approaches are that they enable the modelling of different types of user constraints and they search for an exact solution—a global optimum that satisfies all the constraints. Some declarative approaches are reviewed in Subsection 4.1 of this chapter, more detailed descriptions of approaches using ILP are presented in Subsection 4.2 and approaches using CP in Subsection 4.3.

4.1 Overview

For dissimilarity-based constrained clustering settings, several approaches using SAT, ILP, and CP have been developed. A SAT based framework has been proposed by Davidson et al (2010) for constrained clustering problems with $K = 2$. The assignment of objects to clusters is represented by a binary variable X_i, where $X_i = 1$ (or $X_i = 0$) means the i-th object is assigned to cluster number 1 (or number 0, respectively). Constraints such as must-link, cannot-link, maximum diameter, and minimum split can be expressed by 2-SAT problems. Using binary search, the framework offers both single objective optimisation and bi-objective optimisation. Several single optimisation criteria are integrated: minimising the maximal diameter, maximising the minimal split, minimising the difference between diameters, and minimising the sum of diameters. When optimising multiple objectives, the framework considers minimising the diameter and maximising the split either in a way such that one objective is used as a constraint and the other is optimised under that constraint, or by combining them in a single objective which is the ratio of diameter to split. In order to make the framework more efficient, approximation schemes are also developed to reduce the number of calls in the binary search. CP and ILP based approaches offer flexible frameworks with several choices of optimisation criteria and user constraints (Subsection 4.2 and 4.3).

Generic declarative frameworks have also been investigated in several works for other clustering settings. Conceptual clustering considers objects described by categorical attributes and aims to associate to each cluster a definition expressed by a pattern. CP frameworks have been developed for the K-pattern set mining problem that can be used for conceptual clustering and other pattern mining tasks (e.g. unexpected rules, k-tilling, redescription mining) (Khiari et al, 2010; Guns et al, 2013; Chabert and Solnon, 2017). These frameworks integrate constraints on patterns or groups of patterns as well as different optimisation criteria (Guns et al, 2013). A SAT based framework has also been proposed, which provides a query language

to formalise conceptual clustering tasks (Métivier et al, 2012). The elements of the language are translated into SAT clauses and solved by a SAT solver. An ILP-based framework has been proposed in (Ouali et al, 2016), which also integrates constraints on clustering that enable the modelling of conceptual clustering, soft-clustering, co-clustering, and soft co-clustering. Based on a similarity graph between objects, correlation clustering aims to find a partition that agrees as closely as possible to the similarities. The cost function to be optimised is such that the number of similar points co-clustered is maximised and the number of dissimilar points co-clustered is minimised. The MaxSAT framework has been developed for constrained correlation clustering (Berg and Järvisalo, 2017). In this model, hard-clauses guarantee a well defined partition, must-link and cannot-link constraints, and soft-clauses are used to encode the cost function.

Different from partition clustering, hierarchical clustering constructs a hierarchy of partitions, represented by a dendrogram. A framework developed in (Gilpin and Davidson, 2017) allows to model hierarchical clustering using ILP. Another SAT framework allows to integrate different types of user constraints (Gilpin and Davidson, 2011).

4.2 Integer Linear Programming

Different frameworks using Integer Linear Programming (ILP) have been developed for constrained clustering. Using ILP, constrained clustering problems must be formalised by a linear objective function subject to linear constraints. In the formulation of clustering such as the one used in CP-based approaches, a clustering is defined by an assignment of instances to clusters. For several optimisation criteria, e.g. the within-cluster sum of squares WCSS, this formulation leads to a non-linear objective function. ILP-based approaches therefore use an orthogonal formulation, where a clustering is considered as a subset of the set of all possible clusters.

A cluster is a subset of instances. For a dataset of n instances, the number of possible clusters is 2^n. Let $T = \{1, \ldots, 2^n\}$ be the set of all possibles non-empty clusters. Consider any cluster C_t and let c_t be the *cost* of the cluster, which is defined on C_t depending on the optimisation criterion. In the case of minimising the WCSS, the cost c_t is defined by the sum of squared distances of the instances in C_t to its mean, such that

$$c_t = \frac{1}{2|C_t|} \sum_{o_i, o_j \in C_t} ||o_i - o_j||^2.$$

For each $i \in \{1, \ldots, n\}$, let a_{it} be a constant which is 1 if o_i is in cluster C_t and 0 otherwise. The unconstrained clustering problem is therefore formalised by an integer linear program, such that (du Merle et al, 1999):

$$
\begin{aligned}
\text{minimise} \quad & \sum_{t \in T} c_t x_t, \\
\text{subject to} \quad & \sum_{t \in T} a_{it} x_t = 1, \qquad \forall i \in \{1, \ldots, n\}, \\
& \sum_{t \in T} x_t = K, \\
& x_t \in \{0, 1\}.
\end{aligned}
$$

In this formulation, the first constraint states that each instance o_i must be covered by exactly one cluster (the clustering is therefore a partition of the instances) and the second states that the clustering is formed by K clusters. The variable x_t expresses whether the cluster C_t is chosen in the clustering solution. The number of variables x_t is however exponential w.r.t. the number of instances. Two kinds of ILP-based approaches are therefore developed for constrained clustering: (1) use a column generation approach, where the master problem is restricted to a smaller set $T' \subseteq T$ and columns (clusters) are incrementally added until the optimal solution is proved (Babaki et al, 2014); and (2) restrict the cluster candidates on a subset $T' \subseteq T$ and define the clustering problem on T' (Mueller and Kramer, 2010; Ouali et al, 2016).

An ILP-based approach with column generation for unconstrained minimum sum of squares clustering was introduced by du Merle et al (1999) and improved by Aloise et al (2012). Column generation iterates between solving the restricted master problem and adding one or multiple columns. A column is added to the master problem if it can improve the objective function. If no such column can be found, one is certain that the optimal solution of the restricted master problem is also an optimal solution of the full master problem. Whether a column can improve the objective function can be derived from the dual of the master problem. A column that improves the objective function of the master problem corresponds to a column with a negative reduced cost in the dual. Among all the columns with negative reduced cost, the smallest one is usually searched for, which yields a minimisation subproblem. The column generation approach has been extended to integrate anti-monotone user-constraints (Babaki et al, 2014). A constraint is anti-monotone if it is satisfied on a set of instances S and satisfied on all subsets $S' \subseteq S$. For instance, maximal capacity constraints are anti-monotone but minimal capacity constraints are not. With the observation that many user-constraints can be evaluated on each cluster individually, the user-constraints are not part of the master problem, but only have to be considered when solving the subproblems. These are enforced when solving the subproblem by removing columns corresponding to clusters that do not satisfy the constraints. Subproblems are solved by a branch-and-bound algorithm where an anti-monotone property is used to ensure the correctness of the computed bounds.

The number of cluster candidates in principle is exponential w.r.t. the number of instances. Nevertheless, in some clustering settings such as conceptual clustering, candidates can usually be drawn from a smaller subset T'. Considering a constrained clustering problem on a restricted subset T', Mueller and Kramer (2010) and Ouali et al (2016) develop ILP-based frameworks that can integrate different kinds of user-constraints. The same principle is used, i.e. instance-level and cluster-level con-

straints are enforced to remove cluster candidates that do not satisfy the constraints. Moreover, the frameworks can integrate constraints on clustering and different optimisation criteria. In (Mueller and Kramer, 2010) constraints on clustering can be stated that give, for instance, bounds on the degree of overlap between clusters of the clustering, and bounds on the number of clusters an instance can be grouped into. The best clustering can be found by optimising the mean/minimum/median quality of the clusters. This framework has been used in conceptual clustering for transactional datasets. In a transactional dataset, each instance (transaction) is described by a set of items. Conceptual clustering aims to assign the transactions to homogeneous clusters and to provide each cluster with a distinct description (concept) that characterises all the transactions contained within it. The clusters that comprise the subset T' can be required to correspond to frequent patterns or to closed frequent patterns. The subset T' is therefore precomputed by an algorithm that extracts frequent patterns a priori (Mueller and Kramer, 2010) or closed patterns (e.g. LCM) (Ouali et al, 2016). In the framework developed by Ouali et al (2016), constraints on clustering are also available, for example: at least some instances must be covered, and small overlaps of the clusters are allowed. Besides modelling conceptual clustering, these constrains also enable the modelling of soft clustering (at most some transactions can be uncovered or small overlaps are allowed), co-clustering (clustering that covers both the set of transactions and the set of items, without any overlap on transactions or on items) and soft co-clustering.

4.3 Constraint Programming

A general and declarative framework has been developed for distance-based constrained clustering, based on Constraint Programming (CP) (Dao et al, 2013, 2017). CP is a powerful paradigm for solving combinatory satisfaction or optimisation problems. Modelling the problem in CP consists of its formalisation into a *Constraint Satisfaction Problem (CSP)* or a *Constraint Optimisation Problem (COP)*. A CSP is a triplet $\langle X, \mathrm{Dom}, C \rangle$ where X is a set of variables, $\mathrm{Dom}(x)$ for each $x \in X$ is its domain and C is a set of constraints, each of which expresses a condition on a subset of X. A solution to a CSP is a complete assignment of values from $\mathrm{Dom}(x)$ to each variable $x \in X$ that satisfies all the constraints of C. A COP is a CSP with an objective function to be optimised. An optimal solution of a COP is a solution of the CSP that optimises the objective function.

In general, solving a CSP or a COP is NP-Hard. Nevertheless, the constraint propagation and search strategies (Rossi et al, 2006) used by CP solvers allow a large number of real-world applications to be efficiently solved.

As discussed in Chapter 6 of this volume, the propagation of a constraint c reduces the domain of the variables of c by removing some or all inconsistent values, i.e. values that cannot be part of a solution of c. A propagation scheme is defined for each type of constraint. Different kinds of constraints are available for modelling, they can be elementary constraints expressing arithmetic or logic relations, or global

constraints expressing meaningful n-ary relations. Although equivalent to conjunctions of elementary constraints, global constraints benefit from efficient propagation, performed by a filtering algorithm exploiting results from other domains, e.g. graph theory. Reified constraints are available, which allow a boolean variable to be linked to the truth value of a constraint. A catalogue of global constraints that contains more than 400 inventoried global constraints is maintained by Beldiceanu et al (2005). In a CP solver, two steps—constraint propagation until a stable state is found and branching—are repeated until a solution is found. Different strategies can be used to create and to order branches at each branching point. They can be standard search strategies defined by CP solvers or can be specifically developed.

A CP-based framework developed for distance-based constrained clustering has been developed by Dao et al (2013). This framework enables the modelling of different constrained clustering problems, by specifying an optimisation criterion and by setting the user constraints. The framework is improved by modifying the model and by developing dedicated propagation algorithms for each optimisation criterion (Dao et al, 2017). In this model, the number of clusters K does not need to be fixed beforehand, only bounds are required, i.e. $K_{min} \leq K \leq K_{max}$. The clusters in a partition of K clusters are numbered from 1 to K. In order to express the cluster assignment, a variable G_i is used for each instance $o_i \in \mathcal{O}$, with $\mathrm{Dom}(G_i) = \{1, \ldots, K_{max}\}$. A real valued variable is used to represent the value of the objective function. The model has the following three components.

- Constraints to express a partition. The use of the variables G_i naturally express a partition. Nevertheless, several assignments of the variables $G_1, \ldots, G_n$ can correspond to the same partition, for instance by interchanging the numbers of two clusters. In order to break these symmetries, the constraint precede$([G_1, \ldots, G_n], [1, \ldots, K_{max}])$ is used. This constraint ensures that $G_1 = 1$ and moreover, if $G_i = c$ with $1 < c \leq K_{max}$, there must exist at least an index $j < i$ such that $G_j = c - 1$. The requirement to have at least K_{min} clusters means that all the numbers among 1 and K_{min} must be used in the assignment of the variables G_i. When using the constraint precede, one only needs to require that at least one variable G_i is equal to K_{min}. This is expressed by the relation $\#\{i \mid G_i = K_{min}\} \geq 1$.
- Constraints to express clustering user constraints. All popular user-defined constraints may be straightforwardly integrated. For instance, a must-link (or cannot-link) constraint on o_i and o_j is expressed by $G_i = G_i$ (or $G_i \neq G_j$, respectively). For the minimal size α of clusters, this means that each point must be in a cluster with at least α points (including itself). This is expressed by n constraints: for each $i \in [1, n]$, the assigned value of the variable G_i must then appear at least α times in the array $G_1, \ldots, G_n$, i.e. $\#\{j \mid G_j = G_i\} \geq \alpha$.
- Constraint to express the objective function. Different optimisation criteria are available: minimising the maximal diameter D of the clusters, maximising the minimal split S between clusters, minimising the within-cluster sum of dissimilarities WCSD or minimising the within-cluster sum of squares WCSS. A global optimisation constraint is developed for each criterion along with a filtering algorithm. For instance, if the user chooses to optimise the sum of squares, the variable V will be linked by the constraint WCSS$([G_1, \ldots, G_n], V, d)$.

In order to improve the performance of CP solvers, different search strategies are elaborated for each criterion. For example, a CP-based framework using repetitive branch-and-bound search has been developed (Guns et al, 2016) for the WCSS criterion.

Another interest of the declarative framework is the bi-objective constrained clustering problem. This problem aims to find clusters that are both compact (minimising the maximal diameter) and well separated (maximising the split), under user constraints. In (Dao et al, 2017) it is shown that to solve this problem, the framework can be used by iteratively changing the objective function and adding constraints on the other objective value.

This framework has been extended to integrate the four categories of user constraints on properties (cardinality, density, geometric, and complex logic), in order to make clustering actionable (Dao et al, 2016). Schemes are developed to expresses the categories using CP constraints. For instance, a density constraint provides bounds on the occurrence of some properties on a subset of instances in each cluster. To express this constraint, for each instance $o_i \in \mathcal{O}$ which is eligible (e.g. more than 20 years old), the set of neighbourhood instances $N(i)$ (e.g. persons having the same hobby) is determined. The number of instances of $N(i)$ in the same cluster as o_i can be captured using a variable Z_i, which is linked by the constraint

$$\#\{j \in N(i) \mid G_j = G_i\} = Z_i.$$

Arithmetic constraints are then stated on Z_i to express density constraints. As an example, for the constraint that each person more than 20 years old must be in the same group as at least 5 people sharing the same hobby, the constraint $Z_i \geq 6$ (5 other instances and the instance o_i itself) is included.

Several CP frameworks have been developed tor conceptual clustering (Khiari et al, 2010; Guns et al, 2013; Chabert and Solnon, 2017). These frameworks integrate constraints on patterns or groups of patterns as well as different optimisation criteria, for instance, maximising the minimal size of the clusters, or maximising the minimal size of the patterns defining the clusters. The models are developed using binary variables (Guns et al, 2013) or set variables (Khiari et al, 2010; Chabert and Solnon, 2017).

5 Collaborative Constrained Clustering

Over the last fifty years, a huge number of new clustering algorithms have been developed, and existing methods have been modified and improved (Jain et al, 1999; Rauber et al, 2000; Xu and Wunsch, 2005). This abundance of methods can be explained by the ill-posed nature of the problem—each clustering algorithm is biased by its objective function used to build the clusters. Consequently, different methods can produce very different clustering results from the same data. Furthermore, the same algorithm can produce different results depending upon its parameters and

initialisation. A relatively recent approach to circumvent this problem considers that the information offered by different sources and different clusterings are complementary (Kittler, 1998). A single clustering is produced from the results of methods that have different points of view and each individual clustering opinion is used to find a consensual decision. Thus, the combination of different clusterings may increase their efficiency and accuracy. Each decision can be processed from a different source or media. The final result can be produced directly from the independently obtained results (ensemble clustering) or from the result of a collaborative process (collaborative clustering).

5.1 Ensemble Clustering

Ensemble clustering methods aim to improve the overall quality of the clustering by reducing the bias of each single algorithm (Hadjitodorov and Kuncheva, 2007). An ensemble clustering is composed of two steps. First, multiple clusterings are produced from a set of methods having different points of view. These methods can be different clustering algorithms (Strehl and Ghosh, 2002) or the same algorithm with different parameter values or initialisations (Fred and Jain, 2002). The final result is derived from the independently obtained results by applying a consensus function.

Constraints can be integrated in two manners: each learning agent integrates them in its own fashion; or applying them in the consensus function. The former approach faces an important dilemma: either favor diversity or quality. High quality is desired, but the gain of ensemble clustering is derived from diversity (thus avoiding biased solutions). Clustering from constrained algorithms tends to have a low variance, which implies low diversity (Yang et al, 2017), especially when using the same set of constraints. Therefore the advantage of ensemble clustering is limited.

Implementations of the first approach exist (Yu et al, 2011; Yang et al, 2012). For example Iqbal et al (2012) develop the semi-supervised clustering ensembles by voting (SCEV) algorithm, in which diversity is balanced by using different types of semi-supervised algorithms (i.e. constrained k-Means, COP-KMeans, SP-Kmeans, etc.). In the first step each semi-supervised agent computes a clustering given the data and the set of constraints. It then combines all the results using a voting algorithm after having relabeled and align the different clustering results. The authors propose to integrate a weight for each agents' contributions into the voting algorithm. This weight is a combination of two sub-weights, the first one is defined a priori, based upon the expert's trust of each agent according to the data (i.e. seeded k-Means is more efficient for noise, COP-Means and constraints are more efficient if the data is noise free), the second is also user defined but based upon the user's feedback on the clustering result. As such, the algorithm allows more flexibility and user control over the clustering.

The second approach focuses on applying constraints in the consensus function (Al-Razgan and Domeniconi, 2009; Xiao et al, 2016; Dimitriadou et al, 2002). These

algorithms start by generating the set of clusterings from the clustering agents. The constraints are then integrated in the consensus function, which can be divide in 4 steps:

1. generate a similarity matrix from the set of clusterings;
2. construct a sparse graph from this similarity matrix using the CHAMELEON algorithm—an edge is constructed between two vertices if the value in the similarity matrix is greater than zero for the corresponding elements;
3. partition the graph into a large number of sub-clusters using the METIS method;
4. merge the sub-clusters using an agglomerative hierarchical clustering approach by finding the most similar pair of sub-clusters.

Constraints are integrated during partitioning. Cannot-link constraints are used as priorities for the split operation—sub-clusters that contains a CL constraints are partitioned until the two elements in the constraint are allocated to two different clusters.

5.2 Collaborative Clustering

Collaborative clustering consists in making multiple clustering methods collaborate to reach an agreement on a data partitioning. While ensemble clustering (and consensus clustering (Monti et al, 2003; Li and Ding, 2008)) focuses on merging clustering results, collaborative clustering focuses on iteratively modifying the clustering results by sharing information between them (Wemmert et al, 2000; Gançarski and Wemmert, 2007; Pedrycz, 2002). In consequence it extends ensemble clustering by adding a refinement step before the unification of the results. For instance, in SAMARAH (Wemmert et al, 2000; Gançarski and Wemmert, 2007) each clustering algorithm modifies its results according to all the other clusterings until all the clusterings proposed by the different methods are strongly similar. Thus, they can be more easily unified through a voting algorithm (for example).

Three stages for integrating user constraints in the collaborative process can be identified (Forestier et al, 2010a):

(1) generation of the final result (by labeling the clusters of the final result using label constraints);
(2) directly in the collaborative clustering (in order to guide the collaborative process);
(3) using constrained agents.

Integrating user constraints into the learning agents (3) is complex because it requires extensive modification of each of the clustering methods involved. The complexity of integrating constraints in the collaboration (2) depends on how information is exchanged between the learning agents. Integrating the constraints after collaboration (1), however, does not interfere in the collaborative process, which makes it easier to implement.

To illustrate the second level, the SAMARAH method is first introduced. Then Section 5.2.2 presents the method for integrating constraints into the collaborative process.

5.2.1 SAMARAH: a Framework for Collaborative Multistrategy Clustering

SAMARAH (Forestier et al, 2010a) is based on the principle of mutual and iterative refinement of multiple clustering algorithms. The process can be decomposed into three main steps:

1. The generation of the initial results;
2. The refinement of the different results;
3. The combination of the refined results.

The first step consists of generating the initial results that will be used during the process. In this step, different algorithms or the same algorithm with different parameters can be used. During the refinement stage, each result is compared to the set of results proposed by the other methods, the goal being to evaluate the similarity between the different results in order to observe differences in the clusterings. Once these differences (named conflicts) are identified, the objective is to modify the clusterings to reduce them, in addition to the number of constraints violations, i.e. resolving the conflicts (Forestier et al, 2010b). These are resolved by either merging clusters, splitting clusters, or re-clustering clusters iteratively. This step can be seen as a questioning each result according to the information provided by the other actors in the collaboration and the background knowledge. After multiple iterations of refinement (in which a local similarity criterion is used to evaluate whether the modifications of a pair of results is relevant (Forestier et al, 2010a)), the results are expected to be more similar than before the collaboration began. During the third and final step, the refined results are combined to propose a final and unique result (which is simplified due to the similarity of the results).

5.2.2 Knowledge Integration in the SAMARAH Collaborative Method

During the refinement step of the SAMARAH method, a local similarity criterion $\gamma^{i,j}$ is used to evaluate whether the proposed modification of a pair of results is relevant (Forestier et al, 2010a). This criterion includes a quality criterion δ^i which represents the quality of the result $\mathscr{R}^i$. It therefore balances the refinement between the similarity and the quality of the expected results. It is computed for two aspects of the results: the internal and external qualities. The internal evaluation consists of evaluating the quality of the result through an unsupervised measure. The external evaluation consists of evaluating the quality of the result according to external knowledge, such as an estimation of the number of clusters, some labeled samples, or some constraints.

The original version of SAMARAH included internal knowledge but the only external knowledge was an estimate of the number of clusters. To take into account additional external knowledge, the quality criterion has been extended to measure the level of agreement of the results with different kinds of constraints (Forestier et al, 2010b), such that

$$\delta^i = \sum_{c=1}^{N_c} q_c(\mathscr{R}^i) \times p_c, \tag{4}$$

where N_c is the number of constraints to respect, q_c is the criterion used to evaluate the result according to the c-th constraint ($q_c(\cdot) \in [0,1]$) and p_c is the relative importance given by the user to the c-th constraint ($p_1 + p_2 + \cdots + p_{N_c} = 1$). By default, each constraint is given a weight of $\frac{1}{N_c}$.

Thus, any constraint that can be defined as a function taking its values on $[0,1]$ can be integrated into the process. The method to integrate some frequently encountered constraints are as follows.

Cluster quality constraints are based on the intrinsic quality of clusters, such as inertia or predictivity and include the number of clusters. Criterion such as inertia or compacity need to be balanced with an evaluation of the number of clusters. An example of a criterion that includes the quality of the clusters and the number of clusters is as follows:

$$q_{qb}(\mathscr{R}^i) = \frac{p^i}{n^i} \sum_{k=1}^{n^i} \tau_k^i, \tag{5}$$

where n^i is the number of clusters of $\mathscr{R}^i$, τ_k^i defines the internal quality of the k-th cluster, and p^i is the external quality of the result. The internal quality of the k-th cluster is given by

$$\tau_k^i = \begin{cases} 0, & \text{if } \frac{1}{n_k^i} \sum_{l=1}^{n_k^i} \frac{d(x_{k,l}^i, g_k^i)}{d(x_{k,l}^i, g^i)} > 1, \\ 1 - \frac{1}{n_k^i} \sum_{l=1}^{n_k^i} \frac{d(x_{k,l}^i, g_k^i)}{d(x_{k,l}^i, g^i)}, & \text{otherwise,} \end{cases} \tag{6}$$

where n_k^i is the cardinality of $\mathscr{C}_k^i$, g_k^i is the gravity center of $\mathscr{C}_k^i$, g^i is the gravity center of the closest cluster to $x_{k,l}^i$ and d is the distance function. The measure is computed on each cluster to evaluate the overall quality of the clustering result. To take into account the number of clusters n^i, the criterion p^i is defined as, such that

$$p^i = \frac{n_{\text{sup}} - n_{\text{inf}}}{|n_i - n_{\text{inf}}| + |n_{\text{sup}} - n_i|}, \tag{7}$$

where $[n_{\text{inf}}, n_{\text{sup}}]$ is the range of the expected number of clusters which is given by the user.

Class labeld constraints correspond to the case where a set of labeled samples is available. To evaluate the agreement between results and such constraints, we can use any index which enables us to evaluate the similarity between a clustering and a labeled classification (where all the classes are known, and each object belongs to one

of these classes). To achieve this, it is only necessary to compare results with a given partial partition $\mathbb{R}$ which represents the known labeled objects. In the SAMARAH method, the Rand index (Rand, 1971) or the WG agreement index (Wemmert et al, 2000) is used.

The Rand index is a measure of similarity between two data partitions, such that

$$\text{Rand}(\mathscr{R}^i,\mathbb{R}) = \frac{a+b}{\binom{n}{2}}, \tag{8}$$

where n is the number of objects to classify, a is the number of pairs of objects which are in the same cluster in $\mathscr{R}^i$ and in the known result, and b is the number of pairs of objects which are not in the same cluster in the proposed result $\mathscr{R}^i$ nor in the known result $\mathscr{R}^j$. The sum of these two measurements (a and b) can been seen as the number of times that the two partitions are in agreement. This index takes values in $[0,1]$, where 1 indicates that the two partitions are identical. A constraint $q_{rand}(\mathscr{R}^i)$ can therefore be defined, such that

$$q_{rand}(\mathscr{R}^i) = \text{Rand}(\mathscr{R}^i,\mathbb{R}). \tag{9}$$

The WG agreement index is defined by

$$WG(\mathscr{R}^i,\mathbb{R}) = \frac{1}{n}\sum_{k=1}^{n_i} S\left(C_k^i,\mathscr{R}^j\right)\left|C_k^i\right|, \tag{10}$$

where n is the number of objects to classify and $\mathscr{R}^j$ is the reference partition (e.g. labeled classification, another clustering, etc.). This index takes values in $[0,1]$, where 1 indicates that all the objects in the clustering $\mathscr{R}^i$ are well classified according to the object labels in $\mathscr{R}^j$. A constraint $q_{wg}(\mathscr{R}^i)$ can therefore be defined, such that

$$q_{wg}(\mathscr{R}^i) = WG(\mathscr{R}^i,\mathbb{R}). \tag{11}$$

Link constraints correspond to the case where knowledge is expressed as must-link and cannot-link constraints between objects (see Section 2.2). In this case, the ratio of respected to violated constraints can easily be computed such that

$$q_{link}(\mathscr{R}^i) = \frac{1}{n_r}\sum_{j=1}^{n_r} v(\mathscr{R}^i,l_j), \tag{12}$$

where n_r is the number of constraints between the objects, l_j is a *must-link* or *cannot-link* constraint and $v(\mathscr{R}^i,l_j) = 1$ if $\mathscr{R}^i$ respects the constraint l_j and 0 otherwise.

Note that such constraints can be extracted from class-label constraints. For example, a must-link constraint can be created for all pairs of objects belonging to the same cluster, and a cannot-link constraint can be created for all pairs of objects belonging to different clusters.

6 New Trends

Obtaining useful results with pattern mining methods remains a difficult task. Careful tuning of the algorithm parameters and filtering of the results are needed. This requires considerable effort and expertise from the data analyst. As a consequence, the idea of interactive or exploratory data mining has been proposed (van Leeuwen, 2014). Exploratory data mining looks for models and patterns that explain the data as much as possible by developing user interaction to influence the search and the results.

This section deals with these trends concerning clustering. Scientific challenges (Section 6.1.1), a example of user interaction (Section 6.1.2), and an example of incremental and collaborative clustering (Section 6.1.3) are given. Limitations of the constraint paradigm are sketched and, moving beyond constraints, exploratory data mining is discussed. It is shown that preferences are a way to address pattern mining tasks (Section 6.2.1) and exploratory data mining enables the capturing of implicit preferences (Section 6.2.2). Chapter 3 of Volume 3 describes preferences queries in the field of database.

6.1 Interactive and Incremental Constrained Clustering

6.1.1 Challenges

Preliminary studies have revealed numerous scientific challenges about the objectives mentioned in the previous sections. For example, it is necessary to study and detail the thematic constraints (i.e. of the domain of application) that the expert may formulate to guide the process.

These constraints can be extremely broad and have to be translated into actionable constraints. In the current state of the knowledge, they are generally limited to constraints that can be directly translated into comparison constraints such as ML/CL, labeling constraints, or constraints in terms of cluster number or cluster size. Thus, for example, the following can be accepted: “these two objects seem to be of the same nature”, or “these two ensembles of objects are of the same nature” (ML constraints between all the pairs of objects of the two sets); “these three sets are totally different” (CL constraint on all the pairs of objects from the three sets); “this object is of type C” (labeling constraint); and “a cluster cannot represent more than 20% of the image” (constraints of cluster size).

Generating actionable constraints from a set, however, can rapidly lead to a significant increase of combinatorial complexity. For example, a constraint “of the same nature” on two sets of size N_1 and N_2 respectively will, using the naive approach, generate $N_1(N_1-1)+N_2(N_2-1)+N_1\times N_2$ ML constraints. These are transitive (Wagstaff et al, 2001) however, and therefore the number of constraints needed to satisfy the user requirement can be reduced to $(N_1+N_2)-1$ under the assumption of guaranteed constraint satisfaction. Expressing all the constraints can be very time

consuming, particularly in the case of data mining in big-data problems where the size of sets N_1 and N_2 can be considerable. The following problems therefore need to be tackled.

- How to design algorithms that are able to deal with large constraint sets?
  - Reduce the size of the model by limiting the number of considered elements, for example by sampling or by identifying irrelevant objects.
  - Reduce the number of constraints without loss of quality:
    - sample the constraints or sample the objects under constraint;
    - identify the categories of constraints and study search strategies.
  - Relax the optimality of the solution using a threshold on the execution time (which is easy to choose but cannot guarantee the result's quality).
  - Use a local instead of a global search.
- How to limit the number of constraints—ideally, to define a minimal set of constraints—and how to use incremental approaches to allow a user to give such a set?

6.1.2 Interactive Clustering

In an interactive clustering model, the algorithm proposes a clustering of the data to the user and receives some feedback on the current solution. Taking the feedback into account, the algorithm makes changes to the clustering and proposes the new result to the user. This step is iterated until the user is satisfied with the clustering. The improvements to the clustering given by the user are usually in the form of splitting or merging clusters (Balcan and Blum, 2008; Cutting et al, 1992; Awasthi and Zadeh, 2010; Awasthi et al, 2017). The aim of efficient algorithms is to require as little user interaction as possible. User feedback can also be the rejection of some clusters and to request new ones. The system returns another clustering, which is chosen to fit the data as well as possible, while avoiding the creation of a cluster that is similar to any of those previously rejected. Formalising this in a Bayesian framework, after the user rejects a set of clusters, the prior distribution over model parameters is modified to severely downweight regions of the parameter space that would lead to clusters that are similar to those previously rejected (Srivastava et al, 2016). Another kind of feedback is that associated with each cluster: the user can lock the cluster (it is therefore no longer modified), refine the cluster by adding or removing elements, or change the pairwise distance of elements within a cluster. The distance between elements is recomputed accordingly, the unlocked clusters are reclustered and the process is repeated until no unlocked clusters remain (Coden et al, 2017). Interaction with the user can also be in made at different stages of the clustering process, as in an interactive approach for app search clustering in the Google Play Store, which incorporates human input in a knowledge-graph-based clustering process (Chang et al, 2016). Instead of directly clustering apps, the algorithm extracts topic labels from search results, runs clustering on a semantic graph of the topic labels and assigns search results to topic clusters. The interactive interface lets domain experts

steer the clustering process in different stages: refining the input to the clustering algorithm, steering the algorithm to generate more or less fine-grained clusters, and finally editing topic label clusters and topic labels.

User feedback through constraints on the clustering makes the clustering actionable to some purposes. Let us consider the case where the user already has a clustering, which was obtained by their favourite clustering algorithm. The clustering in general is good but there are a few undesirable properties. One ad hoc way to fix this is to re-run the constrained clustering algorithm but there is no guarantee that the obtained clustering will be as good as the first one. Instead, Kuo et al (2017) propose to *minimally* modify the existing clustering while reaching the desired properties given by user feedback. User feedback can be, for example, splitting or merging some clusters, stating a bound on the diameter of the clusters, or stating a bound on the size of the clusters to be more balanced.

6.1.3 Incremental Constrained Clustering

Many constrained clustering methods require that the complete set of constraints be given before running the algorithm. This is very often unrealistic. For example, when a geographer tries to extract relevant clusters from a remote sensing image, it is almost impossible, given the number of clusters and the large space of possible constraints, to give such constraints a priori. Indeed, experiments show that the user will tend to give "obvious" and not informative constraints (e.g., pixels in the same homogeneous region must be clustered together or pixels of roads and pixels of vegetation cannot be in the same cluster, even if the "colors" of these pixels are sufficient to decide)—these constraints will not impact the algorithm in any way. In other words, the algorithm will find the same result regardless of the constraints. A way to tackle this problem would be to follow the example of interactive supervised learning methods and allow the user to inject constraints based on the results obtained (Davidson et al, 2007). In this manner the algorithm could reduce its uncertainty by obtaining new constraints that are related to uncertain zones (e.g., cluster edges, areas of high object density, etc.). These, highly informative, constraints could be proposed to the user who may validate them, or not, according to their knowledge. The hope is that this approach (selectively choosing constraints) produces significantly better results when compared to randomly choosing constraints. Cohn et al (2003) describes an experiment (introduced by Davidson et al (2007)) concerning document clustering, in which ten constraints incrementally chosen by a user (not by the machine) produced as good results as using between 3000 and 6000 randomly chosen constraints. In the same way our geographer could add label constraints ("The region belongs to the thematic water class?"), ML/CL constraints ("These regions should be together/apart?") or cluster constraint ("Should this cluster be removed?") according to the results obtained and the proposals of the algorithm.

Even with research that is concerned with defining the conditions for the application of such methods (Raj et al, 2013; Vu and Labroche, 2017), a large number of scientific obstacles still remain to be overcome:

- How to evaluate the informativeness of a constraint (some progress in this area has been made by Davidson et al (2006) and Wagstaff et al (2006))?
- How to integrate new constraints while limiting the effects on the results (a strong modification of the result can confuse the user)?
- How to concretely design incremental algorithms, not in terms of new data or interactive cluster modifications but in terms of new constraints?
- How to remove a constraint already taken into account without starting afresh?
- How to deal with inconsistent constraints: should a new constraint be rejected if it is inconsistent with a previously added constraint or should the previously added constraint be removed?

As such, the development of incremental interactive constrained clustering method remains a very challenging research problem.

6.2 Beyond Constraints: Exploratory Data Analysis

6.2.1 From Constraints to Explicit Preferences

The notion of constraints is at the core of numerous works in pattern mining as presented in this chapter. Nevertheless, constraint-based pattern mining assumes that the user is able to express what they are looking for, requires finely tuning thresholds, and a collection of patterns that are often too large to be truly exploited. This picture may explain why preferences in pattern mining become more and more important. Preferences in pattern mining do not arise from nothing. In constraint-based pattern mining, the utility functions measure the interest of a pattern and can be seen as a quantitative preference model (Yao and Hamilton, 2006; Fürnkranz et al, 2012; Geng and Hamilton, 2006). Many other mechanisms have been developed such as mining the most interesting patterns with one measure, top-k patterns (Wang et al, 2005), or more, skyline patterns (Cho et al, 2005; Soulet et al, 2011; van Leeuwen and Ukkonen, 2013); reducing redundancy by integrating subjective interestingness (Gallo et al, 2007; Bie, 2011; De Bie, 2013; van Leeuwen et al, 2016); and putting the pattern mining task to an optimisation problem.

Even though it has been realised for a long time that it is difficult for a data analyst to model their interest in terms of constraints and overcome the well-known thresholding issue, researchers have only recently intensified their study of methods for finding high-quality patterns according to a user's preferences. We shortly introduce the example of the skylines patterns (Cho et al, 2005; Soulet et al, 2011; van Leeuwen and Ukkonen, 2013) which can be seen as a generalisation of the well known top-k patterns (Wang et al, 2005; Ke et al, 2009).

Top-k patterns integrate user preferences in the form of a score in order to limit the number of extracted patterns. By associating each pattern with a rank score, this approach returns an ordered list of the k patterns with the highest score to the user. Nevertheless, top-k patterns suffer from the diversity issue (top-k patterns tend to be similar) and the performance of top-k approaches is often sensitive to the size of the

datasets and to the threshold value, k. Even worst, combining several measures into a single scoring function is difficult.

Skyline patterns introduce the idea of skyline queries (Börzsönyi et al, 2001) into the pattern discovery framework . Such queries have attracted considerable attention due to their importance in multi-criteria decision making, where they are usually called "Pareto efficiency" or "optimality queries". Briefly, in a multidimensional space where a preference is defined for each dimension, a point a dominates another point b if a is better (i.e. more preferred) than b in at least one dimension, and a is not worse than b in every other dimension. For example, a user selecting a set of patterns may prefer a pattern with a high frequency, a large length, and a high confidence. In this case, we say that pattern a dominates another pattern b if frequency$(a) \geq$ frequency(b), length$(a) \geq$ length(b) and confidence$(a) \geq$ confidence(b), where at least one strict inequality holds. Given a set of patterns, the skyline set contains the patterns that are not dominated by any other pattern. Skyline pattern mining is interesting for several reasons. First, skyline processing does not require any threshold tuning. Second, for many pattern mining applications it is often difficult (or impossible) to find a reasonable global ranking function. Thus the idea of finding all optimal solutions in the pattern space with respect to multiple preferences is appealing. Third, the formal property of dominance satisfied by the skyline pattern defines a global interestingness measure with semantics easily understood by the user. While the notion of skylines has been extensively developed in engineering and database applications, it has remained unused for data mining purposes until recently (Cho et al, 2005; Soulet et al, 2011). In this kind of approach, preferences (or measures) are explicitly given by the user.

6.2.2 From Explicit Preferences to Implicit Preferences

All of the approaches introduced in the previous section assume that preferences are explicit and given in the process. In practice, the user only has a vague idea of which patterns could be useful and there is therefore a need to elicit preferences. The recent research field of interactive pattern mining relies on the automatic acquisition of these preferences (van Leeuwen, 2014). Basically, its principle is to repeat a short mining loop centred on the user: (1) the user poses an initial query to the system, which returns an initial result, (2) the user designates components or aspects of this result as (un)desirable/(un)interesting, (3) the system translates the user feedback into a model of the user's preferences and uses this model to adapt its search strategy, (4) a new result is produced and the process returns to step (2). At each iteration, only some patterns are mined and the user has to indicate those which are relevant (Dzyuba et al, 2014) by, for example, liking/disliking, rating, or ranking. The user feedback improves an automatically learned model of preferences that will refine the pattern mining step in the next iteration. A great advantage is that the user does not have to explicitly state their preference model.

Interactive pattern mining raises several challenges. The first being the design of user feedback options. The easiest forms of feedback to take into consideration

are explicit and more or less binary in nature: the requirement that certain instances be grouped together or kept apart, or that particular descriptors are included in a cluster's description. Unfortunately, experts are unlikely to be able to give this kind of feedback, especially early on in the discovery process. A second, less explicit but still easy to use, form of feedback allows the user to designate components of the result, e.g. descriptions, as interesting or uninteresting or to express preferences, denoting a (component of a) result as more interesting than another (Dzyuba et al, 2014). The first form of feedback translates into constraints that can be included by prototypes, meaning that they can be included into the system. The second form requires a notion of equivalence/alternative to returned descriptions to replace uninteresting ones or to allow the user to express a preference over pairs. This requires multiple characterisations and to integrate pairwise comparisons to take preferences over pairs into account. A long-term goal is how to elicit and learn a preference model.

Of equal importance is the design of methods following the principle of *instant data mining* (Boley et al, 2011) to avoid the expert user "checking-out" of the process. Each iteration must be fast and the result must be provided in a concise form so that the user is not overwhelmed with a huge collection of patterns that are impossible to analyse. Instant data mining is based on sampling techniques and provides a representative set of patterns without explicitly searching in the pattern space. These techniques, however, handle a limited set of measures or constraints (Giacometti and Soulet, 2016).

Subjective interestingness is a way of exploiting user feedback to directly influence search. Interactive diverse subgroup discovery (IDSD) (Dzyuba and van Leeuwen, 2013) is an interactive algorithm that allows a user to provide feedback with respect to provisional results to avoid subgroups corresponding to common knowledge, which is usually uninteresting to a domain expert. The beam selection strategy is made interactive on each level of the search, thus the interestingness measure becoming subjective. The one click mining system (Boley et al, 2013) extracts local patterns through a mechanism based on two types of preferences. One is used to allocate the computation time to different mining algorithms, the other is used to learn a utility function to compute a ranking over all mined patterns. Both learning algorithms rely on inputs corresponding to implicit user feedback.

A formal framework for exploratory data mining is proposed by De Bie (2011), who argues that traditional objective quality measures are of limited practical use and proposes a general framework that models background knowledge. Subjective interestingness is formalised by information theory. The principle is to consider prior beliefs, e.g. background information, as constraints on a probabilistic model representing the uncertainty of the data. Given the prior beliefs, the maximum entropy distribution is used to model the data. Then one can compute how informative a pattern is given the current model. This framework follows the iterative data mining process: starting from a MaxEnt model based on prior beliefs, the subjectively most interesting pattern is searched for and added to the model, after which one can start looking for the next pattern. The exact implementation depends on the specific data and pattern types.

Finally, approaches based on the declarative modelling paradigm (see Section 4 in this chapter) help exploratory data analysis. Indeed, the data analyst focuses on the specification of the desired results through constraints (and optionally an optimisation criterion) rather than describing how the solution should be computed. The assumption of the declarative modelling paradigm is that a data analyst is able to express constraints that can be iteratively added to the declarative model.

7 Conclusions and Perspectives

While the development and generalisation of deep learning approaches is revolutionising supervised learning, particularly in the area of decision support, attention is becoming increasingly focused on unsupervised learning. This is due, in part, to the pressing need for methods that allow one to explore large data sets without well-defined preconceptions, and without predefined categories that can be used as labels for training instances. Nevertheless, without any supervision, these approaches can lead to irrelevant results. A way to circumvent this problem is to (re)introduce experts into the analysis process and thus, to define methods capable of taking into account domain knowledge without the fastidious preliminary step of sample annotation.

In this chapter, we have presented the alternative approach to clustering in which the process is guided by user constraints in order to produce more relevant results. Here 'relevant' means more directly matched to the expert's thematic intuition, that is to say to potential thematic classes. Methods derived from this approach have been developed and have demonstrated their effectiveness and applicability in many areas.

Despite the increasing number of methods and tools dedicated to constrained clustering and the surge of interest in constrained clustering, this paradigm is still surprisingly infrequently used. An explanation for this is the issues that remain to be explored and addressed. Without claiming to be exhaustive, this chapter has listed some of the scientific obstacles to be overcome: how to define more expressive operable constraints, for instance constraints involving more objects ("A is closer to B than to C") or conditional constraints ("If A is with B then C cannot be with B")? How to deal with increasing volumes of data, which can lead to an explosion of the number of constraints? How to design incremental interactive methods that are able to deal with incoherent constraints?

Nevertheless, while these issues are important and must be addressed by computer scientists, it is convincingly apparent that the main obstacle preventing the adoption of these approaches is the lack of theoretical and practical understanding of the "translation" of the an expert's knowledge into actionable constraints. As such, research effort should be focused, on the one hand, on the ways of translating domain knowledge into thematic constraints and, on the other hand, on the automatic translation of such constraints into actionable constraints.

This survey chapter has presented the principles of constrained clustering. This exciting field has arrived at a time when solutions to knowledge discovery problems

in big data are needed, and it promises to offer these. The time is therefore ripe for increasing the exposure and use of these methods. All the while, it is also a domain full of questions, some of which go beyond the current theory of statistical learning, and answering these questions promises to stimulate interesting and innovative research directions.

References

Al-Razgan M, Domeniconi C (2009) Clustering ensembles with active constraints. In: Okun O, Valentini G (eds) Applications of Supervised and Unsupervised Ensemble Methods, Springer Berlin Heidelberg, chap 10, pp 175–189

Aloise D, Deshpande A, Hansen P, Popat P (2009) NP-hardness of Euclidean sum-of-squares clustering. Machine Learning 75(2):245–248

Aloise D, Hansen P, Liberti L (2012) An improved column generation algorithm for minimum sum-of-squares clustering. Mathematical Programming 131(1–2):195–220

Alzate C, Suykens J (2009) A regularized formulation for spectral clustering with pairwise constraints. In: Proceedings of the International Joint Conference on Neural Networks, pp 141–148

Anand R, Reddy C (2011) Graph-based clustering with constraints. In: Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp 51–62

Anand S, Bell D, Hughes J (1995) The role of domain knowledge in data mining. In: Proceedings of the International Conference on Information and Knowledge Management, pp 37–43

Awasthi P, Zadeh RB (2010) Supervised clustering. In: Proceedings of the Advances in Neural Information Processing Systems, pp 91–99

Awasthi P, Balcan MF, Voevodski K (2017) Local algorithms for interactive clustering. The Journal of Machine Learning Research 18:1–35

Babaki B, Guns T, Nijssen S (2014) Constrained clustering using column generation. In: Proceedings of the International Conference on AI and OR Techniques in Constriant Programming for Combinatorial Optimization Problems, pp 438–454

Balcan MF, Blum A (2008) Clustering with interactive feedback. In: Proceedings of the International Conference on Algorithmic Learning Theory, pp 316–328

Banerjee A, Ghosh J (2006) Scalable clustering algorithms with balancing constraints. Data Mining and Knowledge Discovery 13(3):365–395

Bar-Hillel A, Hertz T, Shental N, Weinshall D (2003) Learning distance functions using equivalence relations. In: Proceedings of the International Conference on Machine Learning, pp 11–18

Bar-Hillel A, Hertz T, Shental M, Weinshall D (2005) Learning a Mahalanobis metric from equivalence constraints. Journal of Machine Learning Research 6:937–965

Basu S, Banerjee A, Mooney R (2002) Semi-supervised clustering by seeding. In: Proceedings of the International Conference on Machine Learning, pp 19–26

Basu S, Banerjee A, Mooney R (2004a) Active semi-supervision for pairwise constrained clustering. In: Proceedings of the SIAM International Conference on Data Mining, pp 333–344

Basu S, Bilenko M, Mooney R (2004b) A probabilistic framework for semi-supervised clustering. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 59–68

Basu S, Davidson I, Wagstaff K (2008) Constrained Clustering: Advances in Algorithms, Theory, and Applications, 1st edn. Chapman & Hall/CRC

Beldiceanu N, Carlsson M, Rampon JX (2005) Global constraint catalog. Tech. Rep. T2005-08, SICS and EMN Technical Report

Bellet A, Habrard A, Sebban M (2015) Metric Learning. Morgan & Claypool Publishers

Berg J, Järvisalo M (2017) Cost-optimal constrained correlation clustering via weighted partial maximum satisfiability. Artificial Intelligence 244:110–142

Bie TD (2011) Maximum entropy models and subjective interestingness: an application to tiles in binary databases. Data Mining and Knowledge Discovery 23(3):407–446

Bilenko M, Mooney R (2003) Adaptive duplicate detection using learnable string similarity measures. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 39–48

Bilenko M, Basu S, Mooney R (2004) Integrating constraints and metric learning in semi-supervised clustering. In: Proceedings of the International Conference on Machine Learning, pp 11–18

Boley M, Lucchese C, Paurat D, Gärtner T (2011) Direct local pattern sampling by efficient two-step random procedures. In: Proceedings of the ACM SIGKDD International Conference on Knowledge discovery and data mining, pp 582–590

Boley M, Mampaey M, Kang B, Tokmakov P, Wrobel S (2013) One click mining: interactive local pattern discovery through implicit preference and performance learning. In: Proceedings of the ACM SIGKDD Workshop on Interactive Data Exploration and Analytics, pp 27–35

Börzsönyi S, Kossmann D, Stocker K (2001) The skyline operator. In: Proceedings of the International Conference on Data Engineering, pp 421–430

Boulicaut JF, De Raedt L, Mannila H (eds) (2006) Constraint-based mining and inductive databases, Lecture Notes in Artificial Intelligence, vol 3848. Springer-Verlag

Bradley P, Bennett K, Demiriz A (2000) Constrained k-means clustering. Tech. Rep. MSR-TR-2000-65, Microsoft Research

Chabert M, Solnon C (2017) Constraint programming for multi-criteria conceptual clustering. In: Proceedings of the International Conference on Principles and Practice of Constraint Programming, pp 460–476

Chang S, Dai P, Hong L, Sheng C, Zhang T, Chi E (2016) AppGrouper: Knowledge-based interactive clustering tool for app search results. In: Proceedings of the International Conference on Intelligent User Interfaces, pp 348–358

Chen W, Feng G (2012) Spectral clustering: a semi-supervised approach. Neurocomputing 77(1):229–242

Cheng H, Hua K, Vu K (2008) Constrained locally weighted clustering. Proceedings of the VLDB Endowment 1(1):90–101

Cho M, Pei J, Wang H, Wang W (2005) Preference-based frequent pattern mining. International Journal of Data Warehousing and Mining 1(4):56–77

Coden A, Danilevsky M, Gruhl D, Kato L, Nagarajan M (2017) A method to accelerate human in the loop clustering. In: Proceedings of the SIAM International Conference on Data Mining, pp 237–245

Cohn D, Caruana R, Mccallum A (2003) Semi-supervised clustering with user feedback. Tech. Rep. TR2003-1892, Department of Computer Science, Cornell University

Cucuringu M, Koutis I, Chawla S, Miller G, Peng R (2016) Simple and scalable constrained clustering: A generalized spectral method. In: Proceedings of the International Conference on Artificial Intelligence and Statistics, pp 445–454

Cutting D, Pedersen J, Karger D, Tukey J (1992) Scatter/gather: A cluster-based approach to browsing large document collections. In: Proceedings of the international ACM SIGIR conference on Research and development in information retrieval, pp 318–329

Dao TBH, Duong KC, Vrain C (2013) A declarative framework for constrained clustering. In: Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp 419–434

Dao TBH, Vrain C, Duong KC, Davidson I (2016) A framework for actionable clustering using constraint programming. In: Proceedings of the European Conference on Artificial Intelligence, pp 453–461

Dao TBH, Duong KC, Vrain C (2017) Constrained clustering by constraint programming. Artificial Intelligence 244:70–94

Davidson I, Basu S (2007) A survey of clustering with instance level constraints. ACM Transactions on Knowledge Discovery on Data 77(1):1–41

Davidson I, Ravi S (2005) Clustering with constraints: Feasibility issues and the k-means algorithm. In: Proceedings of the SIAM International Conference on Data Mining, pp 138–149

Davidson I, Ravi S (2006) Identifying and generating easy sets of constraints for clustering. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 336–341

Davidson I, Ravi S (2007) The complexity of non-hierarchical clustering with instance and cluster level constraints. Data Mining Knowledge Discovery 14(1):25–61

Davidson I, Wagstaff K, Basu S (2006) Measuring constraint-set utility for partitional clustering algorithms. In: European Conference on Principles of Data Mining and Knowledge Discovery, pp 115–126

Davidson I, Ester M, Ravi S (2007) Efficient incremental constrained clustering. In: Proceedings of the ACM SIGKDD international conference on Knowledge discovery and data mining, pp 240–249

Davidson I, Ravi S, Shamis L (2010) A SAT-based framework for efficient constrained clustering. In: Proceedings of the SIAM International Conference on Data Mining, pp 94–105

De Bie T (2011) An information theoretic framework for data mining. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 564–572

De Bie T (2013) Subjective interestingness in exploratory data mining. In: Proceedings of the International Symposium on Intelligent Data Analysis, pp 19–31

Delattre M, Hansen P (1980) Bicriterion cluster analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-2(4):277–291

Demiriz A, Bennett K, Embrechts M (1999) Semi-supervised clustering using genetic algorithms. In: Proceedings of the Conference on Artificial Neural Networks in Engineering, pp 809–814

Demiriz A, Bennett K, Bradley P (2008) Using assignment constraints to avoid empty clusters in k-means clustering. In: Basu S, Davidson I, Wagstaff K (eds) Constrained Clustering: Advances in Algorithms, Theory, and Applications, 1st edn, Chapman & Hall/CRC, chap 9, pp 201–220

Dimitriadou E, Weingessel A, Hornik K (2002) A mixed ensemble approach for the semi-supervised problem. In: Proceedings of the International Conference on Artificial Neural Networks, pp 571–576

Ding S, Qi B, Jia H, Zhu H, Zhang L (2013) Research of semi-supervised spectral clustering based on constraints expansion. Neural Computing and Applications 22:405–410

Dinler D, Tural M (2016) A survey of constrained clustering. In: M C, K A (eds) Unsupervised Learning Algorithms, Springer, chap 9, pp 207–235

Dzyuba V, van Leeuwen M (2013) Interactive discovery of interesting subgroup sets. In: Proceedings of the International Symposium on Intelligent Data Analysis, pp 150–161

Dzyuba V, van Leeuwen M, Nijssen S, De Raedt L (2014) Interactive learning of pattern rankings. International Journal on Artificial Intelligence Tools 23(6):1460,026

Ester M, Kriegel H, Sander J, Xu X (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the International Conference on Knowledge Discovery and Data Mining, pp 226–231

Fayyad U, Piatetsky-Shapiro G, Smyth P (1996) From data mining to knowledge discovery: an overview. In: Fayyad U, Piatetsky-Shapiro G, Smyth P, Uthurusamy R (eds) Advances in knowledge discovery and data mining, AAAI/MIT Press, chap 1, pp 1–36

Fisher D (1987) Knowledge acquisition via incremental conceptual clustering. Machine Learning 2(2):139–172

Forestier G, Gançarski P, Wemmert C (2010a) Collaborative clustering with background knowledge. Data & Knowledge Engineering 69(2):211–228

Forestier G, Wemmert C, Gançarski P (2010b) Towards conflict resolution in collaborative clustering. In: IEEE International Conference on Intelligent Systems, pp 361–366

Fred ALN, Jain AK (2002) Data clustering using evidence accumulation. Proceedings of the IEEE International Conference on Pattern Recognition pp 276—280

Fürnkranz J, Gamberger D, Lavrač N (2012) Foundations of Rule Learning. Cognitive Technologies, Springer

Gallo A, De Bie T, Cristianini N (2007) Mini: Mining informative non-redundant itemsets. In: Proceedings of the European Conference on Principles of Data Mining and Knowledge Discovery, pp 438–445

Gançarski P, Wemmert C (2007) Collaborative multi-step mono-level multi-strategy classification. Journal on Multimedia Tools and Applications 35(1):1–27

Ganji M, Bailey J, Stuckey P (2016) Lagrangian constrained clustering. In: Proceedings of the SIAM International Conference on Data Mining, pp 288–296

Ge R, Ester M, Jin W, Davidson I (2007) Constraint-driven clustering. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 320–329

Geng L, Hamilton H (2006) Interestingness measures for data mining: A survey. ACM Computing Surveys (CSUR) 38(3)

Giacometti A, Soulet A (2016) Frequent pattern outlier detection without exhaustive mining. In: Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp 196–207

Gilpin S, Davidson I (2011) Incorporating SAT solvers into hierarchical clustering algorithms: an efficient and flexible approach. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 1136–1144

Gilpin S, Davidson I (2017) A flexible ILP formulation for hierarchical clustering. Artificial Intelligence 244:95–109

Gonzalez T (1985) Clustering to minimize the maximum intercluster distance. Theoretical Computer Science 38(2):293–306

Grira N, Crucianu M, Boujemaa N (2006) Fuzzy clustering with pairwise constraints for knowledge-driven image categorization. IEE Proceedings on Vision, Image and Signal Processing (CORE B) 153(3):299–304

Guns T, Nijssen S, De Raedt L (2013) k-Pattern set mining under constraints. IEEE Transactions on Knowledge and Data Engineering 25(2):402–418

Guns T, Dao TBH, Vrain C, Duong KC (2016) Repetitive branch-and-bound using constraint programming for constrained minimum sum-of-squares clustering. In: Proceedings of the European Conference on Artificial Intelligence, pp 462–470

Hadjitodorov ST, Kuncheva LI (2007) Selecting diversifying heuristics for cluster ensembles. Proceedings of the International Workshop on Multiple Classifier Systems pp 200—209

Hansen P, Delattre M (1978) Complete-link cluster analysis by graph coloring. Journal of the American Statistical Association 73(362):397–403

Hansen P, Jaumard B (1997) Cluster analysis and mathematical programming. Mathematical Programming 79(1–3):191–215

Hiep T, Duc N, Trung B (2016) Local search approach for the pairwise constrained clustering problem. In: Proceedings of the Symposium on Information and Communication Technology, pp 115–122

Hoi S, Jin R, Lyu M (2007) Learning nonparametric kernel matrices from pairwise constraints. In: International Conference on Machine Learning, pp 361–368

Hoi S, Liu W, Chang SF (2008) Semi-supervised distance metric learning for collaborative image retrieval. In: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition

Hoi S, Liu W, Chang SF (2010) Semi-supervised distance metric learning for collaborative image retrieval and clustering. ACM Transactions on Multimedia Computing, Communications, and Applications 6(3)

Huang H, Cheng Y, Zhao R (2008) A semi-supervised clustering algorithm based on must-link set. In: Proceedings of the International Conference on Advanced Data Mining and Applications, pp 492–499

Iqbal A, Moh'd A, Zhan Z (2012) Semi-supervised clustering ensemble by voting. In: Proceedings of the International Conference on Information and Communication Systems, pp 1–5

Jain A, Murty M, Flynn P (1999) Data clustering: a review. ACM computing surveys 31(3):264–323

Kamvar S, Klein D, Manning C (2003) Spectral learning. In: Proceedings of the International Joint Conference on Artificial Intelligence, pp 561–566

Ke Y, Cheng J, Yu JX (2009) Top-k correlative graph mining. In: Proceedings of the SIAM International Conference on Data Mining, pp 1038–1049

Khiari M, Boizumault P, Crémilleux B (2010) Constraint programming for mining n-ary patterns. In: Proceedings of the International Conference on Principles and Practice of Constraint Programming, pp 552–567

Kittler J (1998) On combining classifiers. IEEE Transactions on Pattern Analysis and Machine Intelligence 20(3):226–239

Klein D, Kamvar S, Manning C (2002) From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering. In: Proceedings of the International Conference on Machine Learning, pp 307–314

Kopanas I, Avouris S Nikolaosand Daskalaki (2002) The role of domain knowledge in a large scale data mining project. In: Proceedings of the Hellenic Conference on Artificial Intelligence, pp 288–299

Kuhn H, Tucker A (1951) Nonlinear programming. In: Proceedings of the Berkeley Symposium, pp 481–492

Kulis B, Basu S, Dhillon I, Mooney R (2005) Semi-supervised graph clustering: A kernel approach. In: Proceedings of the International Conference on Machine Learning, pp 457–464

Kulis B, Basu S, Dhillon I, Mooney R (2009) Semi-supervised graph clustering: A kernel approach. Machine Learning 74(1):1–22

Kuo CT, Ravi S, Dao TBH, Vrain C, Davidson I (2017) A framework for minimal clustering modification via constraint programming. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 1389–1395

LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521:436–444

van Leeuwen M (2014) Interactive data exploration using pattern mining. In: Interactive Knowledge Discovery and Data Mining in Biomedical Informatics, Lecture Notes in Computer Science, Springer, chap 9, pp 169–182

van Leeuwen M, Ukkonen A (2013) Discovering skylines of subgroup sets. In: Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp 272–287

van Leeuwen M, De Bie T, Spyropoulou E, Mesnage C (2016) Subjective interestingness of subgraph patterns. Machine Learning 105(1):41–75

Li T, Ding C (2008) Weighted consensus clustering. In: Proceedings of the SIAM International Conference on Data Mining, pp 798–809

Li T, Ding C, Jordan M (2007) Solving consensus and semi-supervised clustering problems using nonnegative matrix factorization. In: Proceedings of the IEEE International Conference on Data Mining, pp 577–582

Li Z, Liu J, Tang X (2008) Pairwise constraint propagation by semidefinite programming for semi-supervised classification. In: Proceedings of the International Conference on Machine Learning, pp 576–583

Li Z, Liu J, Tang X (2009) Constrained clustering via spectral regularization. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition, pp 421–428

Lu Z, Carreira-Perpinán M (2008) Constrained spectral clustering through affinity propagation. In: IEEE Conference on Computer Vision and Pattern Recognition, pp 1–8

Lu Z, Ip H (2010) Constrained spectral clustering via exhaustive and efficient constraint propagation. In: Proceedings of the European Conference on Computer Vision, pp 1–14

von Luxburg U (2007) A tutorial on spectral clustering. Statistics and Computing 17(4):395–416

du Merle O, Hansen P, Jaumard B, Mladenović N (1999) An interior point algorithm for minimum sum-of-squares clustering. SIAM Journal on Scientific Computing 21(4):1485–1505

Métivier KP, Boizumault P, Crémilleux B, Khiari M, Loudni S (2012) Constrained clustering using SAT. In: Proceedings of the International Symposium on Advances in Intelligent Data Analysis, pp 207–218

Monti S, Tamayo P, Mesirov J, Golub T (2003) Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data. Machine learning 52(1):91–118

Mueller M, Kramer S (2010) Integer linear programming models for constrained clustering. In: Proceedings of the International Conference on Discovery Science, pp 159–173

Ng M (2000) A note on constrained k-means algorithms. Pattern Recognition 33(3):515–519

Ouali A, Loudni S, Lebbah Y, Boizumault P, Zimmermann A, Loukil L (2016) Efficiently finding conceptual clustering models with integer linear programming. In: Proceedings of the International Joint Conference on Artificial Intelligence, pp 647–654

Pedrycz W (2002) Collaborative fuzzy clustering. Pattern Recognition Letters 23(14):1675–1686

Pelleg D, Baras D (2007) K-means with large and noisy constraint sets. In: Proceedings of the European Conference on Machine Learning, pp 674–682

Raj S, Raj P, Ravindran B (2013) Incremental constrained clustering: A decision theoretic approach. In: Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp 475–486

Rand W (1971) Objective criteria for the evaluation of clustering methods. Journal of the American Statistical Association 66(366):846–850

Rangapuram S, Hein M (2012) Constrained 1-spectral clustering. In: Proceedings of the International Conference on Artificial Intelligence and Statistics, pp 1143–1151

Rauber A, Pampalk E, Paralič J (2000) Empirical evaluation of clustering algorithms. Journal of information and organizational sciences 24(2):195–209

Rossi F, van Beek P, Walsh T (eds) (2006) Handbook of Constraint Programming. Foundations of Artificial Intelligence, Elsevier B.V.

Rutayisire T, Yang Y, Lin C, Zhang J (2011) A modified cop-kmeans algorithm based on sequenced cannot-link set. In: Proceedings of the International Conference on Rough Sets and Knowledge Technology, pp 217–225

Shi J, Malik J (2000) Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(8):888–905

Soulet A, Raïssi C, Plantevit M, Cremilleux B (2011) Mining dominant patterns in the sky. In: Proceedings of the IEEE International Conference on Data Mining, pp 655–664

Srivastava A, Zou J, Adams R, Sutton C (2016) Clustering with a reject option: Interactive clustering as bayesian prior elicitation. In: Proceedings of the ICML Workshop on Human Interpretability in Machine Learning, pp 16–20

Strehl A, Ghosh J (2002) Cluster ensembles—a knowledge reuse framework for combining multiple partitions. Journal of machine learning research 3:583–617

Tan W, Yang Y, Li T (2010) An improved cop-kmeans algorithm for solving constraint violation. In: Proceedings of the International FLINS Conference on Foundations and Applications of Computational Intelligence, pp 690–696

Tang W, Xiong H, Zhong S, Wu J (2007) Enhancing semi-supervised clustering: a feature projection perspective. In: Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining, pp 707–716

Vu VV, Labroche N (2017) Active seed selection for constrained clustering. Intelligent Data Analysis 21(3):537–552

Wagstaff K, Cardie C (2000) Clustering with instance-level constraints. In: Proceedings of the International Conference on Machine Learning, pp 1103–1110

Wagstaff K, Cardie C, Rogers S, Schroedl S (2001) Constrained k-means clustering with background knowledge. In: Proceedings of the International Conference on Machine Learning, pp 577–584

Wagstaff K, Basu S, Davidson I (2006) When is constrained clustering beneficial, and why? In: Proceedings of the National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference

Wang J, Han J, Lu Y, Tzvetkov P (2005) TFP: an efficient algorithm for mining top-k frequent closed itemsets. IEEE Transactions on Knowledge and Data Engineering 17(5):652–663

Wang X, Davidson I (2010a) Active spectral clustering. In: Proceedings of the IEEE International Conference on Data Mining, pp 561–568

Wang X, Davidson I (2010b) Flexible constrained spectral clustering. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 563–572

Wang X, Qian B, Davidson I (2014) On constrained spectral clustering and its applications. Data Mining and Knowledge Discovery 28(1):1–30

Wemmert C, Gançarski P, Korczak J (2000) A collaborative approach to combine multiple learning methods. International Journal on Artificial Intelligence Tools 9(1):59–78

Xiao W, Yang Y, Wang H, Li T, Xing H (2016) Semi-supervised hierarchical clustering ensemble and its application. Neurocomputing 173(3):1362–1376

Xing E, Ng A, Jordan M, Russell S (2002) Distance metric learning learning, with application to clustering with side-information. In: Proceedings of the Advances in Neural Information Processing Systems, pp 521–528

Xu R, Wunsch D (2005) Survey of clustering algorithms. IEEE Transactions on neural networks 16(3):645–678

Yang F, Li T, Zhou Q, Xiao H (2017) Cluster ensemble selection with constraints. Neurocomputing 235:59–70

Yang Y, Tan W, Li T, Ruan D (2012) Consensus clustering based on constrained self-organizing map and improved cop-kmeans ensemble in intelligent decision support systems. Knowledge-Based Systems 32:101—115

Yao H, Hamilton H (2006) Mining itemset utilities from transaction databases. Data & Knowledge Engineering 59(3):603–626

Yi J, Jin R, Jain A, Yang T, Jain S (2012) Semi-crowdsourced clustering: Generalizing crowd labeling by robust distance metric learning. In: Proceedings of the Advances in Neural Information Processing Systems, pp 1772–1780

Yu ZW, Wongb HS, You J, Yang QM, Liao HY (2011) Knowledge based cluster ensemble for cancer discovery from biomolecular data. IEEE Transactions on NanoBioscience 10(2):76—85

Zha H, He X, Ding CHQ, Gu M, Simon HD (2001) Spectral relaxation for k-means clustering. In: NIPS, pp 1057–1064

Zhang T, Ando R (2006) Analysis of spectral kernel design based semi-supervised learning. In: Proceedings of the International Conference on Neural Information Processing Systems, pp 1601–1608

Zhi W, Wang X, Qian B, Butler P, Ramakrishnan N, Davidson I (2013) Clustering with complex constraints - algorithms and applications. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 1056–1062