



Semi-supervised constrained clustering: an in-depth overview, ranked taxonomy and future research directions

Germán González-Almagro^{1,2} · Daniel Peralta³ · Eli De Poorter³ · José-Ramón Cano^{2,4} · Salvador García^{1,2}

Accepted: 29 December 2024 / Published online: 7 March 2025
© The Author(s) 2025

Abstract

Clustering is a well-known unsupervised machine learning approach capable of automatically grouping discrete sets of instances with similar characteristics. Constrained clustering is a semi-supervised extension to this process that can be used when expert knowledge is available to indicate constraints that can be exploited. Well-known examples of such constraints are must-link (indicating that two instances belong to the same group) and cannot-link (two instances definitely do not belong together). The research area of constrained clustering has grown significantly over the years with a large variety of new algorithms and more advanced types of constraints being proposed. However, no unifying overview is available to easily understand the wide variety of available methods, constraints and benchmarks. To remedy this, this study presents in-detail the background of constrained clustering and provides a novel ranked taxonomy of the types of constraints that can be used in constrained clustering. In addition, it focuses on the instance-level pairwise constraints, and gives an overview of its applications and its historical context. Finally, it presents a statistical analysis covering 315 constrained clustering methods, categorizes them according to their features, and provides a ranking score indicating which methods have the most potential based on their popularity and validation quality. Finally, based upon this analysis, potential pitfalls and future research directions are provided.

Keywords Semi-supervised learning · Background knowledge · Pairwise instance-level constraints · Constrained clustering · Taxonomy

1 Introduction

Two major approaches characterize machine learning: supervised learning and unsupervised learning Bishop and Nasrabadi (2006). In supervised learning, the goal is to build a classifier or regressor that, trained with a set of examples (or instances) X and their corresponding output value Y , can predict the value of unseen inputs. In unsupervised learning, only the set of examples X is available, and no output value is provided. In the latter, the goal is to discover some underlying structure in X . For example, in unsupervised clustering the goal is to

infer a mapping from the input to clusters (groups) of similar instances. Generally, the set of examples X is known as the dataset, and the set of output values Y is known as the labels set.

Semi-Supervised Learning (SSL) Chapelle et al. (2010) is the branch of machine learning that tries to combine the benefits of these two approaches. To do so, it makes use of both unlabeled data and labeled data, or other kinds expert knowledge. For example, when considering classification or regression, in addition to a set of labeled data, an additional set of unlabeled data may be available, which can contain valuable information. Similarly, when considering clustering problems, a smaller subset of labeled data (or other types of knowledge about the dataset) may be available. Generally, some kind of information that does not fit within the supervised or unsupervised learning paradigm may be available to perform machine learning tasks. Ignoring or excluding this information does not optimally use all available information, thus the need of SSL Van Engelen and Hoos (2020).

1.1 On the feasibility of semi-supervised learning

With regards to the applicability of SSL, a natural question arises Chapelle et al. (2010): in comparison with supervised and unsupervised learning, can SSL obtain better results? It could be easily inferred that the answer to this question is “yes”, otherwise neither this study nor all the cited before would exist. However, there is an important condition imposed for the answer to be affirmative: the distribution of instances in X must be representative of the true distribution of the data. Formally, the underlying marginal distribution $p(X)$ over the input space must contain information about the posterior distribution $p(Y|X)$. Then, SSL is capable of making use of unlabeled data to obtain information about $p(X)$ and, therefore, about $p(Y|X)$ Van Engelen and Hoos (2020). Luckily, this condition appears to be fulfilled in most real-world learning problems, as suggested by the wide variety of fields in which SSL is successfully applied. Nonetheless, the way in which $p(X)$ and $p(Y|X)$ are related is not always the same. This gives place to the SSL assumptions, introduced in Chapelle et al. (2010) and formalized in Van Engelen and Hoos (2020). A brief summary of these assumptions following Van Engelen and Hoos (2020) is presented, please refer to the cited studies for more details.

- *Smoothness assumption*: two instances that are close in the input space should have the same label.
- *Low-density assumption*: decision boundaries should preferably pass through low-density regions in space.
- *Manifold assumption*: in problems in which data can be represented in Euclidean space, instances in the high-dimensional input space are usually gathered along lower-dimensional structures, known as manifolds: locally Euclidean topological spaces.
- *Cluster assumption*: data points which belong to the same cluster also belong to the same class. This assumption can be seen as a generalization of the other three specific assumptions. As in other machine learning paradigms, the transduction versus induction dichotomy can be made within SSL. Usually, semi-supervised classification methods cope the SSL field, therefore the aforementioned dichotomy is explained in terms of classification as follows:
- *Inductive methods*: inductive methods aim to build a classifier capable of outputting a

label for any instance in the input space. Unlabeled data can be used to train the classifier, but the predictions for unseen instances are independent of each other once the training phase is completed. An example of inductive method in supervised learning is linear regression Van Engelen and Hoos (2020).

- *Transductive methods*: transductive methods do not build a classifier for the entire input space. Their predictions are limited to the data used during the training phase. Transductive methods do not have separated training and testing phases. An example of transductive method in unsupervised learning is hierarchical clustering Van Engelen and Hoos (2020). Classification methods within SSL can be clearly separated following the definitions above. However, when it comes to clustering, this distinction becomes unclear. Clustering methods within the SSL learning paradigm are usually considered to be transductive, as their output is still a set of labels partitioning the dataset and not a classification rule Chapelle et al. (2010). On the other hand, some authors claim that partitional clustering methods can be considered as inductive methods, because their assignment rule can be used to predict the cluster membership of unseen instances. Hierarchical clustering methods would belong to the transductive learning category, as no assignment rule can be derived from them Miyamoto and Terami (2011). The differences between partitional and hierarchical clustering will be formalized later in Sect. 3.

Figure 1 helps us contextualize semi-supervised learning and its derivatives within the overall machine learning landscape. Please note that the figure does not refer to particular learning models, but it focuses on learning paradigms. General SSL literature (Zhu 2005; Chapelle et al. 2010; Zhu and Goldberg 2009) usually divides SSL methods into two categories: semi-supervised classification and semi-supervised clustering. Further dichotomies have been made in later literature. In Van Engelen and Hoos (2020); Zhou (2021) semi-supervised classification methods are taxonomized taking into account the inductive versus transductive dichotomy. Some of the categories found in these taxonomies have been further studied: Subramanya and Talukdar (2014) proposes a taxonomy for graph-based semi-supervised methods, and Triguero et al. (2015) does the same for the self-labeling field. Concerning semi-supervised clustering, Bair (2013) proposes a high level taxonomy with 4 types of methods, while (Davidson and Basu 2007; Basu et al. 2008) focus on the specific area of constrained clustering. The supervised and unsupervised learning paradigms are included in Fig. 1 for the sake of contextualization only. Consequently only classic and widely-known tasks belonging to these areas have been included in the diagram.

1.2 Related work

The semi-supervised clustering area has been widely studied and successfully applied in many fields since its inception. It can be seen as a generalization of the classic clustering problem which is able to include background knowledge into the clustering process (Chapelle et al. 2010). Many types of background knowledge have been considered in semi-supervised clustering (Bair 2013), although the most studied one is the instance-level pairwise must-link and cannot-link constraints (Basu et al. 2008). This type of background knowledge relates instances indicating if they belong to the same class (must-link) or to different classes (cannot-link). The problem of performing clustering in the presence of this

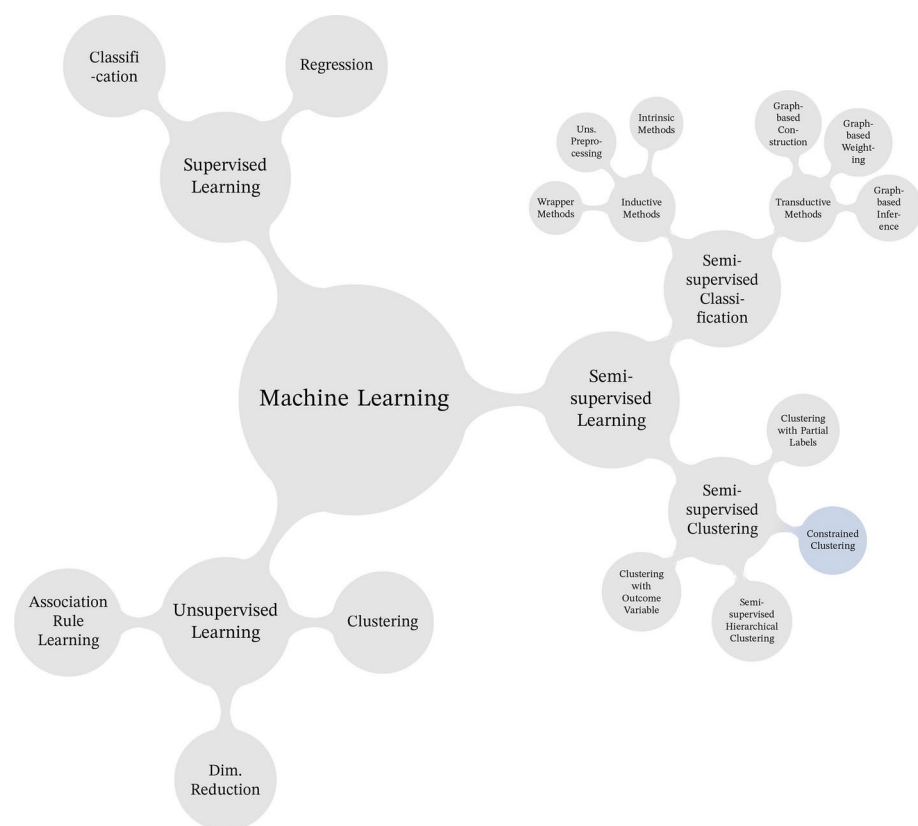


Fig. 1 Mindmap of the machine learning overall landscape

type of background knowledge is referred to in literature as *Constrained Clustering* (CC) (marked in Fig. 1 in blue).

This study carries out a comprehensive review of constrained clustering methods. It also proposes an objective scoring system, which addresses the potential and popularity of existing methods, and can be used to produce a sorted ranking for all of them. To the best of our knowledge, no similar study has been published before. Existing literature is either limited to the theoretical background on the topic, very limited in the number of methods reviewed, or outdated due to the rapid advance of the field. The earliest survey including constrained clustering in the reviewed studies can be found in Grira et al. (2004), although it is very limited in content. In Davidson and Basu (2007), the first survey focusing specifically on constrained clustering is proposed. It introduces many of the foundational concepts of subsequent studies and provides the first dedicated reference on the area. However, this study was published in 2007, and even then it was limited to very few methods. More recent studies extend the scope of the review and provide a taxonomy of methods, such as studies in Dinler and Tural (2016) and Gañçarski et al. (2020). These studies represent a significant improvement over previous attempts to provide a unified constrained clustering reference. However, they are still limited in the corpus of literature they review. Other works focus on a very specific constrained clustering family of algorithms, such as Vrain et al. (2024),

which focuses only on declarative approaches. All of the constrained clustering categories proposed in Dinler and Tural (2016), Vrain et al. (2024), Gañarski et al. (2020) represent only a subset of the much broader taxonomy provided in this study (in Sect. 7). The first book fully devoted to constrained clustering was published in Basu et al. (2008) (2008). It provides unified formal background within the area and detailed studies on state-of-the-art methods.

1.3 Early history of constrained clustering

The Constrained Clustering problem has been rediscovered and renamed throughout years of evolution, firstly in mathematical science, secondly in Computer Science. The first reference to the CC problem was proposed by Harary in Harary (1953) as early as in the year 1953. Harary introduced the *signed graph*, which is an undirected graph with +1 or -1 labels on its edges, respectively indicating similarity or dissimilarity between the vertices they connect. This can be directly translated to the ML and CL constraints that shape the CC problem. Besides, Harary introduced the concept of *imbalance* for a 2-way partitioning of such signed graph, which referred to the number of vertices violated by the partitioning. The aim of Harary was to find highly related groups of vertices within a psychological interpretation of the problem: positive edges correspond to pairs of people who like one another, and negative edges to pairs who dislike one another.

It was not until year 2000 that the name Constrained Clustering made its first appearance by the work of K. Wagstaff and C. Cardie in Wagstaff and Cardie (2000), which is a brief paper that introduces later work by the same authors in which the first two CC algorithms in the history of Computer Science are proposed: COP-COBWEB (Wagstaff and Cardie 2000) and COP-K-Means (Wagstaff et al. 2001) in 2000 and 2001 respectively. These two papers set the precedent for a new area in semi-supervised learning known as Constrained Clustering, providing experimental procedures and baselines to compare with.

On the one hand, and following the trend set by K. Wagstaff and C. Cardie, although in separate studies, S. Basu proposes in 2003 the first two soft constrained approaches to CC in Basu et al. (2003): the PCK-Means and MPCK-Means algorithms. Later, in the year 2005, I. Davidson and S.S. Ravi would propose the first hierarchical approaches to the CC problem (Davidson and Ravi 2005). On the other hand, E. Xing et al. propose the first distance metric learning based approach to CC with their CSI algorithm (Xing et al. 2002), also known in literature simply as Xing's algorithm. Finally, in 2008, S. Basu, I. Davidson and K. Wagstaff joined forces to produce the first book fully dedicated to constrained clustering in Basu et al. (2008).

Research in CC has followed the general trend in Computer Science ever since. Ranging from well-studied classic clustering approaches, such as fuzzy clustering (Li et al. 2018), spectral clustering (Nie et al. 2021) or non-negative matrix factorization (Zong et al. 2018), to modern and general optimization models like classic (Denoeux 2021) or deep (Amirizadeh and Boostani 2021) neural networks and evolutive (González-Almagro et al. 2021) or non-evolutive (González-Almagro et al. 2020) metaheuristic algorithms.

1.4 Remainder of this paper

The rest of this study is organized as follows. Section 2 presents a taxonomy of types of background knowledge with which semi-supervised clustering can work, including equivalencies between them in Subsection 2.8. Section 3 formalizes afterwards the constrained clustering problem, starting with basic background on classic clustering (Subsection 3.1) and pairwise constraints (Subsections 3.2 and 3.3), which is followed by a soft review on the classic applications of constrained clustering (Subsection 3.4). Subsequently, advanced concepts regarding constrained clustering are introduced in Sect. 4. A statistical study on the experimental elements used to demonstrate the capabilities of CC methods is proposed in Sect. 5. This statistical study is used as the basis of the scoring system, which is presented in Sect. 6 and used in subsequent sections to produce a ranking for all reviewed methods. Section 7 proposes a ranked taxonomic review of constrained clustering methods. A statistical analysis of the taxonomy is presented in Sect. 8. Finally, Sect. 9 presents conclusions, criticisms and future research guidelines.

The overall contribution of this study can be summarized as follows:

- We provide both the first taxonomy on the types of background knowledge and the first comprehensive taxonomy on constrained clustering methods. To do so, we review more than 300 methods and classify them according to both existing criteria and newly proposed criteria.
- We provide a statistical analysis on the average experimental procedures, so the reader can get a general understanding of the robustness of the state of the art.
- We provide an objective scoring method for all algorithms reviewed.
- We provide insight on every mid and final class of the taxonomy, reasoning the membership of every method.
- Finally, we present an analysis of the overall taxonomy and provide the reader with future research lines.

2 Clustering with background knowledge

In this section, a comprehensive literature review on the types of background knowledge that have been used by semi-supervised clustering algorithm is carried out. In general terms, 5 families of background knowledge have been identified: partition-level constraints, instance-level constraints, cluster-level constraints, feature-level constraints, and distance constraints. Background information which does not belong to any of the mentioned categories has been placed together in a miscellaneous category. Figure 2 shows a visual representation of this taxonomy. All 5 families are composed by smaller, more specific categories which are detailed below.

2.1 Partition-level constraints

Partition-level constraints refer to restrictions imposed on the partition generated by the semi-supervised clustering algorithm (Liu and Fu 2015; Śmieja and Geiger 2017; Basu et al. 2003; Liu et al. 2017). Their most common form is a subset of labeled data, which is often

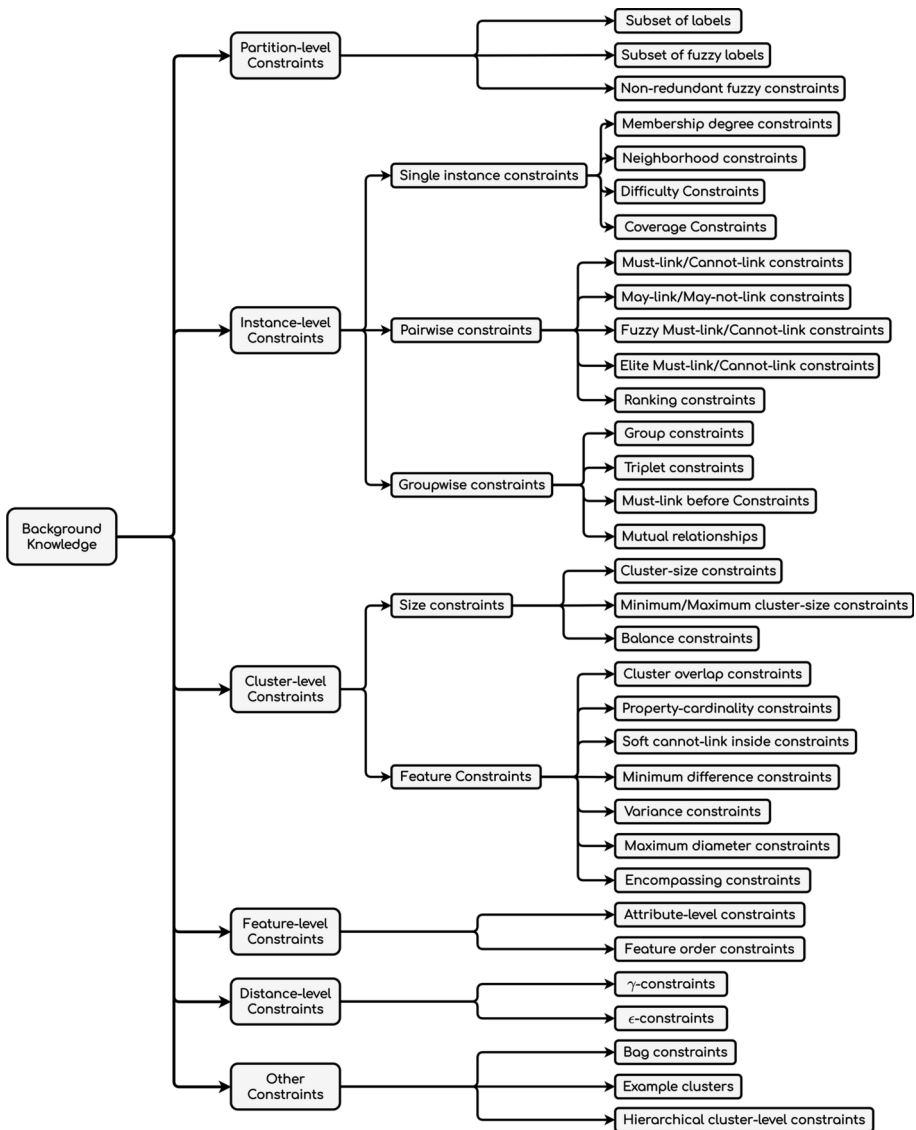


Fig. 2 Taxonomy of types of background knowledge

referred simply as “partition-level constraints”, although other categories within this type of background knowledge can be found:

- *Subset of labels*: they consist of a subset of instances from the dataset for which labels are available. The resulting partition must be consistent with the given labels (Basu et al. 2002; Böhm and Plant 2008; Lelis and Sander 2009; Liu and Wu 2010; Liu et al. 2011; Marussy and Buza 2013; Lan et al. 2014; Treechalong et al. 2015; Choo et al. 2015; Vu and Do 2017; Ienco and Pensa 2018; Liu et al. 2019; Xie and Antoine 2019; Herrmann

and Ultsch 2007).

- *Subset of fuzzy labeled data*: used in fuzzy semi-supervised clustering algorithms. It consists of a subset of instances for which fuzzy labels are provided (Zhang and Lu 2009; Yang and Liu 2015; Antoine et al. 2018; Gan 2019; Mei 2018; Antoine et al. 2021).
- *Non-redundant clusters constraints*: they constraint the output partition so that clusters in it must be orthogonal to each other, therefore maximizing their conditional mutual information and producing non-redundant clusters (Gondek and Hofmann 2007).

2.2 Instance-level Constraints

Instance-level constraints can refer to single instances, pairs of instances or groups of multiple instances. In the case of single instance constraints, they are used to describe particular features of said instances or to restrict the features of the cluster they can belong to:

- *Membership degree constraints*: used in fuzzy semi-supervised clustering algorithm to provide prior membership degrees for some instances (Yasunori et al. 2009; Yin et al. 2012; Zeng et al. 2013).
- *Neighborhood constraints*: they link instances to their neighborhood, with the latter being defined differently for every problem (Nghiem et al. 2020).
- *Instance difficulty constraints*: they are referred to single instances and specify how hard it is to determine the cluster an instance belongs to, so that the semi-supervised clustering algorithm can focus on easy instances first (Zhang et al. 2019).
- *Coverage constraints*: for clustering algorithms which allow instances to belong to multiple clusters at the same time, this type of constraint limits the number of times an instance can be covered by different clusters (Mueller and Kramer 2010). Instance-level pairwise constraints involve pairs of instances and are used to indicate positive or negative relationships. The former refers to features that instances have in common, such as class or relevance, while the latter refers to the opposite case. Even if instance-level must-link and cannot-link are the most common form of instance-level constraints, the latter can be given in multiple ways:
 - *Must-link/Cannot-link constraints*: must-links involve pairs of instances that are known to belong to the same class. Therefore they must belong to the same cluster in the output partition. Cannot-link are used to indicate the opposite (the two instances involved in them are known to belong to different classes and thus they need to be placed in different clusters) Davidson and Basu (2007).
 - *May-link/May-not-link constraints*: they represent soft must-link and cannot-link constraints respectively. This means that they can be violated in the output partition to some extent. They can be used in combination with the hard must-link and cannot-link constraints (Ares et al. 2009).
 - *Fuzzy Must-link/Cannot-link constraints*: pairwise positive/negative relationships with and associated degree of belief (Díaz-Valenzuela et al. 2015).
 - *Elite Must-link/Cannot-link constraints*: refined ML and CL constraints. They have the property of being unarguably satisfied in every optimal partition of the dataset (Jiang et al. 2013).

- *Ranking constraints*: in contexts in which output class labels (clusters) can be ordered, ranking constraints are used to indicate whether an instance should be assigned a class label (cluster) higher than the class label of another instance (Xiao et al. 2015). The last form of instance-level constraints are group constraints, which are used to gather group of instances that are known to share features or to be different to each other in some aspect of their nature. They can also be used to set relative comparisons between a fixed number of constraints. Overall, they can be classified as follows:
- *Group constraints*: also referred to as *grouping information* (Qian et al. 2016; Yu and Shi 2004). They specify the certainty of each or several instances belonging to the same cluster. Note that group constraint cannot be used to specify groups of instances that must not belong to the same cluster (Law et al. 2004).
- *Triplet constraints*: also known as *relative constraints* (Pei et al. 2016; Liu et al. 2011; Amid et al. 2015). They involve three instances: an anchor instance a , a positive instance b , and a negative instance c . A triplet constraint indicates that a is more similar to b than c (Zhang et al. 2019; Ienco and Pensa 2019).
- *Must-link-before*: these are ML constraints specifically designed to be applied in hierarchical clustering setups. They involve triplets of constraints and their basic idea is to link instances positively not only in the output partition, but also in the hierarchy (dendrogram) produced by hierarchical clustering methods (Bade and Nürnberger 2014).
- *Mutual relationships*: they establish a relation in groups of instances that is not known in advance and is determined during the clustering process. For example, a group of instances in the same mutual relation may be determined to belong to the same cluster during the cluster process, or contrarily they may be determined to not belong to the same cluster. Contrary to ML and CL constraints, mutual relations do not specify whether the nature of the relation they describe is positive or negative as part of the prior knowledge (Endo and Hamasuna 2011).

2.3 Cluster-level constraints

Cluster-level constraints are used to restrict a wide variety of features related to clusters without specifying which instances must belong to these clusters. They are considered to be one of the most useful types of background knowledge, as they can convey large amounts of information compared to the amount of expert knowledge available. Size constraints are one of the forms in which cluster-level constraints can be found. They constraint the number of instances that clusters can have in the output partition and can be divided in three categories:

- *Cluster-size constraints*: also called *cardinality constraints* (Gallegos and Ritter 2010). They specify the number of instances each cluster must have in the output partition. The number of instances in a cluster may vary from a cluster to another (Zhu et al. 2010; Tang et al. 2019; Bibi et al. 2023).
- *Maximum/minimum cluster-size constraints*: they specify the maximum/minimum size a cluster can have in the output partition without specifying the exact size of each cluster (Babaki et al. 2014; Dao et al. 2013; Bradley et al. 2000; Lei et al. 2013; Rebollo-Monedero et al. 2013). They may also be referred to as *significance constraints* (Ge et al. 2007).

- *Balance constraints*: also known as *global size constraints* (Zhang et al. 2019) applied in the cluster-level, they try to even the number of instances in every cluster (all cluster should be approximately the same size) Zhong and Ghosh (2003); Banerjee and Ghosh (2006); Zhu et al. (2010); Tang et al. (2019). Apart from the size of the cluster, cluster-level constraints can restrict a wide variety of cluster features, ranging from their shape or separation, to the kind of instances they may contain:
- *Cluster-overlap constraints*: they constraint the amount of overlap between clusters (Nghiem et al. 2020; Mueller and Kramer 2010).
- *Property-cardinality constraints*: they constraint the amount of a specific type of instance a cluster can contain (Nghiem et al. 2020).
- *Soft cannot-link inside cluster constraints*: they require that the number of pairs of instances in a cluster which have a cannot-link constraint among them to be bounded (Babaki et al. 2014).
- *Minimum difference constraints*: applied to pair of clusters, they require clusters to be similar or different to some degree (Babaki et al. 2014).
- *Variance constraints*: they impose maximum or minimum values for the variance clusters must feature in the output partition (Ge et al. 2007).
- *Maximum diameter constraints*: they specify an upper/lower bound on the diameter of the clusters (Dao et al. 2013).
- *Encompassing constraints*: they determine whether clusters are allowed to encompass each other, i.e., they are allowed to form a hierarchy (Mueller and Kramer 2010).

2.4 Feature-level constraints

Feature-level constraints constraint instances by their feature values or directly relate pairs of feature to each other to indicate degrees of importance. Two types of feature-level constraints can be found:

- *Attribute-level constraints*: they constraint the number of possible assignments for instances with specific values for specific features (Nghiem et al. 2020).
- *Feature order constraints*: also called *feature order preferences*. They involve pairs of features and determine which one of them is more important. This is, what features need to be paid more attention to when performing comparisons to decide cluster memberships (Sun et al. 2010).

2.5 Distance constraints

Distance constraints represent a very particular case of constraint-based information, as they relate pair of instances indirectly and in a global way. That means distance constraints can always be translated to instance-level must-link constraints (Davidson and Basu 2007). Two types of distance constraints are defined in literature:

- γ -constraints: also called *minimum margin* (Dao et al. 2013) or *minimum separation* (Davidson et al. 2010). They require the distance between two points of different clusters to be superior to a given threshold called γ (Davidson and Basu 2007; Dao et al.

2013; Davidson and Ravi 2005).

- ϵ -constraints: they require for each instance to have in its neighborhood of radius ϵ at least another point of the same cluster (Davidson and Basu 2007; Dao et al. 2013; Davidson and Ravi 2005).

2.6 Other types of constraints

Finally, authors have proposed forms of background knowledge that do not fit into any of the previous categories:

- *Bag constraints*: specific to the Multi-Instance Learning (MIL) framework, where datasets are given in the form of bags, with each bag containing multiple instances and labels, which are provided only at the bag-level. Bag constraints specify similarities between bags (Pei and Fern 2014). Recent advances in (MIL) have introduced novel approaches to bag constraints, significantly contributing to the robustness and accuracy of MIL models. In Waqas et al. (2023) a Deep Gaussian Mixture Model-based instance relevance estimation method is proposed, which has demonstrated superior performance in various MIL applications by accurately estimating the relevance of instances within a bag. Furthermore, a robust bag classification approach via subspace fuzzy clustering has provided a novel methodology for enhancing the classification accuracy of MIL models in Waqas et al. (2023). Additionally, an ensemble-based instance relevance estimation technique developed in Waqas et al. (2021) further supports the efficacy of using ensemble methods to improve MIL performance. In a similar vein, Carbonneau et al. (2016) explores the use of random subspace instance selection to create robust MIL ensembles, which showed significant improvements in handling diverse and challenging datasets. These studies collectively highlight the importance of bag constraints and offer diverse methodologies to enhance MIL applications.
- *Example clusters*: predefined clusters in the dataset given to the clustering algorithm, which is required to output a partition which is consistent with example clusters. This information can be converted to instance-level pairwise constraints (Hu et al. 2013; Vens et al. 2013).
- *Hierarchical cluster-level constraints*: sometimes also referred to as *ranking constraints* (Ben Ahmed et al. 2012a, b). These constraints are designed to be applied only in semi-supervised hierarchical clustering methods. Given pairs of clusters, they specify which action (merge, split, remove, etc.) must be taken over them in successive steps of the clustering process that builds the output dendrogram (Nogueira et al. 2017).

2.7 Constraints usability

After analyzing the wide variety of forms in which constraints can be given, it is reasonable to ask which type of constraint is more effective for general purposes. There is not in fact a unique answer to this question, as it highly depends on the problem or applications and the type of information available to solve it. In Pei et al. (2016) an empirical setup that tries to answer this question in a reduced semi-supervised environment is proposed. It only considers instance-level pairwise must-link and cannot-link constraints and subsets of labeled data

as available sources of background knowledge. Three questions tried to be answered in the mentioned study, which can be reformulated to include a broader scope as follows:

- Given the same amount of oracle effort, which type of background knowledge is more effective at aiding clustering?
- Which type of constraint is easier to obtain from the oracle?
- Which type of constraint is more reliable? What it is meant here with an oracle is always understood as the source of background knowledge. This oracle can be a human, an automatic classifier, a crowdsourcing setup to gather information from distributed sources, etc. It is essential for any real-world or in-lab application of semi-supervised clustering to address these three questions.

2.8 Equivalencies between types of background knowledge

It is well known that some categories of background knowledge are neither isolated nor hermetic. Some types of constraints can be converted to another in a direct manner. Distance constraints can be translated to must-link constraints (Davidson and Basu 2007), or a subset of labeled data can always be transformed in a set of must-link and cannot-link constraints (Qian et al. 2016). The aim of this section is to provide intuition on all possible transformations without the need of a formal definition/notation for them, as this would require the length of a monography. Previous work on this line has been carried out in Qian et al. (2016), although within a much limited scope regarding the types of background knowledge considered. Figure 3 depicts equivalences found between the types of background knowledge introduced in Sect. 2. Arrows indicate available transformation, so that an arrow from A to B means that the type of knowledge in A can be transformed to the type of knowledge in B e.g.: a subset of labels can always be transformed into a set of ML and CL constraints.

In Fig. 3, only conversions without loss of information are considered, e.g.: fuzzy must-link/cannot-link constraints could be converted to must-link/cannot-link constraints by considering only those whose degree of belief is over 50%, although this would involve

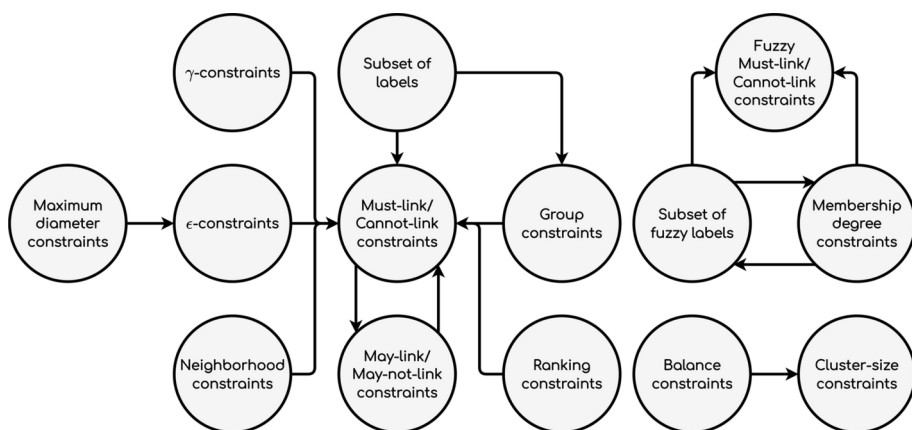


Fig. 3 Graphical representation of direct equivalences between types of background knowledge

losing not only constraints, but also the degree of belief information. Even if these kind of transformations are possible, they are not considered here, as they imply losing information.

3 Instance-level pairwise constrained clustering

Among all types of background knowledge reviewed in Sect. 2, pairwise constraints are undoubtedly one of the most studied topics, particularly basic must-link and cannot-link constraints, as it is shown later in this study. From now on, and for the sake of readability, *must-link and cannot-link constraints will be referred to simply as pairwise constraints*. In this section, the basic concepts of classic clustering and semi-supervised partitional and hierarchical clustering under pairwise constraints are introduced. *This problem is known in literature simply as Constrained Clustering (CC)* Davidson and Basu (2007).

3.1 Background on classic clustering

Partitional clustering can be defined as the task of grouping the instances of a dataset into K clusters. A dataset X consists of n instances, and each instance is described by u features. More formally, $X = \{x_1, \dots, x_n\}$, with the i th instance noted as $x_i = (x_{i,1}, \dots, x_{i,u})$. A typical clustering algorithm assigns a class label l_i to each instance $x_i \in X$. As a result, we obtain the list of labels $L = [l_1, \dots, l_n]$, with $l_i \in \{1, \dots, K\}$, that effectively splits X into K non-overlapping clusters c_i to form a partition called C . The list of labels producing partition C is referred to as L^C . The criterion used to assign an instance to a given cluster is the similarity to the rest of elements in that cluster, and the dissimilarity to the rest of instances of the dataset. This value can be obtained with some kind of distance measurement (Jain et al. 1999).

Hierarchical clustering methods produce an informative hierarchical structure of clusters called dendrogram. Partitions as described above, with a number of clusters ranging from 1 to n , can always be obtained from a dendrogram by just selecting a level from its hierarchy and partitioning the dataset according to its structure. Typically, agglomerative hierarchical clustering methods start with a large number of clusters and iteratively merge them according to some affinity criteria until a stopping condition is reached. Every merge produces a new level in the hierarchy of the dendrogram. Formally, given an initial partition with n_c clusters $C = \{c_1, \dots, c_{n_c}\}$ (usually $n_c = n$), a traditional agglomerative constrained clustering method selects two clusters to merge by applying Eq. 1. Please note that and affinity measure must always be maximized, not minimized.

$$\{c_i, c_j\} = \underset{c_i, c_j \in C, i \neq j}{\operatorname{argmax}} A(c_i, c_j), \quad (1)$$

with $A(\cdot, \cdot)$ being a function used to determine the affinity between the two clusters given as arguments. This function needs to be carefully chosen for every problem, as it greatly affects the result of the clustering process. Some conventional methods to measure affinity between clusters are worth mentioning, such as single linkage, average linkage and complete linkage (Jain et al. 1999). Nevertheless, different measures are used in out-of-lab applications, as the manifold structures usually found in real-world datasets can be hardly captured by the

classic affinity measures mentioned above. Typically, classic partitional clustering methods are algorithmically less complex than hierarchical clustering methods, with the former featuring $\mathcal{O}(n)$ complexity and the latter $\mathcal{O}(n^2)$ (Davidson and Basu 2007).

3.2 Background on pairwise constraints

In most clustering applications, it is common to have some kind of information about the dataset that will be analyzed (Wagstaff et al. 2001). In CC this information is given in the form of pairs of instances that must, or must not, be assigned to the same cluster. We can now formalize these two types of constraints:

- Must-link (ML) constraints $C_=(x_i, x_j)$: instances x_i and x_j from X must be placed in the same cluster. The set of ML constraints is referred to as $C_ =$.
- Cannot-link (CL) constraints $C_ \neq(x_i, x_j)$: instances x_i and x_j from X cannot be assigned to the same cluster. The set of CL constraints is referred to as $C_ \neq$. The goal of constrained clustering is to find a partition (or clustering) of K clusters $C = \{c_1, \dots, c_K\}$ of the dataset X that ideally satisfies all constraints in the union of both constraint sets, called $CS = C_ = \cup C_ \neq$. As in the original clustering problem, the sum of instances in each cluster c_i is equal to the number of instances in X , which we have defined as $n = |X| = \sum_{i=1}^K |c_i|$.

Knowing how a constraint is defined, ML constraints are an example of an equivalence relation; therefore, ML constraints are reflexive, transitive and symmetric. This way, given constraints $C_=(x_a, x_b)$ and $C_=(x_b, x_c)$, then $C_=(x_a, x_c)$ is verified. In addition to this, if $x_a \in c_i$ and $x_b \in c_j$ are related by $C_=(x_a, x_b)$, then $C_=(x_c, x_d)$ is verified for any $x_c \in c_i$ and $x_d \in c_j$ (Davidson and Basu 2007).

It can also be proven that CL constraints do not constitute an equivalence relation. However, analogously, given $x_a \in c_i$ and $x_b \in c_j$, and the constraint $C_ \neq(x_a, x_b)$, then it is also true that $C_ \neq(x_c, x_d)$ for any $x_c \in c_i$ and $x_d \in c_j$ (Davidson and Basu 2007).

Regarding the degree in which constraints need to be met in the output partition/dendrogram of any CC algorithm, a simple dichotomy can be made: hard pairwise constraints must necessarily be satisfied, while soft pairwise constraints can be violated to a variable extent. This distinction is introduced in Davidson and Basu (2007) and adopted by later studies, eventually producing the “may constraints” (may-link/may-not link constraints mentioned in Sect. 2), which can be seen as the formalization of soft constraints. However, the scientific community still refers to “may constraints” as soft constraints in the majority of the cases and the terms may-link and may-not link are used only in cases in which both soft and hard constraints are mixed and can be considered by the same CC algorithm. The major advantages in favor of soft over hard constraints are found in the resiliency to noise in the constraint set, the flexibility on the design of cost/objective functions, and their optimization procedures. The ability to consider soft, hard, or both types of constraints is a defining element for CC methods.

In Davidson et al. (2006) two measures designed to characterize the quality of a given constraint set are proposed: *informativeness* (or informativity (Davidson and Basu 2007)) is used to determine the amount of information in the constraint set that the CC algorithm could determine on its own, and *coherence*, which measures the amount of agreement between the

constraints themselves. These two measures were proposed in early stages of the development of the CC area; however, they have not been used consistently in later studies.

3.3 The feasibility problem

Given that CC adds a new element to the clustering problem, we must consider how it affects the complexity of the problem in both of its forms: partitional and hierarchical. Intuitively, the clustering problem goes from its classic formulation “find the best partition for a given dataset” to its constrained form “find the best partition for a given dataset satisfying all constraints in the constraint set”. The formalization of this problem is tackled in Davidson and Ravi (2005); Davidson and Basu (2007); Davidson and Ravi (2009), where the feasibility problems for partitional and hierarchical CC are defined as in 3.1 and 3.2 respectively, where $CS = C_{\neq} \cup C_{=}$ (the joint constraint set). Given these two definitions, we say that a partition C for a dataset X is feasible when all constraints in CS are satisfied by C . Note that there are constraint sets for which a feasible partition can never be found, e.g., no feasible partition exist for $CS_1 = \{C_{=}(x_1, x_2), C_{\neq}(x_1, x_2)\}$ regardless of the value of K . Similarly, the feasibility of partitions such as $CS_2 = \{C_{\neq}(x_1, x_2), C_{\neq}(x_2, x_3), C_{\neq}(x_1, x_3)\}$ depends on the value of K . In this case, the feasibility problem for CS_2 can be solved for $K = 3$ but not for $K = 2$.

Definition 3.1 *Feasibility Problem for Partitional CC*: given a dataset X , a constraint set CS , and the bounds on the number of clusters $k_l \leq K \leq k_u$, is there a partition C of X with K clusters that satisfies all constraints in CS ? Davidson and Ravi (2005)

In Davidson and Ravi (2005) it is proven that, when $k_l = 1$ and $k_u \geq 3$, the feasibility problem for partitional CC is NP-complete, by reducing it from the Graph K-Colorability problem. It is also proven that it is not harder, so both have the same complexity. Table 1 shows the complexity of the feasibility for different types of constraints.

Definition 3.2 *Feasibility Problem for Hierarchical CC*: given a dataset X , the constraint sets CS , and the symmetric distance measure $D(x_i, x_j) \geq 0$ for each pair of instances: Can X be partitioned into clusters so that all constraints in CS are satisfied? (Davidson and Ravi 2009)

Please note that the definition of the feasibility problem for partitional CC (in Definition 3.1) is significantly different from the definition of the feasibility problem for hierarchical CC (in 3.2). Particularly, the formulation for the hierarchical CC does not include any restriction on the number of clusters K , which is equivalent to considering that any level of the dendrogram can be used to produce the partition that satisfies all constraints (Davidson and Ravi 2009). In Davidson and Ravi (2005) a reduction from the One-in-three 3SAT with positive

Table 1 Feasibility problem complexities for partitional and hierarchical CC and dead-ends found in hierarchical CC Davidson and Ravi (2005)

Constraints	Partitional CC	Hierarchical CC	Dead Ends?
ML	P	P	No
CL	NP-complete	NP-complete	Yes
ML and CL	NP-complete	NP-complete	Yes

literals problem (which is NP-complete) for the problem in Definition 3.2 is used to prove the complexities presented in Table 1 for the hierarchical CC problem. The “Dead Ends?” column refers to whether or not the algorithm can encounter a dead end when operating under a given combination of constraints, and the ML and CL acronyms refer to Must-link and Cannot-link constraints respectively. It is worth mentioning that, for the hierarchical CC problem, the dead-ends problem arises: a hierarchical CC algorithm may find scenarios where no merge/split can be carried out without violating a constraint. Previous solutions based on the transitive closure of the constraint sets have been proposed to this problem, although they imply not generating a full dendrogram (Davidson and Ravi 2005).

Overall, complexity results in Table 1 show that the feasibility problem under CL constraints is intractable, hence constrained clustering is intractable too. This leads to Observation 3.1. For more details on the complexity of constrained clustering please see Davidson and Ravi (2005).

Observation 3.1 *Knowing that a feasible solution exists does not help us find it* Davidson and Ravi (2005). The results from Table 1 imply that the fact that there is a feasible solution for a given set of constraints does not mean it will be easy to find.

With respect to the dead-ends problem, a full dendrogram considering constraints can be obtained by switching from a hard interpretation of constraints to a soft one. This means that every level in the dendrogram tries to satisfy as many constraints as possible, but constraint violations are allowed in order for the algorithm to never reach a dead-end.

Some interesting results, both positive and negative, about the nature of pairwise constraints are proved and discussed in Davidson and Basu (2007), as well as some work-arounds for problems related to the use of constraints in clustering.

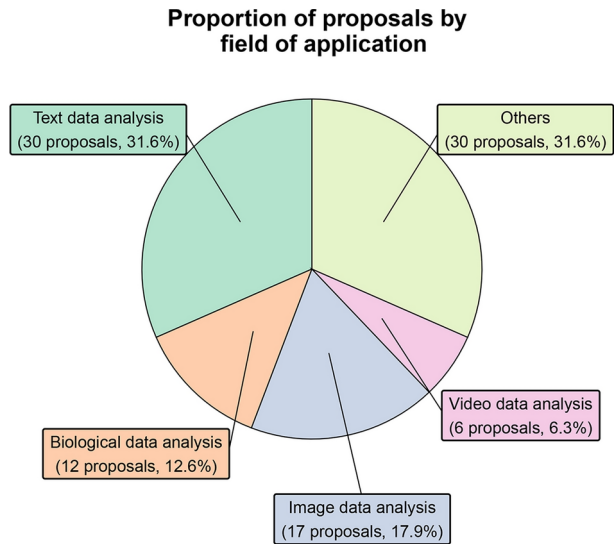
3.4 Applications of constrained clustering

CC has been applied in many fields since its inception, as shown in the various surveys that have been proposed over the years (Dinler and Tural 2016; Gançarski et al. 2020; Davidson and Basu 2007). The first applications are gathered in Davidson and Basu (2007), which include clustering of image data, video and sound data, biological data, text data, web data, and the first application of CC found in Wagstaff et al. (2001), which is lane finding for vehicles in GPS data. Figure 4 shows a summary of the overall CC application field. Please note that, in this study, we do not aim to provide a comprehensive review of constrained clustering applications, but we intend to include classic applications to give the reader a general understanding of its applicability.

Table 2 gathers CC applications, sorting them by application field and indicating the specific purpose of every application. The field with the largest number of publications is text data analysis; within it, document clustering has attracted the most publications. Text data clustering is followed by three other wide application fields, which are biological data analysis, image data analysis and video data analysis.

However, some CC applications are very specific and cannot be grouped into wider application fields. Studies which bring forward this kind of applications are listed in Table 3.

Fig. 4 Piechart showing a summary of the overall CC application field



4 Constrained clustering concepts and structures

Within the CC research field, some concepts and data structures are repeatedly mentioned and used by researchers. The goal of this section is to provide a formal definition of these concepts, as they will be mentioned later and are necessary for the reader to have a good understanding of the methods described later in Sect. 7. From now on, and for the sake of readability and ease of writing, we refer to instances involved in a constraint simply as *constrained instances*, and to ML constraints and CL constraints as simply ML and CL, respectively. Instances involved in ML are referred to as ML-constrained instances and instances involved in CL are referred to as CL-constrained instances.

The Constraint Matrix

This is one of the most, if not the most, basic and most frequently used data structures to store the information contained in the constraint set. It is a symmetric matrix, with as many rows and columns as instances in the dataset, filled with three values: 0 to indicate no constraint between the instances associated with the row and column in which it is stored, 1 is used for ML and -1 is used for CL. Formally, the Constraint Matrix is a matrix $CM_{n \times n}$ filled as in Eq. 2.

$$CM_{[i,j]} = CM_{[j,i]} = \begin{cases} 1 & \text{if } C_{=}(x_i, x_j) \in CS \\ -1 & \text{if } C_{\neq}(x_i, x_j) \in CS \\ 0 & \text{otherwise} \end{cases} . \quad (2)$$

Please note that, following this definition for the constraint matrix, its diagonal may be assigned to all 1 or all 0. That possibility depends on whether ML with the form $C_{=}(x_i, x_i)$ are included or not in CS , respectively. The inclusion of such constraints may be convenient in some cases. Variants of this matrix are also commonly used. In some cases, the constraint matrix can store any value in the range $[-1, 1]$, with negative values indicating the weight or degree of belief for CL and positive values doing so for ML.

Table 2 Comprehensive listing of applications of CC in wide application fields

Field of application	No. of studies	Specific application	No. of studies	References
Text data analysis	30	Document clustering	19	Huang and Lam (2007) Huang and Lam (2009) Ma et al. (2010) Yan et al. (2013) Liu et al. (2013) Ma et al. (2013) Sutanto and Nayak (2014) Hu et al. (2016) Díaz-Valenzuela et al. (2016) Trabelsi and Zaiane (2016) 2016 Balafar et al. (2020) Buatoom et al. (2020) 2020 Van Craenendonck et al. (2017) Hu et al. (2008) Chen et al. (2009) Ma et al. (2010) Zhao et al. (2012) Yan et al. (2013) Gu et al. (2012)
		Text Clustering	4	Ares et al. (2009) Parapar and Barreiro (2012) Song et al. (2012) Zhu et al. (2011)
		Verb clustering	1	Vlachos et al. (2009)
		Word disambiguation	1	Sugiyama and Okumura (2009)
		Microblog clustering	3	Ma et al. (2014) Ma et al. (2015) Ma et al. (2019)
		Document filtering	1	Tang et al. (2005)
		Clustering in online forums	1	Duan et al. (2015)
		Gene Expression	7	Tseng et al. (2008) Maraziotis (2012) Saha et al. (2015) Maraziotis et al. (2006) Maraziotis (2012) Yu et al. (2014) Ceccarelli et al. (2005)
		Gene Clustering	1	Zeng et al. (2007)
		RNA-seq Data Clustering	1	Tian et al. (2021)
		Regulatory Module Discovery	1	Mishra and Gillies (2008)
		Fiber Segmentation	1	Abdala and Jiang (2010)
		Biomolecular Data	1	Yu et al. (2011)
Biological data analysis	12			

Table 2 (continued)

Field of application	No. of studies	Specific application	No. of studies	References
Image data analysis	17	Medical image	2	Verma et al. (2016) Veras et al. (2018)
		Image segmentation	3	Wang and Yang (2018) Guo et al. (2014) Li et al. (2020)
		Image clustering	3	Biswas and Jacobs (2011) Hoi et al. (2010) Biswas and Jacobs (2014)
		Image categorization	2	Grira et al. (2006) Frigui and Meredith (2008)
		Image annotation	2	ISMAIL and Behir (2012) Xiaoguang et al. (2006)
		Image indexing	1	Lai et al. (2014)
		Point of interest mining	1	Bui and Park (2017)
		Multi-target detection	1	Li et al. (2017)
		Face recognition	1	Vu et al. (2019)
		Satellite Image Time Series	1	Lafabregue et al. (2019)
		Extracting moving people	1	Niebles et al. (2008)
		Web Video Categorization	1	Mahmood et al. (2013)
		Face Clustering	2	Wu et al. (2013) 2013 Wu et al. (2017)
		Face Tracking	2	Wu et al. (2013) 2013 Wu et al. (2017)
Video data analysis	6			

Table 3 Comprehensive listing of particular applications of CC

Field of Application	References
Identifying speakers in a conversation through audio data	Xie et al. (2006)
Clustering of software requirements	Duan et al. (2008)
Machinery fault diagnosis	Li et al. (2010)
Patient Segmentation from medical data	Han et al. (2016) Zhang and Chang (2019)
Direct marketing applications	Chang and Tsai (2011) Yu et al. (2021)
	Seret et al. (2014) Akaichi and Wislez (2021)
Group extraction from professional social network	Ben Ahmed et al. (2014)
Clustering of cognitive radio sensor networks	Shah et al. (2014)
District design	Joshi et al. (2011) Song et al. (2021)
Sentiment analysis	Araujo and Kamel (2014) 2014 Xiong and Ji (2015)
Sketch symbol recognition	Tirkaz et al. (2012)
Robot navigation systems	Semnani et al. (2016)
Terrorist community detection	Saidi et al. (2018)
Lane finding for vehicles in GPS data	Wagstaff et al. (2001)
Optimization of rural ecological endowment industry	Zhao and Gai (2022)
Job-shopping scheduling	El-Kholany et al. (2022)
Trace-clustering	De Koninck et al. (2021)
Discovering educational-based life patterns	Zhang et al. (2021)
Oil price prediction	Boesen et al. (2021)
Traffic analysis	Zhang et al. (2019) Malzer and Baum (2021)
Vocabulary maintenance policy for CBR systems	Ben Ayed et al. (2019)
Obstructive sleep apnea analysis	Mai et al. (2018)
Internet traffic classification	Wang et al. (2013)
Social event detection	Sutanto and Nayak (2014)

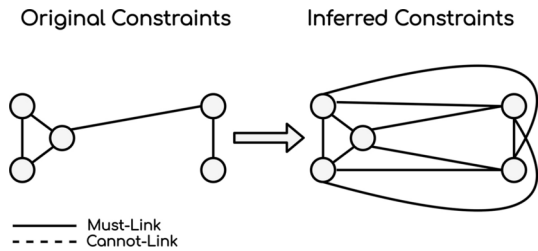
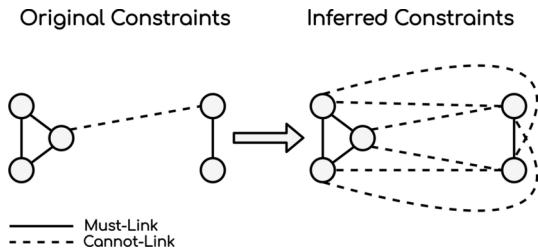
The Constraint List

It is a list, with length equal to the number of constraints, that stores triplets with two values used to specify two instances and a third value used to indicate the type of constraint between them (1 for ML and -1 for CL). Formally, the Constraint List CL contains $|C_{=}|$ triplets with the form $[i, j, 1]$ for ML such that $C_{=}(x_i, x_j)$ and $|C_{\neq}|$ triplets with the form $[i, j, -1]$ for CL such that $C_{\neq}(x_i, x_j)$.

The Constraint List is used in methods in which the number of violated constraints needs to be repeatedly computed over fully formed partitions that are not built incrementally. In these cases, the only option is to iterate over the full constraint set and check individually for every constraint whether it is violated by the partition. This task is performed efficiently iterating over CL , which is $\mathcal{O}(|CS|)$, in contrast with CM , which requires an $\mathcal{O}(n^2)$ computation of the number of violated constraints. However, checking for specific constraint violations in iterative partition building processes can be done in $\mathcal{O}(1)$ with CM , as the indexes of the constrained instances are known and matrices support random access. The same task can be performed over CL , but with the much higher computation cost of $\mathcal{O}(|CS|)$.

The Constraint Graph

It is a weighted, undirected graph with a one vertex per instance in the dataset and one edge per constraint. An edge connects two instances if they are involved in a constraint, with the weight of the edge indicating the type of constraints, using 1 for ML and -1 for CL. Formally, let an undirected weighted graph $G(V, E, W)$ be a finite set of vertices V , a set

Fig. 5 Example of transitive inference of ML constraints**Fig. 6** Example of transitive inference of CL constraints

of edges E over $V \times V$ and a set of weights W for every edge in E . In the constraint graph $CG(V, E, W)$, V is the set of instances in X , and edges $e(x_i, x_j)$ from E are equivalent to constraints in CS , using the weight of the edge $w_{[i,j]}$ as indicator for the type of constraint, i.e., for edge $e(x_i, x_j)$, if $C_=(x_i, x_j) \in CS$ (ML) then $w_{[i,j]} = 1$, and if $C_\neq(x_i, x_j) \in CS$ (CL) then $w_{[i,j]} = -1$ (Yoshida 2010).

The Transitive Closure of the constraint set

It is an augmented set of constraints which can be obtained on the basis of the information contained in the original constraint set, by applying two of its properties which have been introduced in Sect. 3.2, and are formally defined here on the basis of the constraint graph as in Properties 4.1 and 4.2. Graphical examples of these two properties are given in Fig. 5 and Fig. 6, respectively. These two properties can be applied over CG to obtain the transitive closure of the constraint set, which cannot be further augmented without new information.

Proposition 4.1 *Transitive inference of ML:* Let cc_1 and cc_2 be two connected components in CG with only positive edges in it (only ML constraints). Then, if there is a constraint $C_=(x_i, x_j)$ with $x_i \in cc_1$ and $x_j \in cc_2$, then the new constraints $C_=(a, b)$ can be inferred for all $a \in cc_1$ and $b \in cc_2$ (Basu et al. 2008).

Proposition 4.2 *Transitive inference of CL:* Let cc_1 and cc_2 be two connected components in CG with only positive edges in it (only ML constraints). Then, if there is a constraint $C_\neq(x_i, x_j)$ with $x_i \in cc_1$ and $x_j \in cc_2$, then the new constraints $C_\neq(a, b)$ can be inferred for all $a \in cc_1$ and $b \in cc_2$ (Basu et al. 2008).

The Chunklet Graph

This graph structure can be derived from the definition of the constraint graph and the concept of chunklet. Chunklets are defined in Bar-Hillel et al. (2003, 2005) as “subsets of points that are known to belong to the same, although unknown, class”. With this definition,

it is clear that ML connected components can be compared to chunklets, thus the chunklet graph can be obtained on the basis of the constraint graph. This is done by replacing ML connected components in CG by a single vertex which is adjacent to all former neighbors of the connected components (Cheng et al. 2008). In the case that vertices in CG also store position-related information (as in CC), the position of the new vertex is computed as the average of the nodes in the connected component (Yoshida 2010).

The Cluster Skeleton

It is a reduced constraint set which defines the true basic clustering structure of the data. It is obtained by applying the farthest-first scheme to query an oracle about the constraint relating selected instances from the dataset. This is done within an iterative scheme in which membership neighborhoods are created and the farthest instance from all of them is always selected to be queried against at least one instance from every existing neighborhood. If it is constrained to any of those instances by ML, then it is added to that neighborhood and a new ML is created, whereas if it is constrained by CL to all of them, then a new neighborhood is created and related to the other one by CL. The goal of this procedure is to build a constraint set which defines as many disjoint clusters as possible, aiding later CC algorithms determine the number of clusters a feasible partition must have (Basu et al. 2004a).

The Infeasibility

The concept of infeasibility refers to the number of constraints violated by a given partition. It is one of the most used concepts in CC , as many objective/fitness functions include penalty terms that are directly proportional to the number of violated constraints. Given a partition C (an its associated list of labels L^C) and a constraint set CS , the infeasibility can be defined as in Eq. 3, with $\mathbb{1}[\cdot]$ being the indicator function (returns 1 if the input is true and 0 otherwise) González-Almagro et al. (2021).

$$\text{Infs}(C, CS) = \sum_{C=(x_i, x_j) \in CS} \mathbb{1}[l_i^C \neq l_j^C] + \sum_{C \neq (x_i, x_j) \in CS} \mathbb{1}[l_i^C = l_j^C]. \quad (3)$$

The k-NN Graph

Also called k-NNG. It is not an exclusive concept from CC . It has been widely used in classic clustering literature and k-NN based classification. However, it is a very useful tool for CC research, therefore many CC approaches are built based on its definition. The k-NNG is a weighted undirected graph in which vertices represent instances from the dataset and every vertex is adjacent to at most k vertices. An edge is created between vertices u and v if and only if instances associated to u and v have each other in their k-nearest neighbors set. The weight $w(v, u)$ for the edge connecting u and v is defined as the number of common neighbors shared by u and v : $w(v, u) = |NN(u) \cap NN(v)|$, with $NN(\cdot)$ denoting the set of neighbors of the vertex given as argument (Vu et al. 2012).

5 Statistical analysis of experimental elements

In this section, a general view on how CC methods are evaluated and compared is presented. Most studies in CC present one or various new methods that need to be evaluated and proved to be competitive with respect to the state-of-the-art at the time they were proposed. In this section, the three experimental elements used to do so are analyzed: the datasets, the

validity indices, and the competing methods. Table 4 introduces the 15 most frequently used instances of these elements among all the papers analyzed in this study. All statistics presented in this section have been obtained by analyzing 278 studies, which propose a total of 315 methods. Some studies propose more than one method, and some methods are proposed in more than one study, hence the discordance between the number of papers analyzed and the number of proposed methods. Special cases of the experimental elements have not been taken into consideration to obtain the statistics presented in this section. In other words, if a paper uses the Iris dataset for its experiments but removes one of the three classes in the dataset, it is then considered as a single use of the classic Iris dataset, and not listed as a separate dataset. The same can be said for the other two experimental elements. For example, uses of the Pairwise F-measure (PF-measure) are included in the count of the F-measure, and variations on the initialization methods of COP-K-Means are included in the count of the basic COP-K-Means. This is done to obtain more representative and general statistics.

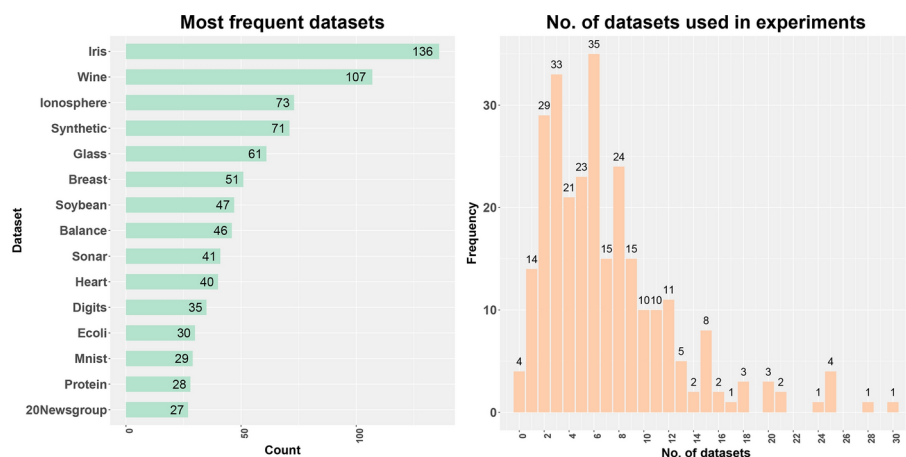
Sections 5.1, 5.2 and 5.3 dive into the statistics of the frequently used experimental setups regarding datasets, validity indices and competing methods, respectively. Section 5.4 presents the most used procedure to artificially generate constraints for benchmarking purposes, and Sect. 5.5 gives a quick note on the use of statistical testing to support conclusions in CC literature.

5.1 Analysis of datasets

A total of 457 different datasets can be identified in the experimental sections of the literature in CC. Figure 7 displays three different statistical measures about the use of these datasets. Figure 7a depicts the same information contained in Table 4, presenting it visually for the sake of ease of understanding. Figure 7b gives a histogram of the number of datasets used in experiments. Lastly, Fig. 7c introduces boxplots featuring the variability on the number of datasets used in different years.

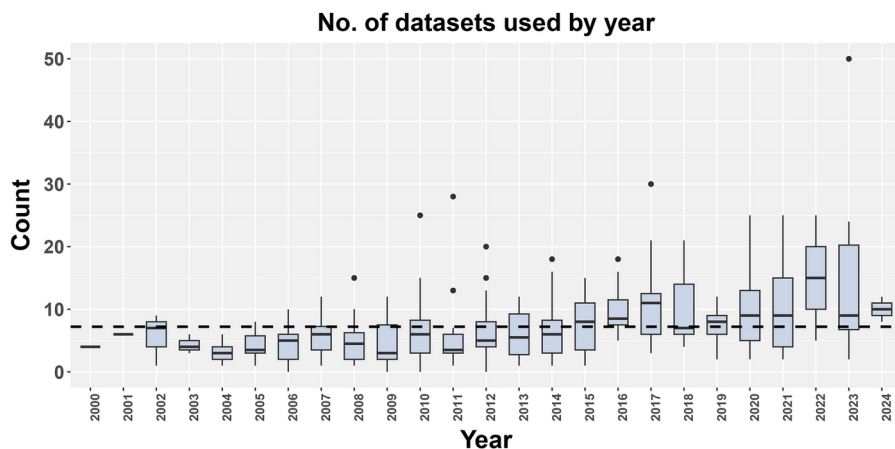
Table 4 Most frequently used experimental setup in CC

Datasets		Competing Methods		Validity Indices	
Name	No. of Uses	Name	No. of Uses	Name	No. of Uses
Iris	136	COP-K-Means	68	NMI	93
Wine	107	K-Means	62	CE	62
Ionosphere	73	MPCK-Means	34	RI	53
Synthetic	71	KKM	26	ARI	51
Glass	61	SSKK	26	F-measure	38
Breast	51	PCK-Means	23	Time	25
Soybean	47	FFQS	19	Unsat	13
Balance	46	RCA	15	Purity	12
Sonar	41	Random	14	Non Standard (NS)	10
Heart	40	NCuts	14	JC	8
Digits	35	CCSR	13	Visual	7
Ecoli	30	CSI	13	CRI	6
Mnist	29	E ² CP	13	FMI	4
Protein	28	Constrained EM	12	Precision	4
20Newsgroup	27	LCVQE	11	V-measure	4



(a) No. of times each dataset in Table 4 is used in experiments

(b) Distribution of the number of datasets used in experiments



(c) Variability of the number of datasets used in every year (the dashed line represents the overall average)

Fig. 7 Statistics about the datasets used in the experimental setups of all papers reviewed

From 7a, it is clear that classification datasets are used as benchmarks for CC methods. The reason for this lies in the lack of specific benchmarks, as very few have been proposed since the inception of the research topic. Classic classification datasets have to be used in order to generate the constraint sets needed by CC methods (see Sect. 5.4). In these cases, labels are never provided to the CC method, but used as the oracle to generate the constraint sets.

Looking at Fig. 7b, it can be concluded that the most frequent number of datasets used in experiment is 6. Please note, the study which uses the most datasets analyzes up to 50 of them in El Amouri et al. (2023). We have decided not to include this study in the figure, as we consider it to be an extreme outlier and it disturbs readability. Most papers use between

1 and 9 datasets. Note how some papers do not use any datasets, therefore they don't carry out any experiments to prove the efficacy of their proposal. Figure 7c shows a consistent increase over the years in the number of datasets used in experiments, probably due to the general growth in computing power, and to the increasing availability of datasets. It also shows how, except for the first few years, there is no consensus on the number of datasets to be used to demonstrate the capabilities of a new method, as boxplots show high variability within each year.

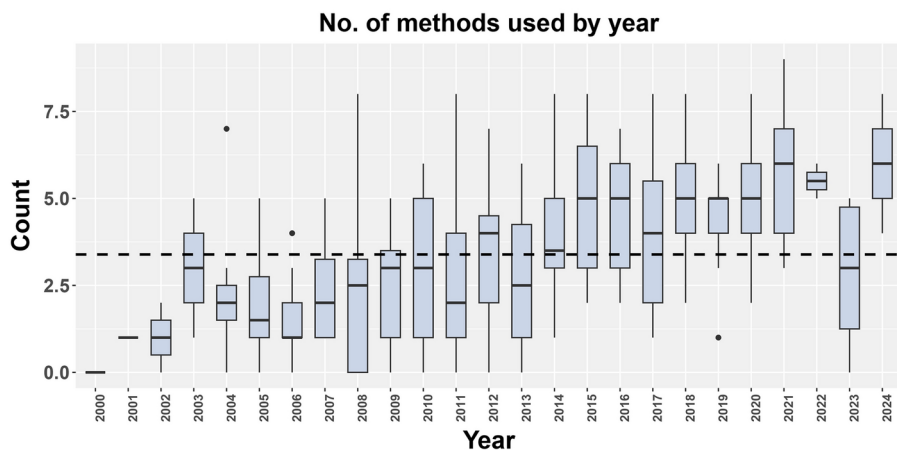
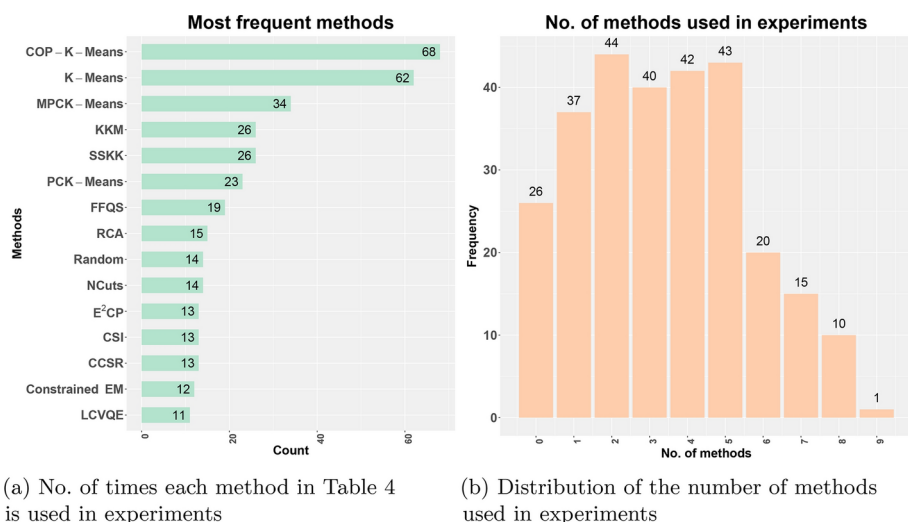
5.2 Analysis of competing methods

Any new CC method has to be proven to be competitive with the methods belonging to the state-of-the-art. Methods belonging to this category change over the years. Nonetheless, Fig. 8a presents a set of methods which are used very frequently, and can be subsequently understood as baseline methods. In fact, most of them correspond to the first proposals in different CC categories, e.g.: COP-K-Means is the first CC method ever proposed, PCK-Means is the first penalty-based CC method, KKM is the first constrained spectral clustering method, FFQS is the first active constrained clustering method, etc. Section 7 presents all these methods within the context of their specific CC category. Algorithms such as K-means or NCuts stand out as well, as they are not CC algorithms but classic clustering algorithms. When the experiments carried out aim to prove not only the capabilities of a new CC method, but also the viability of CC itself (as in the first proposals) or the viability of any new constraint generation method, then comparing with classic clustering algorithm is justified.

Figure 8b shows the distribution of the amount of methods used in experimental setups in CC literature. The most frequent comparison uses only two methods, which is a very low number taking into account the plethora of methods available to compare with (315 particularly). However, comparisons using between 4 and 6 methods are also reasonably frequent, with said frequency decaying from 6 methods to 9, which are used only in a single study. There is a particular fact that may catch our attention: 26 studies chose not to compare against any previous proposals. Given how well established baseline methods are, this should never be allowed in new CC studies. An increasing tendency can be observed in the number of methods used over the years. Likewise with the number of datasets (see Fig. 7c). Accordingly, this can be caused by an increase in computing power over the year and by the increase in the number of available methods to compare against.

Piecharts in Fig. 9 show further statistics about the proportions of methods used in experiments in CC studies. For example, it may be interesting to answer the following question: from all methods used in experiments to compare with, how many of them are CC methods? Fig. 9b answers this question. From all methods compared with (393), only 38.7% (152) are CC methods. The rest of the methods are not necessarily classic clustering methods, they can belong to other fields of SSL or use different types of constraints. This may seem contradictory with respect to what Fig. 7a shows. However, this is not the case. In conjunction, Figs. 9 and 7a evidence that the most frequently used methods are CC methods, even if the number of different classic clustering methods used to compare against is higher than the number of different CC methods.

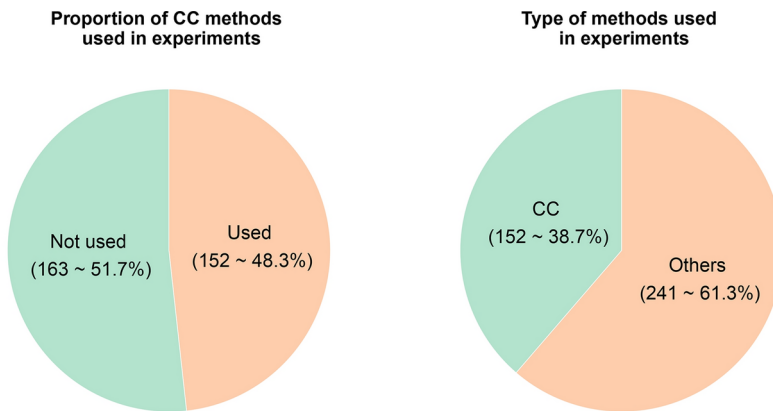
Another interesting question is: from all CC methods proposed over the years, how many of them are used to compare with in later studies? Fig. 9a provides now the answer to this



(c) Variability of the number of methods used in every year (the dashed line represents the overall average)

Fig. 8 Statistics about the methods used in the experimental setups of all papers reviewed

question. From all CC methods proposed (315) in the reviewed studies, 48.3% of them (152) are used in the experimental section of other studies. This indicates that more than half of the proposed methods have never been considered to be compared with by other authors. Of course, this statistic does not take into account the number of years any given method has been available to be used, only the absolute number of uses. However, this should not have a great impact in the proportions.



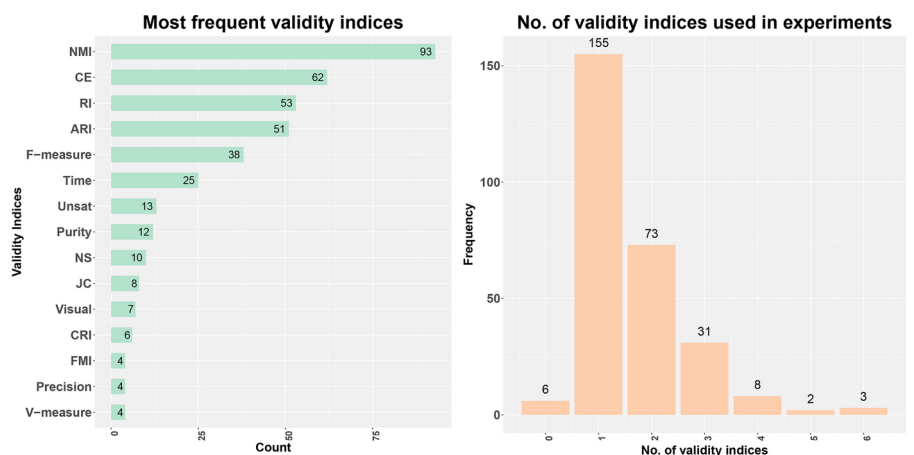
(a) Proportion of CC methods that are used in later studies to compare with.

(b) Proportion of methods used in experiments which are CC or other type of clustering methods.

Fig. 9 Piecharts depicting the usability of all CC methods reviewed in experimental setups

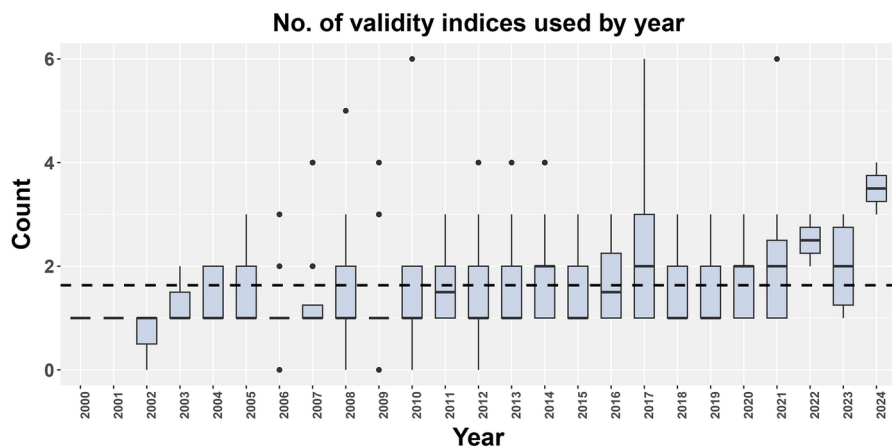
5.3 Analysis of validity indices

Validity indices are used to objectively evaluate the performance of a given method independently of the benchmarks it is tested in. This means that the output value of the validity indices is independent from the features of the benchmarks datasets, such as their size of their number of features in the case of classification datasets. The same analysis performed over the datasets (Sect. 5.1) is performed over the validity indices. Figure 10 shows the statistical summary on the usage of validity indices in CC literature (as it was performed for the datasets). From Fig. 10a it is clear that the most used validity index is the Normalized Mutual Information (NMI), followed by the Clustering Error (EC), the Rand Index (RI), the Adjusted Rand Index (ARI) and the F-measure. Time is used to compare methods a total of 25 times, which represents a very low percentage over the total number of comparisons. Note how Visual validation makes it to the top 15, despite not being an objective and reliable comparison method. Non Standard (NS) measures are used in 10 studies, meaning that the used measure is proposed specifically in the same paper for that specific case or that it is never referred to again in CC literature. Among the 15 most used measures, there is only one specifically designed to compare CC methods: the Unsat. Unsat measures the proportion of constraints violated by the output partition of any given method, and therefore can be used to measure scalability with respect to the number of constraints. In this study, authors want to draw two validity indices to the attention of the reader: the Constrained Rand Index (CRI), proposed in Wagstaff and Cardie (2000), and the Constrained F-measure (CF-measure), proposed in He et al. (2014). These two validity indices are versions of RI and F-measure, respectively, corrected by the number of constraints available. They assume that the higher the number of constraints available, the easier it is to score a high value in classic clustering validity indices, therefore they correct (lower) those values with the size of the constraint set. These two measures are used in very few cases, while they are specifically designed to



(a) No. of times each validity index in Table 4 is used in experiments

(b) Distribution of the number of validity indices used in experiments



(c) Variability of the number of validity indices used in every year (the dashed line represents the overall average)

Fig. 10 Statistics about the validity indices used in the experimental setups of all papers reviewed

benchmark CC methods. This fact is particularly remarkable in the case of the CRI, as it was proposed along with the first CC study ever in Wagstaff and Cardie (2000).

Figure 10b shows that the most common number of validity indices used in CC literature is 1. Using more than one validity index is a healthy practice in any study, as demonstrating the capabilities of a new method in more than one dimension reinforces positive conclusions about it. With respect to Fig. 10c, the variability observed in the other cases (Figs. 7c and 8c) is not present here, as the number of validity indices used is not related to the computation power, and most of them were proposed before the inception of CC.

Some of the validity indices in Table 4 do not need to be specifically defined, as it is the case of the Time or the Visual indices, which are self-explanatory. Others are incidentally

defined, such as the Precision, which is a by-product of the F-measure. Lastly, no general definition can be given for the Non Standard indices. For the rest of them, both a formal and intuitive definition can be found here. From now on, in this section, C refers to the partition generated by any given CC method, and C^* refers to the ground-truth partition. Please note the influence of the use of classification datasets in the selection of validity indices used to evaluate CC methods. All of the validity indices take two partitions as their input, and produce a measure according to their similarity or dissimilarity. Therefore, these validity indices can be used to evaluate the performance of a clustering algorithm only when one of the partitions given as input is the ground-truth partition, which can be obtained for labeled datasets only.

Normalized Mutual Information (NMI)

The NMI is an external validity index that estimates the quality of a partition with respect to a given underlying labeling of the data. In other words, NMI measures how closely a clustering algorithm could reconstruct the underlying label distribution. Taking C as the random variable denoting the cluster assignments of instances (the partition), and C^* as the random variable denoting the underlying class labels, the NMI can be formulated in terms of information theory as in Eq. 4 (Li et al. 2019; Basu et al. 2004b, a).

$$\text{NMI} = 2 \frac{I(C; C^*)}{H(C) + H(C^*)}, \quad (4)$$

where $I(X; Y) = H(X) - H(X|Y)$ is the mutual information between the random variables X and Y , $H(X)$ is the Shannon entropy of X and $H(X|Y)$ is the conditional entropy of X given Y . For more details on NMI please see Dom (2002). The output value range for the NMI is $[0, 1]$, with high values indicating a high level of similarity between the two partitions, and a low value indicating a low level of similarity.

The Clustering Error (CE)

The CE is the negative, unsupervised version of the classic classification accuracy. It measures the proportion of correctly clustered instances by best matching the cluster labels to the ground-truth labels. Given the permutation mapping function $\text{map}(\cdot)$ over the cluster labels, the CE with respect to $\text{map}(\cdot)$ can be computed as in Eq. 5 (Hu et al. 2014; Li et al. 2009; Li and Liu 2009). The best mapping function that permutes clustering labels to match the ground truth labels can be computed by the Kuhn-Munkres algorithm (the Hungarian method) Hu et al. (2014); Kuhn (1955). Please note that the CE validity index is sometimes used in its positive form, which is the clustering accuracy. It can be computed by just changing the condition in the indicator function $\mathbb{1}[\cdot]$ to be negative (replace $=$ by \neq). The output value range for the CE is $[0, 1]$, with high values indicating a low level of accuracy, and a low value indicating a high level of accuracy.

$$\text{CE} = 1 - \frac{1}{n} \sum_{i=1}^n \mathbb{1}[\text{map}(l_i^C) = l_i^{C^*}]. \quad (5)$$

The Rand Index (RI)

The RI measures the degree of agreement between two partitions. It can be used to measure the quality of a partition obtained by any CC algorithm by giving the ground-truth parti-

tion as one of them. Therefore, the two compared partitions are C and C^* . The RI views C and C^* as collections of $n(n-1)/2$ pairwise decisions. For each x_i and x_j in X , they are assigned to the same cluster or to different clusters by a partition. The number of pairings where x_i is in the same cluster as x_j in both C and C^* is taken as a ; conversely, b represents the number of pairings where x_i and x_j are in different clusters. The degree of similarity between C and C^* is computed as in Eq. 6 Rand (1971), where n is the number of instances in X . The output value range for the RI is $[0, 1]$, with high values indicating a high level of agreement between the two partitions, and a low value indicating a low level of agreement.

$$RI = \frac{a + b}{n(n-1)/2}. \quad (6)$$

The RI can be conveniently formulated in terms of the elements of a confusion matrix as well Zhong et al. (2019). Equation 7 defines these elements in terms of cluster memberships in a partition, which can be referred to as: True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN). Equation 8 makes use of these elements to give a new definition for the RI.

$$\begin{aligned} TP &= \{(x_i, x_j) | l_i^{C^*} = l_j^{C^*}, l_i^C = l_j^C, i \neq j\} \\ FP &= \{(x_i, x_j) | l_i^{C^*} = l_j^{C^*}, l_i^C \neq l_j^C, i \neq j\} \\ TN &= \{(x_i, x_j) | l_i^{C^*} \neq l_j^{C^*}, l_i^C \neq l_j^C, i \neq j\} \\ FN &= \{(x_i, x_j) | l_i^{C^*} \neq l_j^{C^*}, l_i^C = l_j^C, i \neq j\} \end{aligned} \quad (7)$$

$$RI = \frac{|TP| + |TN|}{|TP| + |FP| + |TN| + |FN|}. \quad (8)$$

The Adjusted Rand Index (ARI) The ARI is the corrected-for-chance version of the RI. This correction is done by taking into account the expected similarity of all comparisons between partitions specified by the random model to establish a baseline. This modifies the output value range of the original RI, transforming it into $[-1, 1]$ and slightly changing its interpretation. In ARI, a high output value still means a high level of agreement between the two partitions, and a low value means a low level of agreement. However, a value lower than 0 means that the results obtained are worse than those expected from the average random model. Equation 9 gives the formalization for the ARI Hubert and Arabie (1985).

$$ARI = \frac{RI - \text{Expected Index}}{\text{Maximum Index} - \text{Expected Index}}, \quad (9)$$

where Expected Index is the degree of similarity with a random model, Maximum Index is assumed to be 1, and RI is the RI value computed for partitions C and C^* .

The Constrained Rand Index (CRI)

The CRI is a revised version of the RI which includes constraints specifically in its definition. It introduces the concept of free decisions, which are defined as decisions not influenced by constraints. The CRI subtracts the number of available constraints from the numerator and the denominator of the classic RI (Wagstaff and Cardie 2000; Ding et al. 2018). As a result, it only evaluates the performance of the CC methods in the free deci-

sions. Equation 10 formalizes CRI, following the same notation as Eq. 6 (RI). Its results are interpreted as those of RI, but taking into account that the difficulty to obtain values close to 1 increases with the size of the constraint set $|CS|$.

$$\text{CRI} = \frac{a + b - |CS|}{n(n-1)/2 - |CS|}. \quad (10)$$

Pairwise F-measure (PF-measure) The PF-measure is defined as the harmonic mean of pairwise precision and recall, which are classic validity indices adapted to evaluate pairs of instances. For every pair of instances, the decision to cluster this pair into the same or different clusters is considered to be correct if it matches with the underlying class labeling. In other words, the PF-measure gives the matching degree between the obtained partition C and the ground-truth class labels C^* . It can be formalized as in Eq. 12 (Basu et al. 2004a; Li et al. 2019), where Precision and Recall are defined as in Eq. 11, following the notation introduced in Eq. 7 (Zhong et al. 2019). For more details on the PF-measure please see Hripcsak and Rothschild (2005). The output value range for the PF-measure is $[0, 1]$, with high values indicating a high level of agreement between the two partitions, and a low value indicating a low level of agreement.

$$\text{Precision} = \frac{|TP|}{|TP| + |FP|}, \quad \text{Recall} = \frac{|TP|}{|TP| + |FN|}. \quad (11)$$

$$\text{PF-measure} = 2 \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2|TP|}{2|TP| + |FP| + |FN|}. \quad (12)$$

The Constrained Pairwise F-measure (CPF-measure) The CPF-measure is a version of the classic PF-measures that takes constraints into account. It does so by including the number of ML constraints in the computation of the Precision and Recall terms as in Eq. 13. This way, the number of correctly clustered instances is penalized by the number of ML constraints. Subsequently, the higher the number of ML constraints available to perform clustering, the less credit the term TP is given. The final CPF-measure can be computed as in Eq. 14. The output value range for the CPF-measure is $[0, 1]$, and the value is interpreted as in the PF-measure.

$$\text{Precision}' = \frac{|TP| - |C_{=}|}{|TP| + |FP| - |C_{=}|}, \quad \text{Recall}' = \frac{|TP| - |C_{=}|}{|TP| + |FN| - |C_{=}|}. \quad (13)$$

$$\text{CPF-measure} = 2 \frac{\text{Precision}' \times \text{Recall}'}{\text{Precision}' + \text{Recall}'}. \quad (14)$$

The Unsat The Unsat measures the ability of any given CC method to produce partitions satisfying as many constraints as possible. It is computed as the ratio of satisfied constraints as in Eq. 15 (Davidson and Ravi 2005; González-Almagro et al. 2020). It produces a value in the range $[0, 1]$, with a high value indicating a high number of violated constraint, and a low value indicating the contrary.

$$\text{Unsat} = \frac{\text{Infs}(C, CS)}{|CS|}. \quad (15)$$

The Purity This is a classic validity index used to evaluate the performance of clustering methods. It measures the homogeneity of the generated partition, i.e.: the extent to which clusters contain a single class (Zhao and Miller 2005; Halkidi et al. 2012; Xu et al. 2011). It can be computed by determining the most common class of each cluster c_i (with respect to the true labels C^*), which can be done by determining the greatest intersection with respect to the ground-truth partition. The sum of all intersection is then divided by the total number of instances n in the partition C to obtain the Purity value of said partition. Equation 16 formalizes this concept. The output value range for the Purity is $[0, 1]$, with high values indicating high level of resemblance between the two partitions, and a low value indicating a low level of resemblance.

$$\text{Purity} = \frac{1}{n} \sum_{c_i \in C} \max_{c_i^* \in C^*} |c_i \cap c_i^*|. \quad (16)$$

The Jaccard Index (JC)

The JC measures similarity between finite sample sets. It is defined as the size of the intersection divided by the size of the union of the sample sets. However, this definition is inconvenient when JC is applied to measure the quality of a partition. Subsequently, a more useful definition can be given in terms of Eq. 7 as in Eq. 17 (Bae and Bailey 2006; Meilă 2003; Halkidi et al. 2001). Please note that a high value of the CJ in the range $[0, 1]$ indicates high dissimilarity between the two compared partitions, while a low value indicates high similarity.

$$\text{JC} = \frac{|TP|}{|TP| + |FP| + |FN|}. \quad (17)$$

The Folkes-Mallows Index (FMI) The FMI is another classic external validity index used to measure the similarity between two partitions. It is defined as the geometric mean of the Precision and the Recall (Halkidi et al. 2001). It can be formulated as in Eq. 18. The output value range for the FMI is $[0, 1]$, with high values indicating a high level of agreement between the two partitions, and a low value indicating a low level of agreement.

$$\text{FMI} = \sqrt{\frac{|TP|}{|TP| + |FP|} \times \frac{|TP|}{|TP| + |FN|}} = \sqrt{\text{Precision} \times \text{Recall}}. \quad (18)$$

The V-measure. This measure is closely related to the NMI, as it can be viewed as a version of it that computes the normalization of the denominator in Eq. 4 with an arithmetic mean instead of a geometric mean. The V-measure is defined as the harmonic mean of Homogeneity and Completeness, which evaluate a partition in a complementary way Rosenberg and Hirschberg (2007), Vlachos et al. (2008). Homogeneity measures the degree to which each cluster contains instances from a single class of C^* . This value can be computed as in Eq. 19, where $H(X|Y)$ is the conditional entropy of the class distribution of partition X with

respect to partition Y , and $H(X)$ is the Shannon entropy of X . Following the same notation, the Completeness can be defined as in Eq. 20. This can be intuitively interpreted as the degree to which each class is contained in a single cluster. Subsequently, the V-measure is computed as in 21 Rosenberg and Hirschberg (2007). Please note that another aspect to which the V-measure and the NMI are closely related is that the mutual information between two random variables $I(X; Y)$ can always be expressed in terms of the conditional distribution of said variables $H(X|Y)$ as follows: $I(X; Y) = H(X) - H(X|Y)$. The output value range for the V-measure is $[0, 1]$, with high values indicating a high level of similarity between the two partitions, and a low value indicating a low level of similarity.

$$\text{Homogeneity} = 1 - \frac{H(C^*|C)}{H(C^*)}. \quad (19)$$

$$\text{Completeness} = 1 - \frac{H(C|C^*)}{H(C)}. \quad (20)$$

$$\text{V-measure} = 2 \frac{\text{Homogeneity} \times \text{Completeness}}{\text{Homogeneity} + \text{Completeness}}. \quad (21)$$

5.4 Constraint generation methods

The most frequently used procedure to generate constraints is the one proposed in Wagstaff et al. (2001). It is a simple yet effective method to generate a set of constraints based on a set of labels, hence the generalized use of classification datasets as benchmarks in CC literature. It consists of randomly choosing pairs of instances and setting a constraint between them depending on whether their labels are the same (ML constraint) or different (CL constraint).

The way pairs of instances are chosen from the dataset may differ from one study to another. However, two common trends are observed. One first decides the percentage of labeled data the oracle has access to and then generates the complete constraints graph based on those labels. The other one first decides the size of the constraint set, and then extracts random pairs of instances from the complete dataset. On the one hand, the first method is more realistic in the sense that it has limited access to labeled data, although it may bias the solution towards poor local optima if the selected labeled instances are not representative enough of the whole dataset. On the other hand, the second constraint generation method has virtual access to the complete set of labels, as pairs of instances are randomly chosen, and the constraint set may end up involving all instances in the dataset in at least one constraint, which might not be a realistic scenario. Nevertheless, it is less likely to bias the solution towards local optima.

There is no consensus on how many constraints need to be generated in order to evaluate the capabilities of a given CC method. However, some general guidelines can be given. Based on Observation 5.1, it is clear that proper empiric evaluation of CC methods must include an averaging process on the results obtained for different constraint sets, in order to reduce the effects of specific adverse constraint sets.

Observation 5.1 *Specific constraint sets can have adverse effects.* Even if constraint sets are generated on the basis of the true labels, some constraint sets may decrease accuracy when predicting those very labels (Davidson and Basu 2007).

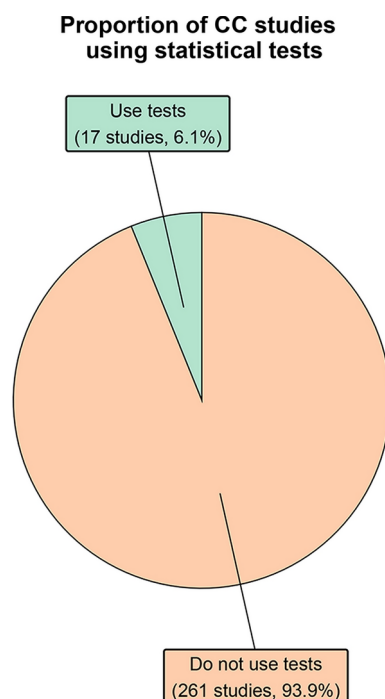
Given Observation 5.2, testing CC methods should include different levels of constraint-based information. This must be done in order to study the scaling capabilities of the proposed method. If a method does not scale the quality of the solutions with the size of the constraint set, any improvement over the solutions obtained with an empty constraint set may be due to random effects.

Observation 5.2 *The accuracy of the predictions scales with the amount of constraint-based information.* The quality of the solution should scale with the size of the constraint set: the more constraint are available, the better the results obtained are Davidson and Basu (2007).

5.5 On the use of statistical tests

Statistical testing is a settled practice in Computer Science. It provides objective evidence of the results of a study, supporting its conclusions, either if the used tests are Null Hypothesis Statistical Tests (NHST) Derrac et al. (2011) or the more recent Bayesian Tests Benavoli et al. (2017). However, this does not seem to be the case in the CC area. As shown in Fig. 11, only 6,1% of the studies (17 out of 278) analyzed in this review use statistical testing to support their conclusions. Authors consider this to be one of the major criticisms of the area of CC. Studies supporting their conclusions on mere average results values for any validity

Fig. 11 Piechart featuring the proportion of CC studies which use statistical tests



index/indices (as it is the case for most of them) should be encouraged to use statistical testing to further objectively prove their hypotheses.

6 Scoring system

The aim of this study is not only to give a taxonomy of constrained clustering methods, but also to provide researchers with tools to decide which methods to use. This section proposes an scoring system that is designed to indicate the potential and popularity of every reviewed method. This system assigns a numerical value to every CC method, which will be later used to rank all 315 of them. This value can be interpreted as a measure for the quality of the method. Three semantically different aspects of every method \mathcal{A} are analyzed to decide its score: the quality of the experimental setup they are tested in ($EQ_{\mathcal{A}}$), the confidence in the results obtained in the experiments ($VQ_{\mathcal{A}}$), and the influence of the method in later studies ($I_{\mathcal{A}}$). As for the formalization of these concepts, it is necessary to define the basic quantifiable elements that can be obtained from a method, which are shown in Table 5. All of them are lists that contain a value or a set of values associated to every method. Therefore, the length of these lists is always equal to the number of methods reviewed in this study. These lists can be accessed in a more precise way, by method or by year. For example: $D_{\mathcal{A}}$ is a single value which refers to the number of datasets used to test method \mathcal{A} , and D^Y is a list of values referring to the number of datasets used to test methods from year Y . Note that $M_{\mathcal{A}}$ is the list of methods used to compare \mathcal{A} , therefore M^Y is a list of lists.

In this study, authors have decided to evaluate each method within its time context, i.e. the year of publication of the method is taken into account to compute its score. This is done to remove the computational capacity component from the scoring system, as the number of datasets or the number of methods used to test new proposal is highly dependent of said parameter (see Fig. 7 and Fig. 8). Moreover, publication requirements and standards change over the years, and tend to become more rigid. Not taking the year of publication into account would greatly benefit recent methods, as their studies have to meet harder publication requirements which are usually related to their novelty and their experimental quality.

6.1 Scoring of the experimental quality

The quality of the experimental setup EQ used to test a method \mathcal{A} can be computed with information that is fully contained in the study which proposes it. Two of the experimental elements introduced in Sect. 5 take part in this procedure: the number of datasets used to

Table 5 Functions to get basic features of methods

Function	Meaning
Y	The list of publication years for all methods
M	The lists of sets of methods used in comparisons
C	The list of number of times a method is used to be compared with in later studies
T	The list of indicators for the use of statistical tests for every method
D	The list of number of datasets used to test every method
V	The list of number of validity indices used to evaluate every method

test the scoring method \mathcal{A} (in list D), and the methods that are used to compare it (in list M). Equation 22 gathers these two basic measures and gives the expression to compute the experimental quality of a scoring method $EQ_{\mathcal{A}}$.

$$EQ_{\mathcal{A}} = \frac{\alpha_1^{Y_{\mathcal{A}}} D'_{\mathcal{A}} + \alpha_2^{Y_{\mathcal{A}}} MS'_{\mathcal{A}}}{\alpha_1^{Y_{\mathcal{A}}} + \alpha_2^{Y_{\mathcal{A}}}}, \quad (22)$$

where the $MS_{\mathcal{A}}$ term is computed on the basis of $M_{\mathcal{A}}$, but taking the publication year of both the scoring method \mathcal{A} and the compared method m into account, as shown in Formula 23. As a result, every compared method m contributes in an inversely proportional way to MS with respect to the difference between the years of publication of the two methods.¹ This way, methods published in years close to the year of publication of \mathcal{A} contribute more to $MS_{\mathcal{A}}$ than methods published long time before \mathcal{A} . In other words: the contribution of every method is proportional to its novelty in the year it is used to make comparisons. Please note that non CC methods are always considered to be published one year before the first CC was published (1999). Subsequently, the contribution of non CC methods to $MS_{\mathcal{A}}$ decays invariably with the years. By doing this, the first CC methods comparing with classic clustering methods are given credit by the comparison, as no CC baseline methods could have been established by that time. However, this is not the case for modern CC methods, which must be compared to other CC methods for said comparison to be meaningful.

$$MS_{\mathcal{A}} = \frac{1}{|M_{\mathcal{A}}|} \sum_{m \in M_{\mathcal{A}}} \frac{1}{Y_{\mathcal{A}} - Y_m}. \quad (23)$$

Both the values of D and MS are normalized within each year following the normalization procedure described in Eq. 24 (min-max normalization), which results in D' and MS' . This is done to lessen the effects that the computation capability context can have in $EQ_{\mathcal{A}}$. Please note that, only with respect to the year grouping aspects, methods published in years 2000–2003 are considered to belong to the same time context, hence they are treated as if they were published in the same year. With this in mind, neither Eq. 23 nor Eq. 28 are affected. This is done to enable within-groups normalization and comparisons, as only 1 method was published in 2000 and 2001, and only 3 were published in 2002 and 2003. These were the years in which the CC research topic was conceived and it was starting to grow in interest (see Sect. 1.3). Subsequently, authors consider this exception to be justified.

$$D'_{\mathcal{A}} = \frac{D_{\mathcal{A}} - \min(D^{Y_{\mathcal{A}}})}{\max(D^{Y_{\mathcal{A}}}) - \min(D^{Y_{\mathcal{A}}})}, \quad MS'_{\mathcal{A}} = \frac{MS_{\mathcal{A}} - \min(MS^{Y_{\mathcal{A}}})}{\max(MS^{Y_{\mathcal{A}}}) - \min(MS^{Y_{\mathcal{A}}})}. \quad (24)$$

The last elements to be introduced from Eq. 23 are the $\alpha_1^{Y_{\mathcal{A}}}$ and $\alpha_2^{Y_{\mathcal{A}}}$ values, which are different for every year. These values are used to determine the influence of the datasets score and the compared methods score in the computation of the experimental quality score. They are computed as in Eq. 25, where $\sigma(\cdot)$ and $\mu(\cdot)$ are functions which return the standard deviation and the mean of the list of values given as argument, respectively. Subsequently, $\alpha_1^{Y_{\mathcal{A}}}$ and $\alpha_2^{Y_{\mathcal{A}}}$ are directly proportional to the standard deviation of the datasets scores and the compared methods scores, respectively. In other words, $\alpha_1^{Y_{\mathcal{A}}}$ and $\alpha_2^{Y_{\mathcal{A}}}$ are used to give

¹ This difference is considered to be 1 for methods published in the same year, in order to avoid divisions by 0.

more importance to disperse measures, which are usually good discriminators, and therefore are better suited to be used in a scoring system.

$$\alpha_1^{Y_A} = \frac{\sigma(D'^{Y_A})}{\mu(D'^{Y_A})}, \quad \alpha_2^{Y_A} = \frac{\sigma(MS'^{Y_A})}{\mu(MS'^{Y_A})}. \quad (25)$$

6.2 Scoring of the validation procedure quality

Once again, the information needed to determine the quality of the validation procedures VQ used to evaluate the results obtained with a method \mathcal{A} is fully contained in the study which proposes it. The two experimental elements (introduced in Sect. 5) that take part in this procedure are: the number of validity indices used to quantify the results obtained by the scoring method \mathcal{A} (in list V), and the indicator of the use of statistical testing procedures (in list T). The list T indicates which methods use statistical tests by giving them a value of 1, whereas the 0 value is assigned to method that do not support their conclusions with statistical tests. Equation 26 shows the expression to compute the validation procedures quality of a scoring method $VQ_{\mathcal{A}}$.

$$VQ_{\mathcal{A}} = V'_{\mathcal{A}} + T_{\mathcal{A}}, \quad (26)$$

where $V'_{\mathcal{A}}$ is the normalized value of $V_{\mathcal{A}}$, which is computed following formula 27. Please note that in this case the min-max normalization does not take the publication year into account (in contrast to Eq. 24), as the number of validity indices used to quantify the results of the proposed methods does not show any tendency with respect to the publication year (see Fig. 10). Authors consider studies which use statistical tests to have a significantly higher confidence rate in their results, hence the strength of the second term in Eq. 26.

$$V'_{\mathcal{A}} = \frac{V_{\mathcal{A}} - \min(V)}{\max(V) - \min(V)}. \quad (27)$$

6.3 Scoring of the influence

The influence I of a given method \mathcal{A} cannot be computed with just the information contained in the study which proposes the method. This aspect of the method refers to how influential it has been in later literature, i.e. how many times method \mathcal{A} has been used to make experimental comparisons. This number differs from the total number of times it has been cited, as a citation does not guarantee that the method is being used to make comparisons. In fact, this is one of the hardest aspects to evaluate, and requires experimental comparisons carried out in a corpus of papers to be self-contained. This means that no method referred in the experimental section of any paper is left out of the corpus. As will be explained in Sect. 7, authors have made sure that this is the case for the taxonomy presented in this study. However, once this information has been obtained, an index for the influence of any given method can be computed as simply as in Eq. 28, where CY refers to the current year, therefore $CY = 2022$. This is, the number of times a method is used in experimental comparisons divided by the number of years it has been available.

$$I_A = \frac{C_A}{CY - Y_A}. \quad (28)$$

6.4 Final scoring

The final scoring S of any given method A can be computed by normalizing and adding up the three partial scores presented in previous sections, and scaling the output range to $[0, 100]$. Equation 29 gives the expression to compute S_A . Please note that none of the partial scores are bounded, hence the need of the min-max normalization step in Eq. 30.

$$S_A = \frac{(EQ'_A + VQ'_A + I'_A) \times 100}{3}. \quad (29)$$

$$\begin{aligned} EQ'_A &= \frac{EQ_A - \min(EQ)}{\max(EQ) - \min(EQ)}. \\ VQ'_A &= \frac{VQ_A - \min(VQ)}{\max(VQ) - \min(VQ)}. \\ I'_A &= \frac{I_A - \min(I)}{\max(I) - \min(I)}. \end{aligned} \quad (30)$$

Finally, authors want to remark that no hand-tuned parameter is needed to compute S_A . Consequently, the probability of introducing any human bias in the scoring system is reduced.

7 Taxonomic review of constrained clustering methods

In this section, a ranked taxonomic classification for a total of 315 CC methods is presented. The starting point to obtain the corpus of CC studies to be reviewed was to run Query 7.1 in the Scopus scientific database.

Query 7.1 *Scopus Query*: (TITLE-ABS-KEY (“constrained clustering”) OR TITLE-ABS-KEY (“semi-supervised clustering”) AND TITLE-ABS-KEY (“constraint” OR “constraints” OR “constrained”))

This is a very general and wide query, which was conceived to make sure that the most of the CC research area was contained in its output. Authors briefly reviewed and evaluated all of outputted papers to remove those which did not belong to the CC research area. Afterwards, a recursive procedure was used to obtain the final corpus to be reviewed: if a study compares its proposal with a CC proposal not included in the corpus, then the newly identified study is included and applied this procedure over. This is done with the aim of producing a self-contained comparison. Figure 12 presents a taxonomic tree, organizing the categories in which the CC landscape may be divided. Particular methods are introduced and discussed in Sects. 7.1 to 7.17, where tables detailing the features of every method can be found.

As Fig. 12 shows, a high-level dichotomy can be made within the CC area: constrained partitionial methods versus constrained distance metric learning (DML) methods (Davidson

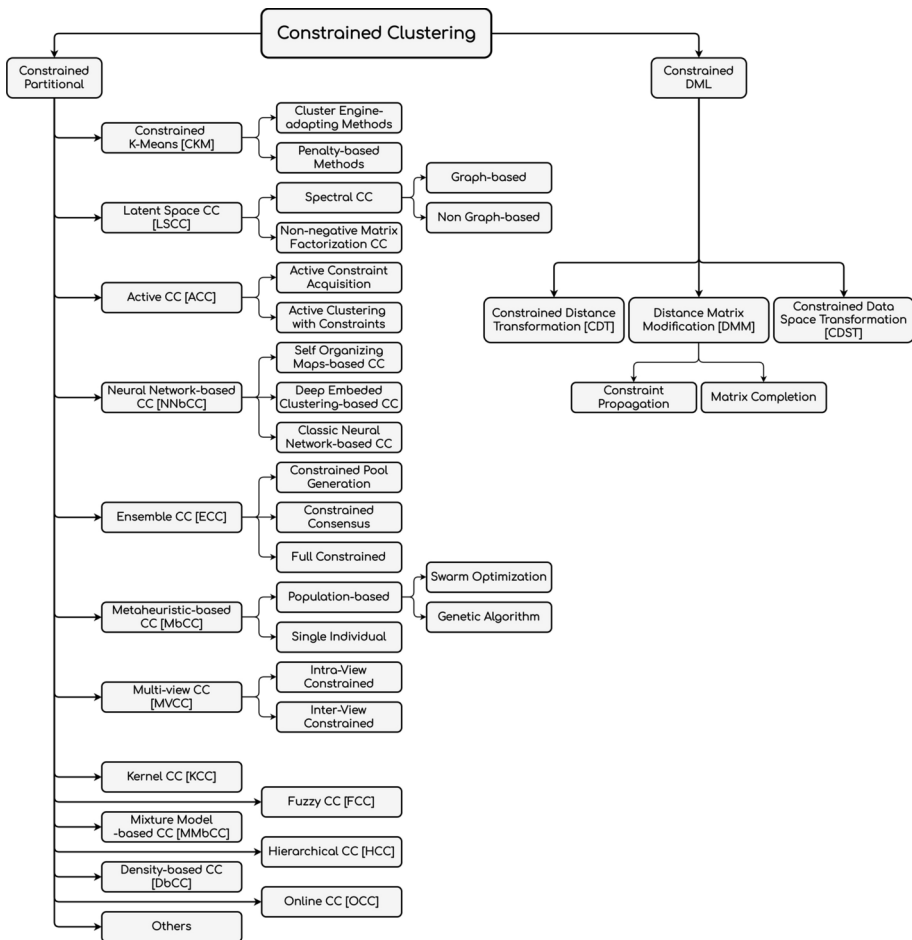


Fig. 12 Taxonomic tree for the CC landscape

and Basu 2007; Basu et al. 2008). The main difference lies in their approach to CC and in their output. In constrained partitional methods, constraints are included into a procedure that progressively builds a partition for the dataset. This is typically done by designing a clustering engine which can deal with constraints or by including constraints in the objective function of a given method, for example, by means of a penalty term. Generally, constrained partitional methods produce a partition of the dataset, which may be accompanied by other by-products of the CC method, such as new constraints or feature weights. On the other hand, constrained DML methods aim to learn a distance metric that reflects the information contained in the constraint set. In general, the learned distances will try to bring ML instances together in the output space, while trying to maximize the distance between CL instances. Generally, constrained DML method do not produce a partition of the dataset, but a new metric, data space or distance matrix. This output can be used to later produce a partition by means of classic clustering algorithms, or even by constrained clustering algorithms. Please note that the difference found between the tree classes of constrained

DML methods is merely conceptual, as the results of all the three of them (new metric/data space/distance matrix) can always be derived from each other using classic DML methods. However, the distinction between the three classes is useful from the point of view of CC, as their approach to the problem is different. The vast majority of CC methods are constrained partitional methods. There are hybrid methods, which combine features inherited from both approaches.

Feature tables in Sects. 7.1 to 7.17 generally include 8 columns:

- The S_A column gives the quality score assigned to each method. It is computed following the scoring system introduced in Sect. 6.
- The *Acronym* column provides the acronym of the method. Bearing in mind that some authors do not name their methods, we have decided to refer to these methods by the initials of their authors' names. However, there are exceptions for this rule, such as methods that are not named by their author but are consistently referred by later literature with a given name. In cases in which two methods have the same name, the year it was proposed in is added at the end of the name to differentiate them.
- The *ID* column assigns a numeric identifier to each method. This number can be used to find the method in Appendix A, where the full name of all methods are listed, along with its identifier and its acronym. Full names are listed only in said appendix for the sake of readability and visualization.
- The *Penalty* column takes two values: "✓" or "×". This indicates whether constraints are included in the method by means of a penalty term in its objective function ("✓") or by other means ("×").
- The *ML* and *CL* columns refer to the type of constraints the method can handle. *Soft* is used for soft constraints, *Hard* is used for hard constraints, *Hybrid* is used for method that can use both hard and soft constraints. If a method cannot handle ML or CL constraints it is indicated with "-".
- The *Hybrid* column indicates if the method belongs to more than one class, specifying the classes it belongs to. The "-" character is used if the method belongs to only one class.
- The *Year* and *Ref* columns provide the year of publication and the reference of the method respectively.

7.1 Constrained K-means

The Constrained K-Means (CKM) category gathers methods that can be considered modifications over the classic K-Means algorithm to include constraints. Their common feature is that all of them use an expectation-minimization (EM) optimization scheme. In the expectation step of an EM scheme, instances are assigned to clusters minimizing the error according to an objective function. In the minimization step, centroids are reestimated according to the assignments made in the expectation step. A plethora of objective functions and centroid update rules has emerged to approach the CC problem, although methods belonging to these category can be divided into two main categories.

7.1.1 Cluster engine-adapting methods

Cluster engine-adapting methods modify one of the two steps (or both) from the EM scheme in order to include constraints. Methods belonging to this category are presented in Table 6. The first and most basic method performing CC this way is COP-K-Means. It modifies the instance to clusters assignment rule from the expectation step (the clustering engine) so that an instance is assigned to the cluster associated to its closer centroid whose assignment does not violate any constraint. Another popular technique in this category consists of performing clustering over the previously computed chunklet graph (which enforces ML), considering only CL in the expectation step. This is how methods like CLAC, CLWC, PCCK-Means, PCBK-Means or SSKMP perform CC. All of them consider hard ML. They differ from each other in the way in which they build their particular chunklet graph, which may contain weighted chunklets (as in CLAC and CLWC), or may rank chunklets in order for them to be examined more efficiently (as in PCCK-Means). Other methods use basic chunklet graph (like PCBK-Means and SSKMP). Some methods include constraints in EM scheme that are not basic K-Means, like SSKMP, which is a constrained version of the K-Medoids algorithm. Based on the COP-K-Means, methods like CLC-K-Means or ICOP-K-Means are designed to solve the dead-ends problem found in the basic algorithm.

7.1.2 Penalty-based methods

Penalty-based methods include constraints by means of a penalty term in the objective function of an EM scheme. These methods are presented in Table 7. Some of them simply modify previous CC or classic clustering algorithms to include a plain penalty term, such as SCOP-K-Means, PCK-Means, S-SCAD. Other methods, like MPCK-Means, also include a metric learning step in the EM scheme, which estimates a cluster-local distance measure for every cluster, and allows them to find clusters with arbitrary shapes. Besides, there are methods which use variable penalty terms, such as HMRF-K-Means, CVQE, LCVQE or CVQE+ that include the distance between constrained instances in it. This is, more relevance is assigned to ML relating distant instances and CL relating close instances. Methods which combine pairwise constraints and other types of constraints have also emerged, like PCS, which includes cluster-size constraints in its EM scheme too. Other methods like GPK-Means use a Gaussian function and the current cluster centroids to infer new constraints in the neighborhood of the original constraints. These new constraints are added to the constraint set and used in subsequent iterations of the EM scheme.

7.2 Latent space CC

Latent space clustering performs clustering in a space which is different from the input space and which is computed on the basis of the dataset, and also the constraint set in Latent Space CC (LSCC). The input to these algorithms is an adjacency matrix defining the topology of the network (or graph) over which clustering needs to be performed. Each row or column may be regarded as the feature or property representation of the corresponding node. Latent space clustering methods first obtain new property representations in a latent space for each node by optimizing different objective functions, and then clusters nodes in that latent space (Yang et al. 2014).

Table 6 Feature table for CKM - Cluster Engine-adapting Methods

S.A	Acronym	ID	Penalty	ML	CL	Hybrid	Year	Refs.
33.33	COP-K-Means	2	×	Hard	Hard	–	2001	Wagstaff et al. (2001)
17.28	PCSK-Means	42	×	Hard	Soft	–	2007	Tang et al. (2007)
7.62	CLAC	55	×	Hard	Hard	Graph-based	2008	Domeniconi et al. (2008)
7.56	MLC-K-Means	57	×	Soft	Soft	–	2008	Huang et al. (2008)
12.54	CLWC	66	×	Hard	Soft	–	2008	Cheng et al. (2008)
7.90	COPGB-K-Means	71	×	Soft	Soft	Graph-based	2008	Rothaus and Jiang (2008)
9.14	PCKK-Means	72	×	Hard	Soft	–	2008	Zhang and Wong (2008)
3.79	SCK-Means	76	×	Hybrid	Hybrid	–	2009	Ares et al. (2009)
5.16	CMSC	90	×	Soft	Soft	–	2009	Tuzel et al. (2009)
6.49	SCKMM	101	×	Soft	Soft	Constrained Distance Transformation	2010	Yin et al. (2010)
6.04	PCBK-Means	102	×	Hard	Soft	–	2010	Yin et al. (2010)
5.78	ICOP-K-Means	104	×	Hard	Hard	–	2010	TAN et al. (2010)
10.24	CLC-K-Means	115	×	Hard	Hard	–	2011	Rutayisire et al. (2011)
5.18	SKMS	190	×	Soft	Soft	–	2014	Anand et al. (2013)
4.94	BCK-Means	255	×	Hard	Hard	–	2019	Le et al. (2018)
8.73	SSKMP	256	×	Hard	Hard	–	2019	Randel et al. (2018)

7.2.1 Spectral CC

Classic spectral clustering algorithms try to obtain the latent space by finding the most meaningful eigenvectors of the adjacency matrix, which are used to define the embedding in which clusters are eventually obtained (Yang et al. 2014). A dichotomy can be made within this category: in graph-based methods the input data is always given in the form of a graph, while in non-graph based the input is an adjacency matrix or a regular dataset, which can be transformed into an adjacency matrix. Please note that this distinction only affects the conceptual level of the spectral CC category, as a graph can always be converted to and adjacency matrix and vice versa, all methods from one category may also be applied in the other category. However, the authors have decided to make this distinction, since the terminology and concepts used in the studies referring to each of them differ greatly and can be misleading if interpreted together.

Graph-based spectral clustering

In graph-based spectral clustering, the input is assumed to be a graph. The goal is to partition the set of vertices of the graph, taking into account the information contained in the vertices themselves and in the edges of the graph. Edges may carry similarity or dissimilarity information regarding the vertices they connect. Some common strategies to perform graph clustering try to maximize the similarity of vertices within a cluster, normalizing the contribution of each to the objective by the size of the cluster in order to balance the size of the clusters. Other methods try to minimize the total cost of the edges crossing the cluster boundaries (Domeniconi et al. 2011). Graph-based methods are particularly suitable to perform CC, as constraints can be naturally represented in their graph form, which is the constraint graph and the chunklet graph (introduced in Sect. 4).

Table 8 gathers graph-based spectral clustering methods. COP-b-coloring and CLAC exemplify the use of chunklets to enforce ML, while including CL by other means. Other methods modify the input graph to include the information contained in the constraint set. For example, PCOG modifies affinities so that ML instances are always placed in the same connected components and removes edges which connect CL instances. CCHAMELEON modifies affinities between constrained instances, making them larger if instances are related by ML and lower in the case of CL. The all-pairs-shortest-path algorithm is used to propagate changes. PAST-Toss uses a spanning tree based technique to perform CC directly over the constraint graph. SCRAWL is the only non-spectral graph-based CC algorithm, as it does not need pairwise similarity/dissimilarity information to perform CC, but graph-related measures instead.

Non graph-based spectral clustering

In these methods, the input is given in the form of an adjacency matrix, or a dataset whose adjacency matrix can be easily obtained. Two techniques are commonly used to include pairwise constraints in these methods. (1) Modifying similarities/dissimilarities in the original adjacency matrix, computing eigenvectors and eigenvalues to obtain the spectral embedding. (2) Using the constraints to directly modify the embedding, obtained on the basis of the original adjacency matrix. Any classic clustering method can be used to obtain the final partition in the new embedding, which can always be mapped to the original data (Yang et al. 2014).

Table 9 shows a list of non-graph-based spectral CC methods. The first spectral CC method is found in KKM/SL, known in the literature by these two acronyms (respectively

Table 7 Feature table for CKM - Penalty-based Methods

<i>S_A</i>	Acronym	ID	Penalty	ML	CL	Hybrid	Year	Refs.
0.00	SCOP-K-Means	3	✓	Soft	Soft	–	2002	Wagstaff (2002)
16.18	PCK-Means	9	✓	Soft	Soft	–	2003	Basu et al. (2003, 2004a)
8.84	HMRF-K-Means	12	✓	Soft	Soft	Constrained Distance Transformation	2004	Basu et al. (2004b)
13.97	GPK-Means	18	✓	Soft	Soft	–	2005	Eaton (2005)
8.76	CVQE	19	✓	Soft	Soft	–	2005	Davidson and Ravi (2005)
8.74	LCVQE	44	✓	Soft	Soft	–	2007	Pelleg and Baras (2007)
6.14	S-SCAD	52	✓	Soft	Soft	–	2007	Frigui and Mahdi (2007)
4.35	SemiStream	127	✓	Soft	Soft	Online CC	2012	Halkidi et al. (2012)
4.34	TDCK-Means	193	✓	Soft	Soft	Online CC	2014	Rizoiu et al. (2014)
4.68	PCS	191	✓	Soft	Soft	–	2014	Zhang et al. (2014)
17.64	PC-HCM-NM	176	✓	Soft	Soft	–	2014	Endo et al. (2014)
2.51	AC-CF-tree	185	✓	Soft	Soft	Active Clustering with Constraints	2014	Lai et al. (2014)
12.47	HSCE	207	✓	Soft	Soft	Non Graph-based & Constrained Pool Generation	2015	Wei et al. (2015, 2018)
2.47	CVQE+	242	✓	Soft	Soft	Active Clustering with Constraints	2018	Mai et al. (2018)
3.00	PCSK-Means(21)	276	✓	Soft	Soft	–	2021	Vouros and Vasilaki (2021)
16.80	fssK-Means	279	✓	Soft	Soft	Time Series	2021	He et al. (2019)
1.45	CDGSC	312	×	Soft	–	–	2023	Shimizu et al. (2024)
8.20	<i>l_p</i> Km	314	×	Soft	Soft	–	2023	Bibi et al. (2023)
21.70	PKM-Mono	311	✓	Soft	Soft	Monotonic	2024	González-Almagro et al. (2024)

obtained from the name of its authors and the title of the study which proposes it). It is based on HMRF, performing CC by modifying the transition probabilities of the field based on the constraints. KKM/SL and AHMRF constitute the only two HMRF-based approaches to spectral CC. Another common technique in spectral CC is learning a kernel matrix based on the dataset over which spectral clustering is later conducted, as in RSCPC, CCSR, CCSKL, LSE or SSCA. This kernel matrix is usually built taking both pairwise distances and constraints into account. With respect to the methods which modify the original adjacency matrix, two strategies are the most used ones: some methods, such as ACCESS or CSC, simply set entries which relate constrained instances to specific fixed values, while other methods, such as NSDR-NCuts or LCPN, use constraint propagation techniques to propagate changes in the affinity matrix once it has been modified.

7.2.2 Non-negative matrix factorization CC

Non-negative Matrix Factorization (NMF) clustering algorithms obtain the new representation of the data by factorizing the adjacency matrix into two non-negative matrices (Yang et al. 2014). These two matrices can be interpreted as the centroids of the partition and the membership degree of each instances to each cluster. By doing this, all instances can be obtained as a linear combination of each column of the centroids matrix, parameterized by its corresponding membership, found in its associated row from the membership matrix. It can be proven that minimizing the difference between the original dataset matrix and the product matrix (computed usually as the Frobenius norm) is equivalent to performing K-Means clustering over the dataset (Zhang et al. 2015).

Table 10 presents a list of NMF-based CC methods. One of the most common strategies to include constraints into the classic NMF-based methods is to modify its objective function. This can be done by means of a penalty term accounting for the number of violated constraints, as in PNMF, SSCsNMF and SS-NMF(08), or by more complex techniques, as in CPSNMF or NMFS. Another popular strategy is forcing affinities between ML instances to be 0 and affinities between CL instances to be 1, as in NMFCC and SymNMFCC.

7.3 Active CC

Active learning is a subfield of machine learning in which algorithms are allowed to choose the data from which they learn. The goal of active learning is to reduce the amount of supervisory information needed to learn, and therefore reduce the human effort and implication in machine learning. In the active learning paradigm, learning methods are provided with an oracle, which is capable of answering a limited number of a specific type of query. For example, in traditional classification, active learning is used to select the best instances to be labeled from a dataset, so the oracle provides the label of the specific queried instance (Settles 2009). In Active CC (ACC), the oracle is queried about the type of constraint relating pairs of instances. The key aspect in any active learning algorithm is how to choose the queries to be presented to the oracle.

Active learning is specially useful in CC. In order to have an explanation for this, we compare the complexity of the answers given by oracles involved in active classification and ACC. In active classification, the oracle is queried about the class of a given instance. This query has a virtually infinite number of answers, as the number of classes in the dataset

Table 8 Feature table for the LSCC - Spectral CC - Graph-based methods

<i>S.A</i>	ID	Acronym	Penalty	ML	CL	Hybrid	Year	Refs.
4.32	38	COP-b-coloring	×	Hard	Hard	–	2007	Elghazel et al. (2007)
3.56	54	PAST-Toss	×	Soft	Soft	Single Individual	2008	Coleman et al. (2008)
7.62	55	CLAC	×	Hard	Hard	Cluster Engine-adapting Methods	2008	Domeniconi et al. (2008)
7.90	71	COPGB-K-Means	×	Soft	Soft	Cluster Engine-adapting Methods	2008	Rothaus and Jiang (2008)
8.29	98	GBSSC	×	Hard	Soft	Dimensionality Reduction	2010	Yoshida (2010); Yoshida and Okatani (2010); Yoshida (2011, 2014)
4.54	112	CCHAMELEON	×	Soft	Soft	–	2011	Anand and Reddy (2011)
11.27	184	SCRAWL	×	Soft	Soft	–	2014	He et al. (2014)
8.05	269	PCOG	×	Hard	Hard	Non Graph-based	2020	Nie et al. (2021)

may be unknown. On the other hand, an oracle involved in CC is queried with two instances and asked about the constraint between them (ML or CL), which is the same as asking whether they belong to the same class. There are only three possible answers to this question: “yes”, “no” or “unknown”. It is clear that the oracle in CC carries out a far simpler job than the one in classification. Let us remember that the oracle is just an abstraction of a knowledge source, which is generally a human user. Querying a human about the relation between instances instead of about their class requires less effort from them and leads to less variability and noise in the queries, as the extensive literature in active CC shows.

Two subcategories can be found in Active CC. In active constraint acquisition, constraints are actively generated before performing constrained clustering, while in active clustering with constraints, both clustering and active constraint generation are performed iteratively at the same time. This way, in active constraint acquisition queries are generated on the basis of the dataset and the current state of the constraint set, while in active clustering with constraints information about the current partition can also be used.

7.3.1 Active constraint acquisition

The immediate result of active constraint acquisition methods is a set of constraints, rather than a partition of the dataset. However, the partition can be obtained using any other CC method by just feeding the generated constraints into it, along with the dataset used to generate the constraints. Only the dataset, the constraint set generated so far, and an initial unconstrained partition are available to perform active constraint learning in the active constraint acquisition paradigm. No CC algorithm is involved in the constraint acquisition step. Table 11 shows a list of active constraint acquisition methods. Columns indicating the type of constraint these methods can handle have been removed, as they are not relevant here. Column “CC Method” has been added, indicating the CC method used to produce a partition based on the constraint generated by every active constraint acquisition method in the experimental section of the studies that propose them.

Many strategies to select the best pair of instances to query to the oracle have been proposed. Some methods start by dividing the dataset into preliminary groups and then use the oracle to query constraints which consolidate that information, such as FFQS, MMFFQS, SSL-EC or LCML. Other methods focus on finding the boundaries in the dataset to select pairs of instances from them, such as ACCESS, ASC(10) or SACS. Besides, there are methods that use classic clustering to obtain preliminary information from the dataset, such as the co-association of instances or the compactness of clusters. DGPC, JDfD, WAKL, MICS, AAA(19), AIPC, ALPCS or ASCENT are some of the methods which use this strategy. Other methods focus on specific features from the constraint themselves in order to evaluate them and select the more informative ones, such as AAVV or KAKB. More complex approaches can be found in AAA(18), which solves the active constraint acquisition problem as an instance of the uncapacitated k-facility location problem, or RWACS, which uses the commute time from graph theory to select queries.

7.3.2 Active clustering with constraints

In active clustering with constraints, a CC procedure and a constraint generation method are applied alternately. The immediate result of these methods are both a partition of the dataset

Table 9 Feature table for the LSCC - Spectral CC - Non Graph-based methods

<i>S_A</i>	Acronym	ID	Penalty	ML	CL	Hybrid	Year	Refs.
14.06	KKM	6	×	Soft	Soft	–	2003	Kamvar et al. (2003)
11.10	TB/SBM	7	×	Hybrid	Hybrid	–	2004	Bie et al. (2004)
12.72	ACCESS	16	×	Soft	Soft	Active Constraint Acquisition	2005	Xu et al. (2005)
6.85	CSC	20	×	Soft	Soft	–	2005	Xu et al. (2005)
12.33	LCPN	61	×	Soft	Soft	Constraint Propagation	2008	Lu and Carreira-Perpinan (2008)
8.40	S3-K-Means	63	✓	Soft	Soft	–	2008	Hu et al. (2008)
3.64	RSCPC	86	×	Soft	Soft	–	2009	Alzate and Suykens (2009)
15.14	CCSR	87	✓	Soft	Soft	–	2009	Li et al. (2009)
6.80	SCLC	88	×	Soft	Soft	–	2009	Xu et al. (2009)
15.70	CCSKL	89	×	Soft	Soft	Kernel CC	2009	Li and Liu (2009)
8.60	CSP	105	×	Hybrid	Hybrid	–	2010	Wang and Davidson (2010)
13.45	ASC	110	×	Any	Any	Active Clustering with Constraints	2010	Wang and Davidson (2010)
5.65	LSE	148	×	Soft	Soft	–	2012	Shang et al. (2012)
3.90	IU-Red	144	×	Soft	Soft	Active Clustering with Constraints	2012	Wauthier et al. (2012)
8.55	SSC-ESE	152	✓	Soft	Soft	–	2012	Jiao et al. (2012)
7.73	NSDR-NCuts	135	×	Soft	Soft	–	2012	Chen and Feng (2012)
13.21	COSC	138	×	Hybrid	Hybrid	–	2012	Rangapuram et al. (2012)
3.62	SSCA	181	×	Soft	Soft	–	2014	Ding et al. (2014)
6.75	FHCSC	183	×	Hybrid	Hybrid	–	2014	Wang et al. (2014)
6.93	CNP-K-Means	187	×	Soft	Soft	Dimensionality Reduction	2014	Wang et al. (2014)
4.43	STSC	188	×	Soft	Soft	Matrix Completion	2014	Wang et al. (2014)
5.95	LXDxD	200	×	Soft	Soft	Non-negative Matrix Factorization CC	2015	Yang et al. (2014)
12.47	HSCE	207	×	Soft	Soft	Constrained Pool Generation & Penalty-based Methods	2015	Wei et al. (2015, 2018)
8.78	FAST-GE	218	×	Soft	Soft	–	2016	Cucuringu et al. (2016)

Table 9 (continued)

S_A	Acronym	ID	Penalty	ML	CL	Hybrid	Year	Refs.
6.56	URASC	227	×	Soft	Soft	Active Clustering with Constraints	2017	Xiong et al. (2016)
16.49	FAST-GE2.0	231	×	Soft	Soft	–	2017	Jiang et al. (2017)
12.06	TL-APICF	233	✓	Soft	Soft	–	2017	Qian et al. (2016)
12.06	TII-APICF	234	✓	Soft	Soft	–	2017	Qian et al. (2016)
1.89	AHMRF	246	×	Soft	Soft	–	2018	Ding et al. (2018)
15.04	SFS ³ EC	257	×	Soft	Soft	Full Constrained	2019	Tian et al. (2019)
18.17	MVCSC	261	×	Soft	Soft	Intra-View Constrained	2019	Chen et al. (2019)
8.05	PCOG	269	×	Hard	Hard	Graph-based	2020	Nie et al. (2021)
14.46	CWLP	313	×	Soft	Soft	–	2024	(Zhang et al. 2024; Shimizu et al. 2022)

Table 10 Feature table for the LSCC - Non-negative Matrix Factorization CC methods

S_A	Acronym	ID	Penalty	ML	CL	Hybrid	Year	Refs.
5.68	NMFS	45	×	Soft	Soft	—	2007	Li et al. (2007)
16.22	SS-NMF(08)	73	×	Soft	Soft	—	2008	Chen et al. (2008)
4.74	OSS-NMF	96	×	Soft	Soft	Co-Clustering	2010	Ma et al. (2010)
11.82	SS-NMF	109	×	Soft	Soft	Co-Clustering	2010	Chen et al. (2009)
4.51	PNMF	123	✓	Soft	Soft	—	2011	Zhu et al. (2011)
6.47	SSC ₃ NMF	129	×	Soft	Soft	—	2012	Jing et al. (2012)
5.95	LXDXD	200	×	Soft	Soft	Non Graph-based	2015	Yang et al. (2014)
4.40	CPSNMF	211	×	Soft	Soft	Constraint Propagation	2016	Wang et al. (2015)
8.20	NMFCC	219	×	Soft	Soft	—	2016	Zhang et al. (2015)
11.78	SymNMFCC	220	×	Soft	Soft	—	2016	Zhang et al. (2015)
5.02	PCPSNMF	251	×	Soft	Soft	—	2018	Wu et al. (2018)
2.85	CMVNMF	254	×	Soft	Soft	Inter-View Constrained & Active Clustering with Constraints	2018	Zong et al. (2018); Zhang et al. (2015)

and a constraint set. These methods usually start by computing an unconstrained partition of the dataset. After this, some criteria are applied to select pairs of instances to query the oracle on the basis of the obtained partition. The answers to these queries are used to generate and save new constraints, which are later used to generate a new partition of the dataset by means of a CC method. Active clustering with constraints methods iterate these steps to produce the final constraint set and the partition. The active constraint generation method can be dependent of the CC method used to produce partitions, in which case they cannot be used separately. On the other hand, some active constraint generation methods are designed to be paired with any CC algorithm.

Table 12 gathers a list of active clustering with constraints methods. A major trend in this category is found in the use of the uncertainty of instances to rank and pair them to select the more uncertain ones and query them to the oracle. The uncertainty is always computed based in the current partition and is usually defined as the probability of an instance belonging to different known clusters. Some methods in this category are: RHWL, IU-Red, ALCSSC, CMKIPCM, AAA, URASC, A-COBS, ADP and ADPE. There are as well other criteria to select pairs of instances to query, such as the utility maximization (SRBR), the maximum expected error reduction (ASC), the partition change maximization (Active-HACC), the ensemble consensus (PT), or the classic informativeness and coherence (A-ITML-K-Means). Cluster-related criteria can also be used to select queries, such as the size and distance between the clusters (CAC, COBRA) or how well defined the frontiers are between them (AFCC, CVQE+). Paradigm-specific criteria are used by some methods, such as CMVNMF, which performs multi-view clustering and selects pairs of instances to query, based on intra-view and inter-view criteria, or AC-CF-tree and COBRAS, which use queries to determine the best cluster merge to perform in hierarchical CC. Similarly, the family of active FIECE-EM use concepts related to the population of individuals it maintains to select the best instances to query.

7.4 Neural network-based CC

Neural Networks (NN) are universal approximators which have been applied in many machine learning tasks, and CC is not an exception. Neural Network-based CC (NNbCC) tackles the CC from the NN perspective in three different ways: through self organizing maps, through deep-embedded clustering and through classic neural networks architectures.

7.4.1 Self organizing maps-based CC

Self organizing maps are NN (usually with fixed topology) whose neurons modify their position in the solution space to organize themselves according to the shape of the clusters. The result is a net whose neurons are grouped in clusters, which can be used to determine the cluster every instance belongs to. Constraints can be included into this process in different ways, as Table 13 shows. Some on them, like SS-FKCN, simply use a penalty term accounting for violated constraints in a classic SOM variant. Others such as PrTM and SSGSOM, reformulate the classic SOM problem and use multiple neuron layers, forcing instances to flow through these to layers to be ultimately assigned to the appropriate cluster. In order to do so, PrTM uses constraint-influenced probabilities to decide how the position of the neurons changes, while SSGSOM adjusts the between-layer weights and the num-

Table 11 Feature table for the ACC - Active Constraint Acquisition methods

<i>S_A</i>	Acronym	ID	CC Methods	Hybrid	Year	Refs.
15.98	FFQS	10	–	–	2004	Basu et al. (2004a)
12.72	ACCESS	16	–	Non Graph-based	2005	Xu et al. (2005)
6.99	DGPC	289	PCK-Means	–	2007	Greene and Cunningham (2007)
13.05	MMFFQS	69	MPCK-Means	–	2008	Mallapragada et al. (2008)
9.28	ASC(10)	106	AHCC, MPCK-Means	–	2010	Vu et al. (2010, 2012) Vu et al. (2010)
7.24	KAKB	290	S3OM	–	2011	Allab and Benabdeslem (2011)
4.63	SSL-EC	131	–	Constraint Propagation	2012	Wang et al. (2012)
5.71	Cons-DBSCAN	137	–	Hierarchical CC & Density-based CC	2012	Zhao et al. (2012)
8.62	JDFD	292	–	–	2013	Duarte et al. (2013)
8.51	SACS	186	Xiang's, RCA, MPCK-Means	–	2014	Abin and Beigy (2014)
6.34	WAKL	293	MPCK-Means	–	2014	Atwa and Li (2014)
4.27	MICS	294	RCA+K-Means	–	2015	Yang et al. (2015)
11.85	CCCPYL	203	DBSCAN, COP-K-Means, KKM, CSI, MPCK-Means	–	2015	Chang and Lin (2015)
6.55	LCML	300	MPCK-Means	–	2016	Cai et al. (2016)
7.12	AAA(18)	296	MPCK-Means, RCA	–	2018	Abin (2016)
5.82	RWACS	297	MPCK-Means, RCA	–	2018	Abin (2017)
9.91	AAA(19)	298	MPCK-Means, RCA	–	2019	Abin (2019)
18.26	AIPC	301	PCK-Means	–	2019	Zhong et al. (2019)
5.65	AAVV	302	MPCK-Means, RCA	–	2020	Abin and Vu (2020)
7.36	ALPCS	299	–	–	2020	Chen and Jin (2020)
3.93	ASCENT	307	MPCK-Means	–	2020	Li et al. (2019)

ber of nodes of the first layer to dynamically correct the violation of constraints. Simpler methods like the S3OM, modify classic SOM for it to carry out only assignments without violating any constraints, similarly to COP-K-Means.

The main difference between SOM-based approaches to CC and the other two approaches (deep embedded clustering and classic neural networks) is that the primary goal of the former is to produce a partition of the dataset, while the latter's is to cast predictions over unseen instances regarding the cluster they belong to. Please note that a partition can be obtained with deep embedded clustering and classic neural networks by feeding the training instances to the trained model.

7.4.2 Deep embedded clustering-based CC

In deep embedded clustering-based CC constraints are included into the classic Deep Embedded Clustering (DEC) model. Table 14 gathers methods which use this approach. SDEC includes constraints into the classic DEC model by using constraints to influence its distance learning step, DCC does so by simply modifying the loss function of DEC with a penalty term. CDEC uses DEC to initialize its encoder, which is finally retrained to finally assign instances to clusters and satisfy the constraints.

7.4.3 Classic neural network-based CC

Lastly, classic neural network-based CC methods are presented in Table 15. Some of them, such as S^3C^2 and CDC, use the siamese neural networks, as they are known, to solve the CC problems in two steps. In the case of S^3C^2 , the siamese neural network is used to solve the two steps in which the CC is decomposed into simpler binary problems, while CDC uses the siamese neural network to perform unsupervised clustering and a triple NN to perform CC. NN-EVCLUS simply implements EVCLUS in an NN setup and uses a penalty term in its loss function to include constraints.

SNNs consist of a NN model designed to learn non-linear similarity measures from pairwise constraints and to generalize the learned criterion to new data pairs. A SNN is a feed-forward multi-layer perceptron whose learning set is defined as triplets composed of two instances and the constraint set between them, using 1 for ML and 0 for CL. In other words, ML instances have an associated target equal to 1, while CL instances have an associated target equal to 0. From the architectural point of view, the SNN has an input layer which accepts pairs of instances, a single hidden layer which contains an even number of units, and an output neuron with sigmoidal activation. The training of the SNN can be performed using the standard backpropagation scheme. Since the metric learned by an SNN cannot be straightforwardly used by a K-Means style algorithm, as the centroids do not necessarily have to be found in the dataset, centroids computation can be embedded in the SNN by using a classic K-Means minimization scheme based on backpropagation. This scheme keeps the weights and biases of the trained SNN fixed and varies the centroid coordinates (seen as free parameters). This is equivalent to redefining the original SNN model by adding a new layer to the network structure whose neuron activation functions correspond to the identity mapping.

Table 12 Feature table for the ACC - Active Clustering with Constraints methods

<i>S_A</i>	Acronym	ID	Penalty	ML	CL	Hybrid	Year	Refs.
7.13	RHWL	43	×	Soft	Soft	Probabilistic Clustering	2007	Huang and Lam (2007)
7.62	PT	56	×	Soft	Soft	Constrained Pool Generation	2008	Domeniconi et al. (2008)
9.44	AFCC	59	×	Soft	Soft	Fuzzy CC	2008	Grira et al. (2008)
13.45	ASC	110	–	Any	Any	Non Graph-based	2010	Wang and Davidson (2010)
4.12	CAC1	113	–	Hard	Soft	Hierarchical CC	2011	Biswas and Jacobs (2011)
3.90	IU-Red	144	×	Soft	Soft	Non Graph-based	2012	Wauthier et al. (2012)
7.86	SRBR	155	×	Soft	Soft	–	2013	Raj and Ravindran (2013)
5.62	A-ITML-K-Means	156	×	Soft	Soft	Constraint Propagation	2013	Rao and Jawahar (2013)
8.93	ALCSCC	180	×	Soft	Soft	–	2014	Xiong et al. (2013)
2.51	AC-CF-tree	185	✓	Soft	Soft	Penalty-based Methods	2014	Lai et al. (2014)
3.54	Active-HACC	189	×	Soft	Soft	Hierarchical CC	2014	Biswas and Jacobs (2014)
16.73	CMKIPCM	202	×	Soft	Soft	Fuzzy CC	2015	Abin and Beigy (2015)
21.19	AAA	221	×	Soft	Soft	Fuzzy CC	2016	Abin (2016)
4.15	COBRA	225	×	Soft	Soft	Hierarchical CC	2017	Van Craenendonek et al. (2017)
6.56	URASC	227	×	Soft	Soft	Non Graph-based	2017	Xiong et al. (2016)
8.76	A-COBS	241	×	Soft	Soft	Constrained Consensus	2017	Van Craenendonek and Blockeel (2017)
2.47	CVQE+	242	×	Soft	Soft	Penalty-based Methods	2018	Mai et al. (2018)
19.51	COBRAS	244	×	Soft	Soft	Hierarchical CC	2018	Craenendonek et al. (2018)
2.85	CMVNMF	254	×	Soft	Soft	Non-negative Matrix Factorization CC & Inter-View Constrained	2018	Zong et al. (2018); Zhang et al. (2015)
15.56	FIECE-EM+BFCU	303	×	Hard	Hard	Genetic Algorithm & Mixture Model-based CC	2020	Fernandes et al. (2019, 2020)
15.56	FIECE-EM+FCU	304	×	Hard	Hard	Genetic Algorithm & Mixture Model-based CC	2020	Fernandes et al. (2019, 2020)
15.56	FIECE-EM+DVO	305	×	Hard	Hard	Genetic Algorithm & Mixture Model-based CC	2020	Fernandes et al. (2019, 2020)
15.56	FIECE-EM+LUC	306	×	Hard	Hard	Genetic Algorithm & Mixture Model-based CC	2020	Fernandes et al. (2019, 2020)

Table 12 (continued)

S_A	Acronym	ID	Penalty	ML	CL	Hybrid	Year	Refs.
17.22	ADPE	281	×	Soft	Soft	Density-based CC & Constrained Pool Generation	2021	Shi et al. (2020)
17.84	ADP	282	×	Soft	Soft	Density-based CC	2021	Shi et al. (2020)
9.09	ACDEC	315	×	Soft	Soft	Deep Embedded Clustering-based CC	2023	Hazratgholizadeh et al. (2023)

Table 13 Feature table for the NNbCC - Self Organizing Maps-based CC methods

S_A	Acronym	ID	Penalty	ML	CL	Hybrid	Year	Refs.
5.74	SS-FKCN	36	✓	Soft	Soft	–	2006	Maraziotis et al. (2006)
10.13	PrTM	77	×	Soft	Soft	–	2009	Benabdeslem and Snoussi (2009)
7.70	S3OM	111	×	Hard	Hard	–	2011	Allab and Benabdeslem (2011)
10.74	CS2GS	201	×	Soft	Soft	Online CC	2015	Allahyar et al. (2015)

Table 14 Feature table for the NNbCC - Deep Embedded Clustering-based CC methods

S_A	Acronym	ID	Penalty	ML	CL	Hybrid	Year	Refs.
10.03	SDEC	260	×	Soft	Soft	–	2019	Ren et al. (2019)
12.74	DCC	265	×	Soft	Soft	–	2020	Zhang et al. (2019, 2021)
2.44	CDEC	275	×	Soft	Soft	–	2021	Amirizadeh and Boostani (2021)
6.10	conDetSEC	310	×	Soft	Soft	Time Series	2023	Ienco and Interdonato (2023)
9.09	ACDEC	315	×	Soft	Soft	Active Clustering with Constraints	2023	Hazratgholizadeh et al. (2023)

Table 15 Feature table for the NNbCC - Classic Neural Network-based CC methods

S_A	Acronym	ID	Penalty	ML	CL	Hybrid	Year	Refs.
7.01	SNN	136	×	Soft	Soft	Constrained Distance Transformation	2012	Maggini et al. (2012)
15.54	S^3C^2	268	×	Soft	Soft	–	2020	Śmieja et al. (2020)
10.09	CDC	274	×	Soft	Soft	–	2021	Cui et al. (2021)
3.56	NN-EVCLUS	284	✓	Soft	Soft	–	2021	Denoeux (2021)

7.5 Ensemble CC

Ensemble clustering methods usually perform clustering in two steps. (1) generating a pool of solutions, whose diversity depends on the method (or methods) used for the generation. (2) taking the pool of solutions as input and producing a single final solution by merging or selecting solutions from the pool. The function in charge of this procedure is called the consensus function. The application of ensemble-based clustering methods on the constrained clustering problem gives place to a new distinction within this category, which classifies Ensemble CC (ECC) methods depending on the step (or steps) in which they consider constraints. Sections 7.5.1 and 7.5.1 present the details on the variants of these concepts.

7.5.1 Constrained pool generation

These ensemble methods use constraints in the pool generation step (the first step), i.e.: the partitions in the pool of solutions are generated with CC methods. Table 16 presents the list of methods belonging to this category. The consensus functions used by these methods do not take constraints into account. Therefore they are not considered in any distinction made within this category. The most commonly used consensus functions are majority voting, NCuts and CSPA.

Table 16 Feature table for the ECC - Constrained Pool Generation methods

<i>S_A</i>	Acronym	ID	Penalty	ML	CL	Hybrid	Year	Refs.
5.33	SMCE	37	×	Soft	Soft	–	2006	Yan and Domeniconi (2006)
7.62	PT	56	×	Soft	Soft	Active Clustering with Constraints	2008	Domeniconi et al. (2008)
10.81	Samarah	99	×	Soft	Soft	–	2010	Forestier et al. (2010)
0.00	SCEV	146	×	Soft	Soft	–	2012	Iqbal et al. (2012)
11.03	MVSCE	164	×	Soft	Soft	–	2013	Chen et al. (2013)
1.19	E ² CPE	170	×	Soft	Soft	–	2013	Lu and Peng (2013)
5.48	SCSC	206	×	Soft	Soft	Online CC	2015	Yi et al. (2015)
12.47	HSCE	207	×	Soft	Soft	Non Graph-based & Penalty-based Methods	2015	Wei et al. (2015, 2018)
22.93	ISSCE	214	×	Soft	Soft	Constraint Propagation	2016	Yu et al. (2015)
20.25	RSSCE	215	×	Soft	Soft	Constraint Propagation	2016	Yu et al. (2015)
6.09	FQH	232	×	Soft	Soft	–	2017	Yang et al. (2017)
7.83	CESCP	252	×	Soft	Soft	Constraint Propagation	2018	Yu et al. (2018)
10.51	DCECP	253	×	Soft	Soft	Constraint Propagation	2018	Yu et al. (2018)
17.22	ADPE	281	×	Soft	Soft	Density-based CC & Active Clustering with Constraints	2021	Shi et al. (2020)

Many methods use the subspace technique, which consists of performing clustering in a new space with a lower number of dimensions than the original space. The technique used to produce different subspaces introduces variability on the pool of solutions. The most common procedure used to generate the subspaces is simply a random sampling of the original features, such as in SCSC, ISSCE, RSSCE, CESC, DCECP or ADPE. However, there are subspace generation methods specifically designed for certain algorithms. An example of this is SMCE, which uses the CSI method to project instances and constraints into multiple low-dimensional subspaces and then learning positive semi-definite matrices therein.

Other methods simply use any previous CC algorithm to produce the pool. The most common way to introduce diversity in the pool is by applying different CC methods to produce different partitions. Methods that use this strategy are SCEV, MVSCE, E²CPE, HSCE or FQH. Another (and less used) method to generate diversity is varying the hyperparameters of a single CC method, as in Samarah.

7.5.2 Constrained consensus

In constrained consensus ensemble methods, constraints are used only in the consensus function to produce a final partition meeting as much constraints as possible. Table 17 gathers the four methods which belong to this category. All of these methods generate the partitions in the pool by means of classic clustering algorithms. This is why the consensus functions used by these methods usually measure the quality of the generated solutions with respect to the constraints by means of a quality index and select the best ones to be finally merged. For example COBS and A-COBS use the infeasibility to select the best partition in the pool, which is generated by any classic clustering method. WECR K-Means runs classic K-Means multiple times with different hyperparameters to generate the pool, then a weighting procedure is used to automatically assign a weight to every partition depending on their local and global quality, which includes the infeasibility. A weighted co-association matrix based consensus approach is then applied to achieve a final partition. Semi-MultiCons builds a tree-like pool or partitions and then applies a normalized score which measures constraint satisfaction if any given merge or split operation between clusters is performed in the tree.

7.5.3 Full constrained

These methods (in Table 18) include constraints in both the pool generation and the consensus steps. They make use of the formulas described before and combine them. On the one hand, SFS³EC, ARSCE and RSEMICE make use of the subspace technique, although they differ in the consensus function. SFS³EC merges partitions in the pool by building a hypergraph which takes partitions and constraints into account and running METIS over this graph to get the final partition. ARSCE computes the affinity graph for every solution in the pool, and uses regularized ensemble diffusion to fuse the similarity information. Finally, RSEMICE assigns a confidence factor to each solution in the pool to build a consensus matrix, which can be interpreted as a graph over which the NCut algorithm is applied (used as the consensus function). On the other hand, COP-SOM-E and Cop-EAC-SL use previous CC methods (ICOP-K-Means and COP-K-Means, respectively) to generate their pool. COP-SOM-E uses a hard constrained version of SOM as the consensus matrix, and Cop-EAC-SL runs the constrained single-link algorithm over a co-association matrix

Table 17 Feature table for the ECC - Constrained Consensus methods

S_A	Acronym	ID	Penalty	ML	CL	Hybrid	Year	Refs.
6.95	COBS	240	×	Soft	Soft	–	2017	Van Craenendonck and Blockeel (2017)
8.76	A-COBS	241	×	Soft	Soft	Active Clustering with Constraints	2017	Van Craenendonck and Blockeel (2017)
6.68	WECR K-Means	278	×	Soft	Soft	–	2021	Lai et al. (2019)
5.39	Semi-MultiCons	288	×	Soft	Soft	–	2022	Yang et al. (2022)

Table 18 Feature table for the ECC - Full Constrained methods

S_A	Acronym	ID	Penalty	ML	CL	Hybrid	Year	Refs.
6.34	Cop-EAC-SL	75	×	Soft	Soft	–	2009	Abdala and Jiang (2009)
3.00	En-Ant	133	×	Soft	Soft	Swarm Optimization	2012	Yang et al. (2012)
7.54	COP-SOM-E	143	×	Hard	Hard	–	2012	Yang et al. (2012)
10.72	RSEMICE	237	✓	Soft	Soft	–	2017	Yu et al. (2017)
15.04	SFS ³ EC	257	×	Soft	Soft	Non Graph-based	2019	Tian et al. (2019)
6.86	ARSCE	263	×	Soft	Soft	–	2020	Luo et al. (2019)

which counts how many times pairs of instances are placed in the same cluster in different partitions.

7.6 Metaheuristics-based CC

Metaheuristics-based CC (MbCC) use metaheuristic algorithms to approach the CC problem. Many distinctions can be made within the metaheuristic algorithms field, in this study the trajectory-based methods versus population-based methods is used to produce to subcategories of CC approaches, as it is the one which results in the more consistent dichotomy.

7.6.1 Population-based

A plethora of metaheuristic methods has been applied to the CC problem. Particularly, population-based methods have shown remarkable success, with evolutive algorithms being the most used ones. A further distinction can be made within these methods: swarm optimization algorithms and genetic algorithms. In swarm optimization algorithms, a population of individuals is used to mimic the behavior of a colony of insects in its natural environment, while in genetic algorithms, the population is evolved according to the rules of natural selection, expecting them to generate the best possible individual (solution).

Swarm Optimization

Table 19 gathers swarm optimization algorithms which tackle the CC problem. All of them are based on ant colonies behavior, with the main differences found in the scheme used to include constraints. MCLA, MELA and CELA are all based in the Leader Ant algorithm and use the same integration scheme. They modify the ant-nest assignment rule so that only feasible assignments are taken into account. MCLA and MELA ensure that they do not violate any ML constraints by using chunklets. They only differ in the type of constraints they can handle. CAC is based on the RWAC algorithm, which tries to simulate the behav-

Table 19 Feature table for the MbCC - Population-based - Swarm Optimization methods

S_A	Acronym	ID	Penalty	ML	CL	Hybrid	Year	Refs.
6.26	MCLA	91	×	Hard	Hard	—	2009	Vu et al. (2009)
12.66	MELA	92	×	Hard	—	—	2009	Vu et al. (2009)
12.66	CELA	93	×	—	Hard	—	2009	Vu et al. (2009)
7.71	CAC	132	×	Soft	Soft	—	2012	Xu et al. (2011, 2013)
3.00	En-Ant	133	×	Soft	Soft	Full Constrained	2012	Yang et al. (2012)

ior of ants in their environment trying to find a place to sleep. Constraints are included in this scheme by modifying attractive and repulsive forces between ants associated to constrained instances. The En-Ant algorithm uses three instances of the Semi-Ant algorithm as the partition generation algorithm in an ensemble setup. Constraints are used both in the generation of the partition pool and in the consensus function. Please note how none of these algorithms modify the fitness function of the ant colony algorithm on which they are based, instead they include constraints by modifying other aspects of the algorithms.

Genetic Algorithm

Genetic strategies used to tackle the CC problems are presented in Table 20. A low-level dichotomy can be made within these methods, they can be either single-objective or multi-objective genetic algorithms. On the one hand, multi-objective algorithms optimize a set of fitness functions all at the same time. Constraints can be naturally included in this paradigm by simply adding the infeasibility as one of the functions to be optimized. This is the case for MOCK (which includes constraints into the classic PESA-II algorithm), PCS (which is based on NSGAII) and ME-MOEA/D_{CC} (which modifies classic MOEA/D). These three proposals also modify a basic aspect of the algorithm they are based on for it to fit better to the CC problem. MOCK implements a constraint-oriented initialization scheme. PCS features an improving procedure applied to the population after the classic operators have been applied. Lastly, ME-MOEA/D_{CC} uses memetic elitism with controlled feedback. On the other hand, single objective genetic algorithms usually optimize a combination of any classic clustering related measure and the infeasibility included as a penalty term. Methods such as COP-HGA, BRKGA+LS or SHADE_{CC} use this strategy. Other proposals, such as Cop-CGA and FIECE-EM, evolve separate populations or subpopulations which have individuals with different solutions qualities and make them interact to generate new individuals. FIECE-EM+BFCU, FIECE-EM+FCU, FIECE-EM+DVO and FIECE-EM+LUC are all active variants of FIECE-EM.

7.6.2 Single individual

Single individual methods focus on modifying and improving a single candidate solution. They start with a single individual which is improved with respect to the fitness function. Simulated annealing, local search, iterated local search or guided local search are examples of single solution metaheuristics. Table 21 lists methods which belong to this category. CCLS and DILS_{CC} use both variants of the classic LS algorithm to find solutions for the CC problem. They use a combination of the intra-cluster mean distance and the infeasibility to build their fitness function. The SemiSync algorithm is a nature-inspired non-evolutionary based on regarding instances as a set of constrained phase oscillators, whose dynamics can be simulated to build a partition. The local interaction of every oscillator with respect to

Table 20 Feature table for the MbCC - Population-based - Genetic Algorithm methods

<i>S_A</i>	Acronym	ID	Penalty	ML	CL	Hybrid	Year	Refs.
15.90	MOCK	30	×	Soft	Soft	–	2006	Handl and Knowles (2006)
9.10	COP-CGA	62	×	Soft	Soft	–	2008	Hong et al. (2008)
10.76	COP-HGA	65	✓	Soft	Soft	–	2008	Hong et al. (2008)
4.31	PSC	126	×	Soft	Soft	–	2012	Ebrahimi and Abadeh (2012)
4.38	CEAC	212	×	Soft	Soft	–	2016	He (2016)
14.72	BRKGA+LS	235	✓	Soft	Soft	–	2017	Oliveira et al. (2017)
13.97	FIECE-EM	247	×	Hard	Hard	Mixture Model-based CC	2018	Covões and Hruschka (2018)
15.56	FIECE-EM+BFCU	303	×	Hard	Hard	Active Clustering with Constraints & Mixture Model-based CC	2020	Fernandes et al. (2019, 2020)
15.56	FIECE-EM+FCU	304	×	Hard	Hard	Active Clustering with Constraints & Mixture Model-based CC	2020	Fernandes et al. (2019, 2020)
15.56	FIECE-EM+DVO	305	×	Hard	Hard	Active Clustering with Constraints & Mixture Model-based CC	2020	Fernandes et al. (2019, 2020)
15.56	FIECE-EM+LUC	306	×	Hard	Hard	Active Clustering with Constraints & Mixture Model-based CC	2020	Fernandes et al. (2019, 2020)
29.08	SHADE _{CC}	285	✓	Soft	Soft	–	2021	González-Almagro et al. (2021)
28.22	ME-MOEA/D _{CC}	286	✓	Soft	Soft	–	2021	González-Almagro et al. (2021)

its neighborhood can be computed over time. Therefore similar instances will synchronize together in groups that can be interpreted as clusters. ML and CL are included by introducing an additional global interaction term.

7.7 Multi-view CC

In many applications, data is collected from different sources in diverse domains, usually involving multiple feature collectors. This data refers to the same reality, although it exhibits heterogeneous properties, which translates into every instance being described by different sets of features. These are the called views, and the problem of performing clustering over instances described by different sets of features is known as multi-view (or multi-source) clustering Yang and Wang (2018). In multi-view clustering, different sources of the data are used to produce a single partition. Constraints can be included into multi-view clustering in different ways and levels, giving place to Multi-View Constrained Clustering (MVCC). With respect to the level in which constraints can be used, two options are available: intra-view constraints and inter-view constraints. Intra-view constraints relate instances which belong to the same view of the data (similarly to constraints in any non MVCC algorithm), while inter-view constraints relate instances which belong to different views, hence encouraging collaboration between the clustering processes applied to them.

7.7.1 Intra-view constrained

Intra-view CC usually performs clustering separately in each view and then tries to find a consensus between the obtained partitions (similarly to what ensemble clustering does with the consensus function). Methods which belong to this category are gathered in Table 22. SMVC models clustering views via multivariate Bayesian mixture distributions located in subspace projections. It includes constraints in the Bayesian learning processes. TVClust and RDPM are both very particular methods, as they view the dataset and the constraint set as different sources of information for the same data, thus performing multi-view clustering with only two views. The dataset is modeled by a Dirichlet Process Mixture model and the constraint set is modeled by a random graph. They aggregate information from the two views through a Bayesian framework and they reach a consensus about the cluster structure through a Gibbs sampler. MVMC independently builds a pairwise similarity matrix for every view and casts the clustering task into a matrix completion problem based on the constraints and the feature information from multiple views. The final pairwise similarity matrix is built iteratively by approaching the independent pairwise similarity matrices in different views to each other. The final partition is obtained by performing spectral clustering of the final similarity matrix. SSCARD is based on classic CARD, which is able to combine multiple weighted sources of relational information (some may be more relevant than others) to produce a partition of the dataset. SSCARD simply includes the PCCA penalty term (see Sect. 7.11) into the classic CARD objective function. Lastly, MVCC is the only intra-view CC method that performs clustering in the different views in a collaborative way. It performs constrained clustering in each view separately, inferring new constraints and transferring them between views using partial mapping. For constraint inference and transfer, a variant of the co-EM algorithm Nigam and Ghani (2000) is used, which is an iterative EM based algorithm that learns a model from multiple views of the data.

Table 21 Feature table for the MbCC - Single individual methods

S_A	Acronym	ID	Penalty	ML	CL	Hybrid	Year	Refs.
3.56	PAST-Toss	54	×	Soft	Soft	Graph-based	2008	Coleman et al. (2008)
3.99	CCLS	223	✓	Soft	Soft	—	2016	Hiep et al. (2016)
11.50	CG+PR+LS	236	×	Soft	Soft	Column Generation	2017	Oliveira et al. (2017)
14.10	SemiSync	258	×	Soft	Soft	—	2019	Zhang et al. (2019)
20.09	DILS _{CC}	270	✓	Soft	Soft	—	2020	González-Almagro et al. (2020)
15.05	FastCCP	308	×	Hard	Hard	—	2023	Gao et al. (2023)

7.7.2 Inter-view constrained

Inter-View CC methods can handle both intra-view and inter-view constraints. Methods which belong to this category are presented in Table 23. UCP uses the results of intra-view constraint propagation to adjust the similarity matrix of each view, and then performs inter-view constraint propagation with the adjusted similarity matrices. Its main drawback is that it is limited to two views. CMVNMf minimizes the loss function of NMF in each view, as well as the disagreement between each pair of views. The disagreement is defined as the difference between feature vectors associated to the same instance in the same view. It should be high if they are CL instances and low if they are ML instances. MSCP can propagate constraints across different data sources by dividing the problem into a series of two-source constraint propagation subproblems, which can be transformed into solving a Sylvester matrix equation, viewed as a generalization of the Lyapunov matrix equation. MCPCP uses a low-rank relation matrix to represent the pairwise constraints between instances from different views. Afterwards it learns the full relation matrix by using a matrix completion algorithm and derives an indicator matrix from it with an iterative optimization process.

7.8 Kernel CC

Kernel methods perform clustering by mapping the data from the original input space to a new feature space, which is usually of higher dimensionality. The key aspect of kernel-based methods is the avoidance of an explicit knowledge of the mapping function, which is achieved by computing dot products in the feature space via a kernel function. A critical aspect for kernel-based methods is the selection of the optimal kernel and its parameters. The basic classic kernel-based clustering method is the Kernel-K-Means algorithm. It performs clustering directly in the feature space by computing pairwise distances and updating centroids, using dot products and the kernel trick (Domeniconi et al. 2011). The goal of Kernel CC (KCC) is to learn a kernel function that maps ML instances close to each other in feature space, while mapping CL instances far apart, and then perform clustering in the feature space.

Table 24 shows a list of KCC methods. The first KCC method can be found in SSKK Kulis et al. (2005, 2009), which is built on the basis of the HMRF-K-Means method. Kernel CC methods are all very similar to each other, with one of the few differences being the way in which they include the kernel parameters in clustering process. Some methods, such as ASSKK or SFFA include this parameter in the optimization process. Therefore they do not need to be specified by the user, not withstanding the higher computational cost. Addition-

Table 22 Feature table for the MVCC - Intra-View Constrained methods

S_A	Acronym	ID	Penalty	ML	CL	Hybrid	Year	Refs.
7.06	SSCARD	49	✓	Soft	Soft	Fuzzy CC	2007	Frigui and Hwang (2007, 2008)
3.42	MVCC	182	×	Soft	Soft	–	2014	Eaton et al. (2014)
3.62	SMVC	195	×	Soft	Soft	–	2014	Günemann et al. (2014)
19.07	TVClust	209	×	Soft	Soft	–	2015	Khashabi et al. (2015)
19.84	RDPM	210	×	Soft	Soft	–	2015	Khashabi et al. (2015)
10.99	MVMC	226	×	Soft	Soft	Matrix Completion & Spectral CC	2017	Zhao et al. (2017)
18.17	MVCSC	261	×	Soft	Soft	Non Graph-based	2019	Chen et al. (2019)

ally, other methods use more than one kernel, such as TRAGEK and ENPAKL. They learn multiple kernels which are later combined in a single one to obtain the global kernel matrix. Some minor differences can be found in ssFS, for example, which is specifically designed to cluster sets of graphs.

7.9 Fuzzy CC

Fuzzy classic clustering represents a hard dichotomy within the clustering area. In fuzzy clustering, instances are allowed to belong to more than one cluster, with a list of probabilities that indicate the likelihood of said instance to belong to every cluster. The set of these lists is called a fuzzy partition, in contrast to crisp (or hard) partitions obtained by non-fuzzy clustering algorithms, where instances are assumed to belong to a single cluster with a 100% probability Baraldi and Blonda (1999). Fuzzy clustering algorithms are usually applied over relational information, meaning that the feature vector describing the instances in the dataset are not needed, only the pairwise relations between them, which can be computed as pairwise distances Ruspini et al. (2019). This makes the fuzzy clustering paradigm specially suitable to be extended in order to include constraints, as constraints are a natural type of relational information.

A list of Fuzzy CC (FCC) methods is proposed in Table 25. Even if this is one of the largest categories in clustering-based CC methods, authors have found that no further significant categorizations can be performed over it. The vast majority of methods in this category include constraints by means of a penalty term, which can be computed with different confidence degrees (Maraziotis 2012). The confidence degree of the penalty refers to the number of possible assignments in which constraints violations are checked. Some methods examine all possible assignments for all constraints, such as SSCARD or AFCC, while others only examine the most probable assignment, like SSFCA, which is equivalent to perform violations checks in the crisp partition. Many methods are based on the PCCA method (see Sect. 7.11), such as AFCC, ACC, SS-CARD, PCsFCM and SS-CLAMP. These methods modify the objective function of PCCA to make it fuzzy or borrow its objective function to use it in a new optimization scheme. Besides, it is worth noting that some fuzzy methods are not relational but evidential. The evidential clustering framework is built around the concept of credal partition, which extends the concepts of crisp and fuzzy partitions and makes it possible to represent not only uncertainty, but also imprecision with respect to the class membership of an instance. Methods such as CECM, CEVCLUS and k-CEVCLUS include constraints into the evidential clustering paradigm, with CEVCLUS and k-CEVCLUS combining both relational and evidential fuzzy clustering features.

7.10 Mixture model-based CC

Mixture models are parametric statistical models which assume that a dataset originates from a weighted sum of several statistical sources. These sources can typically be Gaussian distributions, originating the Gaussian Mixture Model (GMM). However, other statistical distributions (like the Dirichlet distribution) can be used in the mixture models paradigm. In GMM-based clustering, each cluster is associated to a parameterized Gaussian distribution. These parameters are optimized for the final distribution to explain its associated cluster. The result of GMM-based clustering is not a crisp partition, but the probability for each

Table 23 Feature table for the MVCC - Inter-View Constrained methods

S_A	Acronym	ID	Penalty	ML	CL	Hybrid	Year	Refs.
5.19	UCP	167	×	Soft	Soft	Constraint Propagation	2013	Lu and Peng (2013)
4.00	MSCP	169	×	Soft	Soft	Constraint Propagation	2013	Lu and Peng (2013)
8.95	MCPCP	198	×	Soft	Soft	Matrix Completion	2015	Yang et al. (2014)
2.85	CMV/NMF	254	×	Soft	Soft	Non-negative Matrix Factorization CC & Active Clustering with Constraints	2018	Zong et al. (2018); Zhang et al. (2015)

instance to be generated by each of the available optimized Gaussian distributions. EM schemes are one of the most widely used methods to optimize the parameters of the distribution (Dempster et al. 1977). Table 26 gathers Mixture Model-based CC (MMbCC) methods. The most common way to include constraints in GMM-based clustering is by discarding unwanted distributions. For the case of hard CC, this is done by removing all distributions which violate any constraints from the addition of Gaussians, as in Constrained EM or DPMM. In the case of soft CC, the summation of distributions is modified to take these distributions into account to a higher or lower extent depending on the number of violated constraints and on the relevance of constraints themselves. This can be achieved by means of a penalty-style objective function, as in sRLe-GDM-FFS, MCGMM, SCGMM, or by assigning a level of confidence to each constraint, as in MAA or PPC. Some methods use different statistical models (not Gaussian), like DPMM, which uses Dirichlet Processes, or they allow a single cluster to be represented by more than one distribution, as in MCGMM. SPGP is not based on Gaussian mixture models but on Gaussian process classifiers (GPC). Given its similitude with GMM, authors have decided to include it in this category. The main difference between these two paradigms is that GMM are generative models, while GPC are discriminative models.

7.11 Hierarchical CC

Details in classic hierarchical clustering and hierarchical CC have been already introduced in Sect. 3.1. Let us remember that hierarchical clustering methods produce a dendrogram, instead of a partition. Affinity criteria are used to determine cluster merges in every lever of the dendrogram.

Table 27 presents a list of Hierarchical CC (HCC) methods. Many strategies designed to include constraints into hierarchical clustering can be found in this category. Some on them modify the clustering engine of existing methods to include constraints in the process of selecting the clusters to merge, such as COP-COBWEB, C-DBSCAN, CAC1, Cons-DBSCAN or SDHCC. Others transform the dataset in some way for it to include the information contained in the constraint set, such as COBRA, COBRAS, C-DenStream. AHC-CTP includes constraints in the computation of the dissimilarities without using a penalty term, while methods such as 2SHACC and PCCA have to use one. The only divisive hierarchical CC method found by the authors is SDHCC, all of the rest perform hierarchical agglomerative CC.

7.12 Density-based CC

In density-based classic clustering, a cluster is considered to be a set of instances spread in the data space over a contiguous region with high density of instances. Density-based methods separate clusters by identifying regions in the input space with low density of instances, which are usually considered as noise or outliers Kriegel et al. (2011).

Table 28 presents a list of Density-based CC (DbCC) methods. Two main strategies are used to include constraints into density-based clustering methods. The first one consists of modifying the assignation rule for instances to cluster, taking constraints into account, similarly to the way in which it is done in cluster engine-adapting methods. This strategy is used in methods like C-DBSCAN, C-DenStream, Cons-DBSCAN, SDenPeak or SSDC. On

Table 24 Feature table for KCC methods

S_A	Acronym	ID	Penalty	ML	CL	Hybrid	Year	Refs.
20.60	SSKK	23	×	Soft	Soft	—	2005	Kulis et al. (2005, 2009)
6.24	ASSKK	27	×	Soft	Soft	—	2006	Yan and Domeniconi (2006)
9.90	BoostCluster	40	×	Soft	Soft	—	2007	Liu et al. (2007)
15.70	CCSKL	89	×	Soft	Soft	Non Graph-based	2009	Li and Liu (2009)
2.37	SFFA	125	×	Soft	Soft	—	2012	Almeida et al. (2012)
6.97	TRAGEK	141	×	Soft	Soft	—	2012	Chen et al. (2012)
6.97	ENPAKL	142	×	Soft	Soft	—	2012	Chen et al. (2012)
4.15	ssFS	291	✓	Soft	Soft	—	2012	Huang et al. (2012)
5.82	SSSRM	157	×	Hard	Hard	—	2013	Tang and Miyamoto (2013)
5.82	SSSeKRM	158	×	Hard	Hard	—	2013	Tang and Miyamoto (2013)
12.08	SKML	194	×	Soft	Soft	—	2014	Baghshah et al. (2014)

the other hand, some methods use constraints to redefine the density computation method to take them into account, such as in SemiDen or YZWD. In addition, there are methods that simply use constraints to modify the similarity measure or the dataset on the basis of the constraints, which run classic density-based clustering algorithm over them afterwards, such as SSDPC or fssDBSCAN.

7.13 Online CC

Online clustering methods perform clustering over data which varies over time. This is called a data stream. In classic online clustering, new data instances arrive (in the form of chunks or single instances) over time and the goal of the clustering algorithm is to produce a partition of the current set of instances, usually taking into account information obtained from past instances. In online constrained clustering, not only new instances are provided to the method over time, but also constraints and, in some cases, only constraints. Methods performing Online CC (OCC) are gathered in Table 29.

The CME algorithm is an online wrapper for the COP-K-Means algorithms. It gradually forgets past constraints, lowering their effect on the current partition as new data from the data stream arrive. TDCK-Means is built on the basis of classic K-Means and also uses a decay term to handle online constraints. It addresses the temporal nature of the data by adapting the Euclidean distance to take into account both the distance in the multidimensional space and in the temporal space. A penalty term is then added to include constraints, which is more severe for instances closer in time and whose magnitude decays over time. CS2GS also uses constraint weight decay to perform online CC and is based on SOM. The architecture of the neural network used by CS2GS features two layers. The between-layer weights and the number of nodes of the first layer are adapted to dynamically correct the violation of constraints. Using the metrics included in these layers as a reference, the violation of constraints is quantified as network's error. Weights are modified over time based on the error to obtain the new weights. The weight update procedure tries to satisfy the currently violated constraints while keeping the new weights close to the old ones to avoid breaking the old constraints.

O-LCVQE and C-RPCL are both online competitive learners. Competitive learning algorithms are characterized by competition among k neurons, which compete to learn instances. This is known as the winner-take-all (WTA) approach. Competitive learning can be seen as

Table 25 Feature table for FCC methods

S_A	Acronym	ID	Penalty	ML	CL	Hybrid	Year	Refs.
7.06	SSCARD	49	✓	Soft	Soft	Intra-View Constrained	2007	Frigui and Hwang (2007, 2008)
9.44	AFCC	59	×	Soft	Soft	Active Clustering with Constraints	2008	Griira et al. (2008)
0.00	ACC	60	✓	Soft	Soft	—	2008	Frigui and Meredith (2008)
0.00	PCsFCM	78	×	Soft	Soft	—	2009	Kanzawa et al. (2009)
0.00	PCeFCM	79	×	Soft	Soft	—	2009	Kanzawa et al. (2009)
6.17	SCAP	124	✓	Soft	Soft	—	2011	Gao and Wu (2011)
4.82	SSFCA	134	×	Soft	Soft	—	2012	Maraziotis (2012)
7.31	CECM	139	×	Soft	Soft	—	2012	Antoine et al. (2012)
6.41	SS-CLAMP	168	✓	Soft	Soft	—	2013	Melo and Carvalho (2013)
12.45	SS-FCC	163	✓	Soft	Soft	Co-Clustering	2013	Yan et al. (2013)
13.78	SSeFCMCT	162	×	Soft	Soft	—	2013	Hamasuna and Endo (2013)
13.78	SSsFCMCT	161	×	Soft	Soft	—	2013	Hamasuna and Endo (2013)
17.64	PC-eFCM-NM	177	✓	Soft	Soft	—	2014	Endo et al. (2014)
17.64	PC-sFCM-NM	178	✓	Soft	Soft	—	2014	Endo et al. (2014)
5.72	CEVCLUS	192	×	Soft	Soft	—	2014	Antoine et al. (2014)
16.73	CMKPCM	202	×	Soft	Soft	Active Clustering with Constraints	2015	Abin and Beigy (2015)
6.38	SFFD	295	✓	Soft	Soft	—	2015	Li et al. (2015)
21.19	AAA	221	×	Soft	Soft	Active Clustering with Constraints	2016	Abin (2016)
6.84	k-CEVCLUS	245	×	Soft	Soft	—	2018	Li et al. (2018)

performing cluster in the input space, viewing the neurons as centroids and using the Euclidean distance as the competition score. The O-LCVQE is a WTA approach that can only deal with CL constraints by defining the score as in LCVQE. Therefore the winner centroid is computed with regards to the objective function of LCVQE modified to consider only CL constraints. Similarly, the RPCL algorithm Xu et al. (1993) can be modified to include only CL constraints within the WTA, resulting in C-RPCL. The intuition behind this modified algorithm is that, if a CL constraint gets violated by assigning the instance to a given centroid, C-RPCL searches for the nearest rival which does not cause any constraint violations. This nearest rival becomes the winner, and the previous winner prototype is moved away from that instance.

C-DenStream is based on the density-based clustering DenStream, which is the online version of DBSCAN. DenStream performs density-based clustering. it uses the micro-cluster density, which is based on weighting areas of instances in a neighborhood as a result of an exponential decay function over time. C-DenStream includes constraints into the DenStream clustering process by translating instance-level constraints into micro-cluster-level constraints using the micro-cluster membership of each instance in each timestamp. SemiStream builds an initial partition using MPCK-Means and updates it as new chunks of data arrive. The update process consist of performing clustering assigning pairs of constrained instances to clusters, minimizing the cost of said assignment. The SCSC algorithm is the only online cc algorithm that keeps the dataset constant over time and consider the time dimension only over the constraint set. It consists of two main components, an offline procedure to build a convex hull, and an online procedure to update the clustering results when new pairwise constraints are received.

7.14 Others

This section gathers minor CC categories. These categories, shown in Table 30, are considered to be minor because of the number of methods belonging to them (5 or less), or because of the restricted applicability or specificity of said methods. A total of 15 minor CC categories are briefly introduced, for a total of 40 methods.

7.14.1 Constrained co-clustering

Co-clustering methods perform clustering on the column and the rows of a given dataset at the same time, considering them as closely related different sources of information. OSS-NMF extends the classic NMF by introducing constraints and performing clustering by solving a constrained optimization problem. This method is specifically proposed to solve the document clustering task, and does so by performing co-clustering in words and documents simultaneously, and considering both word-level and document-level constraints. RJFM is based on the meta-algorithm called Bregman Co-clustering, which can optimize a large class of objective functions belonging to the Bregman divergences. Its principle is simple: it alternatively refines row and column clusters, while optimizing an objective function that takes both partitions into account. It includes constraints in both columns and rows clustering the same way: it never performs rows or columns assignation breaking CL constraints, and always assigns full cliques of ML constraints. SS-NMF learns a new metric by applying simultaneously distance metric learning and modality selection. The new met-

Table 26 Feature table for MMbCC methods

S_A	Acronym	ID	Penalty	ML	CL	Hybrid	Year	Refs.
5.45	MAA	8	×	Hybrid	Hybrid	–	2004	Law et al. (2004)
17.53	Constrained EM	14	×	Hard	Hard	–	2004	Shental et al. (2004)
4.64	PPC	21	×	Hybrid	Hybrid	–	2005	Lu and Leen (2005, 2007)
20.56	MCGMM	24	×	Soft	Soft	–	2005	Zhao and Miller (2005)
20.25	SCGMM	25	×	Soft	Soft	–	2005	Zhao and Miller (2005)
9.01	SPGP	48	×	Hybrid	Hybrid	–	2007	Lu (2007)
1.45	CDPMM08	58	×	Hard	Hard	–	2008	Vlachos et al. (2008)
7.30	sRLe-GDM-FFS	147	×	Soft	Soft	–	2012	ISMAIL and Behir (2012)
4.71	C ₄ S	238	×	Soft	Soft	–	2017	Śmieja and Wiercioch (2017)
13.97	FIECE-EM	247	×	Hard	Hard	Genetic Algorithm	2018	Covões and Hruschka (2018)
8.18	JDG	249	×	Soft	Soft	–	2018	Covões and Hruschka (2018)
15.56	FIECE-EM+BFCU	303	×	Hard	Hard	Active Clustering with Constraints & Genetic Algorithm	2020	Fernandes et al. (2019, 2020)
15.56	FIECE-EM+FCU	304	×	Hard	Hard	Active Clustering with Constraints & Genetic Algorithm	2020	Fernandes et al. (2019, 2020)
15.56	FIECE-EM+DVO	305	×	Hard	Hard	Active Clustering with Constraints & Genetic Algorithm	2020	Fernandes et al. (2019, 2020)
15.56	FIECE-EM+LUC	306	×	Hard	Hard	Active Clustering with Constraints & Genetic Algorithm	2020	Fernandes et al. (2019, 2020)

ric is used to derive distance matrices over which clustering is finally performed. SS-FCC formulates the CC problem as an optimization problem with an objective function built on the basis of the competitive agglomeration cost with fuzzy terms and constraint-based penalties. It introduces cooperation into the co-clustering process by including two fuzzy memberships, one of them related to columns and other related to rows, which are expected to be highly correlated. The amount of cooperation is delivered by the degree of aggregation, which should be maximized among clusters to accomplish the clustering task.

7.14.2 Alternative clustering based on constraints

Constraints can have multiple uses, other than serving as hints for the clustering process. In alternative clustering, constraints are used to produce different partitions of a single datasets. Alternative clustering methods are not strictly CC methods, as they do not use constraints generated from a side source of information (an oracle), but from the current state of a partition in an iterative clustering process. The COALA takes a partition of a dataset as input, and aims to find a different high-quality partition using constraints. In order to make the obtained partition different from the one provided, CL constraints are created between the instances assigned to the same cluster in the original partition. COALA applies agglomerative hierarchical clustering considering two possible merges in each step, one involves the two closest instances and the other involves the two closest instances that do not violate a CL constraint. Which merge is performed depends on a parameter controlling the trade-off between quality and dissimilarity (with respect to the base partition) of the new partition. COALAcats is the categorical version of COALA. Contrary to COALA, ADFT does not take a partition of a dataset as input. The overall ADFT method can be summarized in five steps. In the first step, a classic clustering method like K-Means is applied to partition the dataset. The second step characterizes this partition by means of ML and CL constraints and obtains a new distance metric based on them (using the CSI method). In the third step, this distance metric is taken as basis to compute an alternative distance measure, by obtaining singular values decomposition of the matrix that defines the metric is obtained, and by computing the Moore-Penrose pseudo-inverse of the stretcher matrix. This effectively flips the stretching and compressing dimensions. After that, the metric matrix is recomposed, multiplying its decomposition. In the fourth step, the newly learned metric is applied on the dataset to transform it, and in the fifth step, the classic clustering algorithm run to obtain the original partition is re-run to obtain a new (and different) one.

7.14.3 CC based on clustering trees

Methods which belong to this category perform clustering by using decision trees. ClusILC uses the Top-Down Induction (TDI) approach to build the clustering tree. While most TDI approaches are based on heuristics local to the node that is being built, ClusILC employs a global heuristic which measures the quality of the entire tree and which takes all instances of the dataset into account. ClusILC's heuristic measures the average variance in the leafs of the tree (normalized by the overall dataset variance) and the proportion of overall violated constraints. ClusILC greedily searches for a tree that minimizes this heuristic using an iterative process that refines the current tree in every iteration by replacing one of its leaves with

Table 27 Feature table for HCC methods

S_A	Acronym	ID	Penalty	ML	CL	Hybrid	Year	Refs.
4.76	COP-COBWEB	1	×	Hard	Hard	–	2000	Wagstaff and Cardie (2000)
19.28	IDSSR	17	×	Hard	Hard	–	2005	Davidson and Ravi (2005, 2009)
4.41	PCCA	22	×	Soft	Soft	–	2005	Girra et al. (2005, 2006)
3.84	C-DBSCAN	39	×	Hard	Hard	Density-based CC	2007	Ruiz et al. (2007)
4.48	C-DenStream	80	×	Hard	Hard	Density-based CC & Online CC	2009	Ruiz et al. (2009)
0.00	AHC-CTP	94	×	Soft	Soft	–	2010	Hamasuna et al. (2010, 2011)
4.12	CAC1	113	×	Hard	Soft	Active Clustering with Constraints	2011	Biswas and Jacobs (2011)
5.07	SDHCC	116	×	Hard	Soft	–	2011	Xiong et al. (2011)
3.44	AHCP	117	×	Hard	Soft	–	2011	Miyamoto and Terami (2011)
3.29	SGID	121	×	Hard	Hard	SAT	2011	Gilpin and Davidson (2011)
5.71	Cons-DBSCAN	137	×	Hard	Hard	Density-based CC & Active Constraint Acquisition	2012	Zhao et al. (2012)
3.54	Active-HACC	189	×	Soft	Soft	Active Clustering with Constraints	2014	Biswas and Jacobs (2014)
4.15	COBRA	225	×	Soft	Soft	Active Clustering with Constraints	2017	Van Craenendonck et al. (2017)
19.51	COBRAS	244	×	Soft	Soft	Active Clustering with Constraints	2018	Craenendonck et al. (2018)
16.38	2SHACC	262	×	Soft	Soft	Constraint Propagation	2020	González-Almagro et al. (2020)
22.52	3SHACC	287	✓	Soft	Soft	Constrained Distance Transformation	2022	González-Almagro et al. (2021)

a subtree consisting of a new test node and two new leaves. This subtree is selected among a set of candidates generated procedurally based on the heuristic described above.

7.14.4 Maximum margin CC

Maximum Margin Clustering (MMC) uses the maximum margin principle adopted in the supervised learning paradigm. It tries to find the hyperplanes that partition the data into different clusters with the largest margins between them. CMMC includes constraints into MMC by adding concave-convex restrictions to the original MMC optimization problem. These restrictions behave as penalties for constraint violation. PMMC introduces a set of loss functions, featuring a strong penalty to partitions violating constraints while operating under the maximum margin principle at the same time. In order to do so, the classical definition of score used in MMC to determine cluster membership is modified to include high penalties for assignments violating constraints. Both CMMC and PMMC use the constrained concave-convex procedure (proposed in Zeng and Cheung (2011)) to solve the non-convex optimization problem they set to address the CC problem. TwoClaCMMC is proposed to overcome some shortcomings of the CMMC algorithm, although it is limited to two-class problems. It modifies the MMC objective function with a penalty term which accounts for constraints violations. TwoClaCMMC differs from CMMC in the formulation of the penalty term. In TwoClaCMMC, the position of the hyperplane with respect to the instances involved in a violated constraint is taken into account, weighting the cost of violating such constraints with respect to said position.

7.14.5 Feature selection

Clustering can be used to perform feature selection over high-dimensionality datasets. YLYM is a clustering method designed to perform feature selection making use of a constraint set. It is composed of three steps. The first two steps consist of the classic expectation and maximization steps from an EM optimization scheme, in which feature saliencies are computed in a completely unsupervised way. The third step is called the tuning step (T-step), which refines saliencies to minimize the feature-wise constraint violation measure, which is computed based on the Jensen-Shannon divergence. The three steps (expectation, maximization and tuning) are performed iteratively until they reach convergence. The proposed method outputs a partition of the dataset and the saliency of every feature.

7.14.6 CC through MapReduce

The MapReduce paradigm is used to address problems in the context of Big Data. The MapReduce paradigm consists of dividing the computational load associated with processing a dataset among multiple processing nodes, in order to decrease the time required to obtain results. MapReduce is based on two operations: (1) The Map operation processes inputs in the form of key-value pairs and generates intermediate key/value pairs received by the Reduce operation (2) The Reduce operation processes all intermediate values associated with the same intermediate key generated by Map. PCKMMR constitutes the first MapReduce approach to CC. It applies the MapReduce approach on the COP-K-Means algorithm. In order to do so, authors propose a Map function which calculates the distance of each

Table 28 Feature table for DbCC methods

S_A	Acronym	ID	Penalty	ML	CL	Hybrid	Year	Refs.
3.84	C-DBSCAN	39	×	Hard	Hard	Hierarchical CC	2007	Ruiz et al. (2007)
4.48	C-DenStream	80	×	Hard	Hard	Hierarchical CC & Online CC	2009	Ruiz et al. (2009)
5.71	Cons-DBSCAN	137	×	Hard	Hard	Hierarchical CC & Active Constraint Acquisition	2012	Zhao et al. (2012)
3.79	SDenPeak	213	×	Soft	Soft	–	2016	Fan et al. (2016)
3.34	SemiDen	224	×	Hard	Hard	–	2017	Atwa and Li (2017)
4.33	YZWD	239	×	Soft	Soft	–	2017	Yang et al. (2017)
5.37	SSDC	243	×	Soft	Hard	–	2018	Ren et al. (2018)
15.94	SSDPC	272	×	Soft	Soft	–	2020	Yan et al. (2021)
16.80	fssDBSCAN	280	×	Soft	Soft	Time Series	2021	He et al. (2019)
17.22	ADPE	281	×	Soft	Soft	Active Clustering with Constraints & Constrained Pool Generation	2021	Shi et al. (2020)
17.84	ADP	282	×	Soft	Soft	Active Clustering with Constraints	2021	Shi et al. (2020)

instance to each centroid and assigns it to the one that minimizes this measure and does not violate any constraint. To avoid interdependence between mappers, the constraints are generated locally to each mapper in each call. The map function returns the centroid/instance pair. The Reduce operation takes as input all instances associated to a centroid and updates the value for that centroid by computing the average of those instances.

7.14.7 SAT-based CC

SAT-based CC approaches formulate the CC problems in terms of logical clauses in conjunctive normal form. They apply general SAT solvers to find a solution, which can find solutions for any problems formulated as a satisfiability problem. This approaches to CC problem can include hard constraints in a very natural way, as well as they usually can handle many types of the constraints described in Sect. 2. ISL is limited to clauses of 2 lals. Several types of problems within the CC framework can be expressed in the form of sets of formulas in closed normal form (CNF), implying that they can be approached with ISL and therefore solved optimally. It is worth noting that ISL is limited to two-class problems ($k=2$), although it can find a solution to these problems in polynomial times (if such solutions exists). SGID performs agglomerative hierarchical clustering given a dataset and a set of constraints formulated in terms of logical clauses in its Horn's normal form. In order to produce a dendrogram, the clauses modeling its properties must also be given to SGID, which allows them to vary in the features of the dendrogram it produces. In JPBMS the declarative modeling principle of constrained programming is used to define a CC problem taking into account the constraint set, the description of the clusters and the clustering process itself. Traditionally, clustering algorithms proceed by iteratively refining queries until a satisfactory solution is found. JPBMS includes the stepwise refinement process in a natural way to focus on more interesting clustering solutions. JBMJ performs CC under the correlational clustering paradigm, where a labeled weighted undirected graph is given to perform clustering. The objective function of correlation clustering clusters the nodes of the graph in a way that minimizes the number of positive edges between different clusters and negative edges within clusters. JBMF formulates this problems in terms of clause satisfiability and applies MaxSAT to obtain an optimal solution. It is able to include several types of constraints by modifying the graph structure and applying specific SAT-translation procedure for some of them. Particularly, instance-level constraints are included by just setting the weight of the edges which connect ML related instances to ∞ and CL related instances to $-\infty$, there is no need to generate specific clauses.

7.14.8 CC through constraint programming

The CC problem can be addressed from the constraint programming (CP) point of view when the classic requirements of a clustering problem formulated as restrictions for a CP are extended with the restrictions regarding ML and CL constraints. TCK does this exactly: it models the classic clustering problem requirements, the constraints, and the clustering optimization criteria, including the within-cluster sums of squares (WCSS), as constraints to be solved by a general constraint programming solver. 3CP extends TCK in the sense that it is more general and it does not need the number of clusters to be specified, only the boundaries of the interval it lies in. Additionally, 3CP is capable of optimizing more than

Table 29 Feature table for OCC methods

S_A	Acronym	ID	Penalty	ML	CL	Hybrid	Year	Refs.
0.34	CME	34	✓	Hard	Hard	–	2006	Moreno et al. (2006)
4.48	C-DenStream	80	×	Hard	Hard	Hierarchical CC & Density-based CC	2009	Ruiz et al. (2009)
4.35	SemiStream	127	✓	Soft	Soft	Penalty-based Methods	2012	Halkidi et al. (2012)
12.20	O-LCVQE	159	✓	–	Soft	Neural Networks-based CC	2013	Covões et al. (2012)
12.20	C-RPCL	160	×	–	Soft	Neural Networks-based CC	2013	Covões et al. (2012)
4.34	TDCK-Means	193	✓	Soft	Soft	Penalty-based Methods	2014	Rizoiu et al. (2014)
10.74	CS2GS	201	×	Soft	Soft	Self Organizing Maps-based CC	2015	Allahyar et al. (2015)
5.48	SCSC	206	×	Soft	Soft	Constrained Pool Generation	2015	Yi et al. (2015)

one clustering criteria at the same time, finding the minimal set of nondominated Pareto solutions. TKC(17) also extends TCK, differing from it in two key aspects. Firstly, It does not need the number of clusters to be specified, only bounds need to be given (as in 3CP). Secondly, three optimization criteria are modeled as CP constraints in TKC(17): minimizing the maximal diameter, maximizing the split between clusters, and minimizing WCSS. Besides, CMSSCCP optimizes the WCSS, but it does so via a global optimization constraint. A lower bound for this criterion is computed using dynamic programming, and a filtering algorithm is proposed to filter objective variables as well as decision variables. As usual, instance-level constraints are modeled in terms of constraint programming, along with the classic clustering problem requirements. JPBMS (also in Sect. 7.14.7) uses the declarative modeling principles of CP to define the CC problem as a SAT problem, which can be solved with a general SAT solver. BBMSC formulates the CC problem in terms of an integer programming problem, rather than in terms of a constraint programming problem (which can be considered as a subtype of the former). It employs the same procedures as in constraint programming, incorporating ML and CL constraints via a weighted penalty term added to the base classic clustering restrictions, alongside with a regularization term to favor smoothness.

7.14.9 CC through column generation

In Column Generation (CG), the minimum sum-of-squares (MSS) problem for clustering is solved optimally. This is done by formulating the problem in terms of an integer linear programming problem. In it, a boolean matrix encoding all possible partitions in its columns is explored to find an optimal solution with respect to a cost function (the MSS in this case) that is applied to the boolean matrix by columns. In practice, the boolean matrix is too large to be computed, therefore it is incrementally built when searching for the optimal solution. The column generation approach derives a master problem from a reduced set of restrictions and iterates between two steps: solving the master problem and adding one or multiple candidate columns to the boolean matrix. A column is a candidate to be included in the restricted master problem if its addition improves the objective function. If no such column can be found, one is certain that the optimal solution of the restricted master problem is also the optimal solution of the full master problem. CCG includes constraints into this framework by modifying the MSS formulation. This way a new restriction which enforces all constraints to be satisfied is added to its classic form. Effectively, this is translated into the clustering process by removing all partitions violating any amount of constraints from the boolean matrix. This way, they are discarded from candidate solutions. The CG+PR+LS solves CC similarly to CCG, however it includes two extra steps: path-relinking algorithms are used to intensify and diversify the search in a group of solutions, and a LS procedure is used to locally improve the final solution.

7.14.10 Information-maximization CC

Information-maximization clustering techniques address the lack of objective model selection and parameter optimization strategies from which most other clustering techniques suffer. In it, a probabilistic classifier is learned so that some information measure between instances and clusters assignments is maximized. 3SMIC includes constraints into the infor-

Table 30 Feature table for CC algorithms belonging to minor categories (Others)

S.A.	Acronym	ID	Category	Penalty	ML	CL	Hybrid	Year	Refs.
5.45	SSAP	85	Affinity Propagation	×	Soft	Soft	–	2009	Givoni and Frey (2009)
9.13	COALA	28	Alternative Clustering	×	Any	Any	–	2006	Bae and Bailey (2006)
9.13	COALAcac	29	Alternative Clustering	×	Any	Any	–	2006	Bae and Bailey (2006)
4.34	ADFT	67	Alternative Clustering	×	Soft	Soft	–	2008	Davidson and Qi (2008)
8.56	ClusILC	46	Clustering Trees	×	Soft	Soft	–	2007	Struyf and Džeroski (2007)
4.74	OSS-NMF	96	Co-Clustering	×	Soft	Soft	Non-negative Matrix Factorization CC	2010	Ma et al. (2010)
2.06	RIFM	97	Co-Clustering	×	Hard	Hard	–	2010	Pensa et al. (2010)
11.82	SS-NMF	109	Co-Clustering	×	Soft	Soft	Non-negative Matrix Factorization CC	2010	Chen et al. (2009)
12.45	SS-FCC	163	Co-Clustering	✓	Soft	Soft	Fuzzy CC	2013	Yan et al. (2013)
11.50	CG+PR+LS	236	Column Generation	×	Soft	Soft	Single Individual	2017	Oliveira et al. (2017)
5.02	CCG	175	Column Generation	×	Hard	Hard	–	2014	Babaki et al. (2014)
7.58	CCPP	264	Constraint Programming	×	Hard	Hard	–	2020	Nghiem et al. (2020)
6.31	TKC(17)	229	Constraint Programming	×	Hard	Hard	–	2017	Duong and Vrain (2017)
2.38	BBMSC	205	Constraint Programming	✓	Soft	Soft	–	2015	Hu et al. (2014)
15.41	CMSSCCP	199	Constraint Programming	×	Soft	Soft	–	2015	Dao et al. (2015)
8.03	3CP	197	Constraint Programming	×	Hard	Hard	–	2015	Calandriello et al. (2014)
9.06	TKC	154	Constraint Programming	×	Hard	Hard	–	2013	Dao et al. (2013)
16.39	cut	84	Constraint Programming	×	Soft	Soft	–	2009	Xia (2009)
4.12	YLYM	70	Feature Selection	×	Soft	Soft	–	2008	Li et al. (2008)
7.53	SCAN	259	HIN	×	Soft	Soft	–	2019	Zhao et al. (2019)
2.61	SCHAIN	230	HIN	×	Soft	Soft	–	2017	Li et al. (2017)
7.00	ElCC	53	Incremental CC	×	Hard	Hard	–	2007	Davidson et al. (2007)
4.42	3SMIC	196	Information Maximization	×	Soft	Soft	–	2014	Calandriello et al. (2014)
5.55	NLPPC	271	Label Propagation	✓	Soft	Soft	–	2020	Bai et al. (2020)
4.88	PCKMMR	114	MapReduce	×	Soft	Soft	–	2011	Lin et al. (2011)
10.99	MVMC	226	Matrix Completion	×	Soft	Soft	Intra-View Constrained & Spectral CC	2017	Zhao et al. (2017)

Table 30 (continued)

<i>S_A</i>	Acronym	ID	Category	Penalty	ML	CL	Hybrid	Year	Refs.
17.07	CMMC	68	Maximum Margin Clustering	×	Soft	Soft	–	2008	Hu et al. (2008)
11.70	PMMC	140	Maximum Margin Clustering	✓	Soft	Soft	–	2012	Zeng and Cheung (2011)
13.16	TwoClCMMC	171	Maximum Margin Clustering	×	Soft	Soft	–	2013	Zeng et al. (2013)
21.70	PCKM-Mono	311	Monotonic	✓	Soft	Soft	Penalty-based Methods	2024	González-Almagro et al. (2024)
8.65	DCPR	222	Probabilistic Clustering	×	Hybrid	Hybrid	–	2016	Pei et al. (2016)
13.70	d-graph	250	Probabilistic Clustering	×	Soft	Soft	–	2018	Šmíjeja et al. (2018)
7.13	RHWL	43	Probabilistic Clustering	×	Soft	Soft	Active Clustering with Constraints	2007	Huang and Lam (2007)
2.98	JPBMS	130	SAT	×	Hard	Hard	–	2012	Métivier et al. (2012)
3.20	JBMJ	228	SAT	×	Hard	Hard	–	2017	Berg and Järvisalo (2017)
3.29	SGID	121	SAT	×	Hard	Hard	Hierarchical CC	2011	Gilpin and Davidson (2011)
4.02	ISL	103	SAT	×	Hard	Hard	–	2010	Davidson et al. (2010)
12.97	CPSC*-PS	151	Spatial CC	×	Soft	Soft	–	2012	Joshi et al. (2011)
12.97	CPSC*	150	Spatial CC	×	Soft	Soft	–	2012	Joshi et al. (2011)
3.54	CPSC	149	Spatial CC	×	Hard	Hard	–	2012	Joshi et al. (2011)
6.10	conDetSEC	310	Time Series	×	Soft	Soft	Deep Embedded Clustering-based CC	2023	Ienco and Interdonato (2023)
16.80	fssK-Means	279	Time Series	✓	Soft	Soft	Penalty-based Methods	2021	He et al. (2019)
16.80	fssDBSCAN	280	Time Series	×	Soft	Soft	Density-based CC	2021	He et al. (2019)
18.80	CDPS	309	Time Series	×	Soft	Soft	Constrained Distance Transformation	2023	El Amouri et al. (2023)

mation-maximization clustering algorithm SMIC. SMIC tries to learn the class-posterior probability in an unsupervised manner so that the mutual information between instances and their class labels (in the final partition) are maximized. Constraints are included into SMIC by modifying the SMI approximator, so that the inner product of the probabilities of constrained instances which belong to the same cluster is maximized in the case of ML and minimized in the case of CL. Furthermore 3SMIC includes a procedure to apply the transitive property of ML efficiently.

7.14.11 CC through heterogeneous information networks

Heterogeneous Information Networks (HINs) are graphs which model real world entities and their relationships with objects and links, where objects can be of different types and whose links represent different kinds of relationships. HINs in which objects are described by attributes (features) are called attributed HIN or AHIN abbreviated. The challenge in AHINs is to perform clustering based not only on attribute similarity, but also based on link similarity. The former can be measured with a conventional distance measure, while the latter is measured with graph-oriented distance measures, such as shortest-path length and random-walk-based, although meta-path are commonly used as well. A meta-path is a sequence of node types that expresses a relation between two objects in an AHIN. SCAN includes constraints in AHIN by means of a penalty term in its similarity function. It computes the similarity of every node pair based on their attribute similarity and the connectedness of the nodes network. The former is obtained by an attribute similarity measure which considers coupling relationship among attributes, while the latter is derived based on the meta-paths connecting the object pair. This similarity is then penalized proportionally to the number of violated constraints. SCHAIN includes constraints in AHINs by first composing a similarity matrix that measures the similarity of every object pair based on the attribute similarity and the network connectedness. SCHAIN assigns a weight to each object attribute and meta-path in composing the similarity matrix. To take constraints into account, SCHAIN uses a penalty function which involves the generated weights and cluster memberships. It employs an iterative, staggered 2-step learning process to determine the optimal weights and cluster assignment as output.

7.14.12 Incremental CC

In incremental CC, a fixed set of instances and a variable constraint set are provided to perform clustering. Modifications over the constraint sets are given to incremental CC algorithms over time. These modifications include the addition and removal of constraints. The goal of Incremental CC is to efficiently update a the current partition, without running the base clustering algorithm used to generate the initial partition. EICC is the only proposal belonging to this category. This method takes as input a single constraint at a time, and depending on the properties of the constraint, it will attempt to greedily optimize a given (not specified) objective function. If the constraint does not result in a significant improvement in the objective function, it is passed over and a new one is chosen (by the user). This algorithm only works when a set of preconditions are met, otherwise it will not produce a partition of the dataset, therefore it cannot be used to address a general constrained clustering problem.

7.14.13 CC through affinity propagation

In Affinity Propagation (AP) a binary grid factor-graph is used to perform clustering and to determine the most representative instances, which are called exemplars and can be compared with the notion of centroid. In order to do so, a number of hidden variables equal to the number of pairwise dissimilarities is defined. Afterwards, an iterative procedure is performed over the hidden variables, updating their value on the basis of their neighborhood variables and hyperparameters. The final value of the hidden variables determines the exemplar instances and the membership of the rest of instances with respect to the exemplars. SSAP includes constraints in AP by introducing a fictitious meta-points for every chunklet in the transitive closure and for every CL instance which is not part of a chunklet. The meta-points allow explicitly enforcing ML constraints and CL constraints, while they also propagate them.

7.14.14 Spatial CC

Spatial clustering is a variant of classical clustering in which instances are not points, but polygons. Classic clustering methods do not work well when applied to spatial clustering because they represent the polygons as points which summarize their features; which are not sufficiently representative of the polygons to obtain a good result. To overcome this problem, the CPSC algorithm is proposed, a spatial clustering algorithm based on the A* algorithm which is able to consider both instance-level constraints (ML and CL) and cluster-level constraints. In order to include the constraints into the clustering process, the heuristic function used by the A* algorithm is designed based on them. CPSC starts by selecting k seeds from the data set (polygons), which will be the initial clusters and will grow through the iterative process. The seeds must be selected in such a way that each of them violates all ML constraints with respect to other seeds, thus ensuring that they will not be grouped in the same cluster. In addition to this, the seeds must satisfy all CL constraints between them. The best k seeds are selected among those that meet these conditions. After that, the A* algorithm starts. The seeds are considered as the initial state, and the target clusters as the goal state. Each cluster is increased by adding polygons to its initial state one by one until it reaches its goal state. At each iteration, the best cluster (with respect to the heuristic) to be augmented and the best polygon to be augmented are selected. This is done to ensure that all clusters grow in parallel and not sequentially, which would affect compactness. The process continues until all polygons have been assigned to a cluster or until a deadlock state is reached, which can occur when two clusters compete for the same polygon and in which case there may be polygons which are not assigned to any cluster in the final partition. To overcome this problem, two other algorithms are proposed: CPSC*, which allows the user to relax the constraints to ensure that all polygons are assigned to a cluster, ensuring convergence, and CPSC*-PS (polygon split), which also allows polygons to be split when strictly necessary.

7.14.15 Probabilistic CC

In probabilistic clustering a probabilistic model is used to describe relationships between instances and their cluster memberships. Constraints are included into this framework by

also describing them in terms of probabilities. RHWL is an active CC method which uses a basic probabilistic CC procedure as the clustering algorithm. It takes advantage of the probabilities computed by this procedure to later use them to decide the pair of instances to query to the oracle. DCPR uses an objective function maximizing the likelihood of the observed constraint labels composed of two terms: (1) the first is the conditional entropy of the empirical cluster label distribution, which is maximized when the cluster labels are uniformly distributed (the clusters are balanced). (2) the second term is the conditional entropy of instance cluster labels for the unlabeled instances, which is minimized when the formed clusters have large separation margin and high confidence for the cluster memberships of unconstrained instances. A variational EM optimization scheme is used to optimized the proposed objective function.

7.15 Constrained distance transformation

Distance transformation methods usually take a standard distance measure as their basis (like as the Euclidean distance) and parameterize it. By modifying this parameters, the distance measure is transformed for it to be adapted to the data and the constraints. Generally, the goal of these methods is to learn a new distance metric bringing ML instances together and setting CL instances apart. Constrained Distance Transformation (CDT) methods are presented in Table 31.

Methods such as CSI and Xiang's learn the weights of a matrix by parameterizing a family of Mahalanobis distances. RCA also learns a Mahalanobis metric. In order to do so it changes the feature space used for data representation by assigning high weights to relevant dimensions and low weights to irrelevant dimensions by means of a global linear transformation. The relevant dimensions are estimated using chunklets. ERCA extends RCA to include CL. It does so by computing a matrix that optimizes the between-class scatter and combining it with the matrix optimizing the within-class scatter. DCA is another extension over RCA to include CL. It does so by looking for a linear transformation which results in an optimal distance metric by maximizing the variance between chunklets and minimizing the variance between instances in the same chunklet. KDCA uses the kernel trick to learn a nonlinear metric distance under the same principles of DCA. MSSB is a modification over RCA that uses a data-dependent regularizer term to avoid the drawbacks that weighting discrimination brings to RCA. From all methods using Mahalanobis distances parameterization, the ITML and A-ITML-K-Means approaches are the only ones using Information Theoretic Metric Learning as its base framework. ITML learns a constrained distance metric by learning a positive-definite matrix that parameterizes a Mahalanobis distance. This matrix is regularized to be as close as possible to a given Mahalanobis distance function, parameterized by another auxiliary matrix. The distance between these two matrices can be quantified via an information-theoretic approach, so it can be computed as the relative entropy between their corresponding multivariate Gaussians. Instance-level constraints are included as linear constraints to the optimization problem, which results in a particular case of the Bregman divergence. Therefore, it can be optimized by the Bregman's method. A-ITML-K-Means is an active clustering with constraints setup for the ITML approach that uses K-means as its clustering algorithm.

It uses a reduced set of selected constraints to perform ITML and applies classic clustering (K-Means) with the newly learned metric in an active clustering setup. ITML learns a

Table 31 Feature table for CDT methods

<i>S_A</i>	Acronym	ID	Penalty	ML	CL	Hybrid	Year	Refs.
19.01	CSI	5	×	Soft	Soft	—	2002	Xing et al. (2002)
20.17	MPCK-Means	11	✓	Soft	Soft	Penalty-based Methods	2003	(Basu et al. 2003; Bilenko et al. 2004)
8.84	HMRF-K-Means	12	✓	Soft	Soft	Penalty-based Methods	2004	Basu et al. (2004b)
15.97	DistBoost	13	×	Soft	Soft	—	2004	Hertz et al. (2004)
13.14	LLMA	15	×	Soft	Soft	—	2004	Chang and Yeung (2004)
15.80	RCA	26	×	Soft	—	Constrained Data Space Transformation	2005	Bar-Hillel et al. (2005)
26.54	ERCA	31	×	Soft	Soft	—	2006	Yeung and Chang (2006)
19.10	DCA	32	×	Soft	Soft	—	2006	Hoi et al. (2006)
16.08	KDCA	33	×	Soft	Soft	—	2006	Hoi et al. (2006)
5.67	Comraf	35	—	Soft	Soft	—	2006	Bekkerman et al. (2006)
19.88	DYYHC	47	×	Soft	—	—	2007	Yeung and Chang (2007)
6.48	ITML	50	×	Soft	Soft	—	2007	Davis et al. (2007)
10.96	SMR	51	×	Soft	Soft	—	2007	Hoi et al. (2007)
14.98	Xiang's	74	×	Soft	Soft	—	2008	Xiang et al. (2008)
11.80	MSSB	83	×	Soft	Soft	—	2009	Baghshah and Shouraki (2009)
16.95	MSBSBS	100	×	Soft	Soft	—	2010	Baghshah and Shouraki (2010a)
6.49	SKMM	101	×	Soft	Soft	Cluster Engine-adapting Methods	2010	Yin et al. (2010)
17.76	MSBSBS(10)	107	×	Soft	Soft	—	2010	Baghshah and Shouraki (2010b)
5.01	LRML	108	×	Soft	Soft	—	2010	Hoi et al. (2010)
18.61	LRKL	118	×	Soft	Soft	—	2011	Baghshah and Shouraki (2011)
7.28	CDIPBY	119	×	Soft	Soft	—	2011	Domeniconi et al. (2011)
7.01	SNN	136	×	Soft	Soft	Classic Neural Network-based CC	2012	Maggini et al. (2012)
7.81	RFD	145	×	Soft	Soft	—	2012	Xiong et al. (2012)
5.62	A-ITML-K-Means	156	×	Soft	Soft	Active Clustering with Constraints	2013	Rao and Jawahar (2013)
7.07	LSCP	204	×	Soft	Soft	Constraint Propagation	2015	Fu et al. (2015)

Table 31 (continued)

S_A	Acronym	ID	Penalty	ML	CL	Hybrid	Year	Refs.
10.40	SSMMHF	216	×	Soft	Soft	–	2016	Johnson et al. (2015)
15.34	AMH-L	266	×	Soft	Soft	–	2020	Abin et al. (2020)
15.34	AMH-NL	267	×	Soft	Soft	–	2020	Abin et al. (2020)
22.52	3SHACC	287	×	Soft	Soft	Hierarchical CC	2022	González-Almagro et al. (2021)
18.80	CDPS	309	×	Soft	Soft	Time Series	2023	El Amouri et al. (2023)

Mahalanobis distance metric under a given set of constraints and instances by minimizing the LogDet divergence between the original distance matrix and an objective distance matrix that is built on the basis of constraints.

HMRf-K-Means approaches the CC problem from a hybrid probabilistic framework based on Hidden Markov Random Fields (HMRf), which is developed to find an EM-optimizable objective function derived from the posterior energy, which is defined by the HMRf. This objective function combines an adaptive distance measure, such as the Bregman divergence or directional similarity measures, and a constraint-violation penalty term. The latter is controlled by a scaling function that assigns more relevance to ML constraints relating distant instances and CL constraints relating close instances. Comraf uses a combinatorial MRF (an MRF in which at least one node is a combinatorial random variable). The Comraf model can be applied to classic clustering by searching for cliques in the Comraf graph and using the mutual information as a potential function. This graph is built on the basis of the interactions between the combinatorial random variables. Comraf can be extended to constrained clustering by incorporating weighted constraints as a penalty term in the objective function that Comraf optimizes.

Other methods such as MPCK-Means iteratively compute cluster-local weights for every feature, effectively creating a new distance metric for every cluster, which can be applied and updated during the clustering process in an EM scheme. LLMA also performs metric learning through locally linear transformations, achieving global consistency via interactions between adjacent local neighborhoods.

SMR builds the graph Laplacian matrix based on pairwise similarities. This matrix is then used to regularize a non-parametric kernel learning procedure in which the learned kernel matrix is forced to be consistent with both pairwise similarities and the constraint sets by minimizing the regularizer that is based on the Laplacian graph. MSBSBS is similar to SMR. However it uses the Karush-Kuhn-Tucker conditions to learn the non-parametric kernel. MSBSBS(10) is an improvement over MSBSBS which combines both the constraint set and the topological structure of the data to learn a non-linear metric. CDJPBY uses a weighted sum to combine multiple kernels. It includes an optimization criterion that allows it to automatically estimate the optimal parameter of the composite Gaussian kernels directly from the data and the constraints.

RDF transforms the metric learning problem into a binary class classification problem and employs random forests as the underlying representation. Constraints are included by replacing the original distance function with a feature map function which transforms each constrained instance, changing their location to reflect the information contained in the constraint set. The transformed data is used as training data for a random forest that evaluates each instance pair. Each tree from the random forest independently classifies the pair as similar or dissimilar, based on the leaf node at which the instance-pair arrives. SSMMHF uses a random forest-based strategy as well. It first builds a model of the data by computing a forest of semi-random cluster hierarchies. Each tree is generated applying a semi-randomized binary semi-supervised maximum-margin clustering (MMC) algorithm iteratively. This way, each tree encodes a particular model of the full semantic structure of the data, so the full structure of the tree can be considered as a weak metric. A final metric model can be produced by merging the output of the forest described above. Constraints are included by modifying the MMC procedure, producing Semi-Supervised MMC. This results in a method that seeks to simultaneously maximize the cluster assignment margin of each point

(as in unsupervised MMC) and an additional set of margin terms reflecting the satisfaction of each pairwise constraint.

SCKMM firstly estimates the optimal value for the parameter of a Gaussian kernel by ascending gradient and obtains an initial partition using PCBKM, which is a modification of the classic K-Means algorithm to include constraints. After this, a distance measure is obtained using the assignments of the last partition, which is used by PCBKM to produce a new partition. SCKMM iterates between steps two and three until it converges.

3SHACC includes a metric learning step in 2SHACC. In first place, it determines the relevance of every constraint in a completely unsupervised manner. Said relevances are used as constraint weights by the Weighted-Learning from Side Information (WLSI) DML method, which is a weighted version of CSI.

7.16 Distance matrix modification

In Distance Matrix Modification (DMM) methods, the CC problem is approached from the DML point of view. Nevertheless these methods work directly with the distance matrix. Their goal is to modify the entries of the distance matrix for it to reflect the information contained in the constraint set, once again resulting in ML instances being brought closer together and CL instances being set apart. These methods do not provide neither a new distance metric nor a new data space, although these two can be obtained from said distance matrix with classic DML techniques.

7.16.1 Constraint propagation

In constraint propagation methods, entries in the distance matrix which correspond to constrained instances are usually first modified. Then, those changes are propagated to the rest of the matrix to a variable extent and using different strategies. Table 32 gathers methods which use constraint propagation to perform CC.

The most simple approach in constraint propagation consists of simply setting distances between ML instances to 0 and to a high value for CL instances in the distance matrix. Afterwards, the all-pairs-shortest-path algorithm is run to propagate the changes to the rest of entries. Methods such as CCL, 2SHACC and CAF use this approach.

E²CP propagates constraints in the distance matrix by taking each of its columns as the initial configuration of a two-class semi-supervised learning problem with respect to the instance associated to the column. The positive class contains the examples which should appear in the same cluster as the instance associated to the column, and the negative class contains the examples that should not. In this way, the constraint propagation problem can be decomposed into a number of subproblems equal to the size of the dataset. After that, the same process is repeated, this time taking rows instead of columns. MSCP extends E²CP to consider multi-source data. It decomposes the problem into a series of two-source constraint propagation subproblems, which can be transformed into solving a Sylvester matrix equation, viewed as a generalization of the Lyapunov matrix equation. SRCP also uses decomposition to propagate constraints, although it decomposes the problem taking into account the full dataset, not only columns or rows. It exploits the symmetric structure of pairwise constraints to develop a constraint propagation approach based on symmetric graph regular-

Table 32 Feature table for DMM - Constraint Propagation methods

S_A	Acronym	ID	Penalty	ML	CL	Hybrid	Year	Refs.
19.06	CCL	4	×	Soft	Soft	–	2002	Klein et al. (2002)
12.33	LCPN	61	×	Soft	Soft	Non Graph-based	2008	Lu and Carreira-Perpinan (2008)
8.22	Lo-NC	82	×	Soft	Soft	–	2009	Yan et al. (2009)
11.69	E ² CP	95	×	Soft	Soft	–	2010	Lu and Ip (2010); Lu and Peng (2013)
5.30	SRCP	120	×	Soft	Soft	–	2011	Fu et al. (2011)
6.89	MMCP	122	×	Soft	Soft	–	2011	Fu et al. (2011)
5.71	LCP	128	×	Soft	Soft	–	2012	He et al. (2012)
4.63	SSL-EC	131	×	Soft	Soft	Active Constraint Acquisition	2012	Wang et al. (2012)
5.19	UCP	167	×	Soft	Soft	Inter-View Constrained	2013	Lu and Peng (2013)
4.00	MSCP	169	×	Soft	Soft	Inter-View Constrained	2013	Lu and Peng (2013)
5.62	A-ITML-K-Means	156	×	Soft	Soft	Active Clustering with Constraints	2013	Rao and Jawahar (2013)
10.31	CAF	166	×	Soft	Soft	Constrained Data Space Transformation	2013	Asafi and Cohen-Or (2013)
4.04	ACC(14)	179	×	Soft	Soft	–	2014	He et al. (2014)
7.07	LSCP	204	×	Soft	Soft	Constrained Distance Transformation	2015	Fu et al. (2015)
4.40	CPSNMF	211	×	Soft	Soft	Non-negative Matrix Factorization CC	2016	Wang et al. (2015)
22.93	ISSCE	214	×	Soft	Soft	Constrained Pool Generation	2016	Yu et al. (2015)
20.25	RSSCE	215	×	Soft	Soft	Constrained Pool Generation	2016	Yu et al. (2015)
7.73	C ³	217	×	Soft	Soft	–	2016	Xu and He (2016)
7.83	CESCP	252	×	Soft	Soft	Constrained Pool Generation	2018	Yu et al. (2018)
10.51	DCECP	253	×	Soft	Soft	Constrained Pool Generation	2018	Yu et al. (2018)
5.21	PCPDAMIR	273	×	Soft	Soft	–	2020	Jia et al. (2020)
16.38	2SHACC	262	×	Soft	Soft	Hierarchical CC	2020	González-Almagro et al. (2020)
4.16	ILMCP	277	×	Soft	Soft	–	2021	Li and Lu (2021)

ization. MMCP propagates constraints in rows and columns separately without decomposition using multi-graph propagation methods.

LCP propagates the influence of the constraints to the unconstrained instances in the dataset proportionally to their similarity with the constrained data. LCP first determines the proportion in which constrained instances influence unconstrained instances using a previously proposed label propagation procedure. Then, intermediate structures, called constrained communities, are defined to include the factional instances that are affected by a constrained instance (including itself). These structures are used to find the range of influence of each constraints without any parameter estimation. ACC(14) performs constraint propagation the same way as LCP, the only difference being that ACC(14) allows overlapping between constrained communities. C^3 propagates constraints proportionally in constrained communities as well, although directly performing clustering on them through their indicator matrix.

PCPDAMR uses both the similarity and the dissimilarity (distance) matrix to perform constraint propagation. It does so to emphasize the difference between ML and CL constraints, which are usually encoded in the same matrix. The constraint propagation is carried out via manifold embedding, in which the inherent manifold structure among the data instances is mapped to their similarity/dissimilarity codings. A regularization term to consider adversarial relations between the two matrices is used to enhance the discriminability of propagated constraints.

Many methods simply use the E^2CP as an intermediate step performed between other CC-related operations. For example, RSSCE, CESSCP and DCECP are all ensemble clustering method which use E^2CP as the CC method to generate different partitions using the random subspace technique. CPSNMF simply performs E^2CP and a penalty-based version of NMF afterwards.

More exotic approaches can be found in LCPN or Lo-NC. LCPN takes the affinity matrix as the covariance matrix of a parameterized Gaussian process with mean 0, effectively connecting the spatial locations of constrained instances and propagating a positive or negative affinity value (depending on whether it is a ML or CL constraint) in those locations. Lo-NC performs a space-level generalization of pairwise constraints by locally propagating the information contained in the constraint set.

7.16.2 Matrix completion

Matrix completion techniques, gathered in Table 33, are used to fill gaps in relational matrices. Its key concept is found in how to build the matrix over which matrix completion is applied. The reason is that matrix completion algorithm itself does not need to be specifically designed for CC, but it only need a matrix with gaps to be filled, so low-rank matrices are preferred.

In CC, the constraint matrix is usually a low-rank matrix, perfectly suitable for constraint propagation. STSC takes advantage of this and proposes the first approach to CC from the Self-taught learning paradigm, where side information is generated by the same algorithm that will make use of it later without the need of an oracle (or human). STSC can augment the set of constraints taking advantage of the low-rank nature of the constraint matrix via matrix completion. Matrix completion methods are able to recover low-rank matrices with high probability by using only a small number of observed entries. STSC performs self-

Table 33 Feature table for DMM - Matrix Completion methods

S_A	Acronym	ID	Penalty	ML	CL	Hybrid	Year	Refs.
6.76	MCCC	165	×	Soft	Soft	–	2013	Yi et al. (2013)
4.43	STSC	188	×	Soft	Soft	Non Graph-based	2014	Wang et al. (2014)
8.95	MCPCP	198	×	Soft	Soft	Inter-View Constrained	2015	Yang et al. (2014)
3.49	LMRPCP	208	×	Soft	Soft	–	2015	Fu (2015)
10.99	MVMC	226	×	Soft	Soft	Intra-View Constrained & Spectral CC	2017	Zhao et al. (2017)

taught constraint augmentation and constrained spectral clustering in an iterative manner. Constraint augmentation is performed via matrix completion over a combination of the low-rank constraint matrix and the affinities in the affinity graph. MCPCP also performs matrix completion over the constraint matrix, with each entry in the matrix being a real number that represents the relevance of the two corresponding instances. MCPCP aims to learn the full relation matrix by using a matrix completion algorithm and derives an indicator matrix from it.

Other methods aim to reconstruct and artificially made low-rank similarity matrix. This is the case of MCCC, which assigns similarity 1 for any pair of instances in the same cluster and 0 otherwise, based on the given constraints and the dataset. It can be proven that this is equivalent to finding the best data partition. A convex optimization problem, whose global solution can be efficiently obtained, can be used to solve this problem. Please note that the aim of MCCC is reconstructing the similarity matrix taking constraints into account, not the constraint matrix.

Lastly, LMRPCP performs matrix completion within a transductive learning framework. These approaches make the data matrix and the label matrix jointly low-rank and simultaneously apply a matrix completion algorithms to them. LMRPCP assumes that the data matrix is a clean low-rank matrix, while the constraint matrix is considered to be noisy and low-rank. The resulting problem is a matrix completion problem which can be solved with an augmented Lagrangian multiplier algorithm. The generated constraints are used to adjust pairwise similarities, over which classic spectral clustering is performed to obtain a partition.

7.17 Constrained data space transformation

Constrained Data Space Transformation (CDST) techniques (gathered in Table 34) seek to transform the space in which the data is embeded so that the new space can include the information contained in the constraint set. This usually involves reducing or augmenting the number of dimensions of said space. The majority of methods which belong to this category seek to reduce the number of dimensions, thus summarizing (and sometimes losing) information from the original data space. Other methods augment the number of dimensions based on the constraints and without any loss of information, although they produce a larger dataset which is usually harder to process by partitional methods.

Most methods in this category perform dimensionality reduction over the original dataset. Some of them are based on the classic PCA algorithm: RCA, CCC-GLPCA and DP-GLPCA are some examples of this. RCA is an ML constrained version of the classic PCA algorithm. It seeks to identify and down-scale global unwanted variability within the data. In order to do so, it changes the feature space used for data representation by assigning high weights to relevant dimensions and low weights to irrelevant dimensions by means of a global linear transformation. The relevant dimensions are estimated using chunklets. CCC-GLPCA includes a regularized term in the objective function of GLPCA (a Graph-Laplacian variant of PCA) to include constraints. This regularizes the similarity between the multiple low-dimensional representations used by GLPCA. DP-GLPCA simply performs classic GLPCA over the modified distance matrix to set distances between instances related by ML and CL to 0 or 1, respectively. Besides, it includes a dissimilarity regularizer which emphasizes CL to expand their influence.

Table 34 Feature table for CDST methods

S_A	Acronym	ID	Penalty	ML	CL	Hybrid	Year	Refs.
15.80	RCA	26	×	Soft	–	Constrained Distance Transformation	2005	Bar-Hillel et al. (2005)
17.63	SCREEN	41	×	Hard	Soft	–	2007	Tang et al. (2007)
16.00	PCP	64	×	Soft	Soft	–	2008	Li et al. (2008)
8.22	RLC-NC	81	×	Soft	Soft	–	2009	Yan et al. (2009)
8.29	GBSSC	98	×	Hard	Soft	Graph-based	2010	Yoshida (2010); Yoshida and Okatani (2010); Yoshida (2011, 2014)
17.73	CPSSAP	153	×	Soft	Soft	–	2012	Wang et al. (2012)
10.31	CAF	166	×	Soft	Soft	Constraint Propagation	2013	Asafi and Cohen-Or (2013)
11.55	MPHS-Linear	172	×	Soft	Soft	–	2013	Gao et al. (2013)
11.55	MPHS-Gauss	173	×	Soft	Soft	–	2013	Gao et al. (2013)
11.55	MPHS-PCP	174	×	Soft	Soft	–	2013	Gao et al. (2013)
6.93	CNP-K-Means	187	×	Soft	Soft	Non Graph-based	2014	Wang et al. (2014)
4.43	CCC-GLPCA	248	×	Soft	Soft	–	2018	Jia et al. (2018)
7.74	DP-GLPCA	283	×	Soft	Soft	–	2021	Jia et al. (2020)

Other methods use diverse procedures to perform dimensionality reduction taking constraints into account. SCREEN includes a step where a constraint-guided feature projection method (called SCREEN_{PROJ}) is used to project the original data in a low-dimensional space. RLC-NC seeks a low-dimensional representation of the data through orthogonal factorizations in which the clustering structure defined by the prior knowledge is strengthened. GBSSC first obtains the chunklet graph, over which a Laplacian process is applied for the graph to reflect CL constraints. The entire resulting graph is then projected onto a lower-dimensional space. CPSSAP uses the constraint projection methods to produce a faithful representation of the constraint set in a lower-dimensional space. The affinity matrix is computed on the basis of the new data space and a classic affinity propagation algorithm is used to produce a partition of the dataset. CNP-K-Means seeks to project the original dataset into a lower-dimensional space, preserving the information contained in the constraint set. In order to do so, it defines a neighborhood for every instance based on a parameterized radius value, used to propagate the influence of constraints from constrained instances without augmenting the constraint set.

The three variants of MPHS perform dimensionality reduction in a more exotic constraint-oriented way. It is inspired by the maximum margin hyperplane of SVM. Intuitively, CL constraints can be used to define hyperplanes between pairs of instances. Given two instances with related by CL constraint, we can compute its mid-perpendicular hyperplane which is perpendicular to the line across the two instances, which is also the maximum margin hyperplane. Since there is more than one CL constraint in the constraint set, multiple mid-perpendicular hyperplanes can be obtained. MPHS contains three main steps. Firstly, it learns a new data representation using the mid-perpendicular hyperplane corresponding to each cannot-link constraint, which can also be regarded as dimensionality reduction. Secondly, it learns individual similarity matrix according to the new data representation

corresponding to each CL. In the end, individual similarity matrices are aggregated into a similarity matrix and then perform kernel k-means. Three variants of MPHS are proposed in Gao et al. (2013): MPHS-linear (for simple and well-structured data), which is performed on original data space, MPHS-Gauss (for complex data) which is performed on Gaussian-kernel induced feature space, and MPHS-PCP which first learns a data-dependent kernel similarity (PCP-kernel Li et al. (2008)) and performs MPHS in PCP-kernel induced feature space.

PCP is one of the few methods which projects the original data into a higher-dimensional space to include constraints. To do so, it learns a mapping over the data graph and maps the data onto a unit hypersphere, where instances related by ML are mapped into the same point and instances related by CL are mapped to be orthogonal. This can be achieved using the kernel trick via semidefinite programming, and has to be done in a high dimensional space, as implementing it in the input space is hard if not unfeasible. Another method which projects the original data into higher dimensions is CAF. It augments the initial space with additional dimensions derived from CL constraints. The instances are augmented with additional features, each of which is defined by one of the given CL constraints. ML constraints are included by modifying the initial distance matrix so that the distance between instances related by ML is the lowest among all pairwise distances (greater than 0) and restoring metricity and the triangle inequality afterwards. Pairwise distances between instances in the augmented space combines both the distances in the original space (modified to include ML) and the distances of instances according to each CL constraint. The distance derivation for the new dimension is based on diffusion maps. The actual clustering is then performed by any classic clustering technique.

8 Statistical analysis of the taxonomy

This section presents relevant statistics on the ranked taxonomy proposed in Sect. 7. The UpSetR package provides the perfect tool to obtain a visualization of the overall taxonomy in the form of a statistical summary, presented in Fig. 14. The left histogram represents the number of methods which belong to every category, while the top histogram considers all hybridizations found between said categories. The categories involved in hybridizations are indicated by the central dot matrix. For example, the left histogram shows that a total of 12 methods belong to the KCC category, and the top histogram shows that 11 of them are purely KCC methods, with a single hybrid method, which the central dot matrix indicates that belongs to both KCC and LSCC category thanks to (which is consistent with the information provided in Table 24).

Figure 14 shows that the ACC category is the most prominent one, while also being the one which represents the most hybridizations. The rationale behind this is that all methods which belong to the active clustering with constraints category (Table 12) always use a CC method from another category as their core CC method. In addition to this, it can be observed that the more successful hybridization are found in MMbCC+MbCC, HCC+ACC, and LSCC+ACC, with up to five hybrids methods in each combination. Please note that, for Constrained DML categories (CDT, CDST and DMM) hybridization never refer to the method used to eventually obtain a partition from their outputs, as these method are never considered to determine their category memberships. The proportion of methods which

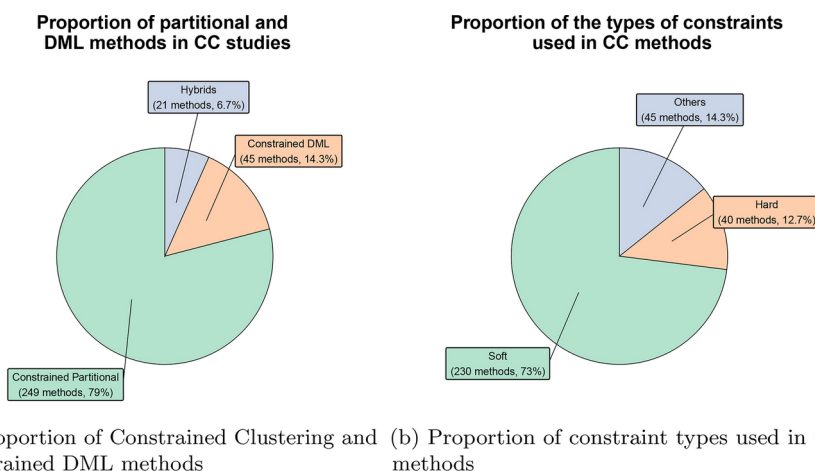


Fig. 13 Piecharts on the proportions of methods in the highest dichotomy of the taxonomy and on the types of constraints

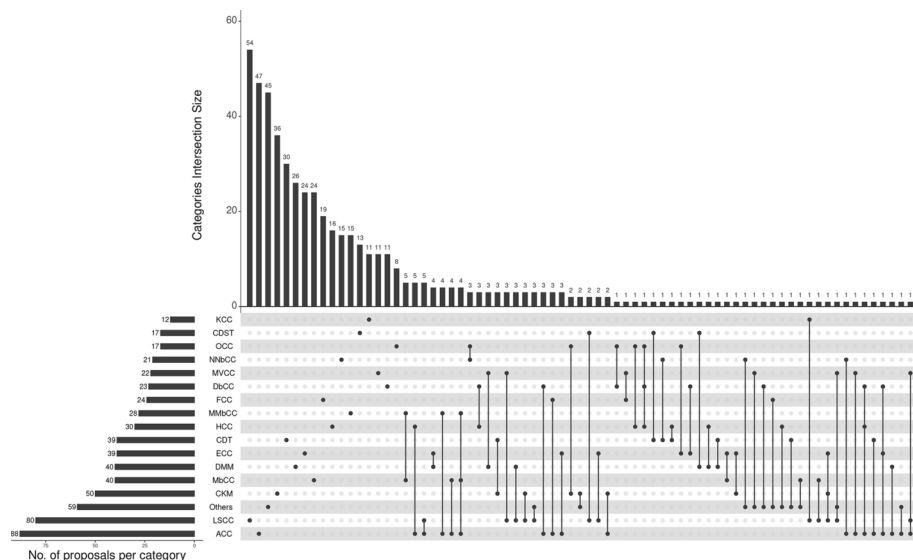


Fig. 14 Summary of taxonomy categories and hybridizations

belong to the classes of the highest level dichotomy in the taxonomy (constrained partitional versus constrained DML) is presented in Fig. 13a, which is introduced next to Fig. 13b, depicting the proportion of the types of constraints used by these methods. From these figures is clear that the vast majority of methods belong to the constrained partitional category and that the use of soft constraints is greatly preferred over any other type of constraints. The “Others” portion in Fig. 13b gathers methods that consider any combination of constraints that are not purely soft nor hard. For example, methods considering only one type of constraints, or using hard ML and soft CL are included in said portion.

Another interesting statistic is presented Fig. 15, which shows a histogram for the number of publications in the CC area sorted by year (please note that only three months of the year 2022 are included in this figure). It is clear that this number increases consistently from 2001 to 2008, when the proficiency of the area becomes inconsistent. Authors firmly believe that this is due to the lack of a solid, general reference in the area, which may help new researchers get a general understanding on the problems and the points of view proposed to tackle them.

A relevant concern within any clustering task is the determination of the number of clusters k . The vast majority of both classic clustering and CC methods simply leaves this step out of the clustering process and takes k as an input hyperparameter. However there are other alternatives to approach this problem. Table 35 gathers CC methods that do not need the number of clusters k to be specified by the user, but they include procedures to determine it in some circumstances. For example, SSFCA can handle both a specification for the number of clusters and the lack of it, as it will try to automatically determine it in such case. The BoostCluster method is basically a wrapper which can be applied to any clustering method in order to include constraints in it, thus handling both cases. Other methods ask the user about an interval in which the k lies in, such as CMSSCCP, 3CP and TKC(17). Besides, there are methods that can only work with a fixed number of clusters (usually $k = 2$), such as CSP and ISL. The rest of the methods do not accept the number of cluster of the output partitions in their inputs, and include procedures to determine it through the clustering process.

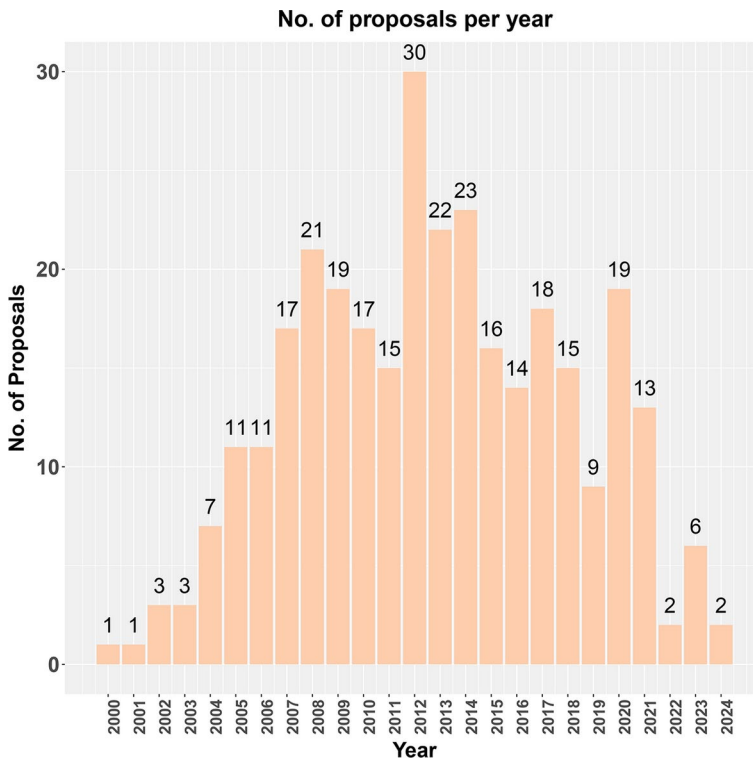


Fig. 15 Number of proposals per year

Table 35 Methods that do not need K to be specified

Acronym	Type	Refs.	Acronym	Type	Refs.
SSFCA	Hybrid	Maraziotis (2012)	URASC	Not Needed	Xiong et al. (2016)
BoostCluster	Hybrid	Liu et al. (2007)	PCCA	Not Needed	Gritra et al. (2005, 2006)
CMSSCCP	Bounded	Dao et al. (2015)	SemiDen	Not Needed	Atwa and Li (2017)
3CP	Bounded	Calandriello et al. (2014)	FIECE-EM+LUC	Not Needed	Fernandes et al. (2019, 2020)
TKC(17)	Bounded	Duong and Vrain (2017)	CECM	Not Needed	Antoine et al. (2012)
CSP	Fixed	Wang and Davidson (2010)	En-Ant	Not Needed	Yang et al. (2012)
ISL	Fixed	Davidson et al. (2010)	CAC	Not Needed	Xu et al. (2011, 2013)
SSDC	Not Needed	Ren et al. (2018)	CELA	Not Needed	Vu et al. (2009)
COBRAS	Not Needed	Craenendonck et al. (2018)	MELA	Not Needed	Vu et al. (2009)
FIECE-EM	Not Needed	Covões and Hruschka (2018)	MCLA	Not Needed	Vu et al. (2009)
JDG	Not Needed	Covões and Hruschka (2018)	SSAP	Not Needed	Givoni and Frey (2009)
Semi-MultiCons	Not Needed	Yang et al. (2022)	ACC	Not Needed	Frigui and Meredith (2008)
ssFS	Not Needed	Huang et al. (2012)	AFCC	Not Needed	Gritra et al. (2008)
SFFD	Not Needed	Li et al. (2015)	COP-b-coloring	Not Needed	Elghazel et al. (2007)
AAVV	Not Needed	Abin and Vu (2020)	MOCK	Not Needed	Handl and Knowles (2006)
FIECE-EM+BFCU	Not Needed	Fernandes et al. (2019, 2020)	MCGMM	Not Needed	Raghuram et al. (2014)
FIECE-EM+FCU	Not Needed	Fernandes et al. (2019, 2020)	COBRA	Not Needed	Ván Craenendonck et al. (2017)
FIECE-EM+DVO	Not Needed	Fernandes et al. (2019, 2020)	ASCENT	Not Needed	Li et al. (2019)
JBMJ	Not Needed	Berg and Jarvisalo (2017)	-	-	-

8.1 Statistic analysis of ranked scores

This section tackles the distribution of the final scores, obtained by applying the methodology introduced in Sect. 6. Firstly, Fig. 16 gives the values for α_1 and α_2 for every year, so the reader can have a better understanding of how the final scores are obtained. Overall, α_1 and α_2 do not show sufficiently significant differences between them for the year of publication to be decisive, although enough to be used as a discriminating factor on a reasonable scale.

Figure 17 gives a visual representation of the detailed scorings for the top 20 best ranked methods. As expected, the best score is obtained by the COP-K-Means method, as it features a high quality experimental setup (S1, S2 and S3) and it is the single most cited CC method ever proposed (S7). This is enough for it to obtain the highest score, even if the paper which proposes it lacks of a proper validation procedure (S4 and S5). Please note that the fact that COP-K-Means is ranked as the best method ever proposed is evidence of the scoring system being resilient with respect to the number of years the method has been available. As a result, one of the oldest CC method reaches top 1 in the ranking, and is followed by a method which was proposed as recently as in 2021, which is SHADE_{CC}. A number of baseline method make it to the top 20, such as ERCA, SSKK, MPCK-Means, DCA or CSI. The reason behind this is the high influence of the I'_A term over the final score S_A . Other methods reach the top 20 by other means, such as a high experimental quality combined with proper validation procedures.

Finally, Fig. 18 shows a histogram with the distribution of all final scores presented in Sect. 7. It is clear that the majority of methods are scored in $[0, 10]$, fewer methods are in the next higher range of values, which is $(10, 20]$, and very few outlier scores are found in the range $(20, 35]$. Please note that the effective output range of the scoring systems is $[0, 33.33]$, as no method fully complies with its standards. This shows the suitability of the proposed scoring system, as any objective raking procedure should place the majority of methods in the low-medium range and few methods in the upper range of the ranking, with those methods being the more remarkable ones for their quality and the robustness of their conclusions.

9 Conclusions, criticisms and future research guidelines

This study presented a systematical review on the Constrained Clustering (CC) research domain. Firstly, a general introduction to the Semi-Supervised Learning (SSL) paradigm is given, after which the specific area of semi-supervised clustering is discussed in detail. The discussed methods within this area are capable of including background knowledge (or incomplete information) about the dataset onto the clustering process. To the best of the authors' knowledge, this study provides the first ever overview and taxonomization of the types of background knowledge (in Sect. 2) that can be included as constraints in semi-supervised clustering. Afterwards, we motivate why out of all types of background knowledge, the instance-level pairwise Must-Link (ML) and Cannot-Link (CL) constraints, are the most successful and prominent types of constraints. Semi-supervised clustering methods that use ML and CL constraints are known as Constrained Clustering (CC). The CC problem is formalized and presented in later sections, and illustrated by giving examples of sev-

		α values																							
		α_2	0.76	0.56	0.61	0.97	1.11	0.72	0.93	1.07	0.73	1.28	0.84	1.79	1.01	1.17	1.5	1.13	0.74	1.06	0.93	0.62	1		
		α_1	0.59	0.77	0.65	0.57	0.61	0.85	0.83	0.84	1.2	0.7	0.65	0.91	0.61	0.77	0.78	0.85	0.51	0.68	0.8	1.12	1		
		2000-03	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2023	2024			

Fig. 16 Values for the weighting parameters α_1 and α_2 used in every year

eral practical fields of applications. Afterwards, advanced CC concepts and structures are described and formalized, allowing to discuss in more detail the advantages and disadvantages of different approaches. Afterwards, a statistical analysis of the experimental elements used in studies which proposes new CC methods has been carried out, revealing the basics research in CC area, as well as the baseline methods, benchmark datasets and suitable validity indices. This overview leads to the proposal of an objective scoring system that captures the potential and relevance of each approach, which is used later to assign every method a score that summarizes its quality, and to produce a ranking of CC methods. Afterwards, a taxonomization of 315 CC methods is conducted, ranking them according to the proposed scoring system and categorizing them in two major families: constrained partitional and constrained DML. These two families are further divided in more specific categories, whose common features and specific methods are described in Sects. 7.1 to 7.17.

The proposed taxonomy can be used to:

- Decide which type of approach and model is best suited to a new constrained clustering problem.
- Compare newly proposed techniques to those belonging to the same family in this taxonomy, so that in can be determined whether the new method represents an improvement over the current state-of-the-art.
- Identify the proposals which best support their conclusions and propose more robust methods, thanks to the scoring system. As any other Computer Science research area, the CC area is not free of flaws and criticism. Having reviewed 278 studies (proposing 315 methods), the authors have identified 5 major problems which affect the vast majority of them. These problems can be summarized as follows:
 - *The lack of a unified, general reference.* There is not an updated, general reference unifying the overall CC literature. This affects the foundations of new proposals, as it is hard to find the state-of-the-art methods that can be used to compare new techniques with. This is illustrated in Fig. 9a, which shows that more than half of the proposed CC methods (51.7%) are never used in subsequent studies. The aim of this study is to address this deficiency.
 - *Low number of application studies.* Even if the main purpose of this study is not to review and gather literature concerning CC applications, it has been difficult for the authors to find application studies other than the 95 presented in Sect. 3.4. This is specially significant, given the high number and diversity found in CC studies which propose new methods.
 - *The lack of extensive experimental comparisons.* From Sect. 5, it is clear that, unfortunately, the CC research area is consistently poor regarding the number of datasets,

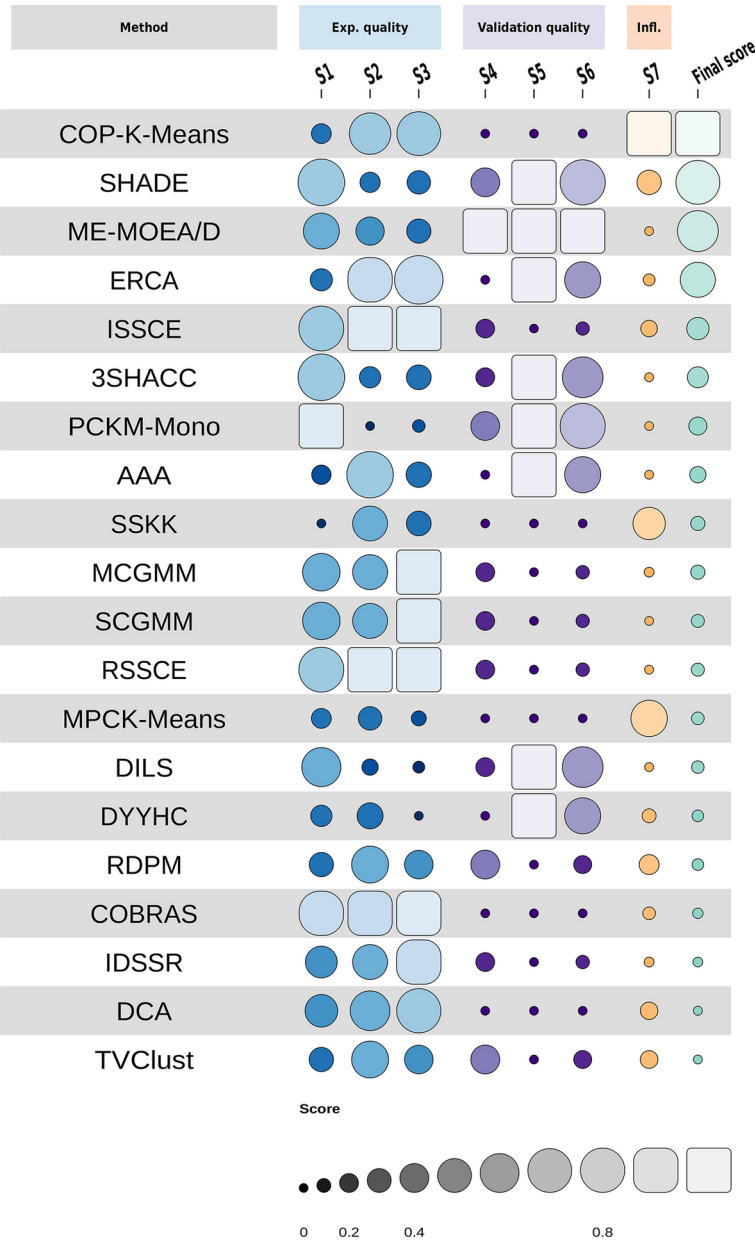


Fig. 17 Detailed scorings for the top 20 ranked methods. Every subscore (S1 to S7) has its own meaning and can be understood individually as follows: the weighted year-normalized number of datasets ($\alpha_1^{Y_A} D'_A$), the weighted year-normalized number of methods ($\alpha_2^{Y_A} MS'_A$), the normalized experimental quality (EQ'_A), the normalized number of validity indices (V'_A), the statistical test usage indicator (T_A), the normalized validation procedure quality (VQ'_A), and the normalized influence (I'_A). Finally, S_A is given in the final score column

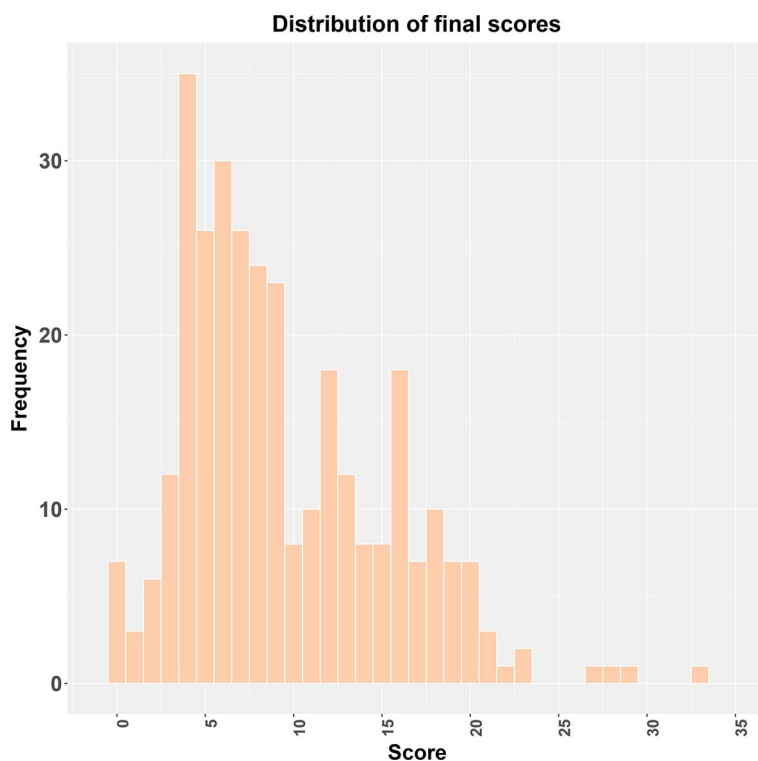


Fig. 18 Distribution of all final scores presented in Sect. 7

validity indices and competing methods used to support their conclusions. Very few methods use more than 10 datasets or 2 validity indices. With respect to the number of competing methods, it is shocking that no study has ever considered more than 9 of them, given that there are, at least, 315. The authors firmly believe that (open-source) code unavailability is responsible for this unsettling fact, as implementation details are rarely given in CC studies.

- *Unavailability of specific standardized CC-oriented datasets and constraint sets.* One of the major flaws of the CC area is the lack of specific datasets, which may be justified, provided that the low number of application papers. However, this is not the case when it comes to the constraint sets. In Sect. 5.4, the constraint generation method used in the vast majority of CC studies was presented. It is clear that this procedure is highly dependent of random effects, as constraints are randomly allocated. For this reason, it is necessary to have access to the specific constraint sets used in a given experimentation if it needs to be reproduced. Unfortunately, constraint sets are only rarely published by authors.
- *Statistically unsupported experimental conclusions.* This is an effect derived from the two previous criticisms, as a low number of standardized experiments is not significant enough to perform statistical testing procedures or to derive generalized conclusions that are broadly applicable. In fact, a very reduced minority of studies support their conclusions using statistical testing. This may greatly affect the confidence future re-

searches may have towards the reviewed studies, reducing their usability in both the development of new methods and their applications in real-world problems. In respect to future research guidelines, new proposals would greatly benefit from avoiding the mentioned flaws as much as possible. New studies should perform extensive experimental comparisons with state-of-the-art methods. A major goal of this overview is to provide easy access to such methods. Similarly, studies would benefit from the use of multiple datasets focusing on a wide range of application domains to make conclusions more generalisable. To this end, authors should always make their datasets and constraint sets public, in order for the result to be reproducible, and preferably also the source code. Supporting conclusions with statistical testing is also essential, as this will increase confidence in the results.

In the landscape of clustering research, there exists a noticeable gap in literature concerning the application of deep learning techniques to constrained clustering problems. Despite the rapid advancement of deep learning in various domains of machine learning, its integration into constrained clustering remains relatively underexplored. The integration of deep learning models has the potential to revolutionize constrained clustering by leveraging the hierarchical and nonlinear relationships within data, which traditional clustering methods may struggle to capture effectively. Encouraging further research in this direction not only promises to advance the theoretical foundations of clustering but also holds significant practical implications for enhancing the accuracy, scalability, and adaptability of clustering algorithms in constrained environments. By bridging this gap, researchers can unlock new avenues for innovation and address complex challenges in diverse application domains.

Future research based on this study can extend the scoring system to evaluate not only quantitative features regarding the quality of the proposal, but also qualitative features, as their novelty or their applicability. This objective would be arduous to achieve with objective standards in mind, as these features greatly depend on the perception of the researchers. Additionally, the constraint equivalences presented in Sect. 2.8 would benefit from formalization. Lastly, the creation of a library of CC baseline algorithms would greatly benefit this research area, making it more accessible to new researchers. Thanks to this study, the mentioned CC baseline algorithms can be chosen using objective criteria.

Full names for all methods reviewed

This appendix gathers Tables 36, 37, 38, 39, 40, 41, 42 and 43, which list all method reviewed in this study, sorted by their identifier, and giving their full names. Methods named after the initials of their authors' names are marked with the word "names". Methods whose name does not refer to neither any acronym nor authors' initials are marked with a hyphen dash ("-").

Table 36 Full names for all methods reviews, sorted by ID (Part I)

ID	Acronym	Full Name
1	COP-COBWEB	CONstrained Partitional - COBWEB
2	COP-K-Means	CONstrained Partitional - K-Means
3	SCOP-K-Means	Soft CONstrained Partitional - K-Means
4	CCL	Constrained Complete-Link
5	CSI	Clustering with Side Information
6	KKM	names
7	TBJSBM	names
8	MAA	names
9	PCK-Means	Pairwise Constrained K-Means
10	FFQS	Farthest First Query Selection
11	MPCK-Means	Metric Pairwise Constrained K-Means
12	HMRf-K-Means	Hidden Markov Random Fields - K-Means
13	DistBoost	–
14	Constrained EM	Constrained Expectation-Minimization
15	LLMA	Locally Linear Metric Adaptation
16	ACCESS	Active Constrained Clustering by Examining Spectral eigenvectorS
17	IDSSR	names
18	GPK-Means	Gaussian Propagated K-Means
19	CVQE	Constrained Vector Quantization Error
20	CSC	Constrained Spectral Clustering
21	PPC	Penalized Probabilistic Clustering
22	PCCA	Pairwise Constrained Competitive Agglomeration
23	SSKK	Semi-Supervised Kernel K-Means
24	MCGMM	Multiple-Component Gaussian Mixture Model
25	SCGMM	Single-Component Gaussian Mixture Model
26	RCA	Relevant Components Analysis
27	ASSKK	Adaptive Semi Supervised Kernel K-Means
28	COALA	Constrained Orthogonal Average Link Algorithm
29	COALAcAt	Constrained Orthogonal Average Link Algorithm (Categorical)
30	MOCK	Multi Objective Clustering with automatic K-determination
31	ERCA	Extended Relevant Components Analysis
32	DCA	Discriminative Component Analysis
33	KDCA	Kernel - Discriminative Component Analysis
34	CME	names
35	Comraf	Combinatorial Markov Random Fields
36	SS-FKCN	Semi-Supervised Fuzzy Kohonen Clustering Network
37	SMCE	Subspace Metric Cluster Ensemble
38	COP-b-coloring	CONstrained Partitional - b-coloring
39	C-DBSCAN	Constraint-driven - DBSCAN
40	BoostCluster	–
41	SCREEN	Semi-supervised Clustering method based on spheRical k-mEans via fEature projection

Table 37 Full names for all methods reviews, sorted by ID (Part II)

ID	Acronym	Full Name
42	PCSK-Means	Pairwise Constrained Spherical K-Means
43	RHWL	names
44	LCVQE	Linear Constrained Vector Quantization Error
45	NMFS	Non-negative Matrix Factorization-based Semi-supervised
46	ClusILC	Clustering with Instance-Level Constraints
47	DYYHC	names
48	SPGP	Semi-supervised Pairwise Gaussian Process classifier
49	SSCARD	Semi-Supervised Clustering and Aggregating Relational Data
50	ITML	Information-Theoretic Metric Learning
51	SMR	names
52	S-SCAD	Semi-Supervised Clustering and Attribute Discrimination
53	EICC	Efficient Incremental Constrained Clustering
54	PAST-Toss	Pick A Spanning Tree - Toss
55	CLAC	Constrained Locally Adaptive Clustering
56	PT	Penta-Training
57	MLC-K-Means	Must-Link Constrained - K-Means
58	CDPMM08	Constrained Dirichlet Process Mixture Models 2008
59	AFCC	Active Fuzzy Constrained Clustering
60	ACC	Adaptive Constrained Clustering
61	LCPN	names
62	COP-CGA	CONstrained Partitional - Clustering Genetic Algorithm
63	S3-K-Means	Semi-Supervised Spectral K-Means
64	PCP	Pairwise Constraint Propagation
65	COP-HGA	CONstrained Partitional - Hybrid Genetic Algorithm
66	CLWC	Constrained Locally Weighted Clustering
67	ADFT	Alternative Distance Function Transformation
68	CMMC	Constrained Maximum Margin Clustering
69	MMFFQS	Min-Max Farthest First Query Selection
70	LYLM	names
71	COPGB-K-Means	CONstrained Partitional Graph-Based - K-Means
72	PCKK-Means	Partial Closure-based Constrained K-Means
73	SS-NMF(08)	Semi-Supervised - Non-negative Matrix Factorization
74	Xiang's	—
75	Cop-EAC-SL	Constrained partitional Evicende ACCumulation Single Link
76	SCK-Means	Soft Constrained K-Means
77	PrTM	Probabilistic Topographic Mapping
78	PCsFCM	Pairwise Constrained standard Fuzzy c-means
79	PCeFCM	Pairwise Constrained entropy Fuzzy c-means
80	C-DenStream	Constrained - Density Stream
81	RLC-NC	Constrained Normalized Cut
82	Lo-NC	Local Normalized Cut
83	MSSB	names
84	cut	—

Table 38 Full names for all methods reviews, sorted by ID (Part III)

ID	Acronym	Full Name
85	SSAP	Semi-Supervised Affinity Propagation
86	RSCPC	Regularized Spectral Clustering with Pairwise Constraints
87	CCSR	Constrained Clustering with Spectral Regularization
88	SCLC	Spectral Clustering with Linear Constraints
89	CCSKL	Constrained Clustering by Spectral Kernel Learning
90	CMSC	Constrained Mean Shift Clustering
91	MCLA	Must-link Cannot-link Leader Ant
92	MELA	Must-link ϵ -link Leader Ant
93	CELA	Cannot-link ϵ -link Leader Ant
94	AHC-CTP	Agglomerative Hierarchical Clustering - Clusterwise Tolerance Pairwise
95	E ² CP	Exhaustive and Efficient Constraint Propagation
96	OSS-NMF	Orthogonal Semi-Supervised - Non-negative Matrix tri-Factorization
97	RJFM	names
98	GBSSC	Graph-Based Semi-Supervised Clustering
99	Samarah	—
100	MSBSBS	names
101	SCKMM	Semi-supervised Clustering Kernel Method based on Metric learning
102	PCBK-Means	Pairwise Constrained Based K-Means
103	ISL	names
104	ICOP-K-Means	Improved COP-K-Means
105	CSP	Constrained SPectral clustering
106	ASC(10)	Ability to Separate between Clusters
107	MSBSBS(10)	names
108	LRML	Laplacian Regularized Metric Learning
109	SS-NMF	Semi-Supervised Nonnegative Matrix Factorization
110	ASC	Active Spectral Clustering
111	S3OM	Semi-Supervised Self Organizing Map
112	CCHAMELEON	Constrained CHAMELEON
113	CAC1	Constrained Active Clustering 1
114	PCKMMR	Parallel COP-K-Means based on MapReduce
115	CLC-K-Means	Cannot-Link Constrained - K-Means
116	SDHCC	Semi-supervised Divisive Hierarchical Clustering of Categorical data
117	AHCP	Agglomerative Hierarchical Clustering with Penalties
118	LRKL	Low-Rank Kernel Learning
119	CDJPBY	names
120	SRCP	Symmetric graph Regularized Constraint Propagation
121	SGID	names
122	MMCP	Multi-Modal Constraint Propagation
123	PNMF	Penalty - Nonnegative Matrix Factorization
124	SCAP	Semi-supervised fuzzy Clustering Algorithm with Pairwise constraints
125	SFFA	names

Table 39 Full names for all methods reviews, sorted by ID (Part IV)

ID	Acronym	Full Name
126	PSC	Pareto based multi objective algorithm for Semi-supervised Clustering
127	SemiStream	—
128	LCP	Local Constraint Propagation
129	SSCsNMF	Semi-supervised symmetriC Non-negative Matrix Factorization
130	JPBMS	names
131	SSL-EC	Semi-Supervised Learning based on Exemplar Constraints
132	CAC	Constrained Ant Clustering
133	En-Ant	Ensemble Ant
134	SSFCA	Semi-Supervised Fuzzy Clustering Algorithm
135	NSDR-NCuts	Near Stranger or Distant Relatives - NCuts
136	SNN	Similarity Neural Networks
137	Cons-DBSCAN	Constrained - DBSCAN
138	COSC	Constrained One Spectral Clustering
139	CECM	Constrained Evidential C-Means
140	PMMC	Pairwise-constrained Maximum Margin Clustering
141	TRAGEK	TRAnsductive Graph Embedding Kernel
142	ENPAKL	Efficient Non-PARametric Kernel Learning
143	COP-SOM-E	COnstrained Partitional - Self Organazing Map - Ensemble
144	IU-Red	Iterative Uncertainty Reduction
145	RFD	Random Forest Distance
146	SCEV	Semi-supervised Clustering Ensemble by Voting
147	sRLe-GDM-FFS	semi-supervised Robust Learning of finite Generalized Dirichlet Mixture models and Feature Subset Selection
148	LSE	Learned Spectral Embedding
149	CPSC	Constrained Polygonal Spatial Clustering
150	CPSC*	Constrained Polygonal Spatial Clustering*
151	CPSC*-PS	Constrained Polygonal Spatial Clustering-Polygon Split
152	SSC-ESE	Semi-Supervised Clustering with Enhanced Spectral Embedding
153	CPSSAP	Constraint Projections Semi-Supervised Affinity Propagation
154	TKC	names
155	SRBR	names
156	A-ITML-K-Means	Active - Information Theoric Metric Learning - K-Means
157	SSKSRM	Semi-Supervised Kernel Switching Regression Models
158	SSSeKRM	Semi-Supervised Sequential Kernel Regression Models
159	O-LCVQE	Online - Linear Constrained Vector Quantization Error
160	C-RPCL	Constrained - Rival Penalized Competitive Learning
161	SSsFCMCT	Semi-Supervised standard Fuzzy C-Means for data with Clusterwise Tolerance by opposite criteria

Table 40 Full names for all methods reviews, sorted by ID (Part V)

ID	Acronym	Full Name
162	SSeFCMCT	Semi-Supervised entropy Fuzzy C-Means for data with Clusterwise Tolerance by opposite criteria
163	SS-FCC	Semi-Supervised - Fuzzy Co-Clustering
164	MVSCE	Majority Voting Semi-supervised Clustering Ensemble
165	MCCC	Matrix Completion based Constraint Clustering
166	CAF	Constraints As Features
167	UCP	Unified Constraint Propagation
168	SS-CLAMP	Semi-Supervised fuzzy C-medoids CLustering Algorithm of relational data with Multiple Prototype representation
169	MSCP	Multi-Source Constraint Propagation
170	E ² CPE	Exhaustive and Efficient Constraint Propagation Ensemble
171	TwoClaCMMC	Two Classes Constrained Maximum Margin Clustering
172	MPHS-Linear	Mid-Perpendicular Hyperplane Similarity - Linear
173	MPHS-Gauss	Mid-Perpendicular Hyperplane Similarity - Gaussian
174	MPHS-PCP	Mid-Perpendicular Hyperplane Similarity - Pairwise Constraint Propagation
175	CCG	Constrained Column Generation
176	PC-HCM-NM	Pairwise Constrained - Hard C-Means - Non Metric
177	PC-eFCM-NM	Pairwise Constrained - entropy Fuzzy C-Means - Non Metric
178	PC-sFCM-NM	Pairwise Constrained - standard Fuzzy C-Means - Non Metric
179	ACC(14)	Adaptive Constrained Clustering
180	ALCSSC	Active Learning of Constraints for Semi-Supervised Clustering
181	SSCA	Semi-supervised Spectral Clustering Algorithm
182	MVCC	Multi-View Constrained Clustering
183	FHCSC	Flexible Highly Constrained Spectral Clustering
184	SCRAWL	Semi-supervised Clustering via RANdom WALK
185	AC-CF-tree	Active Constrained - Clustering Feature - tree
186	SACS	Sequential Approach for Constraint Selection
187	CNP-K-Means	Constraint Neighborhood Projections - K-Means
188	STSC	Self-Taught Spectral Clustering
189	Active-HACC	Active - Hierarchical Agglomerative Constrained Clustering
190	SKMS	Semi-supervised Kernel Mean Shift
191	PCS	PCK-Means with Size constraints
192	CEVCLUS	Constrained EVidential CLUSterng
193	TDCK-Means	Temporal-Driven Constrained K-Means
194	SKML	Spectral Kernel Metric Learning
195	SMVC	Semi-supervised Multi-View Clustering
196	3SMIC	Semi-Supervised Squared-loss Mutual Information Clustering
197	3CP	Constrained Clustering by Constraint Programming
198	MCPCP	Matrix Completion - Pairwise Constraint Propagation

Table 41 Full names for all methods reviews, sorted by ID (Part VI)

ID	Acronym	Full Name
199	CMSSCCP	Constrained Minimum Sum of Squares Clustering by Constraint Programming
200	LXDXD	names
201	CS2GS	Constrained Semi-Supervised Growing SOM
202	CMKIPC	Constrained Multiple Kernels Improved Possibilistic C-Means
203	CCCPYL	names
204	LSCP	Learning Similarity of Constraint Propagation
205	BBMSC	Branch-and-Bound Method for Subspace Clustering
206	SCSC	Semi-supervised Clustering with Sequential Constraints
207	HSCE	Hybrid Semi-supervised Clustering Ensemble
208	LMRPCP	Low-rank Matrix Recovery based Pairwise Constraint Propagation
209	TVClust	Two-Views Clustering
210	RDPM	Relational Dirichlet Process Means
211	CPSNMF	Constrained Propagation for Semi-supervised Nonnegative Matrix Factorization
212	CEAC	Constrained Evolutionary Algorithm for Clustering
213	SDenPeak	Semi-Supervised Density Peak
214	ISSCE	Incremental Semi-Supervised Clustering Ensemble
215	RSSCE	Random Subspace based Semi-supervised Clustering Ensemble
216	SSMMHF	Semi-Supervised Max-Margin Hierarchy Forest
217	C ³	Constrained Community Clustering
218	FAST-GE	Fast-Generalized Spectral Clustering
219	NMFCC	Non-negative Matrix Factorization based Constrained Clustering
220	SymNMFCC	Symmetric Non-negative Matrix Factorization based Constrained Clustering
221	AAA	names
222	DCPR	–
223	CCLS	Constrained Clustering by Local Search
224	SemiDen	Semi-supervised Density-based data clustering
225	COBRA	COntstraint-Based Repeated Aggregation
226	MVMC	Multi-View Matrix Completion
227	URASC	Uncertainty Reducing Active Spectral Clustering
228	JBMJ	names
229	TKC(17)	names
230	SCHAIN	Semi-supervised Clustering in Heterogeneous Attributed Information Networks
231	FAST-GE2.0	Fast-Generalized Spectral Clustering 2.0
232	FQH	names
233	TI-APJCF	Type-I Affinity and Penalty Jointly Constrained Spectral Clustering
234	TII-APJCF	Type-II Affinity and Penalty Jointly Constrained Spectral Clustering
235	BRKGA+LS	Biased Random Key Genetic Algorithm + Local Search
236	CG+PR+LS	Column Generation + Path Relinking + Local Search

Table 42 Full names for all methods reviews, sorted by ID (Part VII)

ID	Acronym	Full Name
237	RSEMICE	Random subspace based SEMI-supervised Clustering Ensemble framework
238	C ₄ S	Constrained Clustering with a Complex Cluster Structure
239	YZWD	names
240	COBS	COnstraint-Based Selection
241	A-COBS	Active - COnstraint-Based Selection
242	CVQE+	Constrained Vector Quantization Error +
243	SSDC	Semi-Supervised DenPeak Clustering
244	COBRAS	COnstraint-Based Repeated Aggregation and Splitting
245	k-CEVCLUS	k - Constrained EvidentialCLUStering
246	AHMRf	A - Hidden Markov Random Field
247	FIECE-EM	Feasible-Infeasible Evolutionary Create & Eliminate - Expectation Maximization
248	CCC-GLPCA	Convex Constrained Clustering - Graph-Laplacian PCA
249	JDG	names
250	d-graph	—
251	PCPSNMF	Pairwise Constraint Propagation-induced Symmetric NMF
252	CESCP	Clustering Ensemble based on Selected Constraint Projection
253	DCECP	Double-weighting Clustering Ensemble with Constraint Projection
254	CMVNMF	Constrained Multi-View NMF
255	BCK-Means	Binary Constrained K-Means
256	SSKMP	Semi-Supervised K-Mediods Problem
257	SFS ³ EC	Stratified Feature Sampling for Semi-Supervised Ensemble Clustering
258	SemiSync	—
259	SCAN	Semi-supervised clustering with Coupled attributes in Attributed heterogeneous information Networks
260	SDEC	Semi-supervised Deep Embedded Clustering
261	MVCSC	Multi-View Constrained Spectral Clustering
262	2SHACC	2 -Stages Hybrid Agglomerative Constrained Clustering
263	ARSCE	Adaptive Regularized Semi-supervised Clustering Ensemble
264	CCPP	Constrained Clustering via Post-Processing
265	DCC	Deep Constrained Clustering
266	AMH-L	names
267	AMH-NL	names
268	S ³ C ²	Semi-Supervised Siamese Classifiers for Clustering
269	PCOG	Pairwise Constrained Optimal Graph
270	DILS _{CC}	Dual Iterative Local Search - Constrained Clustering
271	NLPPC	New Label Propagation with Pairwise Constraints
272	SSDPC	Semi-Supervised Density Peak Clustering
273	PCPDAMR	Pairwise Constraint Propagation with Dual Adversarial Manifold Regularization
274	CDC	Constrained Deep Clustering

Table 43 Full names for all methods reviews, sorted by ID (Part VIII)

ID	Acronym	Full Name
275	CDEC	Constrained Deep Embedded Clustering
276	PCSK-Means(21)	Pairwise Constrained Sparse K-Means
277	ILMCP	Instance Level Multi-modal Constraint Propagation
278	WECR K-Means	WEighted Consensus of Random K-Means ensemble
279	fssK-Means	fast semi-supervised K-Means
280	fssDBSCAN	fast semi-supervised DBSCAN
281	ADPE	Active Density Peak Ensemble
282	ADP	Active Density Peak
283	DP-GLPCA	Dissimilarity Propagation-guided - Graph-Laplacian Principal Component Analysis
284	NN-EVCLUS	Neural Network-based EVIDential CLUStering
285	SHADE _{CC}	Success History-based Adaptive Differential Evolution - Constrained Clustering
286	ME-MOEA/D _{CC}	Memetic Elitist - Multiobjective Optimization Evolutionary Algorithm based on Decomposition - Constrained Clustering
287	3SHACC	3 -Stages Hybrid Agglomerative Constrained Clustering
288	Semi-MultiCons	Semi-supervised Multiple Consensus
289	DGPC	names
290	KAKB	names
291	ssFS	semi-supervised subgraph Feature Selection
292	JDFD	names
293	WAKL	names
294	MICS	Most InformativeConStraints
295	SFFD	Semi-supervised Fuzzy clustering with Feature Discrimination
296	AAA(18)	names
297	RWACS	Random Walk Approach to Constraints Selection
298	AAA(19)	names
299	ALPCS	Active Learning Pairwise Constraint based on Skeletons
300	LCML	names
301	AIPC	Active Informative Pairwise Constraints algorithm
302	AAVV	names
303	FIECE-EM+BFCU	FIECE-EM + Best Feasible Classification Uncertainty
304	FIECE-EM+FCU	FIECE-EM + Feasible Classification Uncertainty
305	FIECE-EM+DVO	FIECE-EM + Distance to Violated Objects
306	FIECE-EM+LUC	FIECE-EM + Largest Unlabeled Clusters
307	ASCENT	—
308	FastCCP	Fast Constrained Clustering Problem
309	CDPS	Constrained DTW-Preserving Shapelets
310	conDetSEC	constrained Deep embedding time SEries Clustering
311	PCKM-Mono	Pairwise Constrained K-Means - Monotonic
312	CDGSC	Clustering with Dynamic Generation of Subordinate Clusters
313	CWLP	Clustering with Weak Label Prior
314	l_p Km	l_p K-Means
315	ACDEC	Active Constrained Deep Embedded Clustering

Acknowledgements This study has been funded by the research projects PID2020-119478GB-I00, A-TIC-434-UGR20 and PREDOC_01648.

Funding Funding for open access publishing: Universidad de Granada/CBUA.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Bishop CM, Nasrabadi NM (2006) Pattern Recognition and Machine Learning, vol 4. Springer, New York
- Chapelle O, Schölkopf B, Zien A (2010) Semi-supervised learning, 1st edn
- Van Engelen JE, Hoos HH (2020) A survey on semi-supervised learning. *Mach Learn* 109(2):373–440
- Miyamoto S, Terami A (2011) Inductive vs. transductive clustering using kernel functions and pairwise constraints. In: 2011 11th International Conference on Intelligent Systems Design and Applications, pp.1258–1264. IEEE
- Zhu XJ (2005) Semi-supervised learning literature survey
- Zhu X, Goldberg AB (2009) Introduction to semi-supervised learning. *Synth Lect Artif Intell Mach Learn* 3(1):1–130
- Zhou Z-H (2021) Semi-supervised learning. In: *Machine Learning*, pp 315–341
- Subramanya A, Talukdar PP (2014) Graph-based semi-supervised learning. *Synth Lect Artif Intell Mach Learn* 8(4):1–125
- Triguero I, García S, Herrera F (2015) Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study. *Knowl Inf Syst* 42(2):245–284
- Bair E (2013) Semi-supervised clustering methods. *Wiley Interdiscipl Rev Comput Stat* 5(5):349–361
- Davidson I, Basu S (2007) A survey of clustering with instance level constraints. *ACM Trans Knowl Discov Data* 1:1–41
- Basu S, Davidson I, Wagstaff K (2008) Constrained clustering: advances in algorithms, theory, and applications
- Grira N, Crucianu M, Boujemaa N (2004) Unsupervised and semi-supervised clustering: a brief survey. *Rev Mach Learn Techniq Process Multim Content* 1:9–16
- Dinler D, Tural MK (2016) A survey of constrained clustering. In: *Unsupervised learning algorithms*, pp 207–235. Springer, Berlin
- Gançarski P, Dao T-B-H, Crémilleux B, Forestier G, Lampert T (2020) Constrained clustering: current and new trends. *A Guided Tour of Artificial Intelligence Research: vol II: AI Algorithms*, pp 447–484
- Vrain C et al (2024) A review on declarative approaches for constrained clustering. *Int J Approx Reason* 109:135
- Harary F (1953) On the notion of balance of a signed graph. *Mich Math J* 2(2):143–146
- Wagstaff K, Cardie C (2000) Clustering with instance-level constraints. *AAAI/IAAI* 1097:577–584
- Wagstaff K, Cardie C (2000) Clustering with instance-level constraints. In: *Proceedings of the Seventeenth International Conference on Machine Learning*, Citeseer, pp 1103–1110
- Wagstaff K, Cardie C, Rogers S, Schrödl S (2001) Constrained k-means clustering with background knowledge. In: *Proceedings of the Eighteenth International Conference on Machine Learning*, pp 577–584. Morgan Kaufmann Publishers Inc., Burlington
- Basu S, Bilenko M, Mooney RJ (2003) Comparing and unifying search-based and similarity-based approaches to semi-supervised clustering. In: *Proceedings of the ICML-2003 workshop on the continuum from labeled to unlabeled data in machine learning and data mining*, Citeseer, pp 42–49
- Davidson I, Ravi S (2005) Agglomerative hierarchical clustering with constraints: theoretical and empirical results. In: *European Conference on Principles of Data Mining and Knowledge Discovery*, pp 59–70. Springer

- Xing EP, Ng AY, Jordan MI, Russell S (2002) Distance metric learning with application to clustering with side-information. *NIPS* 15:12
- Li F, Li S, Denœux T (2018) K-Cevclus: constrained evidential clustering of large dissimilarity data. *Knowl-Based Syst* 142:29–44
- Nie F, Zhang H, Wang R, Li X (2021) Semi-supervised clustering via pairwise constrained optimal graph. In: *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pp 3160–3166
- Zong L, Zhang X, Liu X (2018) Multi-view clustering on unmapped data via constrained non-negative matrix factorization. *Neural Netw* 108:155–171
- Denœux T (2021) NN-EVCLUS: neural network-based evidential clustering. *Inf Sci* 572:297–330
- Amirizadeh E, Boostani R (2021) CDEC: a constrained deep embedded clustering. *Int J Intell Comput Cyber*
- González-Almagro G, Rosales-Pérez A, Luengo J, Cano J-R, García S (2021) ME-MEOA/DCC: Multi-objective constrained clustering through decomposition-based memetic elitism. *Swarm Evol Comput* 66:100939
- González-Almagro G, Luengo J, Cano J-R, García S (2020) DILS: constrained clustering through dual iterative local search. *Comput Operat Res* 104979
- Liu H, Fu Y (2015) Clustering with partition level side information. In: *2015 IEEE International Conference on Data Mining*, pp 877–882. IEEE
- Śmieja M, Geiger BC (2017) Semi-supervised cross-entropy clustering with information bottleneck constraint. *Inf Sci* 421:254–271
- Basu S et al (2003) Semi-supervised clustering: learning with limited user feedback
- Liu H, Tao Z, Fu Y (2017) Partition level constrained clustering. *IEEE Trans Pattern Anal Mach Intell* 40(10):2469–2483
- Basu S, Banerjee A, Mooney R (2002) Semi-supervised clustering by seeding. In: *Proceedings of 19th International Conference on Machine Learning (ICML-2002)*. Citeseer
- Böhm C, Plant C (2008) HISSCLU: a hierarchical density-based method for semi-supervised clustering. In: *Proceedings of the 11th International Conference on Extending Database Technology: Advances in Database Technology*, pp 440–451
- Lelis L, Sander J (2009) Semi-supervised density-based clustering. In: *2009 Ninth IEEE International Conference on Data Mining*, pp 842–847. IEEE
- Liu H, Wu Z (2010) Non-negative matrix factorization with constraints. In: *Twenty-fourth AAAI Conference on Artificial Intelligence*
- Liu H, Wu Z, Li X, Cai D, Huang TS (2011) Constrained nonnegative matrix factorization for image representation. *IEEE Trans Pattern Anal Mach Intell* 34(7):1299–1311
- Marussy K, Buza K (2013) Success: a new approach for semi-supervised classification of time-series. In: *International Conference on Artificial Intelligence and Soft Computing*, pp 437–447. Springer
- Lan L, Guan N, Zhang X, Tao D, Luo Z (2014) Soft-constrained nonnegative matrix factorization via normalization. In: *2014 International Joint Conference on Neural Networks (IJCNN)*, pp 3025–3030. IEEE
- Trechalong K, Rakthanmanon T, Waiyamai K (2015) Semi-supervised stream clustering using labeled data points. In: *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pp. 281–295. Springer
- Choo J, Lee C, Reddy CK, Park H (2015) Weakly supervised nonnegative matrix factorization for user-driven clustering. *Data Min Knowl Disc* 29(6):1598–1621
- Vu V-V, Do H-Q (2017) Graph-based clustering with background knowledge. In: *Proceedings of the Eighth International Symposium on Information and Communication Technology*, pp 167–172
- Ienco D, Pensa RG (2018) Semi-supervised clustering with multiresolution autoencoders. In: *2018 International Joint Conference on Neural Networks (IJCNN)*, pp 1–8. IEEE
- Liu C-L, Hsiao W-H, Chang T-H, Li H-H (2019) Clustering data with partial background information. *Int J Mach Learn Cybern* 10(5):1123–1138
- Xie J, Antoine V (2019) On a new evidential c-means algorithm with instance-level constraints. In: *International Conference on Scalable Uncertainty Management*, pp 66–78. Springer
- Herrmann L, Ultsch A (2007) Label propagation for semi-supervised learning in self-organizing maps. In: *International Workshop on Self-Organizing Maps: Proceedings (2007)*
- Zhang H, Lu J (2009) Semi-supervised fuzzy clustering: a kernel-based approach. *Knowl-Based Syst* 22(6):477–481
- Yang Y, Liu X (2015) A robust semi-supervised learning approach via mixture of label information. *Pattern Recogn Lett* 68:15–21
- Antoine V, Gravouil K, Labroche N (2018) On evidential clustering with partial supervision. In: *International Conference on Belief Functions*, pp 14–21. Springer
- Gan H (2019) Safe semi-supervised fuzzy C-means clustering. *IEEE Access* 7:95659–95664

- Mei J-P (2018) Semisupervised fuzzy clustering with partition information of subsets. *IEEE Trans Fuzzy Syst* 27(9):1726–1737
- Antoine V, Guerrero JA, Xie J (2021) Fast semi-supervised evidential clustering. *Int J Approx Reason* 133:116–132
- Gondek D, Hofmann T (2007) Non-redundant data clustering. *Knowl Inf Syst* 12(1):1–24
- Yasunori E, Yukihiro H, Makito Y, Sadaaki M (2009) On semi-supervised fuzzy c-means clustering. In: 2009 IEEE International Conference on Fuzzy Systems, pp 1119–1124. IEEE
- Yin X, Shu T, Huang Q (2012) Semi-supervised fuzzy clustering with metric learning and entropy regularization. *Knowl-Based Syst* 35:304–311
- Zeng S, Tong X, Sang N, Huang R (2013) A study on semi-supervised FCM algorithm. *Knowl Inf Syst* 35(3):585–612
- Nghiem N-V-D, Vrain C, Dao T-B-H, Davidson I (2020) Constrained clustering via post-processing. In: International Conference on Discovery Science, pp 53–67. Springer
- Zhang H, Basu S, Davidson I (2019) A framework for deep constrained clustering-algorithms and advances. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp 57–72. Springer
- Mueller M, Kramer S (2010) Integer linear programming models for constrained clustering. In: International Conference on Discovery Science, pp 159–173. Springer
- Ares ME, Parapar J, Barreiro Á (2009) Avoiding bias in text clustering using constrained k-means and may-not-links. In: Conference on the Theory of Information Retrieval, pp 322–329. Springer
- Diaz-Valenzuela I, Vila MA, Martin-Bautista MJ (2015) On the use of fuzzy constraints in semisupervised clustering. *IEEE Trans Fuzzy Syst* 24(4):992–999
- Jiang H, Ren Z, Xuan J, Wu X (2013) Extracting elite pairwise constraints for clustering. *Neurocomputing* 99:124–133
- Xiao Y, Liu B, Hao Z (2015) A maximum margin approach for semisupervised ordinal regression clustering. *IEEE Transactions on Neural Networks and Learning Systems* 27(5):1003–1019
- Qian P, Jiang Y, Wang S, Su K-H, Wang J, Hu L, Muzic RF (2016) Affinity and penalty jointly constrained spectral clustering with all-compatibility, flexibility, and robustness. *IEEE Transact Neural Netw Learn Syst* 28(5):1123–1138
- Yu SX, Shi J (2004) Segmentation given partial grouping constraints. *IEEE Trans Pattern Anal Mach Intell* 26(2):173–183
- Law MH, Topchy A, Jain AK (2004) Clustering with soft and group constraints. In: Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR), pp 662–670. Springer
- Pei Y, Fern XZ, Tjahja TV, Rosales R (2016) Comparing clustering with pairwise and relative constraints: a unified framework. *ACM Transact Knowl Discover Data (TKDD)* 11(2):1–26
- Liu EY, Zhang Z, Wang W (2011) Clustering with relative constraints. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 947–955
- Amid E, Gionis A, Ukkonen A (2015) A kernel-learning approach to semi-supervised clustering with relative distance comparisons. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp 219–234. Springer
- Ienco D, Pensa RG (2019) Deep triplet-driven semi-supervised embedding clustering. In: International Conference on Discovery Science, pp 220–234. Springer
- Bade K, Nürnberger A (2014) Hierarchical constraints. *Machine learning* 94(3):371–399
- Endo Y, Hamasuna Y (2011) Fuzzy c-means clustering with mutual relation constraints. In: International Conference on Knowledge-Based and Intelligent Information and Engineering Systems, pp 131–140. Springer
- Gallegos MT, Ritter G (2010) Using combinatorial optimization in model-based trimmed clustering with cardinality constraints. *Comput Stat Data Anal* 54(3):637–654
- Zhu S, Wang D, Li T (2010) Data clustering with size constraints. *Knowl-Based Syst* 23(8):883–889
- Tang W, Yang Y, Zeng L, Zhan Y (2019) Size constrained clustering with MILP formulation. *IEEE Access* 8:1587–1599
- Bibi A, Alqahtani A, Ghanem B (2023) Constrained clustering: general pairwise and cardinality constraints. *IEEE Access* 11:5824–5836
- Babaki B, Guns T, Nijssen S (2014) Constrained clustering using column generation. In: International Conference on Integration of Constraint Programming, Artificial Intelligence, and Operations Research, pp 438–454. Springer
- Dao T-B-H, Duong K-C, Vrain C (2013) A declarative framework for constrained clustering. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp 419–434. Springer
- Bradley PS, Bennett KP, Demiriz A (2000) Constrained k-means clustering. Technical report, MSR-TR-2000-65, Microsoft Research

- Lei K, Wang S, Song W, Li Q (2013) Size-constrained clustering using an initial points selection method. In: International Conference on Knowledge Science, Engineering and Management, pp 195–205. Springer
- Rebollo-Monedero D, Solé M, Nin J, Forné J (2013) A modification of the k-means method for quasi-supervised learning. *Knowl-Based Syst* 37:176–185
- Ge R, Ester M, Jin W, Davidson I (2007) Constraint-driven clustering. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 320–329
- Zhong S, Ghosh J (2003) A unified framework for model-based clustering. *J Mach Learn Res* 4:1001–1037
- Banerjee A, Ghosh J (2006) Scalable clustering algorithms with balancing constraints. *Data Min Knowl Disc* 13(3):365–395
- Tang W, Yang Y, Zeng L, Zhan Y (2019) Optimizing MSE for clustering with balanced size constraints. *Symmetry* 11(3):338
- Sun J, Zhao W, Xue J, Shen Z, Shen Y (2010) Clustering with feature order preferences. *Intell Data Anal* 14(4):479–495
- Davidson I, Ravi S, Shamis L (2010) A sat-based framework for efficient constrained clustering. In: Proceedings of the 2010 SIAM International Conference on Data Mining, pp 94–105. SIAM
- Pei Y, Fern XZ (2014) Constrained instance clustering in multi-instance multi-label learning. *Pattern Recogn Lett* 37:107–114
- Waqas M, Tahir MA, Qureshi R (2023) Deep gaussian mixture model based instance relevance estimation for multiple instance learning applications. *Appl Intell* 53(9):10310–10325
- Waqas M, Tahir MA, Khan SA (2023) Robust bag classification approach for multi-instance learning via subspace fuzzy clustering. *Expert Syst Appl* 214:119113
- Waqas M, Tahir MA, Qureshi R (2021) Ensemble-based instance relevance estimation in multiple-instance learning. In: 2021 9th European Workshop on Visual Information Processing (EUVIP), pp 1–6. IEEE
- Carboneau M-A, Granger E, Raymond AJ, Gagnon G (2016) Robust multiple-instance learning ensembles using random subspace instance selection. *Pattern Recogn* 58:83–99
- Hu P, Vens C, Verstrynghe B, Blockeel H (2013) Generalizing from example clusters. In: International Conference on Discovery Science, pp 64–78. Springer
- Vens C, Verstrynghe B, Blockeel H (2013) Semi-supervised clustering with example clusters. In: Proceedings of the 5th International Conference on Knowledge Discovery and Information Retrieval and the 5th International Conference on Knowledge Management and Information Sharing, pp 45–51
- Ben Ahmed E, Nabli A, Gargouri F (2012) Shacun: Semi-supervised hierarchical active clustering based on ranking constraints. In: Industrial Conference on Data Mining, pp 194–208. Springer
- Ben Ahmed E, Nabli A, Gargouri F (2012) Towards quantitative constraints ranking in data clustering. In: International Conference on Database and Expert Systems Applications, pp 121–128. Springer
- Nogueira BM, Tomas YKB, Marcacini RM (2017) Integrating distance metric learning and cluster-level constraints in semi-supervised clustering. In: 2017 International Joint Conference on Neural Networks (IJCNN), pp 4118–4125. IEEE
- Jain AK, Murty MN, Flynn PJ (1999) Data clustering: a review. *ACM Comput Surv (CSUR)* 31(3):264–323
- Davidson I, Wagstaff KL, 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. Springer
- Davidson I, Ravi S (2005) Clustering with constraints: Feasibility issues and the k-means algorithm. In: Proceedings of the 2005 International Conference on Data Mining, pp 138–149. SIAM
- Davidson I, Ravi SS (2009) Using instance-level constraints in agglomerative hierarchical clustering: theoretical and empirical results. *Data Min Knowl Disc* 18(2):257–282
- Huang R, Lam W (2007) Semi-supervised document clustering via active learning with pairwise constraints. In: Seventh IEEE International Conference on Data Mining (ICDM 2007), pp 517–522. IEEE
- Huang R, Lam W (2009) An active learning framework for semi-supervised document clustering with language modeling. *Data Knowl Eng* 68(1):49–67
- Ma H, Zhao W, Tan Q, Shi Z (2010) Orthogonal nonnegative matrix tri-factorization for semi-supervised document co-clustering. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp 189–200. Springer
- Yan Y, Chen L, Tjhi W-C (2013) Fuzzy semi-supervised co-clustering for text documents. *Fuzzy Sets Syst* 215:74–89
- Liu C-L, Hsiao W-H, Lee C-H, Chen C-H (2013) Clustering tagged documents with labeled and unlabeled documents. *Inform Process Manag* 49(3):596–606
- Ma H, Zhao W, Shi Z (2013) A nonnegative matrix factorization framework for semi-supervised document clustering with dual constraints. *Knowl Inf Syst* 36(3):629–651
- Sutanto T, Nayak R (2014) The ranking based constrained document clustering method and its application to social event detection. In: International Conference on Database Systems for Advanced Applications, pp 47–60. Springer

- Hu Y, Milios EE, Blustein J (2016) Document clustering with dual supervision through feature reweighting. *Comput Intell* 32(3):480–513
- Díaz-Valenzuela I, Loia V, Martín-Bautista MJ, Senatore S, Vila MA (2016) Automatic constraints generation for semisupervised clustering: experiences with documents classification. *Soft Comput* 20(6):2329–2339
- Trabelsi A, Zaïane OR (2016) Mining contentious documents. *Knowl Inf Syst* 48(3):537–560
- Balafar M, Hazratgholizadeh R, Derakhshi M (2020) Active learning for constrained document clustering with uncertainty region. *Complexity* 2020
- Buatoom U, Kongprawechanon W, Theeramunkong T (2020) Document clustering using k-means with term weighting as similarity-based constraints. *Symmetry* 12(6):967
- Van Craenendonck T, Dumančić S, Blockeel H (2017) Cobra: a fast and simple method for active clustering with pairwise constraints. In: *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pp 2871–2877
- Hu G, Zhou S, Guan J, Hu X (2008) Towards effective document clustering: a constrained k-means based approach. *Inform Process Manag* 44(4):1397–1409
- Chen Y, Wang L, Dong M (2009) Semi-supervised document clustering with simultaneous text representation and categorization. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp 211–226. Springer
- Zhao W, He Q, Ma H, Shi Z (2012) Effective semi-supervised document clustering via active learning with instance-level constraints. *Knowl Inf Syst* 30(3):569–587
- Yan Y, Chen L, Tjhi W-C (2013) Semi-supervised fuzzy co-clustering algorithm for document categorization. *Knowl Inf Syst* 34(1):55–74
- Gu J, Feng W, Zeng J, Mamitsuka H, Zhu S (2012) Efficient semisupervised medline document clustering with mesh-semantic and global-content constraints. *IEEE Transact Cyber* 43(4):1265–1276
- Parapar J, Barreiro Á (2012) Language modelling of constraints for text clustering. In: *European Conference on Information Retrieval*, pp 352–363. Springer
- Song Y, Pan S, Liu S, Wei F, Zhou MX, Qian W (2012) Constrained text coclustering with supervised and unsupervised constraints. *IEEE Trans Knowl Data Eng* 25(6):1227–1239
- Zhu Y, Jing L, Yu J (2011) Text clustering via constrained nonnegative matrix factorization. In: *2011 IEEE 11th International Conference on Data Mining*, pp 1278–1283. IEEE
- Vlachos A, Korhonen A, Ghahramani Z (2009) Unsupervised and constrained dirichlet process mixture models for verb clustering. In: *Proceedings of the Workshop on Geometrical Models of Natural Language Semantics*, pp 74–82
- Sugiyama K, Okumura M (2009) Semi-supervised clustering for word instances and its effect on word sense disambiguation. In: *International Conference on Intelligent Text Processing and Computational Linguistics*, pp 266–279. Springer
- Ma H, Jia M, Shi Y, Hao Z (2014) Semi-supervised nonnegative matrix factorization for microblog clustering based on term correlation. In: *Asia-Pacific Web Conference*, pp 511–516. Springer
- Ma H, Jia M, Zhao W, Lin X (2015) Semi-supervised microblog clustering method via dual constraints. In: *International Conference on Knowledge Science, Engineering and Management*, pp 360–369. Springer
- Ma H, Zhang D, Jia M, Lin X (2019) A term correlation based semi-supervised microblog clustering with dual constraints. *Int J Mach Learn Cybern* 10(4):679–692
- Tang N, Vemuri VR (2005) User-interest-based document filtering via semi-supervised clustering. In: *International Symposium on Methodologies for Intelligent Systems*, pp 573–582. Springer
- Duan C, Dumitru H, Cleland-Huang J, Mobasher B (2015) User-constrained clustering in online requirements forums. In: *International Working Conference on Requirements Engineering: Foundation for Software Quality*, pp 284–299. Springer
- Tseng VS, Chen L-C, Kao C-P (2008) Constrained clustering for gene expression data mining. In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp 759–766. Springer
- Maraziotis IA (2012) A semi-supervised fuzzy clustering algorithm applied to gene expression data. *Pattern Recogn* 45(1):637–648
- Saha S, Alok AK, Ekbal A (2015) Use of semisupervised clustering and feature-selection techniques for identification of co-expressed genes. *IEEE J Biomed Health Inform* 20(4):1171–1177
- Maraziotis IA, Dragomir A, Bezerianos A (2006) Semi supervised fuzzy clustering networks for constrained analysis of time-series gene expression data. In: *International Conference on Artificial Neural Networks*, pp 818–826. Springer
- Yu Z, Chen H, You J, Wong H-S, Liu J, Li L, Han G (2014) Double selection based semi-supervised clustering ensemble for tumor clustering from gene expression profiles. *IEEE/ACM Trans Comput Biol Bioinf* 11(4):727–740
- Ceccarelli M, Maratea A (2005) Semi-supervised fuzzy c-means clustering of biological data. In: *International Workshop on Fuzzy Logic and Applications*, pp 259–266. Springer

- Zeng E, Yang C, Li T, Narasimhan G (2007) On the effectiveness of constraints sets in clustering genes. In: 2007 IEEE 7th International Symposium on BioInformatics and BioEngineering, pp 79–86. IEEE
- Tian T, Zhang J, Lin X, Wei Z, Hakonarson H (2021) Model-based deep embedding for constrained clustering analysis of single cell RNA-seq data. *Nat Commun* 12(1):1–12
- Mishra A, Gillies D (2008) Semi supervised spectral clustering for regulatory module discovery. In: International Workshop on Data Integration in the Life Sciences, pp 192–203. Springer
- Abdala DD, Jiang X (2010) Fiber segmentation using constrained clustering. In: International Conference on Medical Biometrics, pp 1–10. Springer
- Yu Z, Wongb H-S, You J, Yang Q, Liao H (2011) Knowledge based cluster ensemble for cancer discovery from biomolecular data. *IEEE Trans Nanobiosci* 10(2):76–85
- Verma H, Agrawal R, Sharan A (2016) An improved intuitionistic fuzzy c-means clustering algorithm incorporating local information for brain image segmentation. *Appl Soft Comput* 46:543–557
- Veras R, Aires K, Britto L, et al (2018) Medical image segmentation using seeded fuzzy c-means: A semi-supervised clustering algorithm. In: 2018 International Joint Conference on Neural Networks (IJCNN), pp 1–7. IEEE
- Wang Z, Yang Y (2018) A non-iterative clustering based soft segmentation approach for a class of fuzzy images. *Appl Soft Comput* 70:988–999
- Guo J, Cheong L-F, Tan RT, Zhou SZ (2014) Consistent foreground co-segmentation. In: Asian Conference on Computer Vision, pp 241–257. Springer
- Li Y, Liu Y, Cui B, Sun C, Ji X, Zhang J, Li B, Chen H, Zhang J, Wang Y, et al (2020) Spatial constrained k-means for image segmentation. In: Proceedings of SAI Intelligent Systems Conference, pp 662–672. Springer
- Biswas A, Jacobs D (2011) Large scale image clustering with active pairwise constraints. In: International Conference in Machine Learning 2011 Workshop on Combining Learning Strategies to Reduce Label Cost, 2. Citeseer
- Hoi SC, Liu W, Chang S-F (2010) Semi-supervised distance metric learning for collaborative image retrieval and clustering. *ACM Transact Multim Comput Commun Appl (TOMM)* 6(3):1–26
- Biswas A, Jacobs D (2014) Active image clustering with pairwise constraints from humans. *Int J Comput Vision* 108(1):133–147
- Grira N, Crucianu M, Boujmaa N (2006) Fuzzy clustering with pairwise constraints for knowledge-driven image categorisation. *IEE Proceed Vision Image Signal Process* 153(3):299–304
- Frigui H, Meredith J (2008) Image database categorization under spatial constraints using adaptive constrained clustering. In: 2008 IEEE International Conference on Fuzzy Systems (IEEE World Congress on Computational Intelligence), pp 2268–2276. IEEE
- ISMAL MMB, Bchir O, (2012) Automatic image annotation based on semi-supervised clustering and membership-based cross media relevance model. *Int J Pattern Recognit Artif Intell* 26(06):1255009
- Xiaoguang R, Pingbo Y, Nenghai Y (2006) Image annotations based on semi-supervised clustering with semantic soft constraints. In: Pacific-Rim Conference on Multimedia, pp 624–632. Springer
- Lai HP, Visani M, Boucher A, Ogier J-M (2014) A new interactive semi-supervised clustering model for large image database indexing. *Pattern Recogn Lett* 37:94–106
- Bui T-H, Park S-B (2017) Point of interest mining with proper semantic annotation. *Multim Tools Appl* 76(22):23435–23457
- Li T, Prieta Pintado F, Corchado JM, Bajo J (2017) Multi-source homogeneous data clustering for multi-target detection from cluttered background with misdetection. *Appl Soft Comput* 60:436–446
- Vu V-V, Do H-Q, Dang V-T, Do N-T (2019) An efficient density-based clustering with side information and active learning: a case study for facial expression recognition task. *Intell Data Anal* 23(1):227–240
- Lafabregue B, Weber J, Gançarski P, Forestier G (2019) Deep constrained clustering applied to satellite image time series. In: ECML/PKDD Workshop on Machine Learning for Earth Observation Data (MACLEAN), Würzburg, Germany. <https://hal.archives-ouvertes.fr/hal-02326219>
- Niebles JC, Han B, Ferencz A, Fei-Fei L (2008) Extracting moving people from internet videos. In: European Conference on Computer Vision, pp 527–540. Springer
- Mahmood A, Li T, Yang Y, Wang H (2013) Semi-supervised clustering ensemble evolved by genetic algorithm for web video categorization. In: International Conference on Advanced Data Mining and Applications, pp 1–12. Springer
- Wu B, Zhang Y, Hu B-G, Ji Q (2013) Constrained clustering and its application to face clustering in videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 3507–3514
- Wu B, Hu B-G, Ji Q (2017) A coupled hidden Markov random field model for simultaneous face clustering and tracking in videos. *Pattern Recogn* 64:361–373
- Wu B, Lyu S, Hu B-G, Ji Q (2013) Simultaneous clustering and tracklet linking for multi-face tracking in videos. In: Proceedings of the IEEE International Conference on Computer Vision, pp 2856–2863

- Xie Y, Liu M, Yao Z, Dai B (2006) Improved two-stage wiener filter for robust speaker identification. In: 18th International Conference on Pattern Recognition (ICPR'06), vol. 4, pp 310–313. IEEE
- Duan C, Cleland-Huang J, Mobasher B (2008) A consensus based approach to constrained clustering of software requirements. In: Proceedings of the 17th ACM Conference on Information and Knowledge Management, pp 1073–1082
- Li N, Li Y, Liu H, Liu C (2010) An evolving machinery fault diagnosis approach based on affinity propagation algorithm. In: International Conference on Intelligent Robotics and Applications, pp 654–664. Springer
- Han L, Luo S, Wang H, Pan L, Ma X, Zhang T (2016) An intelligible risk stratification model based on pairwise and size constrained K-means. *IEEE J Biomed Health Inform* 21(5):1288–1296
- Zhang J, Chang D (2019) Semi-supervised patient similarity clustering algorithm based on electronic medical records. *IEEE Access* 7:90705–90714
- Chang H-C, Tsai H-P (2011) Group RFM analysis as a novel framework to discover better customer consumption behavior. *Expert Syst Appl* 38(12):14499–14513
- Yu S-m, Du Z-j, Zhang X, Luo H, Lin X (2021) Trust cop-kmeans clustering analysis and minimum-cost consensus model considering voluntary trust loss in social network large-scale decision-making. *IEEE Transactions on Fuzzy Systems*
- Seret A, Verbraken T, Baesens B (2014) A new knowledge-based constrained clustering approach: theory and application in direct marketing. *Appl Soft Comput* 24:316–327
- Akaichi I, Wislez P (2021) Pairwise constrained clustering and robust regression: A case study on french enterprise activities and expenses data. In: 2021 International Conference of Women in Data Science at Taif University (WiDSTaif), pp 1–7. IEEE
- Ben Ahmed E, Nabli A, Gargouri F (2014) Group extraction from professional social network using a new semi-supervised hierarchical clustering. *Knowl Inf Syst* 40(1):29–47
- Shah GA, Alagöz F, Fadel EA, Akan OB (2014) A spectrum-aware clustering for efficient multimedia routing in cognitive radio sensor networks. *IEEE Trans Veh Technol* 63(7):3369–3380
- Joshi D, Soh L-K, Samal A (2011) Redistricting using constrained polygonal clustering. *IEEE Trans Knowl Data Eng* 24(11):2065–2079
- Song J, Xing H, Zhang H, Xu Y, Meng Y (2021) An adaptive network-constrained clustering (ANCC) model for fine-scale urban functional zones. *IEEE Access* 9:53013–53029
- Araujo R, Kamel MS (2014) Audio-visual emotion analysis using semi-supervised temporal clustering with constraint propagation. In: International Conference Image Analysis and Recognition, pp 3–11. Springer
- Xiong S, Ji D (2015) Exploiting capacity-constrained k-means clustering for aspect-pharse grouping. In: International Conference on Knowledge Science, Engineering and Management, pp 370–381. Springer
- Tirkaz C, Yanikoglu B, Sezgin TM (2012) Sketched symbol recognition with auto-completion. *Pattern Recogn* 45(11):3926–3937
- Semnani SH, Basir OA, Van Beek P (2016) Constrained clustering for flocking-based tracking in maneuvering target environment. *Robot Auton Syst* 83:243–250
- Saidi F, Trabelsi Z, Ghazela HB (2018) A novel approach for terrorist sub-communities detection based on constrained evidential clustering. In: 2018 12th International Conference on Research Challenges in Information Science (RCIS), pp 1–8. IEEE
- Zhao Y, Gai Z (2022) Transformation and optimization of rural ecological endowment industry chain based on constrained clustering algorithm. *Scientific Programming*, 2022
- El-Kholany M, Schekotihin K, Gebser M (2022) Decomposition-based job-shop scheduling with constrained clustering. In: International Symposium on Practical Aspects of Declarative Languages, pp 165–180. Springer
- De Koninck P, Nlissen K, Baesens B, Snoeck M, De Weerd J et al (2021) Expert-driven trace clustering with instance-level constraints. *Knowl Inf Syst* 63(5):1197–1220
- Zhang Y, Melnykov V, Melnykov I (2021) Semi-supervised clustering of time-dependent categorical sequences with application to discovering education-based life patterns. *Statistical Modelling*, 1471082–21989170
- Boesen T, Haber E, Hoversten GM (2021) Data-driven semi-supervised clustering for oil prediction. *Comput Geosci* 148:104684
- Zhang Y, Zhao S, Sang Y (2019) Towards unknown traffic identification using deep auto-encoder and constrained clustering. In: International Conference on Computational Science, pp 309–322. Springer
- Malzer C, Baum M (2021) Constraint-based hierarchical cluster selection in automotive radar data. *Sensors* 21(10):3410
- Ben Ayed S, Elouedi Z, Lefevre E (2019) Cevm: constrained evidential vocabulary maintenance policy for cbr systems. In: International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, pp 579–592. Springer

- Mai ST, Amer-Yahia S, Bailly S, Pépin J-L, Chouakria AD, Nguyen KT, Nguyen A-D (2018) Evolutionary active constrained clustering for obstructive sleep apnea analysis. *Data Sci Eng* 3(4):359–378
- Wang Y, Xiang Y, Zhang J, Zhou W, Wei G, Yang LT (2013) Internet traffic classification using constrained clustering. *IEEE Trans Parallel Distrib Syst* 25(11):2932–2943
- Yoshida T (2010) Performance evaluation of constraints in graph-based semi-supervised clustering. In: *International Conference on Active Media Technology*, pp 138–149. Springer
- Bar-Hillel A, Hertz T, Shental N, Weinshall D (2003) Learning distance functions using equivalence relations. In: *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pp 11–18
- Bar-Hillel A, Hertz T, Shental N, Weinshall D, Ridgeway G (2005) Learning a mahalanobis metric from equivalence constraints. *J Mach Learn Res* 6(6)
- Cheng H, Hua KA, Vu K (2008) Constrained locally weighted clustering. *Proceed VLDB Endow* 1(1):90–101
- Basu S, Banerjee A, Mooney RJ (2004) Active semi-supervision for pairwise constrained clustering. In: *Proceedings of the 2004 SIAM International Conference on Data Mining*, pp 333–344. SIAM
- González-Almagro G, Suárez JL, Luengo J, Cano J-R, García S (2021) 3shacc: Three stages hybrid agglomerative constrained clustering. *Neurocomputing*
- Vu V-V, Labroche N, Bouchon-Meunier B (2012) Improving constrained clustering with active query selection. *Pattern Recogn* 45(4):1749–1758
- El Amouri H, Lampert T, Gañarski P, Mallet C (2023) Constrained DTW preserving shapelets for explainable time-series clustering. *Pattern Recogn* 143:109804
- He P, Xu X, Hu K, Chen L (2014) Semi-supervised clustering via multi-level random walk. *Pattern Recogn* 47(2):820–832
- Li Y, Wang Y, Yu D-J, Ye N, Hu P, Zhao R (2019) Ascent: active supervision for semi-supervised learning. *IEEE Trans Knowl Data Eng* 32(5):868–882
- Basu S, Bilenko M, Mooney RJ (2004) A probabilistic framework for semi-supervised clustering. In: *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp 59–68
- Dom BE (2002) An information-theoretic external cluster-validity measure. In: *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence*, pp 137–145
- Hu H, Feng J, Zhou J (2014) Exploiting unsupervised and supervised constraints for subspace clustering. *IEEE Trans Pattern Anal Mach Intell* 37(8):1542–1557
- Li Z, Liu J, Tang X (2009) Constrained clustering via spectral regularization. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp 421–428. IEEE
- Li Z, Liu J (2009) Constrained clustering by spectral kernel learning. In: *2009 IEEE 12th International Conference on Computer Vision*, pp 421–427. IEEE
- Kuhn HW (1955) The Hungarian method for the assignment problem. *Naval Res Logist Q* 2(1–2):83–97
- Rand WM (1971) Objective criteria for the evaluation of clustering methods. *J Am Stat Assoc* 66(336):846–850
- Zhong G, Deng X, Xu S (2019) Active informative pairwise constraint formulation algorithm for constraint-based clustering. *IEEE Access* 7:81983–81993
- Hubert L, Arabie P (1985) Comparing partitions. *J Classif* 2(1):193–218
- Ding S, Jia H, Du M, Xue Y (2018) A semi-supervised approximate spectral clustering algorithm based on HMRF model. *Inf Sci* 429:215–228
- Hripesak G, Rothschild AS (2005) Agreement, the F-measure, and reliability in information retrieval. *J Am Med Inform Assoc* 12(3):296–298
- Zhao Q, Miller DJ (2005) Mixture modeling with pairwise, instance-level class constraints. *Neural Comput* 17(11):2482–2507
- Halkidi M, Spiliopoulou M, Pavlou A (2012) A semi-supervised incremental clustering algorithm for streaming data. In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp 578–590. Springer
- Xu X, Pan Z, He P, Chen L (2011) Constrained clustering via swarm intelligence. In: *International Conference on Intelligent Computing*, pp 404–409. Springer
- Bae E, Bailey J (2006) Coala: A novel approach for the extraction of an alternate clustering of high quality and high dissimilarity. In: *Sixth International Conference on Data Mining (ICDM'06)*, pp 53–62. IEEE
- Meilă M (2003) Comparing clusterings by the variation of information. In: *Learning Theory and Kernel Machines*, pp 173–187. Springer
- Halkidi M, Batistakis Y, Vazirgiannis M (2001) On clustering validation techniques. *J Intell Inform Syst* 17(2):107–145
- Rosenberg A, Hirschberg J (2007) V-measure: A conditional entropy-based external cluster evaluation measure. In: *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pp 410–420
- Vlachos A, Ghahramani Z, Korhonen A (2008) Dirichlet process mixture models for verb clustering. In: *Proceedings of the ICML Workshop on Prior Knowledge for Text and Language*

- Derrac J, García S, Molina D, Herrera F (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm Evol Comput* 1(1):3–18
- Benavoli A, Corani G, Demšar J, Zaffalon M (2017) Time for a change: a tutorial for comparing multiple classifiers through Bayesian analysis. *J Mach Learn Res* 18(1):2653–2688
- Tang W, Xiong H, Zhong S, Wu J (2007) Enhancing semi-supervised clustering: a feature projection perspective. In: *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp 707–716
- Domeniconi C, Al-Razgan M (2008) Penta-training: Clustering ensembles with bootstrapping of constraints. In: *Workshop on Supervised and Unsupervised Ensemble Methods and Their Applications*, p 47 (2008)
- Huang H, Cheng Y, Zhao R (2008) A semi-supervised clustering algorithm based on must-link set. In: *International Conference on Advanced Data Mining and Applications*, pp 492–499. Springer
- Rothaus K, Jiang X (2008) Constrained clustering by a novel graph-based distance transformation. In: *2008 19th International Conference on Pattern Recognition*, pp 1–4. IEEE
- Zhang S, Wong H-S (2008) Partial closure-based constrained clustering with order ranking. In: *2008 19th International Conference on Pattern Recognition*, pp 1–4. IEEE
- Tuzel O, Porikli F, Meer P (2009) Kernel methods for weakly supervised mean shift clustering. In: *2009 IEEE 12th International Conference on Computer Vision*, pp 48–55. IEEE
- Yin X, Chen S, Hu E, Zhang D (2010) Semi-supervised clustering with metric learning: an adaptive kernel method. *Pattern Recogn* 43(4):1320–1333
- TAN W, YANG Y, LI T (2010) An improved cop-kmeans algorithm for solving constraint violation. In: *Computational Intelligence: Foundations and Applications*, pp 690–696. World Scientific
- Rutayisire T, Yang Y, Lin C, Zhang J (2011) A modified cop-kmeans algorithm based on sequenced cannot-link set. In: *International Conference on Rough Sets and Knowledge Technology*, pp 217–225. Springer
- Anand S, Mittal S, Tuzel O, Meer P (2013) Semi-supervised kernel mean shift clustering. *IEEE Trans Pattern Anal Mach Intell* 36(6):1201–1215
- Le HM, Eriksson A, Do T-T, Milford M (2018) A binary optimization approach for constrained k-means clustering. In: *Asian Conference on Computer Vision*, pp 383–398. Springer
- Randel R, Aloise D, Mladenović N, Hansen P (2018) On the k-medoids model for semi-supervised clustering. In: *International Conference on Variable Neighborhood Search*, pp 13–27. Springer
- Wagstaff KL (2002) *Intelligent Clustering with Instance-level Constraints*. Cornell University USA
- Bilenko M, Basu S, Mooney RJ (2004) Integrating constraints and metric learning in semi-supervised clustering. In: *Proceedings of the Twenty-first International Conference on Machine Learning*, p 11
- Eaton ER (2005) *Clustering with propagated constraints*. PhD thesis, University of Maryland, Baltimore County
- Pelleg D, Baras D (2007) K-means with large and noisy constraint sets. In: *European Conference on Machine Learning*, pp 674–682. Springer
- Frigui H, Mahdi R (2007) Semi-supervised clustering and feature discrimination with instance-level constraints. In: *2007 IEEE International Fuzzy Systems Conference*, pp 1–6. IEEE
- Rizoiu M-A, Velcin J, Lallich S (2014) How to use temporal-driven constrained clustering to detect typical evolutions. *Int J Artif Intell Tools* 23(04):1460013
- Zhang S, Wong H-S, Xie D (2014) Semi-supervised clustering with pairwise and size constraints. In: *2014 International Joint Conference on Neural Networks (IJCNN)*, pp 2450–2457. IEEE
- Endo Y, Kinoshita N, Iwakura K, Hamasuna Y (2014) Hard and fuzzy c-means algorithms with pairwise constraints by non-metric terms. In: *International Conference on Modeling Decisions for Artificial Intelligence*, pp 145–157. Springer
- Wei S, Li Z, Zhang C (2015) A semi-supervised clustering ensemble approach integrated constraint-based and metric-based. In: *Proceedings of the 7th International Conference on Internet Multimedia Computing and Service*, pp 1–6
- Wei S, Li Z, Zhang C (2018) Combined constraint-based with metric-based in semi-supervised clustering ensemble. *Int J Mach Learn Cybern* 9(7):1085–1100
- Mai ST, Amer-Yahia S, Chouakria AD, Nguyen KT, Nguyen A-D (2018) Scalable active constrained clustering for temporal data. In: *International Conference on Database Systems for Advanced Applications*, pp 566–582. (2018). Springer
- Vouras A, Vasilaki E (2021) A semi-supervised sparse k-means algorithm. *Pattern Recogn Lett* 142:65–71
- He G, Pan Y, Xia X, He J, Peng R, Xiong NN (2019) A fast semi-supervised clustering framework for large-scale time series data. *IEEE Transact Syst Man Cybern Syst* 51(7):4201–4216
- Shimizu S, Sakayauchi S, Shibata H, Takama Y (2024) Dynamic generation of subordinate clusters based on Bayesian information criterion for must-link constrained k-means. *J Adv Comput Intell Intell Inform* 28(1):86–93

- González-Almagro G, Sánchez-Bermejo P, Suarez JL, Cano J-R, García S (2024) Semi-supervised clustering with two types of background knowledge: fusing pairwise constraints and monotonicity constraints. *Inform Fusion* 102:102064
- Yang L, Cao X, Jin D, Wang X, Meng D (2014) A unified semi-supervised community detection framework using latent space graph regularization. *IEEE Transact Cybern* 45(11):2585–2598
- Domeniconi C, Peng J, Yan B (2011) Composite kernels for semi-supervised clustering. *Knowl Inf Syst* 28(1):99–116
- Elghazel H, Benabdeslem K, Dussauchoy A (2007) Constrained graph b-coloring based clustering approach. In: *International Conference on Data Warehousing and Knowledge Discovery*, pp 262–271. Springer
- Coleman T, Saunderson J, Wirth A (2008) A local-search 2-approximation for 2-correlation-clustering. In: *European Symposium on Algorithms*, pp 308–319. Springer
- Yoshida T, Okatani K (2010) A graph-based projection approach for semi-supervised clustering. In: *Pacific Rim Knowledge Acquisition Workshop*, pp 1–13 (2010). Springer
- Yoshida T (2011) Pairwise constraint propagation for graph-based semi-supervised clustering. In: *International Symposium on Methodologies for Intelligent Systems*, pp 358–364. Springer
- Yoshida T (2014) A graph-based approach for semisupervised clustering. *Comput Intell* 30(2):263–284
- Anand R, Reddy CK (2011) Graph-based clustering with constraints. In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp 51–62. Springer
- Kamvar K, Sepandar S, Klein K, Dan D, Manning M, Christopher C (2003) Spectral learning. In: *International Joint Conference of Artificial Intelligence*. Stanford InfoLab
- Bie TD, Suykens J, Moor BD (2004) Learning from general label constraints. In: *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pp 671–679. Springer
- Xu Q, Wagstaff KL, et al (2005) Active constrained clustering by examining spectral eigenvectors. In: *International Conference on Discovery Science*, pp 294–307. Springer
- Xu Q, Desjardins M, Wagstaff K (2005) Constrained spectral clustering under a local proximity structure assumption. In: *FLAIRS Conference*. Citeseer
- Lu Z, Carreira-Perpinan MA (2008) Constrained spectral clustering through affinity propagation. In: *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pp 1–8. IEEE
- Alzate C, Suykens JA (2009) A regularized formulation for spectral clustering with pairwise constraints. In: *2009 International Joint Conference on Neural Networks*, pp 141–148. IEEE
- Xu L, Li W, Schuurmans D (2009) Fast normalized cut with linear constraints. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp 2866–2873. IEEE
- Wang X, Davidson I (2010) Flexible constrained spectral clustering. In: *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp 563–572
- Wang X, Davidson I (2010) Active spectral clustering. In: *2010 IEEE International Conference on Data Mining*, pp 561–568. IEEE
- Shang F, Jiao L, Liu Y, Wang F (2012) Learning spectral embedding via iterative eigenvalue thresholding. In: *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, pp 1507–1511
- Wauthier FL, Jojic N, Jordan MI (2012) Active spectral clustering via iterative uncertainty reduction. In: *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp 1339–1347
- Jiao L, Shang F, Wang F, Liu Y (2012) Fast semi-supervised clustering with enhanced spectral embedding. *Pattern Recogn* 45(12):4358–4369
- Chen W, Feng G (2012) Spectral clustering: a semi-supervised approach. *Neurocomputing* 77(1):229–242
- Rangapuram SS, Hein M (2012) Constrained 1-spectral clustering. In: *Artificial Intelligence and Statistics*, pp 1143–1151. PMLR
- Ding S, Jia H, Zhang L, Jin F (2014) Research of semi-supervised spectral clustering algorithm based on pairwise constraints. *Neural Comput Appl* 24(1):211–219
- Wang X, Qian B, Davidson I (2014) On constrained spectral clustering and its applications. *Data Min Knowl Disc* 28(1):1–30
- Wang H, Li T, Li T, Yang Y (2014) Constraint neighborhood projections for semi-supervised clustering. *IEEE Transact Cybern* 44(5):636–643
- Wang X, Wang J, Qian B, Wang F, Davidson I (2014) Self-taught spectral clustering via constraint augmentation. In: *Proceedings of the 2014 SIAM International Conference on Data Mining*, pp 416–424. SIAM
- Cucuringu M, Koutis I, Chawla S, Miller G, Peng R (2016) Simple and scalable constrained clustering: a generalized spectral method. In: *Artificial Intelligence and Statistics*, pp 445–454. PMLR
- Xiong C, Johnson DM, Corso JJ (2016) Active clustering with model-based uncertainty reduction. *IEEE Trans Pattern Anal Mach Intell* 39(1):5–17

- Jiang C, Xie H, Bai Z (2017) Robust and efficient computation of eigenvectors in a generalized spectral method for constrained clustering. In: *Artificial Intelligence and Statistics*, pp 757–766. PMLR
- Tian J, Ren Y, Cheng X (2019) Stratified feature sampling for semi-supervised ensemble clustering. *IEEE Access* 7:128669–128675
- Chen C, Qian H, Chen W, Zheng Z, Zhu H (2019) Auto-weighted multi-view constrained spectral clustering. *Neurocomputing* 366:1–11
- Zhang J, Fan R, Tao H, Jiang J, Hou C (2024) Constrained clustering with weak label prior. *Front Comp Sci* 18(3):183338
- Shimizu S, Sakayauchi S, Shibata H, Takama Y (2022) Proposal of dynamic generation of subordinate clusters based on bayesian information criterion for must-link constrained k-means. In: *The 10th Int. Symp. on Computational Intelligence and Industrial Applications (ISCIIA2022)*, Session
- Zhang X, Zong L, Liu X, Yu H (2015) Constrained NMF-based multi-view clustering on unmapped data. In: *Twenty-Ninth AAAI Conference on Artificial Intelligence*
- Li T, Ding C, Jordan MI (2007) Solving consensus and semi-supervised clustering problems using nonnegative matrix factorization. In: *Seventh IEEE International Conference on Data Mining (ICDM 2007)*, pp 577–582. IEEE
- Chen Y, Rege M, Dong M, Hua J (2008) Non-negative matrix factorization for semi-supervised data clustering. *Knowl Inf Syst* 17(3):355–379
- Chen Y, Wang L, Dong M (2009) Non-negative matrix factorization for semisupervised heterogeneous data coclustering. *IEEE Trans Knowl Data Eng* 22(10):1459–1474
- Jing L, Yu J, Zeng T, Zhu Y (2012) Semi-supervised clustering via constrained symmetric non-negative matrix factorization. In: *International Conference on Brain Informatics*, pp 309–319. Springer
- Wang D, Gao X, Wang X (2015) Semi-supervised nonnegative matrix factorization via constraint propagation. *IEEE Transact Cybern* 46(1):233–244
- Zhang X, Zong L, Liu X, Luo J (2015) Constrained clustering with nonnegative matrix factorization. *IEEE Transact Neural Netw Learn Syst* 27(7):1514–1526
- Wu W, Jia Y, Kwong S, Hou J (2018) Pairwise constraint propagation-induced symmetric nonnegative matrix factorization. *IEEE Transact Neural Netw Learn Syst* 29(12):6348–6361
- Settles B (2009) Active learning literature survey
- Greene D, Cunningham P (2007) Constraint selection by committee: An ensemble approach to identifying informative constraints for semi-supervised clustering. In: *European Conference on Machine Learning*, pp 140–151. Springer
- Mallapragada PK, Jin R, Jain AK (2008) Active query selection for semi-supervised clustering. In: *2008 19th International Conference on Pattern Recognition*, pp 1–4. IEEE
- Vu V-V, Labroche N, Bouchon-Meunier B (2010) An efficient active constraint selection algorithm for clustering. In: *2010 20th International Conference on Pattern Recognition*, pp 2969–2972. IEEE
- Vu V-V, Labroche N, Bouchon-Meunier B (2010) Boosting clustering by active constraint selection. *ECAI* 10:297–302
- Allab K, Benabdeslem K (2011) Constraint selection for semi-supervised topological clustering. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp 28–43. Springer
- Wang H, Li T, Li T, Yang Y (2012) Exemplars-constraints for semi-supervised clustering. In: *International Conference on Advanced Data Mining and Applications*, pp 115–126. Springer
- Duarte JM, Fred AL, Duarte FJF (2013) A constraint acquisition method for data clustering. In: *Iberoamerican Congress on Pattern Recognition*, pp 108–116. Springer
- Abin AA, Beigy H (2014) Active selection of clustering constraints: a sequential approach. *Pattern Recogn* 47(3):1443–1458
- Atwa W, Li K (2014) Active query selection for constraint-based clustering algorithms. In: *International Conference on Database and Expert Systems Applications*, pp 438–445 (2014). Springer
- Yang Y, Qi Z, Liu H (2015) Selective domain information acquisition to improve segmentation quality. In: *Proceedings of the 17th International Conference on Electronic Commerce 2015*, pp 1–8
- Chang C-C, Lin P-Y (2015) Active learning for semi-supervised clustering based on locally linear propagation reconstruction. *Neural Netw* 63:170–184
- Cai L, Yu T, He T, Chen L, Lin M (2016) Active learning method for constraint-based clustering algorithms. In: *International Conference on Web-Age Information Management*, pp 319–329. Springer
- Abin AA (2016) Querying beneficial constraints before clustering using facility location analysis. *IEEE Transact Cybern* 48(1):312–323
- Abin AA (2017) A random walk approach to query informative constraints for clustering. *IEEE Transact Cybern* 48(8):2272–2283
- Abin AA (2019) Querying informative constraints for data clustering: an embedding approach. *Appl Soft Comput* 80:31–41

- Abin AA, Vu V-V (2020) A density-based approach for querying informative constraints for clustering. *Expert Syst Appl* 161:113690
- Chen DW, Jin YH (2020) An active learning algorithm based on Shannon entropy for constraint-based clustering. *IEEE Access* 8:171447–171456
- Grira N, Crucianu M, Boujemaa N (2008) Active semi-supervised fuzzy clustering. *Pattern Recogn* 41(5):1834–1844
- Raj SRP, Ravindran B (2013) Incremental constrained clustering: A decision theoretic approach. In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp 475–486. Springer
- Rao V, Jawahar C (2013) Semi-supervised clustering by selecting informative constraints. In: *International Conference on Pattern Recognition and Machine Intelligence*, pp 213–221. Springer
- Xiong S, Azimi J, Fern XZ (2013) Active learning of constraints for semi-supervised clustering. *IEEE Trans Knowl Data Eng* 26(1):43–54
- Abin AA, Beigy H (2015) Active constrained fuzzy clustering: a multiple kernels learning approach. *Pattern Recogn* 48(3):953–967
- Abin AA (2016) Clustering with side information: Further efforts to improve efficiency. *Pattern Recogn Lett* 84:252–258
- Van Craenendonck T, Blockeel H (2017) Constraint-based clustering selection. *Mach Learn* 106(9):1497–1521
- Craenendonck TV, Dumančić S, Wolputte EV, Blockeel H (2018) COBRAS: Interactive clustering with pairwise queries. In: *International Symposium on Intelligent Data Analysis*, pp 353–366. Springer
- Fernandes MC, Covões TF, Pereira ALV (2019) Active learning for evolutionary constrained clustering. In: *2019 8th Brazilian Conference on Intelligent Systems (BRACIS)*, pp 162–167. IEEE
- Fernandes MC, Covões TF, Pereira ALV (2020) Improving evolutionary constrained clustering using active learning. *Knowl-Based Syst* 209:106452
- Shi Y, Yu Z, Cao W, Chen CP, Wong H-S, Han G (2020) Fast and effective active clustering ensemble based on density peak. *IEEE Transact Neural Netw Learn Syst* 32(8):3593–3607
- Hazratgholizadeh R, Balafar M, Derakhshi M (2023) Active constrained deep embedded clustering with dual source. *Appl Intell* 53(5):5337–5367
- Benabdeslem K, Snoussi J (2009) A probabilistic approach for constrained clustering with topological map. In: *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pp 413–426. Springer
- Allahyar A, Yazdi HS, Harati A (2015) Constrained semi-supervised growing self-organizing map. *Neurocomputing* 147:456–471
- Ren Y, Hu K, Dai X, Pan L, Hoi SC, Xu Z (2019) Semi-supervised deep embedded clustering. *Neurocomputing* 325:121–130
- Zhang H, Zhan T, Basu S, Davidson I (2021) A framework for deep constrained clustering. *Data Min Knowl Disc* 35(2):593–620
- Ienco D, Interdonato R (2023) Deep semi-supervised clustering for multi-variate time-series. *Neurocomputing* 516:36–47
- Maggini M, Melacci S, Sarti L (2012) Learning from pairwise constraints by similarity neural networks. *Neural Netw* 26:141–158
- Śmieja M, Struski Ł, Figueiredo MA (2020) A classification-based approach to semi-supervised clustering with pairwise constraints. *Neural Netw* 127:193–203
- Cui Y, Zhang X, Zong L, Mu J (2021) Maintaining consistency with constraints: a constrained deep clustering method. In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp 219–230. Springer
- Yan B, Domeniconi C (2006) Subspace metric ensembles for semi-supervised clustering of high dimensional data. In: *European Conference on Machine Learning*, pp 509–520. Springer
- Forestier G, Gañcarski P, Wemmert C (2010) Collaborative clustering with background knowledge. *Data Knowl Eng* 69(2):211–228
- Iqbal AM, Moh'd A, Khan Z (2012) Semi-supervised clustering ensemble by voting. Preprint at [arXiv:1208.4138](https://arxiv.org/abs/1208.4138)
- Chen D, Yang Y, Wang H, Mahmood A (2013) Convergence analysis of semi-supervised clustering ensemble. In: *2013 IEEE Third International Conference on Information Science and Technology (ICIST)*, pp 783–788. IEEE
- Lu Z, Peng Y (2013) Exhaustive and efficient constraint propagation: a graph-based learning approach and its applications. *Int J Comput Vision* 103(3):306–325
- Yi J, Zhang L, Yang T, Liu W, Wang J (2015) An efficient semi-supervised clustering algorithm with sequential constraints. In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp 1405–1414
- Yu Z, Luo P, You J, Wong H-S, Leung H, Wu S, Zhang J, Han G (2015) Incremental semi-supervised clustering ensemble for high dimensional data clustering. *IEEE Trans Knowl Data Eng* 28(3):701–714
- Yang F, Li T, Zhou Q, Xiao H (2017) Cluster ensemble selection with constraints. *Neurocomputing* 235:59–70

- Yu Z, Luo P, Liu J, Wong H-S, You J, Han G, Zhang J (2018) Semi-supervised ensemble clustering based on selected constraint projection. *IEEE Trans Knowl Data Eng* 30(12):2394–2407
- Lai Y, He S, Lin Z, Yang F, Zhou Q, Zhou X (2019) An adaptive robust semi-supervised clustering framework using weighted consensus of random k k-means ensemble. *IEEE Trans Knowl Data Eng* 33(5):1877–1890
- Yang T, Pasquier N, Precioso F (2022) Semi-supervised consensus clustering based on closed patterns. *Knowl-Based Syst* 235:107599
- Abdala DD, Jiang X (2009) An evidence accumulation approach to constrained clustering combination. In: *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pp 361–371. Springer
- Yang Y, Wang H, Lin C, Zhang J (2012) Semi-supervised clustering ensemble based on multi-ant colonies algorithm. In: *International Conference on Rough Sets and Knowledge Technology*, pp 302–309. Springer
- Yang Y, Tan W, Li T, Ruan D (2012) Consensus clustering based on constrained self-organizing map and improved cop-k-means ensemble in intelligent decision support systems. *Knowl-Based Syst* 32:101–115
- Yu Z, Kuang Z, Liu J, Chen H, Zhang J, You J, Wong H-S, Han G (2017) Adaptive ensembling of semi-supervised clustering solutions. *IEEE Trans Knowl Data Eng* 29(8):1577–1590
- Luo R, Yu Z, Cao W, Liu C, Wong H-S, Chen CP (2019) Adaptive regularized semi-supervised clustering ensemble. *IEEE Access* 8:17926–17934
- Vu V-V, Labroche N, Bouchon-Meunier B (2009) Leader ant clustering with constraints. In: *2009 IEEE-RIVF International Conference on Computing and Communication Technologies*, pp 1–8. IEEE
- Xu X, Lu L, He P, Pan Z, Chen L (2013) Improving constrained clustering via swarm intelligence. *Neurocomputing* 116:317–325
- Handl J, Knowles J (2006) On semi-supervised clustering via multiobjective optimization. In: *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, pp 1465–1472
- Hong Y, Kwong S, Wang H, Ren Q, Chang Y (2008) Probabilistic and graphical model based genetic algorithm driven clustering with instance-level constraints. In: *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, pp. 322–329. IEEE
- Hong Y, Kwong S, Xiong H, Ren Q (2008) Genetic-guided semi-supervised clustering algorithm with instance-level constraints. In: *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation*, pp 1381–1388
- Ebrahimi J, Abadeh MS (2012) Semi supervised clustering: a pareto approach. In: *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pp 237–251. Springer
- He Z (2016) Evolutionary k-means with pair-wise constraints. *Soft Comput* 20(1):287–301
- Oliveira RM, Chaves AA, Lorena LAN (2017) A comparison of two hybrid methods for constrained clustering problems. *Appl Soft Comput* 54:256–266
- Covões TF, Hruschka ER (2018) Classification with multi-modal classes using evolutionary algorithms and constrained clustering. In: *2018 IEEE Congress on Evolutionary Computation (CEC)*, pp 1–8. IEEE
- González-Almagro G, Luengo J, Cano J-R, García S (2021) Enhancing instance-level constrained clustering through differential evolution. *Appl Soft Comput* 107435
- Hiep TK, Duc NM, Trung BQ (2016) Local search approach for the pairwise constrained clustering problem. In: *Proceedings of the Seventh Symposium on Information and Communication Technology*, pp 115–122
- Zhang Z, Kang D, Gao C, Shao J (2019) Semisync: Semi-supervised clustering by synchronization. In: *International Conference on Database Systems for Advanced Applications*, pp 358–362. Springer
- Gao J, Tao X, Cai S (2023) Towards more efficient local search algorithms for constrained clustering. *Inf Sci* 621:287–307
- Yang Y, Wang H (2018) Multi-view clustering: a survey. *Big Data Min Anal* 1(2):83–107
- Nigam K, Ghani R (2000) Analyzing the effectiveness and applicability of co-training. In: *Proceedings of the Ninth International Conference on Information and Knowledge Management*, pp 86–93
- Frigui H, Hwang C (2007) Adaptive concept learning through clustering and aggregation of relational data. In: *Proceedings of the 2007 SIAM International Conference on Data Mining*, pp 90–101. SIAM
- Frigui H, Hwang C (2008) Fuzzy clustering and aggregation of relational data with instance-level constraints. *IEEE Trans Fuzzy Syst* 16(6):1565–1581
- Eaton E, Desjardins M, Jacob S (2014) Multi-view constrained clustering with an incomplete mapping between views. *Knowl Inf Syst* 38(1):231–257
- Günemann S, Färber I, Rüdiger M, Seidl T (2014) SMVC: semi-supervised multi-view clustering in subspace projections. In: *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp 253–262
- Khashabi D, Wieting J, Liu JY, Liang F (2015) Clustering with side information: From a probabilistic model to a deterministic algorithm. Preprint at [arXiv:1508.06235](https://arxiv.org/abs/1508.06235)

- Zhao P, Jiang Y, Zhou Z-H (2017) Multi-view matrix completion for clustering with side information. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp 403–415. Springer
- Lu Z, Peng Y (2013) Unified constraint propagation on multi-view data. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol 27, pp 640–646
- Yang Z, Hu Y, Liu H, Chen H, Wu Z (2014) Matrix completion for cross-view pairwise constraint propagation. In: Proceedings of the 22nd ACM International Conference on Multimedia, pp 897–900
- Kulis B, Basu S, Dhillon I, Mooney R (2005) Semi-supervised graph clustering: a kernel approach. In: Proceedings of the 22nd International Conference on Machine Learning, pp 457–464
- Kulis B, Basu S, Dhillon I, Mooney R (2009) Semi-supervised graph clustering: a kernel approach. *Mach Learn* 74(1):1–22
- Yan B, Domeniconi C (2006) An adaptive kernel method for semi-supervised clustering. In: European Conference on Machine Learning, pp 521–532. Springer
- Liu Y, Jin R, Jain AK (2007) Boostcluster: boosting clustering by pairwise constraints. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 450–459
- Almeida SGM, Coelho FGF, Guimarães FG, Braga AP (2012) A general approach for adaptive kernels in semi-supervised clustering. In: International Conference on Intelligent Data Engineering and Automated Learning, pp 508–515. Springer
- Chen C, Zhang J, He X, Zhou Z-H (2012) Non-parametric kernel learning with robust pairwise constraints. *Int J Mach Learn Cybern* 3(2):83–96
- Huang X, Cheng H, Yang J, Yu JX, Fei H, Huan J (2012) Semi-supervised clustering of graph objects: A subgraph mining approach. In: International Conference on Database Systems for Advanced Applications, pp 197–212. Springer
- Tang H, Miyamoto S (2013) Semi-supervised sequential kernel regression models with pairwise constraints. In: International Conference on Modeling Decisions for Artificial Intelligence, pp 166–178. Springer
- Baghshah MS, Afsari F, Shouraki SB, Eslami E (2014) Scalable semi-supervised clustering by spectral kernel learning. *Pattern Recogn Lett* 45:161–171
- Baraldi A, Blonda P (1999) A survey of fuzzy clustering algorithms for pattern recognition. i. *IEEE Transact Syst Man Cybern Part B (Cybernetics)* 29(6):778–785
- Ruspini EH, Bezdek JC, Keller JM (2019) Fuzzy clustering: a historical perspective. *IEEE Comput Intell Mag* 14(1):45–55
- Kanzawa Y, Endo Y, Miyamoto S (2009) Some pairwise constrained semi-supervised fuzzy c-means clustering algorithms. In: International Conference on Modeling Decisions for Artificial Intelligence, pp 268–281. Springer
- Gao C-F, Wu X-J (2011) A new semi-supervised clustering algorithm with pairwise constraints by competitive agglomeration. *Appl Soft Comput* 11(8):5281–5291
- Antoine V, Quost B, Masson M-H, Denoeux T (2012) CECM: Constrained evidential c-means algorithm. *Comput Stati Data Anal* 56(4):894–914
- Melo FM, Carvalho FdA (2013) Semi-supervised fuzzy c-medoids clustering algorithm with multiple prototype representation. In: 2013 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), pp 1–7. IEEE
- Hamasuna Y, Endo Y (2013) On semi-supervised fuzzy c-means clustering for data with clusterwise tolerance by opposite criteria. *Soft Comput* 17(1):71–81
- Antoine V, Quost B, Masson M-H, Denoeux T (2014) CEVCLUS: evidential clustering with instance-level constraints for relational data. *Soft Comput* 18(7):1321–1335
- Li L, Garibaldi JM, He D, Wang M (2015) Semi-supervised fuzzy clustering with feature discrimination. *PLoS ONE* 10(9):0131160
- Dempster AP, Laird NM, Rubin DB (1977) Maximum likelihood from incomplete data via the EM algorithm. *J Roy Stat Soc: Ser B (Methodol)* 39(1):1–22
- Shental N, Bar-Hillel A, Hertz T, Weinshall D (2004) Computing gaussian mixture models with EM using equivalence constraints. *Adv Neural Inf Process Syst* 16(8):465–472
- Lu Z, Leen T (2005) Semi-supervised learning with penalized probabilistic clustering. In: Saul L, Weiss Y, Bottou L (Eds) *Advances in Neural Information Processing Systems*, vol 17. MIT Press. <https://proceedings.neurips.cc/paper/2004/file/8dc5983b8c4ef1d8fed5f325f9a65511-Paper.pdf>
- Lu Z, Leen TK (2007) Penalized probabilistic clustering. *Neural Comput* 19(6):1528–1567
- Lu Z (2007) Semi-supervised clustering with pairwise constraints: A discriminative approach. In: *Artificial Intelligence and Statistics*, pp 299–306. PMLR
- Śmieja M, Wiercioch M (2017) Constrained clustering with a complex cluster structure. *Adv Data Anal Classif* 11(3):493–518
- Grira N, Crucianu M, Boujema N (2005) Semi-supervised fuzzy clustering with pairwise-constrained competitive agglomeration. In: *The 14th IEEE International Conference on Fuzzy Systems, 2005. FUZZ'05.*, pp 867–872. IEEE

- Ruiz C, Spiliopoulou M, Menasalvas E (2007) C-DBSCAN: Density-based clustering with constraints. In: International Workshop on Rough Sets, Fuzzy Sets, Data Mining, and Granular-soft Computing, pp 216–223. Springer
- Ruiz C, Menasalvas E, Spiliopoulou M (2009) C-DENSTREAM: Using domain knowledge on a data stream. In: International Conference on Discovery Science, pp 287–301. Springer
- Hamasuna Y, Endo Y, Miyamoto S (2010) Semi-supervised agglomerative hierarchical clustering using clusterwise tolerance based pairwise constraints. In: International Conference on Modeling Decisions for Artificial Intelligence, pp 152–162. Springer
- Hamasuna Y, Endo Y, Miyamoto S (2011) Semi-supervised agglomerative hierarchical clustering with ward method using clusterwise tolerance. In: International Conference on Modeling Decisions for Artificial Intelligence, pp 103–113. Springer
- Xiong T, Wang S, Mayers A, Monga E (2011) Semi-supervised parameter-free divisive hierarchical clustering of categorical data. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp 265–276. Springer
- Miyamoto S, Terami A (2011) Constrained agglomerative hierarchical clustering algorithms with penalties. In: 2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011), pp 422–427. IEEE
- Gilpin S, Davidson I (2011) Incorporating sat solvers into hierarchical clustering algorithms: an efficient and flexible approach. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 1136–1144
- González-Almagro G, Suarez JL, Luengo J, Cano J-R, García S (2020) Agglomerative constrained clustering through similarity and distance recalculation. In: International Conference on Hybrid Artificial Intelligence Systems, pp 424–436. Springer
- Kriegel H-P, Kröger P, Sander J, Zimek A (2011) Density-based clustering. Wiley interdisciplinary reviews: data mining and knowledge discovery 1(3):231–240
- Fan W-Q, Wang C-D, Lai J-H (2016) SDENPEAK: semi-supervised nonlinear clustering based on density and distance. In: 2016 IEEE Second International Conference on Big Data Computing Service and Applications (BigDataService), pp 269–275. IEEE
- Atwa W, Li K (2017) Constraint-based clustering algorithm for multi-density data and arbitrary shapes. In: Industrial Conference on Data Mining, pp 78–92. Springer
- Yang Y, Li Z, Wang W, Tao D (2017) An adaptive semi-supervised clustering approach via multiple density-based information. Neurocomputing 257:193–205
- Ren Y, Hu X, Shi K, Yu G, Yao D, Xu Z (2018) Semi-supervised denpeak clustering with pairwise constraints. In: Pacific Rim International Conference on Artificial Intelligence, pp 837–850. Springer
- Yan S, Wang H, Li T, Chu J, Guo J (2021) Semi-supervised density peaks clustering based on constraint projection. Int J Comput Intell Syst 14(1):140–147
- Xu L, Krzyzak A, Oja E (1993) Rival penalized competitive learning for clustering analysis, RBF net, and curve detection. IEEE Trans Neural Networks 4(4):636–649
- Moreno CR, Spiliopoulou M, Menasalvas E (2006) User constraints over data streams. Knowl. Discov. from Data Streams, 117
- Covões TF, Hruschka ER, Ghosh J (2012) Competitive learning with pairwise constraints. IEEE Transact Neural Netw Learn Syst 24(1):164–169
- Givoni I, Frey B (2009) Semi-supervised affinity propagation with instance-level constraints. In: Artificial Intelligence and Statistics, pp 161–168. PMLR
- Davidson I, Qi Z (2008) Finding alternative clusterings using constraints. In: 2008 Eighth IEEE International Conference on Data Mining, pp 773–778. IEEE
- Struyf J, Džeroski S (2007) Clustering trees with instance level constraints. In: European Conference on Machine Learning, pp 359–370. Springer
- Pensa RG, Boulicaut J-F, Cordero F, Atzori M (2010) Co-clustering numerical data under user-defined constraints. Stat Anal Data Min ASA Data Sci J 3(1):38–55
- Duong K-C, Vrain C et al (2017) Constrained clustering by constraint programming. Artif Intell 244:70–94
- Dao T-B-H, Duong K-C, Vrain C (2015) Constrained minimum sum of squares clustering by constraint programming. In: International Conference on Principles and Practice of Constraint Programming, pp 557–573. Springer
- Calandriello D, Niu G, Sugiyama M (2014) Semi-supervised information-maximization clustering. Neural Netw 57:103–111
- Xia Y (2009) A global optimization method for semi-supervised clustering. Data Min Knowl Disc 18(2):214–256
- Li Y, Dong M, Ma Y (2008) Feature selection for clustering with constraints using Jensen–Shannon divergence. In: 2008 19th International Conference on Pattern Recognition, pp 1–4. IEEE

- Zhao J, Xiao D, Hu L, Shi C (2019) Coupled semi-supervised clustering: Exploring attribute correlations in heterogeneous information networks. In: Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data, pp 95–109. Springer
- Li X, Wu Y, Ester M, Kao B, Wang X, Zheng Y (2017) Semi-supervised clustering in attributed heterogeneous information networks. In: Proceedings of the 26th International Conference on World Wide Web, pp 1621–1629
- Davidson I, Ravi S, Ester M (2007) Efficient incremental constrained clustering. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 240–249
- Bai L, Wang J, Liang J, Du H (2020) New label propagation algorithm with pairwise constraints. *Pattern Recogn* 106:107411
- Lin C, Yang Y, Rutayisire T (2011) A parallel cop-k-means clustering algorithm based on mapreduce framework. In: Knowledge Engineering and Management, pp 93–102. Springer
- Hu Y, Wang J, Yu N, Hua X-S (2008) Maximum margin clustering with pairwise constraints. In: 2008 Eighth IEEE International Conference on Data Mining, pp 253–262. IEEE
- Zeng H, Cheung Y-M (2011) Semi-supervised maximum margin clustering with pairwise constraints. *IEEE Transact Knowl Data Eng* 24(5):926–939
- Zeng H, Song A, Cheung YM (2013) Improving clustering with pairwise constraints: a discriminative approach. *Knowl Inf Syst* 36(2):489–515
- Śmiejka M, Myronov O, Tabor J (2018) Semi-supervised discriminative clustering with graph regularization. *Knowl-Based Syst* 151:24–36
- Métivier J-P, Boizumault P, Crémilleux B, Khiari M, Loudni S (2012) Constrained clustering using sat. In: International Symposium on Intelligent Data Analysis, pp 207–218. Springer
- Berg J, Järvisalo M (2017) Cost-optimal constrained correlation clustering via weighted partial maximum satisfiability. *Artif Intell* 244:110–142
- Hertz T, Bar-Hillel A, Weinshall D (2004) Boosting margin based distance functions for clustering. In: Proceedings of the Twenty-first International Conference on Machine Learning, p 50
- Chang H, Yeung D-Y (2004) Locally linear metric adaptation for semi-supervised clustering. In: Proceedings of the Twenty-First International Conference on Machine Learning. ICML '04, p 20. Association for Computing Machinery, New York. <https://doi.org/10.1145/1015330.1015391>
- Yeung D-Y, Chang H (2006) Extending the relevant component analysis algorithm for metric learning using both positive and negative equivalence constraints. *Pattern Recogn* 39(5):1007–1010
- Hoi SC, Liu W, Lyu MR, Ma W-Y (2006) Learning distance metrics with contextual constraints for image retrieval. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), vol. 2, pp 2072–2078. IEEE
- Bekkerman R, Sahami M, Learned-Miller E (2006) Combinatorial Markov random fields. In: European Conference on Machine Learning, pp 30–41. Springer
- Yeung D-Y, Chang H (2007) A kernel approach for semisupervised metric learning. *IEEE Trans Neural Networks* 18(1):141–149
- Davis JV, Kulis B, Jain P, Sra S, Dhillon IS (2007) Information-theoretic metric learning. In: Proceedings of the 24th International Conference on Machine Learning, pp 209–216
- Hoi SC, Jin R, Lyu MR (2007) Learning nonparametric kernel matrices from pairwise constraints. In: Proceedings of the 24th International Conference on Machine Learning, pp 361–368
- Xiang S, Nie F, Zhang C (2008) Learning a Mahalanobis distance metric for data clustering and classification. *Pattern Recogn* 41(12):3600–3612
- Baghshah MS, Shouraki SB (2009) Metric learning for semi-supervised clustering using pairwise constraints and the geometrical structure of data. *Intell Data Anal* 13(6):887–899
- Baghshah MS, Shouraki SB (2010) Kernel-based metric learning for semi-supervised clustering. *Neurocomputing* 73(7–9):1352–1361
- Baghshah MS, Shouraki SB (2010) Non-linear metric learning using pairwise similarity and dissimilarity constraints and the geometrical structure of data. *Pattern Recogn* 43(8):2982–2992
- Baghshah MS, Shouraki SB (2011) Learning low-rank kernel matrices for constrained clustering. *Neurocomputing* 74(12–13):2201–2211
- Xiong C, Johnson D, Xu R, Corso JJ (2012) Random forests for metric learning with implicit pairwise position dependence. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 958–966
- Fu Z, Lu Z, Ip HH, Lu H, Wang Y (2015) Local similarity learning for pairwise constraint propagation. *Multim Tools Appl* 74(11):3739–3758
- Johnson DM, Xiong C, Corso JJ (2015) Semi-supervised nonlinear distance metric learning via forests of max-margin cluster hierarchies. *IEEE Trans Knowl Data Eng* 28(4):1035–1046
- Abin AA, Bashiri MA, Beigy H (2020) Learning a metric when clustering data points in the presence of constraints. *Adv Data Anal Classif* 14(1):29–56

- Klein D, Kamvar SD, Manning CD (2002) From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering. Technical report, Stanford
- Yan S, Wang H, Lee D, Giles CL (2009) Pairwise constrained clustering for sparse and high dimensional feature spaces. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp 620–627. Springer
- Lu Z, Ip HH (2010) Constrained spectral clustering via exhaustive and efficient constraint propagation. In: European Conference on Computer Vision, pp 1–14. Springer
- Fu Z, Lu Z, Ip H, Peng Y, Lu H (2011) Symmetric graph regularized constraint propagation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol 25, pp 350–355
- Fu Z, Ip HH, Lu H, Lu Z (2011) Multi-modal constraint propagation for heterogeneous image clustering. In: Proceedings of the 19th ACM International Conference on Multimedia, pp 143–152
- He P, Xu X, Chen L (2012) Constrained clustering with local constraint propagation. In: European Conference on Computer Vision, pp 223–232. Springer
- Asafi S, Cohen-Or D (2013) Constraints as features. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 1634–1641
- He P, Xu X, Zhang L, Zhang W, Li K, Qian H (2014) Constrained community clustering. In: International Conference on Intelligent Computing, pp 797–802. Springer
- Xu X, He P (2016) Improving clustering with constrained communities. *Neurocomputing* 188:239–252
- Jia Y, Liu H, Hou J, Kwong S (2020) Pairwise constraint propagation with dual adversarial manifold regularization. *IEEE Transact Neural Netw Learn Syst* 31(12):5575–5587
- Li Y, Lu H (2021) Multi-modal constraint propagation via compatible conditional distribution reconstruction. *Neurocomputing* 426:185–194
- Yi J, Zhang L, Jin R, Qian Q, Jain A (2013) Semi-supervised clustering by input pattern assisted pairwise similarity matrix completion. In: International Conference on Machine Learning, pp 1400–1408. PMLR
- Fu Z (2015) Pairwise constraint propagation via low-rank matrix recovery. *Comput Visual Media* 1(3):211–220
- Li Z, Liu J, Tang X (2008) Pairwise constraint propagation by semidefinite programming for semi-supervised classification. In: Proceedings of the 25th International Conference on Machine Learning, pp 576–583
- Wang H, Nie R, Liu X, Li T (2012) Constraint projections for semi-supervised affinity propagation. *Knowledge-Based Syst* 36:315–321
- Gao S, Zu C, Zhang D (2013) Learning mid-perpendicular hyperplane similarity from cannot-link constraints. *Neurocomputing* 113:195–203
- Jia Y, Kwong S, Hou J, Wu W (2018) Convex constrained clustering with graph-laplacian pca. In: 2018 IEEE International Conference on Multimedia and Expo (ICME), pp 1–6. IEEE
- Jia Y, Hou J, Kwong S (2020) Constrained clustering with dissimilarity propagation-guided graph-Laplacian PCA. *IEEE Transact Neural Netw Learn Syst* 32(9):3985–3997
- Raghuram J, Miller DJ, Kesidis G (2014) Instance-level constraint-based semisupervised learning with imposed space-partitioning. *IEEE Transact Neural Netw Learn Syst* 25(8):1520–1537

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Germán González-Almagro^{1,2} · Daniel Peralta³ · Eli De Poorter³ · José-Ramón Cano^{2,4} · Salvador García^{1,2}

✉ Germán González-Almagro
germagalmagro@ugr.es

Daniel Peralta
daniel.peralta@ugent.be

Eli De Poorter
eli.depoorter@ugent.be

José-Ramón Cano
jrcano@ujaen.es

Salvador García
salvagl@decsai.ugr.es

- ¹ Department of Computer Science and Artificial Intelligence, University of Granada, Granada, Andalucía, Spain
- ² Data Science and Computational Intelligence Andalusian Institute, DaSCI, Granada, Andalucía, Spain
- ³ IDLab, Department of Information Technology, Ghent University - imec, Ghent, Belgium
- ⁴ Department of Computer Science, University of Jaén, Jaén, Andalucía, Spain