Noname manuscript No.
(will be inserted by the editor)

A Lagrangian-based score for assessing the quality of pairwise constraints in semi-supervised clustering

Rodrigo Randel · Daniel Aloise ·
Simon J. Blanchard · Alain Hertz

Received: 04/23/20 / Accepted: 09/02/21

Abstract Clustering algorithms help identify homogeneous subgroups from data. In some cases, additional information about the relationship among some subsets of the data exists. When using a semi-supervised clustering algorithm, an expert may provide additional information to constrain the solution based on that knowledge and, in doing so, guide the algorithm to a more useful and meaningful solution. Such additional information often takes the form of a cannot-link constraint (i.e., two data points cannot be part of the same cluster) or a must-link constraint (i.e., two data points must be part of the same cluster). A key challenge for users of such constraints in semi-supervised learning algorithms, however, is that the addition of inaccurate or conflicting constraints can decrease accuracy and little is known about how to detect whether expert-imposed constraints are likely incorrect. In the present work, we introduce a method to score each must-link and cannot-link pairwise constraint as likely incorrect. Using synthetic experimental examples and real data, we show that the resulting impact score can successfully identify individual constraints that should be removed or revised.

Keywords clustering · semi-supervised · pairwise constraints · constraint selection · Lagrangian duality.

Rodrigo Randel
Département de Génie Informatique et Génie Logiciel, Polytechnique Montréal, Montréal, Québec, Canada
E-mail: rodrigo.randel@polymtl.ca

Daniel Aloise
Département de Génie Informatique et Génie Logiciel, Polytechnique Montréal, Montréal, Québec, Canada
E-mail: daniel.aloise@polymtl.ca

Simon J. Blanchard
McDonough School of Business, Georgetown University, Washington, DC, USA
E-mail: sjb247@georgetown.edu

Alain Hertz
Département de Mathématiques et de Génie Industriel, Polytechnique Montréal, Montréal, Québec, Canada
E-mail: alain.hertz@polymtl.ca

1 Introduction

A common typology is to consider machine learning algorithms as being of one of two paradigms: (i) *unsupervised learning*, when the objective is to provide the best underlying description of the data when no label information is available; (ii) *supervised learning*, when the objective is to use labeled training data to create an input-output function to map inputs to those labels[1]. Thus, in both cases, the objective is to identify a classification function but the paradigms differ in whether labels are available for all the training data points (supervised learning) or none of the training data points (unsupervised learning). Both learning paradigms face challenges. Although supervised learning techniques can obtain minimal error measures, the labels it requires are time-consuming/expensive to generate as, in most cases, a human expert must act as an annotator. As for unsupervised learning, it suffers from assumptions on the underlying structure of the dataset that are imposed when selecting a specific algorithm to work with it.

Semi-supervised learning presents a third paradigm for which one can incorporate limited information about how training data points should be related to one another. For instance, one may not know precisely all the labels of all the data points as in supervised learning, but one may know that some subsets of points belong (or do not belong) to the same classes. Thus, in *semi-supervised learning*, one can generate a classification function using both labeled and unlabeled data. Typically, incomplete labeling information is obtained from the knowledge of domain experts who provide a set of constraints that the classification function must satisfy (Zhu et al., 2009; Anil et al., 2015). Performing the supervision through expert-provided constraints thus aims to combine the advantages of unsupervised and supervised learning.

To formally illustrate how semi-supervised learning incorporates such external knowledge, we do so by building on the most popular unsupervised learning model: clustering. Given a set $O = \{o_1, \dots, o_n\}$ of n unlabeled data points in a s-dimensional space, clustering methods identify subsets of data points, called clusters, which are homogeneous or well separated (Hansen and Jaumard, 1997). Among clustering methods, *partitioning* focuses on splitting O into k clusters ($P_k = \{C_1, C_2, \dots, C_k\}$) such that:

(i) $C_j \neq \emptyset$ for all $j = 1, \dots, k$,

(ii) $C_i \cap C_j = \emptyset$ for all $1 \leq i < j \leq k$, and

(iii) $\bigcup_{j=1}^{k} C_j = O$,

and where the set of all k-partitions of O is denoted $\mathcal{P}(O, k)$. If the number of clusters k is known, and thus fixed, clustering can be formulated as a mathematical optimization problem whose objective function $f : \mathcal{P}(O, k) \to \mathbb{R}$,

[1] We focus on discrete labels (e.g., classes) for simplicity of exposition, although there are numerous unsupervised (e.g., latent trait models) and supervised models (e.g., regression) which focus on continuous outcomes.

usually called *clustering criterion*, defines the optimal solution for the problem given by the following (e.g. Christou, 2011):

$$\min\{f(P) : P \in \mathcal{P}(O, k)\}. \tag{1}$$

The choice of function f is critical to how homogeneity and separation will be expressed in the resulting clusters. For example, homogeneity of a cluster can be measured by its *diameter* (i.e., the maximum dissimilarity between two data points part of the same cluster) and separation can be measured by the *split* (i.e., the minimum dissimilarity between two points part of different clusters). Such clustering criteria can be expressed in the form of thresholds, min-sum or max-sum functions. For example, the *minimum sum-of-squares clustering* criterion (MSSC), in which is based the optimization performed by the popular k-means algorithm, seeks to minimize the sum of squared distances from each data point to the representative of the cluster to which it belongs. In minimizing the sum of squared distances, the criterion indirectly imposes a constraint on the output that all clusters have a spherical shape. The user of the algorithm rarely has evidence or external data to support that choice.

In *Semi-Supervised Clustering*, the domain expert's information is used to circumvent the potential shortcomings associated with the choice of a particular clustering model. It has been suggested (Anil et al., 2015) that a domain expert could provide, whenever possible, auxiliary information regarding the data distribution, thus leading to better clustering solutions that are more in line with their knowledge, beliefs, and expectations. In this context, a different kind of assumption about the data distribution is made. Specifically, it is often assumed that a non-zero subset of objects have cluster labels that are known due to external knowledge. This type of supervision is called *pointwise information* and is usually easy to incorporate in existing unsupervised clustering algorithms (Aggarwal, 2015), for instance, by using pre-determined labels for the initialization of an existing unsupervised clustering algorithm like k-means (Basu et al., 2002). As an expert may not have knowledge of precise label assignments but rather the pairwise similarity between data points, a form of supervision that is more likely to be used by experts is to provide information regarding whether two points can (or cannot) belong to the same clusters (i.e., *must-link* and *cannot-link* constraints, respectively). Formally, a must-link constraint for data points o_i and o_j requires that o_i and o_j must be assigned to the same cluster, and a cannot-link constraint on the same data points requires that o_i and o_j must be assigned to different clusters. The definition and integration of such constraints when reasoning on background knowledge allows the user to incorporate extra requirement as well as directing the clustering model output in a declarative way (Grossi et al., 2017a).

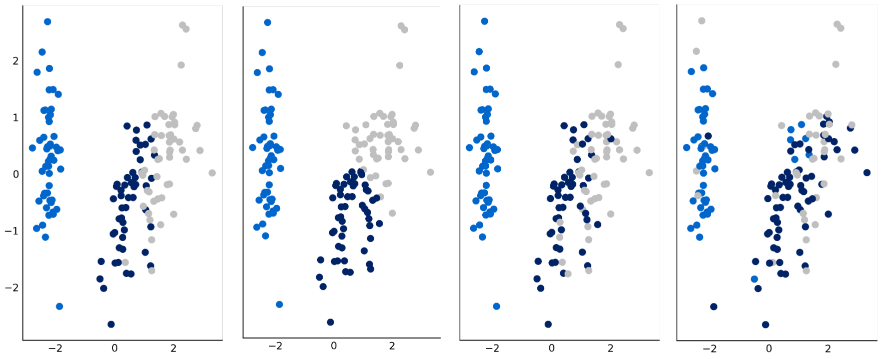
Moreover, such information that experts have to provide is common to many types of applications. Basu et al. (2006) discuss an example in the context of clustering protein sequences in which it is easy to identify proteins that co-occur in other proteins (i.e., must-link constraints) even if the class label is unknown or uncertain for these proteins. In image segmentation applications, cannot-link constraints are added for pixels that are in very distant regions of an image or when there is a frontier visible to the expert's eye. Kim et al. (2013) provide an example of how managers may have prior knowledge to impose constraints into Bayesian mixture models to render solutions that are eventually actionable by businesses. Nonetheless, working with pairwise constraints is typically more complex than incorporating pointwise information, and the problem of whether it is possible to satisfy a given set of cannot-link constraints with k clusters is NP-complete (Davidson and Ravi, 2005).

It would be sensible to assume that if input data is augmented by that of an expert, it should improve clustering performance. However, the presence of inaccurate or conflicting pairwise constraints has been shown to degrade it (Davidson et al., 2006; Davidson and Ravi, 2006). Degradation can be because it is generally assumed that when an expert provides information, the expert must be correct. However, in many cases, the labels provided by experts themselves are subject to errors of human judgments (e.g., a single human judge determines whether two proteins must co-occur). Such human judgment errors are especially likely when multiple experts are used to arrive at a consensus judgment. As the accuracy of constraints imposed to the algorithm ultimately impacts clustering accuracy (Ares et al., 2012), and that inaccuracy of constraints can occur due to human judgment errors and is an important problem, methods that can help users identify which constraints are likely to be subject to errors should be helpful in improving accuracy (Anil et al., 2015).

To illustrate the consequences of having inaccurate constraints, we show in Figure 1 clustering solutions from the two principal components of an application to the Iris dataset (Fisher, 1936). Figure 1(a) illustrates the ground-truth partition, whereas Figure 1(b) shows the optimal partition obtained with MSSC. Whereas MSSC recovers perfectly the cluster depicted in light blue, it does not well separate the two other clusters. Figure 1(c) illustrates the partition obtained by using the popular COP-Kmeans algorithm (Wagstaff et al., 2001) executed with a random set of 60 correct pairwise constraints extracted from the ground-truth partition. We observe that it is more consistent with the ground-truth partition. However, we also show in Figure 1(d) that a solution with 10 erroneous constraints can significantly deteriorate the performance of a clustering algorithm to a point that is worse than when no constraint was imposed.

Our objective in this paper is to provide a method for quantifying the likely accuracy of pairwise constraints. Specifically, we define an impact score for each pairwise constraint based on the solution of the dual of a integer program. In doing so, we provide a quantitative measure (i.e., Lagrangian-based impact score) that can help a user identify which must-link or cannot-link constraints degrade the clustering solution and should be removed or revised.

The rest of the paper is organized as follows. Section 2 provides an overview of prior research regarding the difficulty of substantiating whether a constraint set is informative. Then, section 3 presents the proposed impact score, and section 4 reports our experiments regarding the effectiveness of the score. Finally, concluding remarks are given in the last section of the paper.



(a) Ground-truth partition

(b) Solution produced by MSSC

(c) Solution produced by COP-Kmeans with 60 correct pairwise constraints

(d) Solution produced by COP-Kmeans with 60 correct and 10 erroneous pairwise constraints

Fig. 1 Illustration of the effects of clustering in the presence of erroneous constraints. The solution obtained with COP-Kmeans and 60 correct constraints in (c) is closer to the the ground-truth partition (a) than the unsupervised MSSC solution presented in (b). In contrast, the insertion of 10 erroneous constraints deteriorates the clustering solution as shown in (d).

2 Constraint inclusions in learning models

When using semi-supervised clustering (SSC), obtaining useful constraints is challenging as relying on domain experts can be difficult to scale for large classification problems (Wagstaff, 2007). One approach taken is the use of *active learning methods* which automatically generate constraints to reduce the amount of information that a domain expert needs to provide. Yet, even active learning methods require some a-priori domain knowledge provided by an expert to identify the additional (or redundant) constraints. For example, the widely used PCKmeans (Basu et al., 2004) identifies the pairs of data points which are farthest from each other and queries an *oracle* to determine whether a cannot-link constraint should be added. The oracle is a function that analyzes the known pairwise constraints to investigate if the dissimilarity between the queried pair of data points is sufficient to impose a new cannot-link constraint. In Mallapragada et al. (2008), the authors also use the similarity between a pair of data points as a proxy for the confidence level that one should have in adding a must-link constraint. In Xiong et al. (2014), the authors uses pairwise constraints to build neighborhoods of data points in the same cluster (must-link constraints) and neighborhoods of points in different clusters (cannot-link constraints). Then, they use an active learning method to expand these neighborhoods by selecting informative points and querying the oracle about their relationship with their neighbors. In both Mallapragada et al. (2008) and Xiong et al. (2014), it is important to note that the active learning methods must still begin with a small set of pairwise information that are assumed to be correct and direct the algorithm in the correct course (Xiong et al., 2017).

Regardless of whether constraint information was originated from the domain expert or was generated by an active learning method, there is no guarantee that its inclusion will improve the clustering solutions. As such, one must have a way to identify whether the added constraints are helpful. Davidson et al. (2006) propose two measures that evaluate the *informativeness* and *coherence* of a constraint set. Informativeness aims to capture the incremental effect of adding the constraints to a solution. Specifically, informativeness is operationalized as the fraction of pairwise constraints that are violated once added to a clustering solution obtained without any constraints. The higher is the proportion of violated constraints, the more informative is the constraint set. Coherence is a measure of the agreement of a constraint set based on the adopted dissimilarity metric. Specifically, it aims to identify pairs of constraints, one must-link and one cannot-link constraint, which overlap when the constraint vectors (i.e., vectors connecting their associated points) are projected onto each other. Figure 2 illustrates two constraints with an overlapping segment when the cannot-link vector is projected onto the must-link vector. The constraint set with the highest proportion of null projections (when there is no overlapping segment) is considered as the most coherent set. For both measures, the idea is that constraint sets with the higher informativeness and coherence should improve the clustering solution. Wagstaff (2007) has found partial support for this hypothesis, suggesting that more properties related with the utility of pairwise constraints should be further developed.

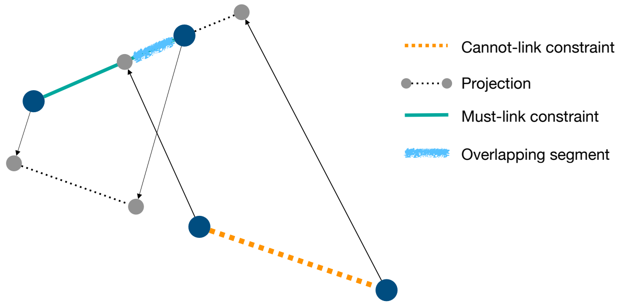


Fig. 2 Illustration of Coherence measure proposed by Davidson et al. (2006): projection of must-link and cannot-link constraint vectors onto each other.

Informativeness and coherence are not the only measures available to evaluate the helpfulness of constraints. For instance, Davidson (2012) proposes two other measures. For the first, he suggests counting the number of feasible clustering solutions using Markov Chain Monte Carlo samplers - with the goal of eliminating constraints which are difficult to satisfy and whose inclusions often leads to few feasible clustering solutions across the samplers. For the second, he suggests to eliminate constraints based on the fractional chromatic number of the constraint graph. The constraint graph contains one vertex for each data point and an edge for each cannot-link constraint. Data points involved in one or more must-link constraints are merged into a single vertex. As determining the chromatic number of this graph is equivalent to determining the minimum number of clusters required to make the problem feasible, and as finding the chromatic number of a graph is a NP-hard problem, the author suggests to solve a *linear relaxation* of the problem in which every vertex can be associated with more than one color (i.e., more than one cluster) to identify constraints to eliminate. As a final step, the second approach proceeds to pruning constraints by the following: if a vertex has many fractional colors, i.e., it is part of many independent sets, the constraints associated with the vertex are not hard to satisfy and can remain. However, if a vertex is part of only one independent set (i.e., its assignment is not fractional), the associated constraints are hard to satisfy and should be removed.

We have outlined three existing measures (fractional chromatic number, informativeness, and coherence). An important commonality is that all three measures focus on identifying good constraint sets based on the ability to satisfy them. More importantly, they cannot speak to the quality of individual pairwise constraints contained in the proposed constraint sets. As such, such measures cannot speak to how constraints interact, and thus cannot help assess the global quality of each constraint for the target clustering model. In the next section, we introduce our Lagrangian-based impact score to assess the individual quality of each pairwise constraint.

3 A Lagrangian-based scoring of the effect of individual pairwise constraints

Consider the following general integer programming formulation of a semi-supervised clustering problem:

$$Z = \min_{X} \quad f(x) \tag{2}$$

subject to

$$x_i^c + x_j^c \leq 1 \qquad \forall (o_i, o_j) \in \mathcal{CL}, \quad \forall c = 1, \ldots, k \tag{3}$$

$$x_i^c - x_j^c = 0 \qquad \forall (o_i, o_j) \in \mathcal{ML}, \quad \forall c = 1, \ldots, k \tag{4}$$

$$x_i^c \in \{0,1\} \qquad \forall i = 1, ..., n; \quad \forall c = 1, \ldots, k \tag{5}$$

where f is the clustering criterion to be minimized, and where every binary decision variables x_i^c of the solution space X indicates whether data point o_i is assigned to cluster c. Typically, X is composed of the set $\mathcal{P}(O, k)$ of all k-partitions of O for a given k predetermined number of clusters. In such a model, pairwise constraints are included via (3) and (4) where $\mathcal{CL}$ and $\mathcal{ML}$ represent the sets of pairs of data objects involved in cannot-link and must-link constraints, respectively.

To avoid situations where constraints (3) and (4) are satisfied with equality, we can replace them by the following equivalent constraints where ϵ is any real number in $]0,1[$:

$$x_i^c + x_j^c \leq 1 + \epsilon \qquad \forall (o_i, o_j) \in \mathcal{CL}, \quad \forall c = 1, \dots, k \tag{3'}$$

$$x_i^c - x_j^c \leq \epsilon \qquad \forall (o_i, o_j) \in \mathcal{ML}, \quad \forall c = 1, \dots, k \tag{4'}$$

$$x_j^c - x_i^c \leq \epsilon \qquad \forall (o_i, o_j) \in \mathcal{ML}, \quad \forall c = 1, \dots, k. \tag{4''}$$

The choice of function f has a significant impact on the computational complexity of any clustering problem. Whereas, for example, split maximization is polynomially solvable in time $O(n^2)$ (Delattre and Hansen, 1980), diameter minimization is NP-hard for more than two clusters (Brucker, 1978). For example, from Huygen's theorem (Edwards and Cavalli-Sforza, 1965), MSSC is expressed within (2)-(5) by:

$$f(x) = \sum_{c=1}^{k} \frac{\displaystyle\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \|o_i - o_j\|^2 x_i^c x_j^c}{\sum_{i=1}^{n} x_i^c}, \tag{6}$$

which is a non-convex quadratic function, making (2)-(5) NP-hard even for two clusters in general Euclidean dimension (Aloise et al., 2009). For the k-medoids model (see e.g. Kaufman and Rousseeuw (2009)), f can be expressed by:

$$f(x) = \sum_{c=1}^{k} \sum_{i=1}^{n} \sum_{j=1}^{n} \|o_i - o_j\| x_i^c y_j^c, \tag{7}$$

after adding binary variables y_i^c which are equal to 1 if the object o_i is the medoid for cluster c, and 0 otherwise, and constraints:

$$x_i^c \leq y_i^c \quad \forall i, j = 1, \dots, n \tag{8}$$

$$\sum_{i}^{n} y_i^c = k. \tag{9}$$

The k-medoids model is also NP-hard (Kariv and Hakimi, 1979). Algorithms for finding the optimal solution of the problem for large data sets are presented by Avella et al. (2007) and García et al. (2011), while efficient heuristics are proposed by Hansen et al. (2009) and Resende and Werneck (2007).

Classical Lagrangian duality theory associates penalty terms, named *Lagrangian multipliers*, to the problem constraints. Applied to SSC, regardless of the choice of clustering criterion f, the Lagrangian function $L(\eta, \lambda, \gamma)$ associated with the above integer programming problem is obtained by introducing penalty terms $\eta_{ij}^c, \lambda_{ij}^c$ and γ_{ij}^c for the violation of constraints (3'), (4'), and (4"). Specifically, the Lagrangian function is defined as follows:

$$
\begin{aligned}
L(\eta,\lambda,\gamma) = \min_X \Bigg(f(x) &+ \sum_{(o_i,o_j)\in\mathcal{CL}} \sum_{c=1}^{k} \eta_{ij}^c (1+\epsilon - x_i^c - x_j^c) \\
&+ \sum_{(o_i,o_j)\in\mathcal{ML}} \sum_{c=1}^{k} \lambda_{ij}^c (\epsilon + x_i^c - x_j^c) \\
&+ \sum_{(o_i,o_j)\in\mathcal{ML}} \sum_{c=1}^{k} \gamma_{ij}^c (\epsilon + x_j^c - x_i^c) \Bigg)
\end{aligned} \tag{10}
$$

and the dual of the integer program (2)-(5) can be expressed as follows:

$$
L_D = \max_{\eta,\lambda,\gamma\leq 0} L(\eta,\lambda,\gamma) \tag{11}
$$

where η, λ and γ correspond to its dual variables. The weak duality theorem (see e.g. Bertsimas and Tsitsiklis (1997)) asserts that L_D is the best lower bound for the optimal value Z of the integer program (2)-(5).

To illustrate how the Lagrangian function penalizes constraint violations, consider a cannot-link constraint $(o_i, o_j) \in \mathcal{CL}$ and a cluster $c \in \{1, \ldots, k\}$. Given that $\eta_{ij}^c \leq 0$, we penalize situations where $x_i^c + x_j^c > 1$ (i.e., the corresponding constraint (3) is violated). If $x_i^c + x_j^c \leq 1$, we have $1 + \epsilon - x_i^c - x_j^c > 0$ and the optimal value L_D is therefore obtained by setting $\eta_{ij}^c = 0$. Analogously, for a must-link constraint $(o_i, o_j) \in \mathcal{ML}$, both λ_{ij}^c and γ_{ij}^c are equal to 0 in an optimal solution of the dual problem when $x_i^c = x_j^c$, while exactly one of λ_{ij}^c and γ_{ij}^c is strictly negative (and the other one is equal to 0) when $x_i^c \neq x_j^c$.

3.1 Scoring constraints from the dual's information

The difference between Z and L_D is the *duality gap*. The values of the dual variables in an optimal solution of the dual problem provide information about the difficulty to satisfy a constraint and are of particular usefulness when the duality gap is small which is often the case in clustering models (Kochetov and Ivanenko, 2005; Aloise et al., 2010).

To illustrate, consider any cannot-link constraint $(o_u, o_v) \in \mathcal{CL}$. Assume that the constraints (3') imposing $x_u^c + x_v^c \leq 1 + \epsilon$ for all $c \in \{1 \ldots k\}$ are replaced by the following constraints:

$$
x_u^c + x_v^c \leq 1 + \epsilon + b \qquad \forall c = 1, \ldots, k \tag{12}
$$

In doing so, we added a non-negative value b to the right-hand side of the cannot-link constraints which involve objects o_u and o_v. As b increases, the cannot-link constraint for data objects o_u and o_v becomes more relaxed. Let us denote Z^b the optimal solution value of this modified problem, with $Z^b = Z$ for $b = 0$, and $Z^b \leq Z$, otherwise.

The objective function of the Lagrangian function, parameterized in b, is given by:

$$L_{uv}^{\mathcal{CL}}(\eta, \lambda, \gamma, b) = L(\eta, \lambda, \gamma) + \sum_{c=1}^{k} b\eta_{uv}^{c} \tag{13}$$

which is a lower bound to Z^b. Its partial derivative

$$\frac{\partial L_{uv}^{\mathcal{CL}}(\eta, \lambda, \gamma, b)}{\partial b} = \sum_{c=1}^{k} \eta_{uv}^{c}. \tag{14}$$

provides then an approximation of the effect on Z^b of deactivating the cannot-link constraint for data objects o_u and o_v. Likewise, given a must-link constraint $(o_u, o_v) \in \mathcal{ML}$, we add a positive value b to the right-hand side of the must-link constraints (4') and (4") for objects o_u and o_v. The Lagrangian function $L_{uv}^{\mathcal{ML}}(\eta, \lambda, \gamma, b)$ becomes:

$$L_{uv}^{\mathcal{ML}}(\eta, \lambda, \gamma, b) = L(\eta, \lambda, \gamma) + \sum_{c=1}^{k} b(\lambda_{uv}^{c} + \gamma_{uv}^{c}) \tag{15}$$

and the approximated effect on Z^b of deactivating the must-link constraint between data points o_u and o_v is given by:

$$\frac{\partial L_{uv}^{\mathcal{ML}}(\eta, \lambda, \gamma, b)}{\partial b} = \sum_{c=1}^{k} (\lambda_{uv}^{c} + \gamma_{uv}^{c}). \tag{16}$$

Negative values for the partial derivatives (14) and (16) suggest that a user can likely improve Z if the constraints are removed from the SSC model. Zero values for the partial derivatives suggest that the corresponding constraint is intrinsic to the underlying structure of the data or is redundant due to the inclusion of other constraints.

Based on these observation, we propose the following impact score $\mathcal{I}_{uv}$ for a pairwise constraint associated with objects o_u and o_v:

$$\mathcal{I}_{uv} = \begin{cases} \displaystyle\sum_{c=1}^{k} \eta_{uv}^{c} & \text{if } (o_u, o_v) \in \mathcal{CL} \\ \displaystyle\sum_{c=1}^{k} (\lambda_{uv}^{c} + \gamma_{uv}^{c}) & \text{if } (o_u, o_v) \in \mathcal{ML}. \end{cases} \tag{17}$$

In the next section, we discuss how to solve the dual problem (11) to calculate the impact score (17).

3.2 Solving the dual problem

The *sub-gradient optimization algorithm* (Shor et al., 1985; Held et al., 1974) is a widely used technique for optimizing non-differentiable optimization problems such as (11). To minimize a function $g : U \subset \mathbb{R} \to \mathbb{R}$, the domain variables are iteratively updated by setting

$$w \leftarrow w + \alpha_\ell \mathfrak{s}(w), \tag{18}$$

where $w \in U$ and $\mathfrak{s}(w)$ is any subgradient of $g(w)$, i.e., any vector that satisfies the inequality $g(y) \geq g(w) + \mathfrak{s}^T(y - w)$ for all $y \in U$. The step size for the ℓ-th iteration is defined by α_ℓ.

Algorithm 1 Subgradient method for optimizing the dual problem (11)

```
Initialize variables $\eta^c_{uv}, \lambda^c_{uv}$, and $\gamma^c_{uv}$ to 0.
Set the upper bound $\bar{Z}^*$ on $Z$ equal to the value of the best availabe feasible solution.
for all $\ell = 1$ to $m$ do
    Lower bounding step.
    Use current values of the dual variables and equation (10) to determine a lower bound solution $x$ of cost $Z$.
    if $Z$ is the largest lower bound ever found  then
        Save the dual variables in vectors $\eta_{best}, \lambda_{best}$ and $\gamma_{best}$.
    end if
    Upper bounding step.
    Let $\mathcal{R}$ be a routine able to transform any solution $x \in X$ into a feasible solution to (3)-(5) Run $\mathcal{R}(x)$ to obtain an upper bound solution of cost $\bar{Z}$. If $\bar{Z} < \bar{Z}^*$ then set $\bar{Z}^* \leftarrow \bar{Z}$.
    Updating step.
    $\alpha_\ell = \frac{1}{\sqrt{\ell}}$
    for all $(o_u, o_v) \in \mathcal{CL}$ and all $c \in \{1, \ldots, k\}$ do
        $\eta^c_{uv} \leftarrow \eta^c_{uv} + \alpha_\ell \frac{(\bar{Z}^* - Z)}{\sum_{(i,j) \in \mathcal{CL}} \sum_{c'=1}^{k} (1 + \epsilon - x^{c'}_i - x^{c'}_j)^2} (1 + \epsilon - x^c_u - x^c_v).$
    end for
    for all $(o_u, o_v) \in \mathcal{ML}$ and all $c \in \{1, \ldots, k\}$ do
        $\lambda^c_{uv} \leftarrow \lambda^c_{uv} + \alpha_\ell \frac{(\bar{Z}^* - Z)}{\sum_{(i,j) \in \mathcal{ML}} \sum_{c'=1}^{k} (\epsilon + x^{c'}_i - x^{c'}_j)^2} (\epsilon + x^c_u - x^c_v)$
        $\gamma^c_{uv} \leftarrow \gamma^c_{uv} + \alpha_\ell \frac{(\bar{Z}^* - Z)}{\sum_{(i,j) \in \mathcal{ML}} \sum_{c'=1}^{k} (\epsilon + x^{c'}_j - x^{c'}_i)^2} (\epsilon + x^c_v - x^c_u).$
    end for
end for
```

Algorithm 1 describes the steps of the sub-gradient method for solving (11). The algorithm begins by defining initial values for the Lagrangian multipliers $\eta^c_{uv}, \lambda^c_{uv}$ and γ^c_{uv}. As a common practice when working with Lagrangian relaxation, we initialize these penalty terms with zero (Fisher, 1981), which means that we impose no prior cost on the objective function. We next make the resonable assumption that there are solutions that satisfy constraints (3)-(5) and that it is not difficult to determine some of them. The initial upper bound $\bar{Z}^*$ on Z is thus set equal to the value of the best available feasible solution. Then, the algorithm begins its main loop wherein three steps take place for a predefined number m of iterations. In the first step, a lower bound for (2)-(5) is obtained by solving model (10) with fixed values of the Lagrangian multipliers. In other words, this step aims to solve the unsupervised clustering problem with predefined penalty terms for violating pairwise constraints. If the lower bound obtained is the best obtained so far, values of the Lagrangian multipliers are stored in vectors $\eta_{best}, \lambda_{best}$, and γ_{best}. The next step uses the lower bound solution to recover a feasible solution to (2)-(5). This routine can be as simple as the procedure described in Algorithm 2. This algorithm cannot offer any guarantee that it will converge to a feasible solution, because the problem of determining whether such a solution exists is NP-complete. Convergence is however ensured in our case thanks to our reasonable assumption that it is not difficult to generate solutions that satisfy constraints (3)-(5). If a situation arises for which it is difficult to recover feasibility, we can stop Algorithm 2 after a time limit of a few seconds and thus give up updating the upper bound $\bar{Z}^*$. Finally, the last step updates the dual variables with respect to their subgradient for a step size α_ℓ which is updated at each iteration with a decreasing rule.

Algorithm 2 Routine for restoring feasibility

```
for each violated must-link constraints $(o_i, o_j) \in \mathcal{ML}$ do
    Move $o_i$ and $o_j$ to the best cluster w.r.t. $f$.
end for
while at least one cannot-link constraint is violated do
    Choose a data point $o_i$ at random among those involved in a violated cannot-link constraint, and let $c$ be the cluster that contains $o_i$.
    Move $o_i$ to the best cluster $c' \neq c$ w.r.t. $f$, prioritizing the clusters that do not contain a data point $o_j$ with $(o_i, o_j) \in \mathcal{CL}$.
    if $o_i$ is involved in must-link constraints with other data points then
        Move these data points to cluster $c'$ (where $o_i$ has also been moved).
    end if
end while
```

An execution of this algorithm produces optimal values for the dual variables, and these values are used to compute the impact score $\mathcal{I}_{uv}$ for each pairwise constraint. Unfortunately, solving (10) to optimality might be NP-hard for a wide variety of clustering criteria. Thus, for the lower bounding step of Algorithm 1, one likely must resort to heuristics or valid relaxations to find good approximations.

4 Computational Experiments

To evaluate the usefulness of the impact score defined in (17), we first report experiments conducted with synthetic data. Second, we compare our method with naïve approaches. Third, we evaluate the proposed method with real datasets and discuss the convergence of our algorithm. Lastly, we demonstrate the ability of the proposed methodology to identify the best constraint sets when a collection of constraint sets is available using real data. All datasets are available on a public repository: `https://github.com/rodrigorandel/ssc_lagrangian_score`.

4.1 Experiments with synthetic data

The first experiment follows the fractional factorial experimental design similar to that used in Blanchard et al. (2012) and Santi et al. (2016). The process involves generating 500 two-dimensional datasets with known clustering solutions (i.e., ground-truth labels). Having a set of known ground-truth labels allows the generation of constraint sets with *correct* and *erroneous* pairwise information. The parameters used to generate these datasets are given in Table 1: for every dataset, we first randomly choose its size n and its number k of clusters in $\{100, 200, 300, 400, 500\}$ and $\{2, 5, 10, 15\}$, respectively. Second, we generate p pairwise constraints, q among them being erroneous, and the other $p-q$ being correct, with p chosen at random in $\{\frac{5n}{100}, \frac{10n}{100}, \frac{15n}{100}, \frac{20n}{100}\}$ and q in $\{\lceil\frac{5p}{100}\rceil, \lceil\frac{10p}{100}\rceil, \lceil\frac{15p}{100}\rceil, \lceil\frac{20p}{100}\rceil\}$. The results was 17415 pairwise constraints, among which 2219 (12.7%) are erroneous. Although on a real application the amount of erroneous constraints is expected to be smaller (i.e. less than 10%), this experiment also aimed to investigate more complex configuration, and thus, the ratio q of erroneous constraints was allowed up to 20%.

The data generation mechanism is as follows. For each cluster k of each dataset, we first draw coordinates x_k and y_k from a normal distribution $\mathcal{N}(0, 5)$. Then, the x and y coordinates of each data point associated with cluster k are obtained by sampling $\mathcal{N}(x_k, 0.5)$ and $\mathcal{N}(y_k, 0.5)$ respectively. The pairwise constraints (correct and erroneous) are randomly generated with an equal number of cannot-link and must-link constraints. More precisely, the erroneous constraints are obtained by flipping their meaning in the ground-truth, i.e., given a pair of data points, a cannot-link constraint is created if the points have the same ground-truth label. Otherwise, a must-link constraint is created.

Table 1 Experimental Design.

Characteristics	**Values**
Size n of the dataset	$\{100, 200, 300, 400, 500\}$
Number k of clusters	$\{2, 5, 10, 15\}$
Number p of pairwise constraints (as a percentage of n)	$\{5\%, 10\%, 15\%, 20\%\}$
Number q of erroneous constraints (as a percentage of p)	$\{5\%, 10\%, 15\%, 20\%\}$

For each one of these 500 two-dimensional datasets, we use the sub-gradient optimization method in Algorithm 1 with $m = 1000$ (number of iterations) and $\epsilon = 0.5$. The Euclidean distance is considered as dissimilarity metric between data points. For data clustering, we use the k-medoids model (Kaufman and Rousseeuw, 2009). To accelerate the lower bounding step, we opt for relaxing the integrality constraints (5) by $x_i^c \in [0, 1]$ for all $i = 1, \ldots, n$ and $c = 1, \ldots, k$, and equation (10) is then solved using CPLEX 12.8. Algorithm 2 is used to restore feasibility at the upper bounding step of the sub-gradient algorithm. Upon completion of the optimization, we consider every pair of data points o_u and o_v associated with a pairwise constraint and compute the impact score $\mathcal{I}_{uv}$ according to (17), using $\eta_{best}, \lambda_{best}$ and γ_{best}. If $\mathcal{I}_{uv} < 0$, the constraint associated with the pair (o_u, o_v) is predicted as erroneous, whereas if $\mathcal{I}_{uv} = 0$, the constraint is predicted as correct.

To assess the accuracy of the proposed impact score, we begin by computing the true positive, true negative, false positive and false negative counts across all the constraints: a correct constraint predicted as correct is a *true positive* (TP), an erroneous constraint predicted as erroneous is a *true negative* (TN), an erroneous constraint predicted as correct is a *false positive* (FP), and a correct constraint predicted as erroneous is a *false negative* (FN). Using these numbers, we can evaluate the accuracy of the proposed impact score via the three following standard measures:

- Precision $= \frac{TN}{TN+FN}$;
- Recall $= \frac{TN}{TN+FP}$;
- F1-score $= 2\frac{\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}}$.

Across all datasets, we counted $TN = 2205$, $TP = 15130$, $FN = 66$, and $FP = 14$ which provide a Precision of 0.97, a Recall of 0.99 and a F1-score of 0.98. These numbers clearly demonstrate that the proposed Lagrangian-based impact score is able to assess the informativeness of pairwise constraints, as only 0.63% of erroneous constraints and 0.43% of correct constraints were misclassified. We also investigated why some correct pairwise constraints were mistakenly predicted as erroneous. We found that the majority of these false negatives are attributable to an overlapping of two or more clusters in the ground-truth data. In such situations, the clustering model prefers to merge data objects belonging to different classes, which presumably yields cannot-link constraints to be predicted as incorrect.

In these experiments, we assumed that the number of clusters k was known to the user. It is interesting to note that the proposed Lagrangian-based impact score can also offer a mechanism to provide information about the number of clusters likely present in the grouth-truth data generating mechanism. Indeed, one can consider the proportion of pairwise constraints predicted as erroneous as a tool to predict the right number of clusters, following the idea that a high number of erroneous constraints is an indication that an incorrect number of clusters was adopted by the model. To illustrate, Figure 3 shows the fraction of constraints predicted as erroneous for the experimental datasets with five clusters. The proposed algorithm was executed for each of these instances by varying the number k of clusters from 2 to 10. We observe that the lowest ratio is reached with $k = 5$. We can also observe that the F1-score is maximized with $k = 5$, which provides support for the suggestion of the proportion of constraints predicted as erroneous by the impact score as an additional tool for selecting the right number of clusters. Likewise, if a very large amount of pairwise constraints are classified as erroneous by our impact score, this could indicate to the clustering analyst that the model used is not the most adequate for the data.

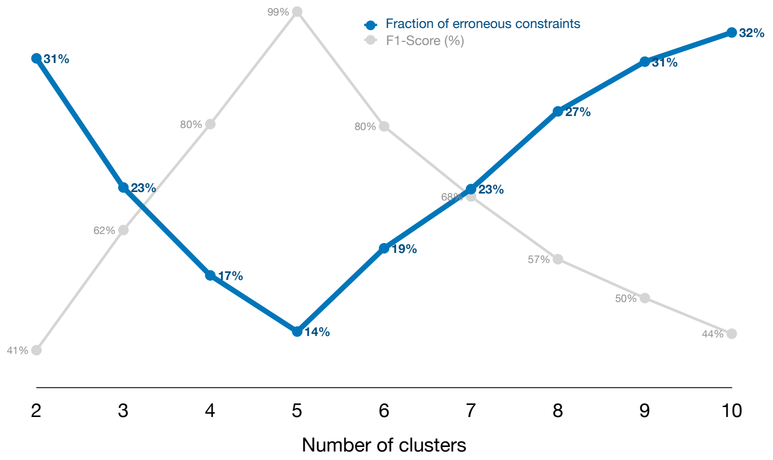


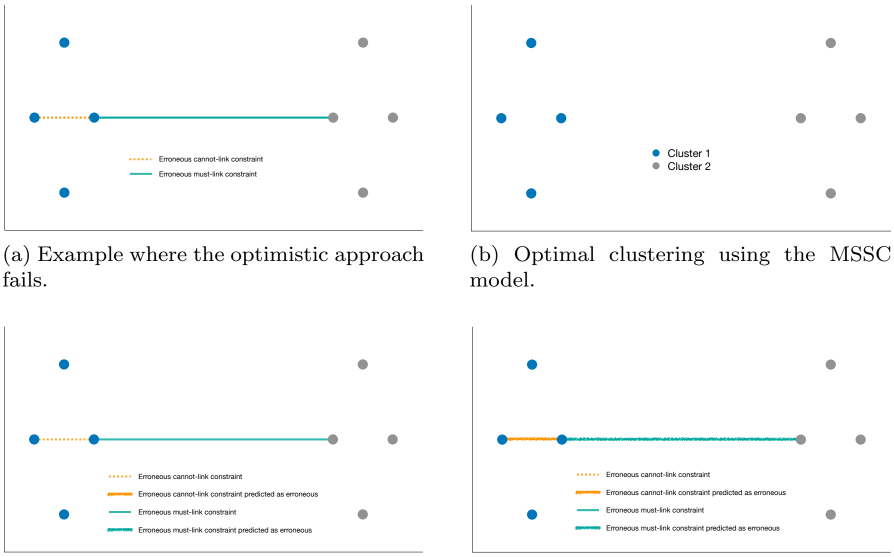
Fig. 3 Fraction of constraints predicted as erroneous and F1-Score obtained by our impact score as a function of the number of clusters. A small proportion of constraints predicted as erroneous suggests the appropriated number of clusters.

4.2 Comparison with optimistic and pessimistic naïve approaches

Whereas we believe that the proposed approach is easy to implement, it may be that some naïve approaches that do not require solving the dual can achieve the same level of accuracy on individual pairwise constraint predictions. We detail here two such (baseline) approaches, and evaluate their performance on the same synthetic datasets.

The optimistic approach. Let $\mathcal{C} = \mathcal{CL} \cup \mathcal{ML}$ denote the constraint set. Assuming that the semi-supervision provided by the expert is correct, the optimistic approach first solves the integer program (2)-(5) for the whole set $\mathcal{C}$ and considers its optimal value Z_B as the *base cost* of the objective function. Then, for each constraint $(o_u, o_v) \in \mathcal{C}$, the integer program is solved again, but with $C' = C \setminus \{(o_u, o_v)\}$ as constraint set which allows an updated optimal value denoted Z_{uv}. The impact score of the optimistic approach is defined as $\mathcal{I}^o_{uv} = Z_{uv} - Z_B$, and we use it as follows. If $\mathcal{I}^o_{uv} < 0$, the constraint associated with the pair (o_u, o_v) is predicted as erroneous. If $\mathcal{I}^o_{uv} = 0$, the constraint is redundant and predicted as correct.

With this approach, even if a constraint is erroneous, removing it from the constraint set may have no impact on the solution cost because the clustering solution can be tied up by other constraints (i.e., assignments will not change). To illustrate, Figure 4(a) shows one erroneous must-link constraint and one erroneous cannot-link constraint. The optimal MSSC partition is shown in Figure 4(b). However, if one tries to partition the illustrated data with COP-k-means taking into account the two erroneous constraints, the data point that contains both cannot-link and must-link constraints is misclassified. The problem with the optimistic approach is thus that if the erroneous must-link constraint is discarded, the solution obtained remains unchanged (i.e., $Z_{uv} = Z_B$) due to the erroneous cannot-link constraint, and the opposite also holds. Consequently, the optimistic approach would yield two false positives by predicting both erroneous constraints as correct (Figure 4(c)). For comparison, the execution of the proposed Lagrangian-based method correctly predicts both constraints as erroneous (Figure 4(d)), and the optimal clustering solution produced by MSSC can thus be retrieved.



(a) Example where the optimistic approach fails.

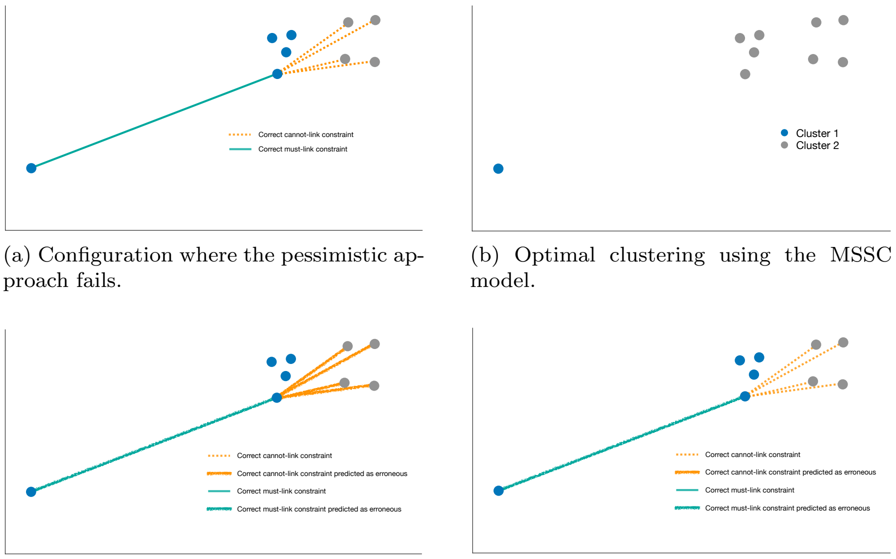
(b) Optimal clustering using the MSSC model.

(c) Predictions using the optimistic approach.

(d) Predictions using the Lagrangian-based impact score.

Fig. 4 Illustration of a case where the optimistic approach fails to identify erroneous constraints. In this example, both the erroneous cannot-link constraint and the erroneous must-link constraint are predicted as correct by the baseline method.

The pessimistic approach. The pessimistic approach begins by assuming that all constraints are erroneous. It begins by defining the base cost Z_B by solving the integer program without any pairwise constraint. Then, for every $(o_u, o_v) \in \mathcal{C}$, the integer program is solved again with only (o_u, o_v) as pairwise constraint and the updated score is denoted Z_{uv}. The impact score of the pessimistic approach is defined as $\mathcal{I}^p_{uv} = Z_B - Z_{uv}$, and we use it as follows. If $\mathcal{I}^p_{uv} < 0$, the constraint associated with the pair (o_u, o_v) is predicted as erroneous. If $\mathcal{I}^p_{uv} = 0$, the constraint is redundant and predicted as correct.



(a) Configuration where the pessimistic approach fails.

(b) Optimal clustering using the MSSC model.

(c) Predictions using the pessimistic approach.

(d) Predictions using the Lagrangian-based impact score.

Fig. 5 Illustration of a case where the pessimistic approach fails to identify correct constraints. In this example, all constraints are incorrectly predicted as erroneous by the baseline method.

With this approach, only one constraint is considered at a time. It is thus possible that every constraint is predicted as erroneous whereas the combination of several constraints would show that they are all correct. To illustrate, consider the data points in Figure 5(a) for which all pairwise constraints are correct. Still adopting the k-means clustering criterion, all constraints would be predicted as erroneous given that the optimal unsupervised clustering solution groups the eight data points on the right into a unique cluster. Doing so leaves a single data point alone, as illustrated in Figure 5(b). Separating the single point produces a low cost for Z_B, which leads to $Z_{uv} > Z_B$ for all $(o_u, o_v) \in \mathcal{C}$. As shown in Figure 5(c), the pessimistic approach yields five false negatives given that the five correct constraints are predicted as erroneous. However, as shown in Figure 5(d), the Lagrangian-based method only predicts the must-link constraint as erroneous. It does so because when the blue data point associated to the cannot-link constraints is grouped with the blue data point at the bottom left, the cannot-link constraints are no longer necessary. The fact that the Lagrangian-based impact score is computed while simultaneously considering all constraints allows correct identification.

The proposed Lagrangian-based impact score $\mathcal{I}_{uv}$ can be seen as a combination of both the pessimistic and optimistic approaches. By considering the whole constraint set, the Lagrangian-based impact score can identify redundant constraints that would be predicted as incorrect in situations like the one shown in Figure 5(a). Besides, it does not experience tied solutions as the one illustrated in Figure 4(a), where erroneous constraints are predicted as correct by the optimist approach. In some scenarios, the optimist and pessimistic approaches may behave in a complimentary fashion as the false positives predicted by the optimistic approach would be correctly predicted as erroneous by the pessimistic approach, whereas the false negatives predicted by the latter would be correctly predicted as correct by the optimistic approach.

It is important to note that the use of heuristics to compute Z_B and Z_{uv} could lead to situations where the impact scores $\mathcal{I}^o_{uv}$ and $\mathcal{I}^p_{uv}$ are slightly smaller than 0, whereas optimal values would have given non-negative scores and thus opposite predictions. To mitigate such a risk, we can adapt the prediction process as follows. Let $s^{\mathcal{CL}}$ and $s^{\mathcal{ML}}$ be the smallest scores reached by a constraint in $\mathcal{CL}$ and $\mathcal{ML}$ respectively. The impact scores $\mathcal{I}^o_{uv}$ and $\mathcal{I}^p_{uv}$ are normalized by dividing by $s^{\mathcal{CL}}$ if $(o_u, o_v) \in \mathcal{CL}$, and by $s^{\mathcal{ML}}$ if $(o_u, o_v) \in \mathcal{ML}$. All normalized impact scores are now at most equal to 1, and a constraint is predicted as erroneous if and only if its normalized impact score is larger than a given threshold τ. We tested this modification of the algorithm via 1000 different values for τ and we report in Figure 6 the F1-scores obtained when using the normalized impact scores. The optimistic approach reaches its maximum F1-score with $\tau = 0.15$, whereas the best F1-score of the pessimistic approach is reached with $\tau = 0$. We have also determined the best threshold value τ for the Lagrangian-based approach based on normalized impact scores, with

$$s^{\mathcal{CL}} = \min_{(o_u,o_v)\in\mathcal{CL}} \mathcal{I}_{uv} \qquad \text{and} \qquad s^{\mathcal{ML}} = \min_{(o_u,o_v)\in\mathcal{ML}} \mathcal{I}_{uv}.$$

As was the case for the pessimistic approach, the best results are obtained with $\tau = 0$.

In Figure 7, we compare the pessimistic and optimistic (with $\tau = 0.15$) approaches with the Lagrangian-based method, for the same 500 experimental datasets. The values of Z_B and Z_{uv} for the two baseline approaches were obtained with the Variable Neighborhood Search (VNS) designed in Randel et al. (2019) for the k-medoids clustering model. VNS is a metaheuristic method that systematically explores increasing neighborhoods from the current solution in order to escape from local optima. In the case of clustering the VNS neighborhoods can be defined by increasing the number of data points that have changed their cluster membership. VNS increases its neighborhood exploration whenever its local descent is not able to find a better solution inside the current neighborhood (see e.g. Costa et al. (2017); Hansen et al. (2009)).

For each baseline method, we give the Precision, Recall and F1-score measures. We find that both the optimistic and pessimistic approaches produce results that are inferior to that of the proposed Lagrangian-based method. As expected, we see from the Recall values that the optimistic approach yields

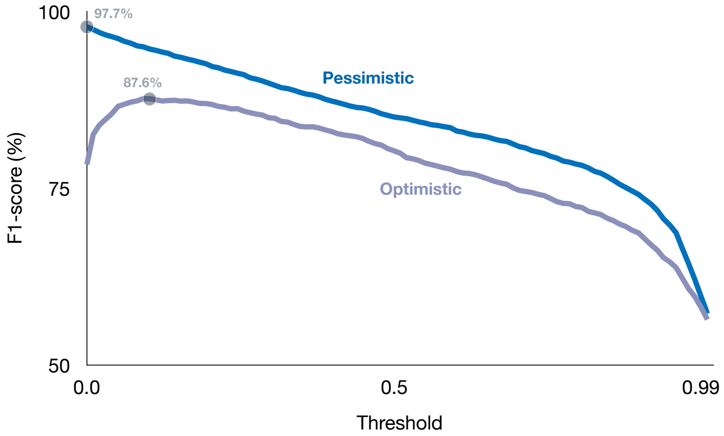


Fig. 6 F1-score obtained by the baseline approaches when predicting erroneous constraints as a function of the threshold (τ). The latter is used for filtering slightly negative scores.

more false positives than the other methods (i.e., erroneous constraints predicted as correct). The pessimistic approach obtains fair results, but with slightly worse classification scores than the Lagrangian-based approach.

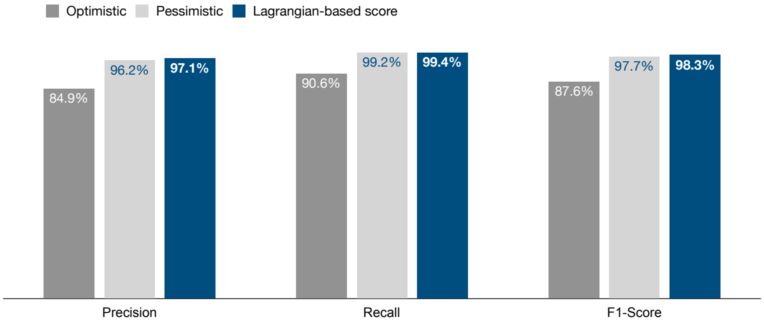


Fig. 7 Performance comparison between the two baseline approaches and the Lagrangian-based method.

4.3 Performance and convergence on real data

In the next series of experiments, we analyze the algorithm's performance for a set of real datasets. The objective of these experiments is threefold: (i) investigate whether the sub-gradient algorithm converges, i.e., verify if the relaxed model (10) can approximate the original problem (2)-(5); (ii) check if the proposed methodology succeeds in determining which constraints are erroneous; and (iii) observe execution time.

We consider eight benchmark datasets, listed in Table 2, that are available at the UCI Machine Learning Repository (Dua and Graff, 2017). For each, as with the synthetic data experiment, we generated p pairwise constraints from which $q = \lceil \frac{1}{3} p \rceil$ are erroneous and $p - q$ are correct with respect to the known ground-truth partitions. We have considered $p = \lceil \frac{15n}{100} \rceil$ and $p = \lceil \frac{20n}{100} \rceil$ which give two constraint sets for every dataset. The final set of constraints for $p = \lceil \frac{20n}{100} \rceil$ is obtained by adding new constraints to the set used for $p = \lceil \frac{15n}{100} \rceil$.

Table 2 Benchmark real datasets

	Samples	Classes	Features
Iris	150	3	4
Wine	178	3	13
Glass	214	3	10
Ionosphere	351	2	34
Control	600	6	60
Balance	625	3	4
Cardiotocography	2126	10	23
Optical	3823	10	61

We obtained the impact scores considering MSSC as underlying clustering model. The lower bounding step of the subgradient method was obtained by solving (10) with a simple adaptation of the k-means heuristic. In particular, instead of iteratively assigning data points to their closest centers, each data point is assigned to the cluster that yields the largest reduction in the expression:

$$\begin{aligned}\sum_{c=1}^{k} \frac{\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \|o_i - o_j\|^2 x_i^c x_j^c}{\sum_{i=1}^{n} x_i^c} &+ \sum_{(o_i,o_j)\in\mathcal{CL}} \sum_{c=1}^{k} \eta_{ij}^c (1 + \epsilon - x_i^c - x_j^c) \\ &+ \sum_{(o_i,o_j)\in\mathcal{ML}} \sum_{c=1}^{k} \lambda_{ij}^c (\epsilon + x_i^c - x_j^c) \\ &+ \sum_{(o_i,o_j)\in\mathcal{ML}} \sum_{c=1}^{k} \gamma_{ij}^c (\epsilon + x_j^c - x_i^c).\end{aligned}$$

The upper bound solution was computed only once with COP-k-means at the start of the algorithm. For these experiments, the stopping criterion of the algorithm was the execution time. Table 3 presents the time allocated to each dataset. For comparison, the baseline methods decribed in Section 4.2 were tested on the same instances. Specifically, COP-k-means was used to test each constraint individually, each run having a time limit fixed to $T_b = \frac{1}{p} T_s$, where T_s is the time limit of the subgradient method.

The results are summarized in Table 3 where we report the final dual gaps at the end of the subgradient algorithm, as well as the F1-scores of our

Table 3 Results for the selected benchmark datasets.

			Impact Score		Baselines (F1-score)	
	Time(s)	p	Gap	F1-score	Pessimistic	Optimistic
Iris	120	23	0.000%	0.799	0.799	0.615
		30	0.000%	0.758	0.733	0.647
Wine	120	27	0.000%	0.705	0.666	0.555
		36	0.000%	0.702	0.685	0.611
Glass	120	33	0.000%	0.842	0.800	0.736
		43	0.001%	0.790	0.790	0.723
Ionosphere	120	53	0.001%	0.727	0.606	0.545
		71	0.001%	0.725	0.691	0.658
Control	240	90	0.000%	0.757	0.709	0.612
		120	0.002%	0.705	0.691	0.685
Balance	240	94	0.002%	0.704	0.704	0.612
		125	0.003%	0.684	0.671	0.624
Cardiotocography	1800	319	0.006%	0.693	0.648	0.608
		426	0.007%	0.659	0.595	0.583
Optical	1800	574	0.005%	0.734	0.723	0.688
		765	0.005%	0.721	0.707	0.696

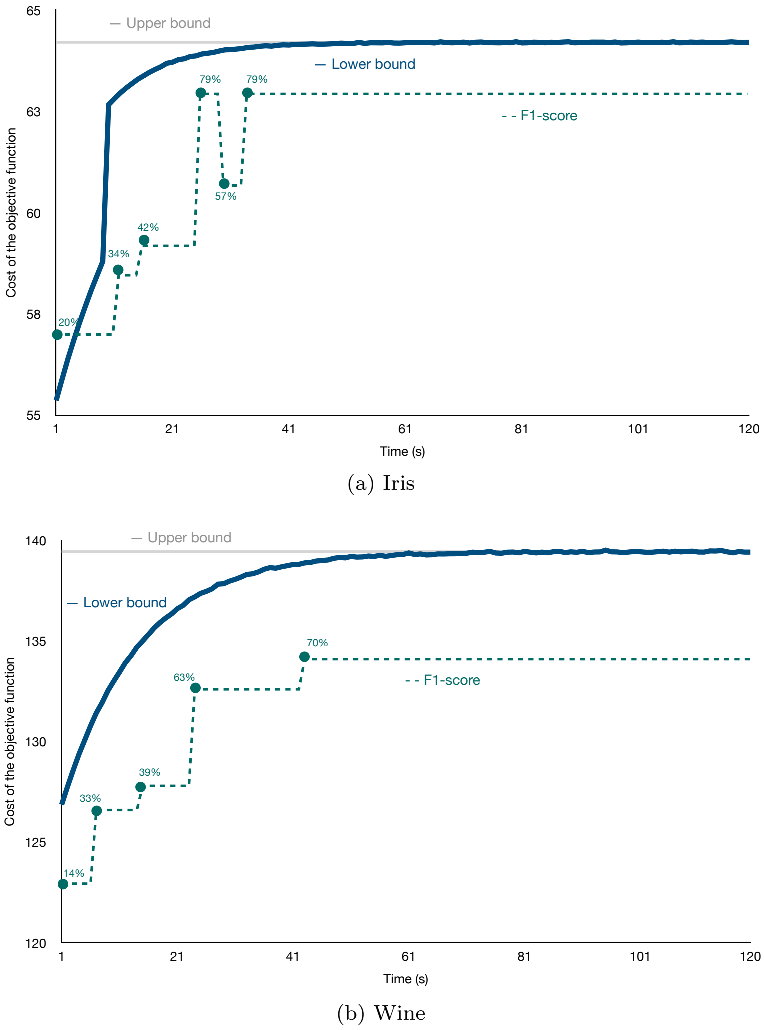
proposed impact score and those obtained by the pessimistic and optimistic approaches. We note from the table that the final dual gaps are quite negligible (max. 0.007%) which means that the final dual values are a rich source of information for the considered clustering problem. Additionally, such small gaps demonstrate that Algorithm 1 converges well in all tested instances. In sum, we find that our Lagrangian-based impact score seems to better assess quality of pairwise constraints than the baseline approaches.

Finally, Figure 8 illustrates the convergence of the subgradient algorithm for the Iris and Wine datasets with $p = \lceil \frac{15n}{100} \rceil$. The figure shows in blue the evolution of the lower bound as the algorithm progresses, and in dark green the evolution of the F1-score based on the values of the dual variables. The algorithm is able to quickly tighten the gap between the upper and lower bounds, suggesting that it could be stopped earlier. Given that the the F1-score shows that stopping our algorithm prematurely may lead to very bad results, it may be ill-advised to do so. A less compromised stopping condition might be to stop the algorithm after the obtained lower bounds appear to stabilize.

4.4 Evaluation of entire constraint sets

As last experiment, we show how to use our Lagrangian-based impact score to evaluate the quality of entire constraint sets. To do so, we use the four datasets Iris, Wine, Glass and Ionosphere (see Table 2)

We begin by noting that pairwise constraints are ultimately used in semi-supervised clustering to guide clustering methods towards obtaining groups



(a) Iris

(b) Wine

Fig. 8 Illustration of the algorithm's evolution for datasets Iris and Wine. The figure shows an upward progression of the F1-score associated to predicting erroneous constraints, and a duality gap reduction as the subgradient algorithm progresses.

that agree with expert knowledge, more typically when unsupervised clustering methods fail to obtain clusters that bear face validity. However, we argue that (sets of) experts might be wrong or uncertain about data relationships and interpretation. Besides, erroneous pairwise constraints can be inadvertently added to a clustering model as a result of a data artifact or noise.

The ultimate goal of our Lagrangian-based impact score is to determine a list of constraints that are merit reviewing. If one follows our methodology to calculate impact scores, constraints most needing of review would be those with the smallest impact score (remember that our impact scores are always non-positive). When faced with a poorly scored (i.e., very negative) pairwise constraint, an expert may decide to either discard it from the constraint set or to keep it, expecting to improve the clustering method ability to retrieve the intended data structures.

To provide an objective assessment of whether impact scores can be useful at selecting pairwise constraints to review, we will use the standard Adjusted Rand Index (ARI) (Hubert and Arabie, 1985), which is defined as follows. Let $X_1, \ldots, X_k$ be the ground-truth partition of a dataset of n points into k clusters, and let $Y_1, \ldots, Y_k$ be the partition obtained by solving (2)-(5) with constraint set $\mathcal{C}$. Also, let $a_i = |X_i|$ and $b_i = |Y_i|$ for all $i = 1, \ldots, k$, and let $c_{ij} = |X_i \cap Y_j|$ for all i and j in $\{1, \ldots, k\}$. The ARI is then computed as follows:

$$\text{ARI} = \frac{\sum_{ij}\binom{n_{ij}}{2} - (\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2})/\binom{n}{2}}{\frac{1}{2}(\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}) - (\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2})/\binom{n}{2}}.$$

Further, let us define $\mathcal{I}(\mathcal{C})$ as the impact score of a constraint set $\mathcal{C}$ calculated as the sum of the impact scores over all constraints, that is

$$\mathcal{I}(\mathcal{C}) = \sum_{(o_u, o_v) \in \mathcal{C}} \mathcal{I}_{uv}.$$

In the first part of our experiments, we generate for each benchmark dataset 100 constraint sets, each composed by 25 randomly selected *correct* constraints. As such, these pairwise constraints are all supposed to increase the clustering performance (e.g. ARI). Once our methodology is applied, each pairwise constraint is given an impact score. Recall that pairwise constraints with negative impact scores are those that are most inconsistent with the unsupervised clustering solution. Such negatively scored (but correct) constraints are then called to be reexamined by the expert who should keep them within the clustering model as they incorporate the expert's knowledge in the clustering solution.

Figure 9 shows for the k-medoids model ARIs with standard box-and-whisker plots, when the whole collection of 100 constraint sets is used, and when only the 50 constraint sets with *smallest* impact score are used. As mentioned in Section 2, Davidson et al. (2006) propose to evaluate the quality of a constraint set by using a coherence measure. We also show in Figure 9 the ARIs for each data set when using the 50 constraint sets with highest coherence measure. We can observe that the Lagrangian-based impact score performed better on the task of identifying the correct constraint sets which are more helpful to guide the algorithm towards the ground-truth partition. In fact, the impact score was always capable of finding the best constraint set from the entire collection of 100 constraint sets. More precisely, the best constraint set, i.e. that one yielding the best ARI, was ranked #1 by our impact score for the Iris and Wine datasets, #2 for Glass and #4 for Ionosphere.

Finally, we repeat the same approach except that we now generate constraint sets composed of 25 randomly generated *erroneous* constraints - constraints which should eventually be discarded by an expert after review. Figure

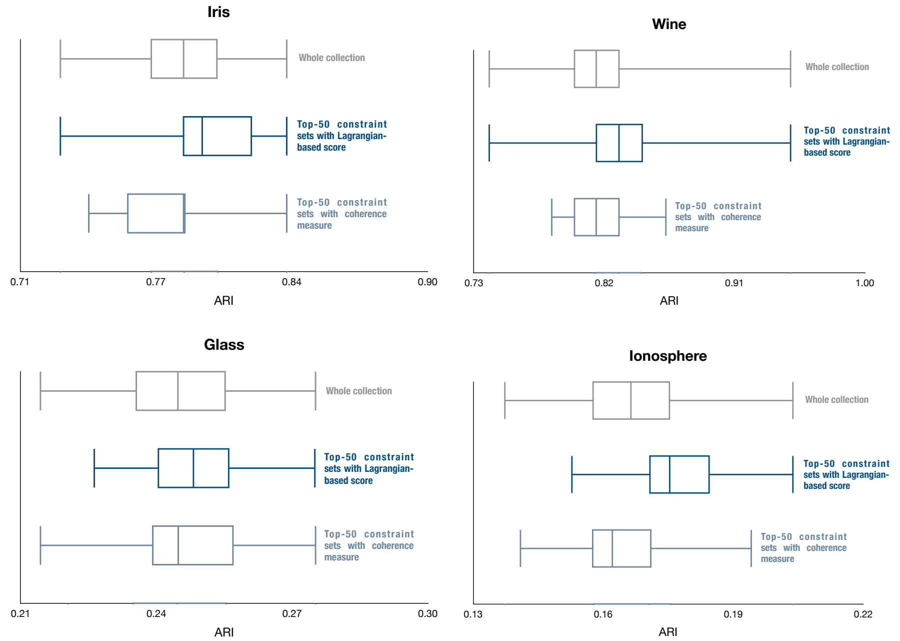


Fig. 9 Comparison of the ARIs for the whole collection of 100 constraint sets of 25 **correct** constraints, and for the top 50 constraint sets selected by the impact score and Davidson's coherence measure.

10 shows ARIs with standard box-and-whisker plots, when the whole collection of 100 constraint sets is used, and when only the 50 constraint sets with *highest* impact score and smallest coherence measure are used.

The Lagrangian-based impact score proved to be more effective at identifying the most degrading sets of erroneous constraints. For all datasets, its top 50 selection included the constraint set with the highest ARI, in addition of having obtained the highest median ARI, overall. Furthermore, its worst selected constraint set (lowest ARI) was always better than the worst set selected by the coherence measure within its top 50.

As further analysis, we indicate in Table 4 the proportions of cannot-link (columns CL) and must-link (columns ML) constraints in the selected sets. We observe that these proportions are very similar for the three experimented methods. Hence, the gain in performance obtained by using the Lagrangian-based score rather than the coherence measure seems to be due to the quality of the selected constraints.

In summary, we have shown that the impact score obtained from the proposed Lagrangian-based model is capable of detecting the most informative constraint sets, rejecting those that degrade the clustering performance and keeping those that help finding the unknown group structures.

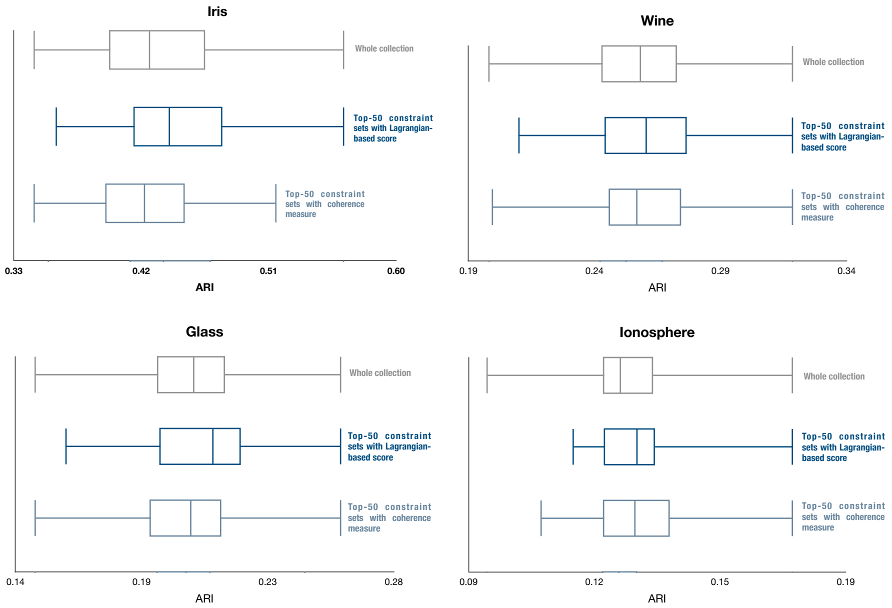


Fig. 10 Comparison of the ARIs for the whole collection of 100 constraint sets of 25 **erroneous** constraints, and for the top 50 constraint sets selected by the impact score and Davidson's coherence measure.

Table 4 Proportion of cannot-link and must-link constraints in the selected sets.

	Iris		Wine		Glass		Ionosphere	
	CL	ML	CL	ML	CL	ML	CL	ML
Whole Collection	33.6%	66.4%	34.8%	65.2%	24.6%	75.4%	53.4%	46.6%
Top 50 with Lagrangian score	33.2%	66.8%	34.5%	65.5%	23.1%	76.9%	54.2%	45.8%
Top 50 with coherence score	32.3%	67.7%	34.0%	66.0%	24.0%	76.0%	53.6%	46.4%

5 Conclusion

We proposed a Lagrangian-based procedure and impact score for assessing the quality of semi-supervision in clustering. The procedure addresses an important issue in semi-supervised clustering applications: the incorporation by experts of constraints which degrade the clustering solution. To help experts identify which pairwise constraints from a set should be revised, the technique estimates the quality of pairwise constraints by exploiting the dual variables of the Lagrangian relaxation of a constrained integer programming formulation of the clustering problem. The impact of each pairwise constraint is computed using a sub-gradient algorithm that optimizes the Lagrangian relaxation. To demonstrate the effectiveness of our approach, we conducted several experiments on synthetic and real data. We also compared our approach to that of prior methods, which do not enable the evaluation of individual pairwise constraints of a set but rather evaluate the set as a whole. We find across these experiments that the method is robust.

In summary, our approach provides valuable information regarding the usefulness of pairwise clustering constraints. The quality of this information however depends on how much time the sub-gradient algorithm is allowed to run in order to refine that information. Besides, our methodology is arguably connected to the ability of the chosen clustering model to recover the underlying structure of the data. Therefore, our results are expected to be more reliable if an appropriated clustering model is adopted.

Finally, we would like to remark that although our discussion in this paper is focused on data partitioning with hard semi-supervised pairwise constraints, i.e., which must be satisfied, our impact score can be adapted, for instance, to *fuzzy clustering* (Bezdek, 1981; Pinheiro et al., 2020), for which the assignment variables are relaxed allowing the data points to belong to more than one cluster with different membership degrees that represent the likelihood of the data point belonging to that cluster. Moreover, another option is to use the impact score in conjunction with algorithms to *soft-constrained clustering* models in which the pairwise constraints, namely should-link and should-not-link, do not need to be necessarily satisfied (Campello et al., 2013; Grossi et al., 2017b). We believe that in this setting our impact score might serve as a warm-start information to SSC algorithms providing them in advance which are the most critical constraints to be first explored for violation.

Acknowledgements The authors would like to thank three anonymous referees for their valuable and constructive suggestions. This research was enabled in part by support provided by Calcul Québec (`https://www.calculquebec.ca`) and Compute Canada (`www.computecanada.ca`).

References

Aggarwal CC (2015) Data Mining. Springer International Publishing, DOI 10.1007/978-3-319-14142-8

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

Aloise D, Hansen P, Liberti L (2010) An improved column generation algorithm for minimum sum-of-squares clustering. Mathematical Programming 131(1-2):195–220, DOI 10.1007/s10107-010-0349-7

Anil J, Rong J, Radha C (2015) Semi-Supervised Clustering, CRC Press, book section Semi-Supervised Clustering. DOI 10.1201/b19706-26

Ares ME, Parapar J, Barreiro A (2012) An experimental study of constrained clustering effectiveness in presence of erroneous constraints. Information Processing & Management 48(3):537–551, DOI 10.1016/j.ipm.2011.08.006

Avella P, Sassano A, Vasil'ev I (2007) Computational study of large-scale p-median problems. Mathematical Programming 109(1):89–114, DOI 10.1007/s10107-005-0700-6

Basu S, Banerjee A, Mooney RJ (2002) Semi-supervised clustering by seeding. In: Proceedings of the Nineteenth International Conference on Machine Learning, Morgan Kaufmann Publishers Inc., 656012, pp 27–34

Basu S, Banerjee A, Mooney RJ (2004) Active Semi-Supervision for Pairwise Constrained Clustering, Society for Industrial and Applied Mathematics, pp 333–344. DOI 10.1137/1.9781611972740.31

Basu S, Bilenko M, Banerjee A, Mooney RJ (2006) Probabilistic semi-supervised clustering with constraints. Semi-supervised learning pp 71–98

Bertsimas D, Tsitsiklis J (1997) Introduction to Linear Optimization, 1st edn. Athena Scientific

Bezdek J (1981) Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York, NY

Blanchard SJ, Aloise D, DeSarbo WS (2012) The heterogeneous p-median problem for categorization based clustering. Psychometrika 77(4):741–762, DOI 10.1007/s11336-012-9283-3

Brucker P (1978) On the complexity of clustering problems. In: Henn R, Korte B, Oettli W (eds) Optimization and Operations Research, Springer Berlin Heidelberg, Berlin, Heidelberg, pp 45–54, DOI 10.1007/978-3-642-95322-4_5

Campello RJ, Moulavi D, Zimek A, Sander J (2013) A framework for semi-supervised and unsupervised optimal extraction of clusters from hierarchies. Data Mining and Knowledge Discovery 27(3):344–371

Christou IT (2011) Coordination of cluster ensembles via exact methods. IEEE Trans Pattern Anal Mach Intell 33(2):279–93, DOI 10.1109/TPAMI.2010.85

Costa LR, Aloise D, Mladenović N (2017) Less is more: basic variable neighborhood search heuristic for balanced minimum sum-of-squares clustering. Information Sciences 415:247–253

Davidson I (2012) Two approaches to understanding when constraints help clustering. In: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '12, ACM, 2339734, pp 1312–1320, DOI 10.1145/2339530.2339734

Davidson I, Ravi SS (2005) Clustering with constraints: Feasibility issues and the k-means algorithm. In: SDM, Society for Industrial and Applied Mathematics, DOI 10.1137/1.9781611972757.13

Davidson I, Ravi SS (2006) Identifying and generating easy sets of constraints for clustering. In: Proceedings of the 21st national conference on Artificial intelligence - Volume 1, AAAI Press, 1597593, pp 336–341

Davidson I, Wagstaff KL, Basu S (2006) Measuring constraint-set utility for partitional clustering algorithms. In: Fürnkranz J, Scheffer T, Spiliopoulou M (eds) Knowledge Discovery in Databases: PKDD 2006, Springer Berlin Heidelberg, Berlin, Heidelberg, pp 115–126

Delattre M, Hansen P (1980) Bicriterion cluster analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-2(4):277–291, DOI 10.1109/TPAMI.1980.4767027

Dua D, Graff C (2017) UCI machine learning repository. URL `http://archive.ics.uci.edu/ml`

Edwards AWF, Cavalli-Sforza LL (1965) A method for cluster analysis. Biometrics 21(2):362–375, URL http://www.jstor.org/stable/2528096

Fisher ML (1981) The lagrangian relaxation method for solving integer programming problems. Management Science 27(1):1–18, DOI 10.1287/mnsc.27.1.1, URL https://doi.org/10.1287/mnsc.27.1.1, https://doi.org/10.1287/mnsc.27.1.1

Fisher RA (1936) The use of multiple measurements in taxonomic problems. Annals of eugenics 7(2):179–188

García S, Labbé M, Marín A (2011) Solving large p-median problems with a radius formulation. INFORMS Journal on Computing 23(4):546–556, DOI 10.1287/ijoc.1100.0418

Grossi V, Romei A, Turini F (2017a) Survey on using constraints in data mining. Data Mining and Knowledge Discovery 31(2):424–464, DOI 10.1007/s10618-016-0480-z

Grossi V, Romei A, Turini F (2017b) Survey on using constraints in data mining. Data mining and knowledge discovery 31(2):424–464

Hansen P, Jaumard B (1997) Cluster analysis and mathematical programming. Mathematical Programming 79(1-3):191–215, DOI 10.1007/bf02614317

Hansen P, Brimberg J, Urošević D, Mladenović N (2009) Solving large p-median clustering problems by primal–dual variable neighborhood search. Data Mining and Knowledge Discovery 19(3):351–375

Held M, Wolfe P, Crowder HP (1974) Validation of subgradient optimization. Mathematical Programming 6(1):62–88, DOI 10.1007/bf01580223

Hubert L, Arabie P (1985) Comparing partitions. Journal of Classification 2(1):193–218, DOI 10.1007/BF01908075

Kariv O, Hakimi SL (1979) An algorithmic approach to network location problems. II: the p-medians. SIAM J Appl Math 37:539–560, DOI 10.1137/0137041

Kaufman L, Rousseeuw PJ (2009) Finding groups in data: an introduction to cluster analysis, vol 344. John Wiley & Sons

Kim S, Blanchard SJ, DeSarbo WS, Fong DK (2013) Implementing managerial constraints in model-based segmentation: extensions of Kim, Fong, and DeSarbo (2012) with an application to heterogeneous perceptions of service quality. Journal of Marketing Research 50(5):664–673

Kochetov Y, Ivanenko D (2005) Computationally Difficult Instances for the Uncapacitated Facility Location Problem, Springer US, Boston, MA, pp 351–367. Operations Research/Computer Science Interfaces Series

Mallapragada PK, Jin R, Jain AK (2008) Active query selection for semi-supervised clustering. In: 2008 19th International Conference on Pattern Recognition, IEEE, pp 1–4, DOI 10.1109/ICPR.2008.4761792

Pinheiro DN, Aloise D, Blanchard SJ (2020) Convex fuzzy k-medoids clustering. Fuzzy Sets and Systems 389:66–92

Randel R, Aloise D, Mladenović N, Hansen P (2019) On the k-medoids model for semi-supervised clustering. In: Sifaleras A, Salhi S, Brimberg J (eds) Variable Neighborhood Search, Springer International Publishing, Cham, pp 13–27

Resende MGC, Werneck RF (2007) A fast swap-based local search procedure for location problems. Annals of Operations Research 150(1):205–230, DOI 10.1007/s10479-006-0154-0

Santi É, Aloise D, Blanchard SJ (2016) A model for clustering data from heterogeneous dissimilarities. European Journal of Operational Research 253(3):659–672

Shor NZ, Kiwiel KC, Ruszcayǹski A (1985) Minimization methods for non-differentiable functions. Springer-Verlag

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, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, ICML '01, pp 577–584

Wagstaff KL (2007) Value, cost, and sharing: Open issues in constrained clustering. In: Džeroski S, Struyf J (eds) Knowledge Discovery in Inductive Databases, Springer Berlin Heidelberg, Berlin, Heidelberg, pp 1–10

Xiong C, Johnson DM, Corso JJ (2017) Active clustering with model-based uncertainty reduction. IEEE Transactions on Pattern Analysis and Machine Intelligence 39(1):5–17, DOI 10.1109/TPAMI.2016.2539965

Xiong S, Azimi J, Fern XZ (2014) Active learning of constraints for semi-supervised clustering. IEEE Transactions on Knowledge and Data Engineering 26(1):43–54, DOI 10.1109/tkde.2013.22

Zhu X, Goldberg AB, Brachman R, Dietterich T (2009) Introduction to Semi-Supervised Learning. Morgan and Claypool Publishers