Les Cahiers du GERAD ISSN: 0711–2440

A Lagrangian-based score for assessing the quality of pairwise constraints in SSC

R. Randel, D. Aloise,
S. J. Blanchard, A. Hertz

G–2019–96

December 2019

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

A Lagrangian-based score for assessing the quality of pairwise constraints in SSC

Rodrigo Randel [a]

Daniel Aloise [a,b]

Simon J. Blanchard [c]

Alain Hertz [b,d]

[a] *Département de Génie Informatique et Génie Logiciel, Polytechnique Montréal (Québec) Canada, H3C 3A7*

[b] *GERAD, Montréal (Québec), Canada, H3T 2A7*

[c] *McDonough School of Business, Georgetown University, Washington, DC 20057, USA*

[d] *Département de Mathématiques et de Génie Industriel, Polytechnique Montréal (Québec) Canada, H3C 3A7*

rodrigo.randel@polymtl.ca
daniel.aloise@polymtl.ca
sjb247@georgetown.edu
alain.hertz@polymtl.ca

December 2019
Les Cahiers du GERAD
G–2019–96

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 guide the algorithm to a more useful and meaningful solution. For instance, he may specify that two points cannot be part of the same cluster (i.e., cannot-link constraint) or two points must be part of the same clusters (i.e., must-link constraint). A key challenge for users of 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 wrong. In the present work, we propose a method to score each must-link and cannot-link pairwise constraint and help users identify which constraints should be amended or removed. Using synthetic experimental examples and real data, we show that the scoring method can successfully identify constraints that should be removed.

Keywords: Clustering, semi-supervised, pairwise constraints, Lagrangian duality

Résumé : Les algorithmes de partitionnement de données aident à identifier des sous-groupes homogènes en ce sens que les données de chaque groupe partagent des caractéristiques communes. Dans certains cas, on dispose d'information supplémentaire sur la relation entre certains sous-ensembles de données. Par exemple, lors de l'utilisation d'un algorithme de partitionnement semi-supervisé, un expert peut fournir des informations supplémentaires pour contraindre la solution recherchée en fonction de ses connaissances et guider ainsi l'algorithme vers une solution plus significative. L'expert peut ainsi spécifier des contraintes par paires en ce sens qu'il peut imposer que deux points ne fassent pas partie d'un même groupe ou, qu'au contraire, ces deux points doivent impérativement faire partie d'un même groupe. Un défi majeur pour les utilisateurs d'algorithmes d'apprentissage semi-supervisés, cependant, est que l'ajout de contraintes inexactes ou conflictuelles peut diminuer la précision du partitionnement généré et on sait peu de choses sur la façon de détecter si les contraintes imposées par des experts sont éventuellement erronées. Dans le présent travail, nous proposons une méthode permettant d'évaluer individuellement chacune des contraintes par paires et aider ainsi les utilisateurs à identifier celles qui doivent être modifiées ou supprimées. À l'aide d'exemples expérimentaux synthétiques et de données réelles, nous montrons que la méthode d'évaluation proposée permet d'identifier avec succès les contraintes erronées.

Acknowledgments: This research was enabled in part by support provided by Calcul Québec (https://www.calculquebec.ca) and Compute Canada (www.computecanada.ca).

1 Introduction

A common typology is to allocate 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 and are successfully applied in many data analysis tasks, the labels are usually scarce and very time-consuming/expensive to generate, since this requires, in most cases, a human expert acting as the 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.

In a third paradigm, it is possible that there is 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. In *semi-supervised learning*, one can generate a classification function using both labeled and unlabeled data. Typically, this is done by incorporating knowledge from 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 in this fashion aims to combine the advantages of unsupervised and supervised learning into a powerful and inexpensive technique.

To 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 $O = \{o_1, \ldots, o_n\}$ 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 partitioning O into k clusters ($P_k = \{C_1, C_2, \ldots, C_k\}$) such that:

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

(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 $f : \mathcal{P}(O, k) \to \mathbb{R}$, 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 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 threshold min-sum or max-sum functions. For example, the *minimum sum-of-squares clustering* technique (MSSC), which is based on 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

[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.

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. 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 the clustering performance (Davidson et al., 2006; Davidson and Ravi, 2006). This 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 is 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).

In order to reduce the possible negative effects of constraints sets in constrained clustering, Zhang et al. (2019) recently proposed a deep learning framework. Assuming that all constraints are correct, which was indeed the case in their experiment (the constraints are generated randomly from the ground-truth partition), good clustering results are obtained. However, it is important to note that in the presence of erroneous constraints, since no mechanism is proposed to identify such constraints, their algorithm can suffer from contradictory and inaccurate information, degrading the quality of the solutions, because their method aims to learn a representation of the data which respects all the constraints, making no distinction between correct and incorrect.

To illustrate the consequences of having inaccurate constraint, 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

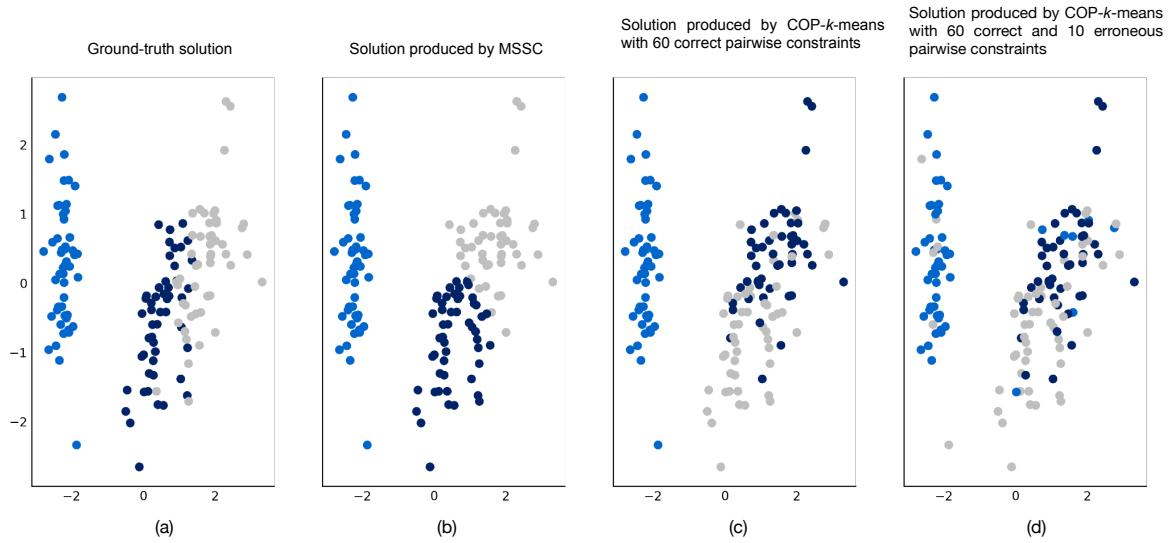


Figure 1: Partitions obtained with and without correct or erroneous pairwise constraints.

well separate the two other clusters. Figure 1(c) illustrates the partition obtained by using the popular COP-k-means 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.

The objective of 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.

2 Constraint inclusions in learning models

When using semi-supervised learning clustering (SSC), a big challenge is to identify useful constraints as relying on domain experts can be difficult on large data (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. Still, active learning methods usually 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. This 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 the work conducted by Mallapragada et al. (2008), this idea is enhanced to use the similarity between a pair of data points as the confidence level to create a must-link constraint. The work of Xiong et al. (2014) 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, an active learning method expands these neighborhoods by selecting informative points and querying the oracle about their relationship with their neighbors. In both cases, the active learning algorithms must begin with a small set of pairwise information that will serve as the foundation to increase the supervision and direct the algorithm in the correct course (Xiong et al., 2017).

The requirement of background information, regardless of whether it originated from the domain expert or was generated by an active learning method, can lead to less desirable 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 is a measure of the incremental information provided by adding a pairwise constraint to the clustering model. 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, with an overlapping segment when the constraint vectors (i.e., vectors connecting their associated points) are projected onto the 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 support for this hypothesis, but only for some dataset, suggesting that more properties related with the utility of pairwise constraints should be developed.

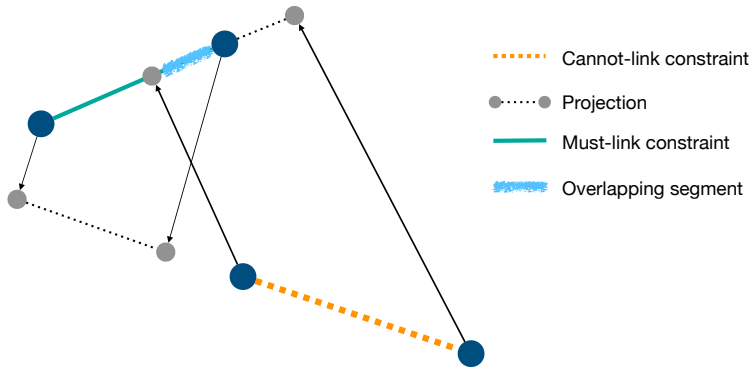


Figure 2: Illustration of Coherence from Davidson et al. (2006): Projection of must-link and cannot-link constraint vectors onto each other.

Informativeness and coherence are not the only measures available to evaluate the helpfulness of constraints. For instance, Davidson (2012) proposes two other approaches. For the first, he suggests counting the number of feasible clustering solutions using Markov Chain Monte Carlo samplers - the idea being to eliminate 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 use the fractional chromatic number of the *constraint graph* to identify constraints to eliminate. 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 since 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). This problem is known as the fractional chromatic number since each vertex can have a fractional assignment of colors. As a last 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.

All three approaches (informativeness, coherence and fractional chromatic number) focus on identifying good constraint sets based on the ability to satisfy them, and are build to quantity constraint sets and not individual pairwise constraints. A consequence is that such techniques for extracting local information on how the constraints interact, as well as techniques for assessing the global impact and effectiveness of each constraint for the target clustering model are not explored in the literature.

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_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 $]0, 1[$:

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

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

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

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).

Classical Lagrangian duality theory associates penalty terms, named *Lagrangian multipliers*, to the problem constraints. Applied to clustering, 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:

$$\begin{aligned} L(\eta, \lambda, \gamma) = \min_{X} f(x) &+ \sum_{(o_i, o_j) \in \mathcal{CL}} \sum_{c=1}^{k} \eta_{ij}^c (1 + \epsilon - x_i^c - x_j^c) \\ &+ \sum_{(o_i, o_j) \in \mathcal{ML}} \sum_{c=1}^{k} \lambda_{ij}^c (\epsilon + x_i^c - x_j^c) \\ &+ \sum_{(o_i, o_j) \in \mathcal{ML}} \sum_{c=1}^{k} \gamma_{ij}^c (\epsilon + x_j^c - x_i^c) \end{aligned} \tag{6}$$

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

$$L_D = \max_{\eta, \lambda, \gamma \leq 0} L(\eta, \lambda, \gamma) \tag{7}$$

for which 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 dual 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:

$$x_u^c + x_v^c \leq 1 + \epsilon + b \qquad \forall c = 1, \ldots, k \tag{8}$$

In doing so, we added a positive value b to the right-hand side of the cannot-link constraints which involve objects o_u and o_v. When $b \geq 1 - \epsilon$, the original cannot-link constraint is deactivated (i.e., completely relaxed) as any value for x_u^c and x_v^c, for all $c \in \{1 \ldots k\}$, satisfies the new constraints. However, if $b < 1 - \epsilon$, o_u and o_v cannot belong to the same cluster according to both the old and new constraints. This replacement of constraints leads to the following new Lagrangian function $L_{uv}^{\mathcal{CL}}(\eta, \lambda, \gamma, b)$:

$$L_{uv}^{\mathcal{CL}}(\eta, \lambda, \gamma, b) = L(\eta, \lambda, \gamma) + \sum_{c=1}^{k} b\eta_{uv}^c \tag{9}$$

for which

$$\frac{\partial L_{uv}^{\mathcal{CL}}(\eta, \lambda, \gamma, b)}{\partial b} = \sum_{c=1}^{k} \eta_{uv}^c. \tag{10}$$

The value calculated in (10) provides an approximation of the effect on Z 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 and the resulting constraints are deactivated if and only if $b \geq 1 - \epsilon$. The Lagrangian function $L_{uv}^{\mathcal{ML}}(\eta, \lambda, \gamma, b)$ becomes:

$$L_{uv}^{\mathcal{ML}}(\eta, \lambda, \gamma, b) = L(\eta, \lambda, \gamma) + \sum_{c=1}^{k} b(\lambda_{uv}^c + \gamma_{uv}^c) \tag{11}$$

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

$$\frac{\partial L_{uv}^{\mathcal{ML}}(\eta, \lambda, \gamma, b)}{\partial b} = \sum_{c=1}^{k} (\lambda_{uv}^c + \gamma_{uv}^c). \tag{12}$$

Negative values for the partial derivatives (10) and (12) suggest that a user can likely improve Z if the constraints are removed from the semi-supervised clustering 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 therefore propose the following impact score $\mathcal{I}_{uv}$ for a pairwise constraint associated with objects o_u and o_v:

$$\mathcal{I}_{uv} = \begin{cases} \sum_{c=1}^{k} \eta_{uv}^c & \text{if } (o_u, o_v) \in \mathcal{CL} \\ \sum_{c=1}^{k} (\lambda_{uv}^c + \gamma_{uv}^c) & \text{if } (o_u, o_v) \in \mathcal{ML}. \end{cases} \tag{13}$$

In the next section, we discuss how to solve the dual problem (7) to calculate the impact score (13).

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 (7). To minimize a function $g : U \subset \mathbb{R} \to \mathbb{R}$, the domain variables are iteratively updated by setting

$$w \leftarrow w + \alpha_\ell \mathfrak{s}(w), \tag{14}$$

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 (7).

Choose initial values for every variable $\eta_{uv}^c, \lambda_{uv}^c$, and γ_{uv}^c (for example, all these values can be set equal to 0), and set $\bar{Z}^* \leftarrow -\infty$ (best upper bound).
for all $\ell = 1$ to m **do**
  Lower bounding step.
  Use current values of the dual variables and equation (6) to determine a lower bound solution $\underline{x}$ of cost $\underline{Z}$.
  if $\underline{Z}$ is the largest lower bound ever found **then**
    Save the dual variables in vectors $\eta_{best}, \lambda_{best}$ and γ_{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 $\mathcal{R}(\underline{x})$ 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**
    $\eta_{uv}^c \leftarrow \eta_{uv}^c + \alpha_\ell \frac{(\bar{Z}^* - \underline{Z})}{\sum_{(i,j)\in\mathcal{CL}} \sum_{c'=1}^{k} (1+\epsilon-\underline{x}_i^{c'}-\underline{x}_j^{c'})^2} (1 + \epsilon - \underline{x}_u^c - \underline{x}_v^c)$.
  end for
  for all $(o_u, o_v) \in \mathcal{ML}$ and all $c \in \{1, \ldots, k\}$ **do**
    $\lambda_{uv}^c \leftarrow \lambda_{uv}^c + \alpha_\ell \frac{(\bar{Z}^* - \underline{Z})}{\sum_{(i,j)\in\mathcal{ML}} \sum_{c'=1}^{k} (\epsilon+\underline{x}_i^{c'}-\underline{x}_j^{c'})^2} (\epsilon + \underline{x}_u^c - \underline{x}_v^c)$
    $\gamma_{uv}^c \leftarrow \gamma_{uv}^c + \alpha_\ell \frac{(\bar{Z}^* - \underline{Z})}{\sum_{(i,j)\in\mathcal{ML}} \sum_{c'=1}^{k} (\epsilon+\underline{x}_j^{c'}-\underline{x}_i^{c'})^2} (\epsilon + \underline{x}_v^c - \underline{x}_u^c)$.
  end for
end for

Algorithm 1 describes the steps of the sub-gradient method for solving (7). It begins by defining initial values for the Lagrangian multipliers $\eta_{uv}^c, \lambda_{uv}^c$ and γ_{uv}^c. 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 (6) 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 second step uses the lower bound solution to recover a feasible solution to (3)–(5). This solution is an upper bound for the original SSC problem. 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.

An execution of this algorithm produces optimal values for the variables of the dual problem, and these values are used to compute the impact score $\mathcal{I}_{uv}$ for each pairwise constraint. Unfortunately, solving (6) 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 to find good approximations.

4 Computational experiments

To evaluate the usefulness of the impact score defined in (13), we first report experiments conducted with synthetic data. Second, we compare our method with naive approaches. Third, 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. For reproducibility purposes, 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 used in Blanchard et al. (2012). 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 $\{2, 5, 10, 15\}$. Second, we generate p correct pairwise constraints, with p randomly chosen in $\{\frac{5n}{100}, \frac{10n}{100}, \frac{15n}{100}, \frac{20n}{100}\}$ Third, we generate q erroneous constraints, with q randomly chosen 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 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 coordintes 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.

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 clustering algorithm, we use the k-medoids clustering model (Randel et al., 2019). 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 (6) is then solved using CPLEX 12.8. The routine suggested in Randel et al. (2019) is used to restore feasibility at the upper bounding step of our 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 (13), 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 $= 2\frac{\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}}$.

The results are summarized in Figure 3. Across all datasets, we counted $TN = 2205$, $TP = 15130$, $FN = 66$, and $FT = 14$ 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 wrong.

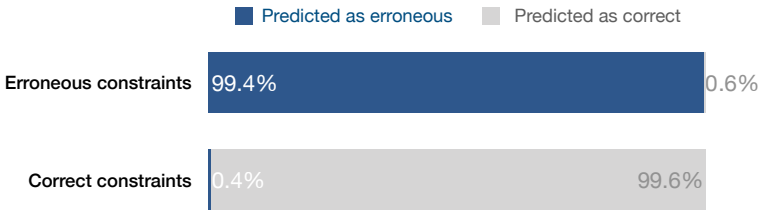


Figure 3: Predictions made with the Lagrangian-based score.

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. 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 a wrong number of clusters was adopted by the model. To illustrate, Figure 4 shows the ratio 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.

4.2 Comparison to optimistic and pessimistic naive approaches

Whereas we believe that the proposed approach is easy to implement, it may be that some naive 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

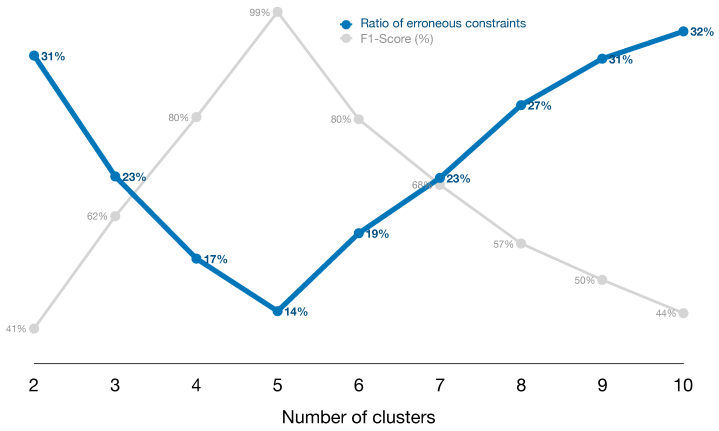


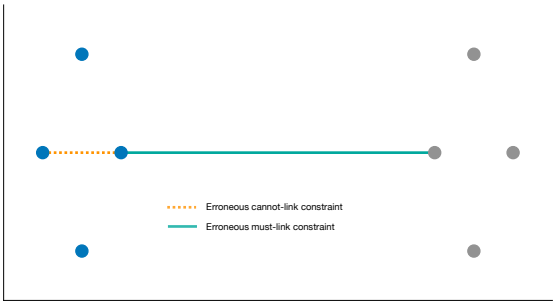
Figure 4: Estimating the number of clusters by counting the number of constraints predicted as erroneous.

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 if $\mathcal{I}^o_{uv} > 0$, the constraint is predicted as correct.

One can easily see the challenge faced 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 may be tied up by other constraints (i.e., the assignments will not change). To illustrate, Figure 5(a) shows one erroneous must-link constraint and one erroneous cannot-link constraint. The optimal partition obtained with MSSC is shown in Figure 5(b). If one adopts the popular k-means clustering criterion, the data point that contains both cannot-link and must-link constraints is misclassified. The problem with the optimistic approach is 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 5(c)). For comparison, the execution of the proposed Lagrangian-based method correctly predicts both constraints as erroneous (Figure 5(d)), and the optimal clustering solution produced by MSSC can thus be retrieved.

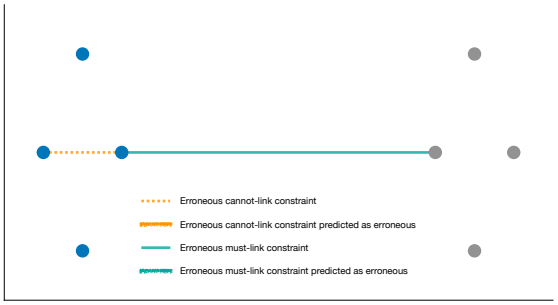
The pessimistic approach. The pessimistic approach begins by assuming that all constraints are erroneous. It begins by defining the base cost Z_B by solving the integer program without any pairwise constraint. Then, for every $(o_u, o_v) \in \mathcal{C}$, the integer program is solved again with only (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 $\mathcal{I}^p_{uv} > 0$, it is predicted as correct.

One can also easily see the challenge faced with this approach. When only one constraint is considered at a time, it is 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 6(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 6(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 6(c), the pessimistic approach yields five false negatives given that the five correct constraints are predicted as erroneous. However, as shown in Figure 6(d), the Lagrangian-based method only predicts the must-link constraint as erroneous. It is able to do so because when the blue data point associated to the cannot-link constraints is grouped with the top-bottom blue data point in the left, the cannot-link constraints are no longer necessary. The fact that the Lagrangian-based impact score is computed considering all constraints together allows correct identification in this case.

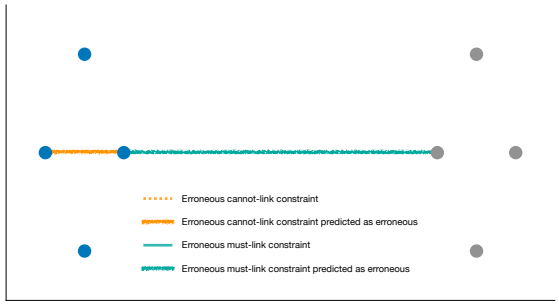


(a) Example where the optimistic approach fails.

(b) Optimal clustering using the MSSC model.

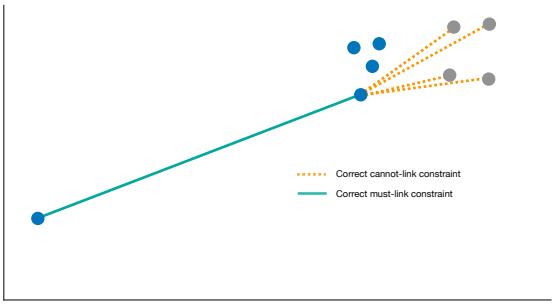


(c) Predictions using the optimistic approach.

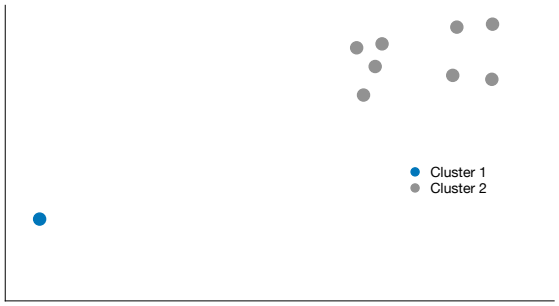


(d) Predictions using the Lagrangian-based score.

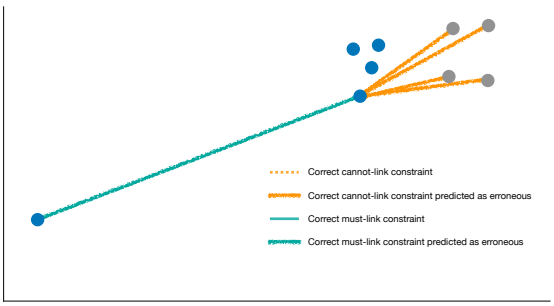
Figure 5: Illustration of a case where the optimistic approach fails to identify erroneous constraints.



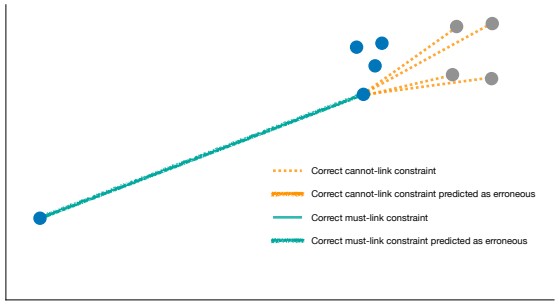
(a) Configuration where the pessimistic approach fails.



(b) Optimal clustering using the MSSC model.



(c) Predictions using the pessimistic approach.



(d) Predictions using the Lagrangian-based score.

Figure 6: Illustration of a case where the pessimistic approach fails to identify correct constraints.

In a way, 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 6(a). Besides, it does not experience tied solutions as the one illustrated in Figure 5(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 7 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 $\tau = 0$.

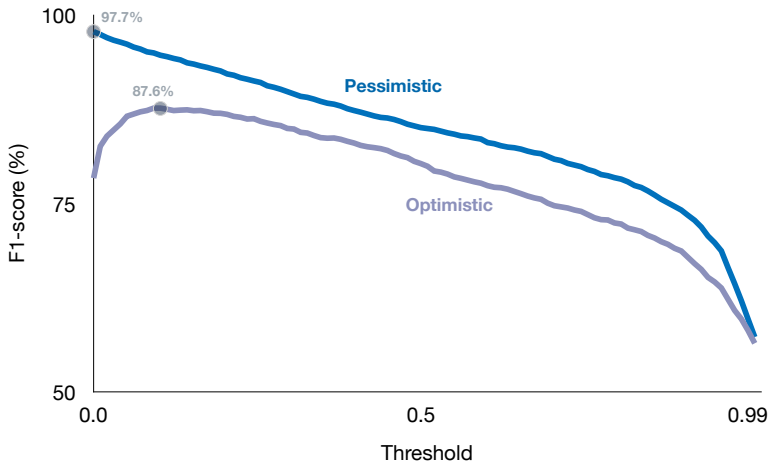


Figure 7: Identifying a good threshold to filter slightly negative scores for the baseline approaches.

In Figure 8, 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 designed in Randel et al. (2019) for the k-medoids clustering model. For each 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 Precision values that the optimistic approach yields 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.

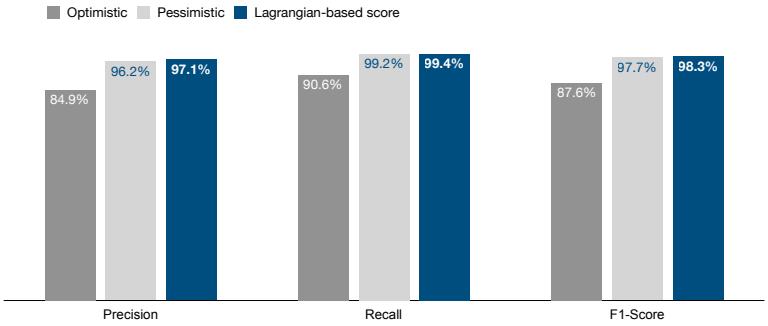


Figure 8: Comparison of the two baseline approaches with the Lagrangian-based method.

4.3 Evaluating entire constraint sets at once

As last experiment, we show how to use our Lagrangian-based impact score to evaluate the quality of a proposed constraint set on real data. For this purpose, we consider the four following classical benchmark datasets available at the UCI Machine Learning Repository (Dua and Graff, 2017): Iris ($n = 150, k = 3$, dimension $= 4$), Wine ($n = 178, k = 3$, dimension $= 13$), Glass ($n = 214, k = 3$, dimension $= 10$) and Ionosphere ($n = 351, k = 2$, dimension $= 34$). For each dataset, a collection of 100 constraint sets was generated, each one containing 25 randomly generated erroneous pairwise constraints.

The quality score $q(\mathcal{C})$ of a constraint set $\mathcal{C}$ is defined as the sum over all constraints of the impact scores, that is

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

Given that all values I_{uv} are non-positive, the above quality score has smaller values for constraint sets with higher total negative impact on the clustering solution. We therefore claim that the best constraint sets are those that provide the highest quality score $q(\mathcal{C})$.

To evaluate whether the highest quality score is helpful, we have to measure the impact of imposing a constraint set to a clustering problem. For this purpose, we 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 computed as follows :

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

Figure 9 shows the ARI with standard box-and-whisker plots, for each dataset, when the whole collection of 100 constraint sets is used, and when only the 50 constraint sets with highest quality 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 show in Figure 9 the ARI for each data set when using the 50 constraint sets with highest coherence measure.

We observe that the use of the Lagrangian-based score for selecting less impactful erroneous constraint sets is effective. For all datasets, we obtained a better average ARI than that obtained when using the coherence measure of Davidson et al. (2006). Moreover, our method always selects within its top 50 constraint sets the one with highest average ARI. Additionally, the worst constraint set included by our technique within its top 50 is always better than the worst set selected by the coherence

measure. It is also worth noting that our method never selects within its top 50 constraint sets the worst possible set regarding the average ARI metric.

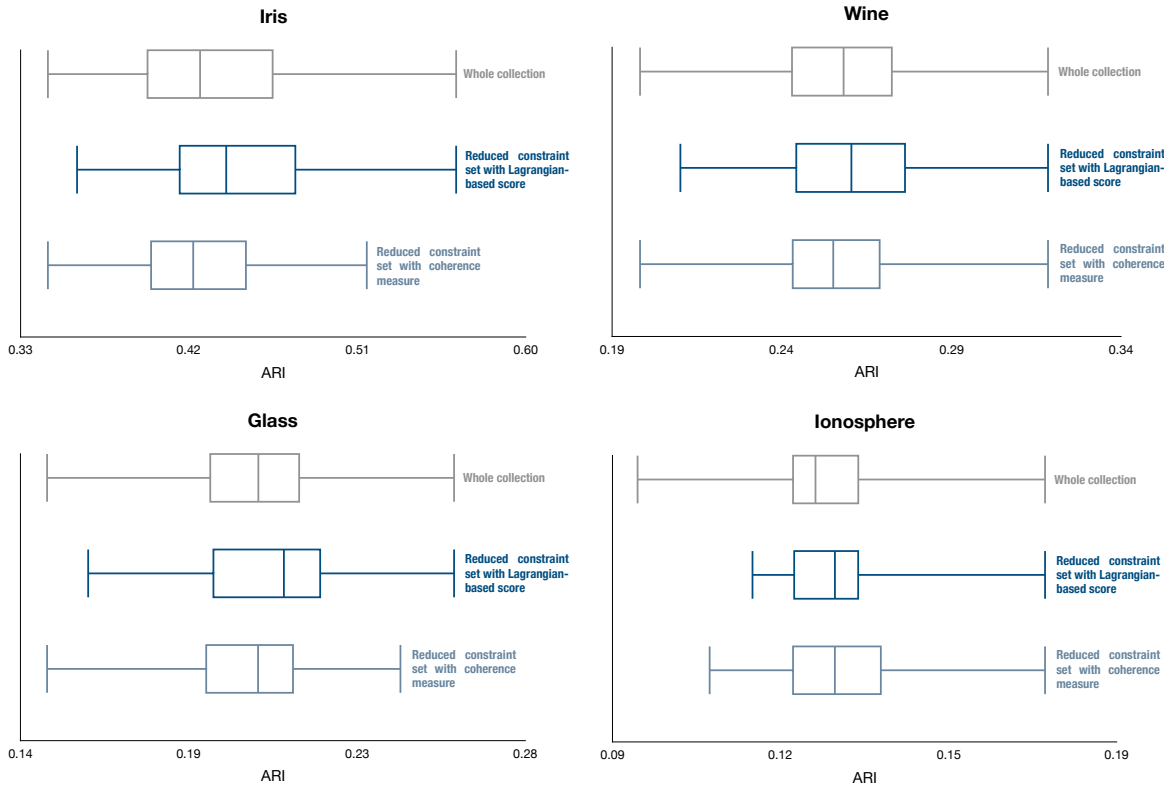


Figure 9: Comparison of the Adjusted Rand Index for several data sets and different constraint sets.

5 Concluding remarks

We proposed a Lagrangian-based procedure 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 removed, 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 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.

References

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

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

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

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

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

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

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

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

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

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

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

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

Zhang H, Basu S, Davidson I (2019) Deep constrained clustering – algorithms and advances. CoRR abs/1901.10061, URL http://arxiv.org/abs/1901.10061, 1901.10061

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