POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Optimization methods to enhance constraint-based semi-supervised clustering

RODRIGO ALVES RANDEL
Département de génie informatique et génie logiciel

Thèse présentée en vue de l'obtention du diplôme de *Philosophiæ Doctor*
Génie informatique

Septembre 2021

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Cette thèse intitulée :

Optimization methods to enhance constraint-based semi-supervised clustering

présentée par **Rodrigo ALVES RANDEL**
en vue de l'obtention du diplôme de *Philosophiæ Doctor*
a été dûment acceptée par le jury d'examen constitué de :

Gilles PESANT, président
Daniel ALOISE, membre et directeur de recherche
Alain HERTZ, membre et codirecteur de recherche
Pierre HANSEN, membre et codirecteur de recherche
Thibaut VIDAL, membre
Emilio CARRIZOSA, membre externe

DEDICATION

To my beloved wife, Juliana, who
has been my source of love,
support and strength ...

ACKNOWLEDGEMENTS

First and foremost, I thank God for blessing me with the opportunity of pursuing this Ph.D. and for his infinite mercy and love in always take care of my life.

My most sincere thanks go to Professors Daniel Aloise, Alain Hertz, and Pierre Hansen. It has been a great honor to be mentored by you. Over the last few years, you have been a great source of inspiration from where I have learned so much. I am truly grateful for all the support, teaching, and advice you have given me. I feel truly fortunate to have worked under your supervision.

I want to express my deepest thanks to my family: my wife Juliana, my parents Marco and Nazaré, and my brother Rafael. It's hard to put into words how much your support, encouragement, and love were fundamental to conclude this work. I am eternally grateful for every sacrifice you made for me. The accomplishment of this doctorate is as much mine as yours.

To all my friends, know that you were essential in my life. You made my journey incredibly fun. Thank you so much for always being there, for always listening to me when I needed it, and for all the laughs and memories we shared. You were a great source of motivation and always made my life lighter and enjoyable.

I was very fortunate to work alongside incredible people. To the collaborators in the works composing this thesis, Nenad Mladenović and Simon J. Blanchard, I appreciate your availability and excellent work. To my colleagues and labmates Thiago, Irving, Leandro, Kim, Théo, Laurent, Clara, Quentin, Allyson and Luciano, I will always be grateful for all your help, shared ideas, advice, and chess matches. Working with you was an immense pleasure.

I would like to extend my gratitude to professors Gilles Pesant, Thibaut Vidal, and Emilio Carrizosa for accepting the invite to compose the jury of my thesis.

Finally, with a heavy heart, I dedicate this thesis in memory of my late aunt Silvia Randel. I will be ever grateful for all the support that she has given me throughout my life and for the encouragement she has given me in pursuing this doctorate. *Tia Silvinha*, it's a deep regret that you will not be there to see me graduate.

RÉSUMÉ

Le clustering est une technique importante de l'analyse des données non supervisée qui permet de récupérer automatiquement la structure sous-adjacente des données. Au cours des deux dernières décennies, il a été démontré que les performances des modèles de clustering peuvent être considérablement améliorées lorsque la tâche est assistée par des informations secondaires, généralement fournies par des experts du domaine. Par conséquent, faire du clustering en présence de connaissances supplémentaires devrait produire des solutions plus conformes aux hypothèses des experts concernant la distribution des données, et ainsi permettre d'obtenir une description des données plus fiable. Cette technique, qui a suscité beaucoup d'intérêt ces dernières années, est connue sous le nom de *clustering semi-supervisé*. Une manière courante d'exprimer les informations secondaires est au moyen de contraintes sur les paires d'objets, appelées formellement contraintes de *must-link* et de *cannot-link*, qui indiquent si une paire d'objets doit être ou ne doit pas être dans les mêmes clusters. Dans cette thèse, nous avons étudié et proposé des techniques analytiques et des algorithmes qui peuvent être utilisés pour améliorer les performances des modèles de clustering semi-supervisés basés sur des contraintes sur les paires d'objets.

Notre première contribution concerne l'utilisation du modèle des k-médoïdes dans un cadre semi-supervisé. Nous avons observé que, bien qu'il s'agisse d'un modèle de clustering connu, il n'avait pas été exploré auparavant pour tirer parti des informations secondaires. Par conséquent, motivés par les avantages du modèle des k-médoïdes, tels que sa flexibilité pour travailler avec des données métriques ou non métriques et sa robustesse aux valeurs aberrantes, nous avons formulé un modèle des k-médoïdes semi-supervisé avec des contraintes sur les paires d'objets, et proposé une heuristique de recherche de voisinage variable pour l'optimiser. Nous avons démontré que le modèle des k-médoïdes pouvait atteindre de bonnes performances de clustering, et que l'algorithme proposé est efficace pour trouver des solutions optimales ou quasi optimales pour le problème considéré.

Dans la deuxième contribution de cette thèse, nous abordons la question de la réalisation d'un clustering semi-supervisé en présence de contraintes erronées. Il est bien connu que certaines contraintes sur les paires d'objets peuvent nuire à la tâche de clustering, car les informations secondaires fournies par les experts sont sujettes à des erreurs humaines. Cependant, en raison de la nature non supervisée du clustering, les utilisateurs de l'application manquent de mécanismes pour identifier si une contrainte donnée est incorrecte, et par conséquent, ont pour seule option d'utiliser toutes les contraintes fournies. Pour atténuer ce problème, nous

avons proposé une méthode pour évaluer quantitativement la qualité des contraintes sur les paires d'objets. Ainsi, nous explorons l'information duale obtenue à partir d'une relaxation lagrangienne de modèles de clustering semi-supervisés basés sur des contraintes, et avons défini un score en utilisant l'impact estimé des contraintes sur les paires d'objets dans la fonction objectif du clustering. Ce faisant, nous avons conçu un outil pour aider les experts du domaine à identifier les contraintes de must-link ou cannot-link impactant négativement la solution du clustering, et qui devrait être examinées.

Enfin, la dernière contribution de cette thèse étend l'exploration de l'information duale du travail précédent pour améliorer les algorithmes d'apprentissage de métriques basés sur le clustering. Lorsque des contraintes sur les paires d'objets sont disponibles, l'objectif de l'apprentissage de la métrique pour le clustering est de trouver des transformations des données telles que les distances entre les paires d'objets de données associées à une contrainte de must-link soient réduites et que les distances entre les paires d'objets de données associées à une contrainte de cannot-link sont augmentées. Nous avons observé que certains défis se posent, notamment, celui de trouver et appliquer des transformations qui préservent autant que possible les propriétés géométriques des données. D'autres défis consistent à choisir la mesure de dissimilarité la plus appropriée pour apprendre la transformation des données, et à determiner quelles sont les contraintes les plus avantageuses à utiliser pour redimensionner les données. Nous avons développé des techniques analytiques et des algorithmes qui exploitent l'information duale pour résoudre ces problèmes, et nous avons démontré que les algorithmes d'apprentissage de métriques bénéficient de leur utilisation.

ABSTRACT

Clustering is an essential unsupervised data analysis technique for automatically retrieving data underlying structures. In the past two decades, it has been demonstrated that the clustering performance can be significantly improved when the task is assisted by side information, which is usually provided by domain experts. As a consequence, clustering in the presence of background knowledge should yield solutions that better suit the experts assumptions regarding the data distribution, thus yielding a more reliable data description. This technique, which has recently attracted much interest, is known as *semi-supervised clustering*. A common way of expressing side information is by means of pairwise constraints, namely *must-link* and *cannot-link* constraints, which indicate whether a pair of data objects must or must not be in the same cluster. In this thesis, we studied and proposed novel analytical techniques and algorithms that enhance the performance of constraint-based semi-supervised clustering models.

Our first contribution concerns with the use of the k-medoids model in a semi-supervised paradigm. We observed that although k-medoids is a well-known clustering model, it has not been previously explored to profit from side information. Therefore, motivated by the advantages of the k-medoids model such as its flexibility for working with metric or non-metric data and its robustness to outliers, we formulated a semi-supervised k-medoids model with pairwise constraints and proposed a Variable Neighborhood Search heuristic for optimizing it. We demonstrated that the k-medoids model is able to achieve good clustering performance, and that the proposed algorithm is efficient in finding optimal or near-optimal solutions for the problem under consideration.

In the second contribution of this thesis, we address the issue of performing semi-supervised clustering in the presence of erroneous constraints. It is well known that some pairwise constraints can harm the clustering task, as the side information provided by experts are subject to human errors. However, due to the unsupervised nature of clustering techniques, application users lack of mechanisms to identify incorrect constraints, and therefore, are left with the sole option of using all provided pairwise constraints. To mitigate this problem, we proposed a method to assess the quality of pairwise constraints quantitatively. We explored the dual information obtained from a Lagrangian relaxation of constraint-based semi-supervised clustering models and defined a score using the estimated impact of pairwise constraints on the clustering objective function. In doing so, we introduced a tool to help domain experts identify which must-link or cannot-link constraints are negatively impacting the clustering

solution and should be reviewed.

Finally, the last contribution of this thesis extends the exploration of dual information from the previous work to enhance clustering-based distance metric learning algorithms. When pairwise constraints are available, the objective of distance metric learning for clustering is to find data transformations, such that the distances between pairs of data objects associated with must-link constraints are reduced, while the distances between pairs of data objects associated with cannot-link constraints are increased. We observed that some challenges arise in doing so, including (i) how to find and apply transformations that would preserve as much as possible the geometrical properties of the data, (ii) which is the most appropriated notion of dissimilarity for learning the data transformation, and (iii) which are the most beneficial constraints one should use to rescale the data. We developed analytical techniques and algorithms that exploit dual information for addressing those issues and demonstrated that distance metric learning algorithms benefit from using them.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ACRONYMS

MSSC	Minimum Sum-of-Squares Clustering
DEC	Deep Embedded Clustering
VNS	Variable Neighborhood Search
SSC	Semi-Supervised Clustering
SSMSSC	Semi-Supervised Minimum-Sum-of-Squares Clustering
SSKMP	Semi-Supervised K-Medoids Problem
ARI	Adjusted Rand Index
TF-IDF	Term Frequency–Inverse Document Frequency

CHAPTER 1 INTRODUCTION

Machine learning techniques are mainly studied under two paradigms: (i) *unsupervised learning*, where no background knowledge is available a priori, and whose objective is generally set to capture the best underlying description of the data; and (ii) *supervised learning*, where the data samples are assisted by ground-truth observations, allowing machine learning models to accurately foresee output values for unseen data samples. Although supervised learning techniques can achieve minimal prediction errors and are successfully applied in numerous data analysis tasks, ground-truth information is often scarce and expensive to obtain [3, 4]. In contrast, the ability of unsupervised learning techniques to automatically identify the inherent data structure make them advantageous for exploratory analysis. However, due to the lack of background information, unsupervised models are usually difficult to evaluate or compare, as well as often being subject to assumptions regarding the data distribution.

In between these two learning concepts emerges another machine learning branch: the one composed of *semi-supervised learning* models. The principal idea of this learning paradigm is to combine the advantages of unsupervised and supervised learning, i.e., the accessibility of unsupervised learning and the background knowledge support of supervised learning, into a powerful and inexpensive approach. Therefore, instead of working plainly with labeled data, semi-supervised learning is intended to create techniques that would learn from both unlabeled and labeled data, where, normally, the latter is scarce. In other words, semi-supervised learning concerns how to combine unlabeled and labeled data, exploring the partial background knowledge, often provided by a domain specialist, to enhance machine learning models. In fact, most semi-supervised learning strategies strive to extend either unsupervised or supervised learning to include this additional expertise [5].

Despite the fact that semi-supervised techniques emerged several decades ago [6, 7, 8], their growth in popularity is recent, and mostly due to the increasing volume of generated data [9]. Since the vast majority of this data is unlabeled and acquiring high-quality labeled information can be costly and time-consuming [10], semi-supervised learning became particularly relevant to a wide variety of application domains [9, 11]. Therefore, the ability to design models that can learn from a reduced amount of side information establishes semi-supervised learning as a machine learning branch of great interest [12]. In addition, a parallel motivation in the research field of semi-supervised learning is its close relation to the fundamental problem of understanding human and animal cognitive processes, since many of them are weakly supervised [10].

1.1 The Clustering Problem

In this thesis, special attention is dedicated to the study of semi-supervised techniques for *clustering*, one of the most popular unsupervised data analysis tasks [4]. Formally, for a given set of data points, clustering methods address the problem of finding subsets, named clusters, that are homogeneous and well separated [13]. Homogeneity asserts that data points assigned to the same cluster are expected to be similar, and separation suggests that data points with distinct cluster memberships should differ one from another. A common requisite of a clustering model is the definition of an appropriate *clustering criterion* that best captures homogeneity and separation for a given dataset. For example, the homogeneity of a cluster can be measured by its *diameter* (i.e., the maximum dissimilarity between two data points of the same cluster), and separation can be measured by the *split* (i.e., the minimum dissimilarity between two data points in different clusters).

Over the years, several clustering strategies have been proposed to address the difficulty of recovering the best data description. Among them, one of the most frequently used approaches is partitioning. Formally, given a set $O = \{o_1, \ldots, o_n\}$ of n data points in an s-dimensional space, partitioning focuses on splitting O into k clusters $C_1, C_2, \ldots, C_k$ such that:

1. $C_j \neq \emptyset$ for all $j = 1, \ldots, k$,
2. $C_i \cap C_j = \emptyset$ for all $1 \leq i < j \leq k$, and
3. $\bigcup_{j=1}^{k} C_j = O$

If the number of clusters k is known, and thus fixed, clustering can be formulated as a mathematical optimization problem whose objective function $f : \mathcal{P}(O, k) \rightarrow \mathbb{R}$, usually called clustering criterion, defines the optimal solution for the problem given by the following [14]:

$$\min\{f(P) : P \in \mathcal{P}(O, k)\}, \tag{1.1}$$

where the set of all k-partitions of O is denoted $\mathcal{P}(O, k)$.

The simplest manner of defining f is using a *representative-based* approach [15]. For this family of clustering models, each cluster designates a cluster representative. Hence, point-cluster proximities are calculated in terms of the dissimilarities between each data point and the cluster representatives, each data point being assigned to its most similar representative. This strategy relies on the assumption that, if good cluster representatives are obtained,

then high-quality clusters can be discovered by assigning the data points to their closest representatives according to some dissimilarity function. Therefore, representative-based clustering algorithms seek to find the set of representatives elements, $\mathcal{Y} = \{y_i, \ldots, y_k\}$, such that the following objective function is minimized:

$$Z = \sum_{i=1}^{n} \min_{c \in \{1 \ldots k\}} d(o_i, y_c), \tag{1.2}$$

where $d(o, y)$ accounts for the dissimilarity between the data point $o \in O$ and a representative point $y \in \mathcal{Y}$. In fact, relying directly on the intuitive notion of dissimilarity is exactly what makes representative-based algorithms the simplest clustering approach [15]. Furthermore, an interesting observation from (1.2) is that if the optimal set of representatives $\mathcal{Y}$ is known a priori, then the optimal assignment may be easy to determine, and vice versa. Many representative-based algorithms indeed explore this circular property by an iterative procedure where potential representatives and candidate assignments are used to improve one another. Such a popular method is the well-known minimum-sum-of-squares-clustering (MSSC) model, popularly known as the k-means model [16]. Given a starting set of k representatives, k-means works by executing the following two steps until convergence:

1. **Assign step**: Assign each data point to its closest representative in $\mathcal{Y}$ in terms of Euclidean distances. Let the corresponding clusters be $C_1 \ldots C_k$.

2. **Optimizing step**: Determine the optimal representative y_c for each cluster C_c that minimizes the sum of squared distances within the clusters. As a result, y_c is updated to the cluster's centroid, i.e., the mean point in the s-dimensional space considering all points assigned to C_c.

1.2 Semi-Supervised Clustering

As a consequence of the lack of external information to validate the clustering solutions, the user of an application typically selects a clustering algorithm off-the-shelf, hoping that it is capable of retrieving a correct group description of the data. For instance, in the k-means model, the objective is to minimize the sum of the squared Euclidean distances between data points and the cluster representatives. This criterion can express both homogeneity and separation [17], but it results in clusters having spherical shapes due to variance minimization.

The incorporation of semi-supervision can guide the learning process and increase the generalization performance of classification models [18]. Thus, semi-supervised learning can help clustering models in the task of recognizing hidden patterns in data more in accordance with

the domain experts' knowledge and beliefs. As a result, experts have a manner of introducing their own data assumptions. The clustering process driven by this side information is called *semi-supervised clustering*. Since different clusterings models may not be equally useful from an application-specific perspective [15], semi-supervised clustering can reduce the impact caused by prior assumptions imposed by such models.

The concept of semi-supervised learning has been intensively studied and successfully applied to many clustering algorithms. Initially, they were designed to incorporate *cluster labels* as background knowledge. This type of supervision is called pointwise information and is usually easier to incorporate since it can be used more naturally in conjunction with existing clustering algorithms [15]. For example, pointwise information may be used to create an initial set $\mathcal{Y}$ of clusters' representatives, which next could be used as a starting point for a clustering algorithm such as k-means [19].

Nevertheless, a more conventional way of formulating side information is through pairwise constraints, where the user provides information regarding the relationship between a pair of data objects. In this sense, a *must-link* constraint indicates that two data points are similar and, therefore, must be assigned to the same cluster. Likewise, a *cannot-link* constraint indicates that a pair of points must be assigned to different clusters. Pairwise constraints arise naturally in many applications, e.g., image retrieval [20], and, in many circumstances, are more practical to obtain than the class labels, since the true label may not be known beforehand [21].

Without loss of generality, for a given set of must-link, $\mathcal{ML}$, and cannot-link, $\mathcal{CL}$, constraints, we can express any semi-supervised clustering model by the following integer programming formulation:

$$Z = \min\ f(\mathbf{x}) \tag{1.3}$$

subject to

$$\sum_{c=1}^{k} x_i^c = 1 \qquad \forall i = 1 \ldots n \tag{1.4}$$

$$x_i^c + x_j^c \leq 1 \qquad \forall (o_i, o_j) \in \mathcal{CL},\ \forall c = 1, \ldots, k \tag{1.5}$$

$$x_i^c - x_j^c = 0 \qquad \forall (o_i, o_j) \in \mathcal{ML},\ \forall c = 1, \ldots, k \tag{1.6}$$

$$x_i^c \in \{0, 1\} \qquad \forall i = 1, ..., n;\ \forall c = 1, \ldots, k \tag{1.7}$$

where f is the clustering criterion to be minimized and the binary decision variables x_i^c refer to the assignment of point o_i to cluster c. Constraints (1.4) ensure that each data point is assigned to exactly one cluster. Cannot-link constraints are expressed by (1.5) and must-link

constraints by (1.6).

Incorporating semi-supervision (e.g., pairwise constraints (1.5) and (1.6)) into the clustering model can restrict the solution space of the original problem. Thus, semi-supervised clustering may also be referred to as *constrained clustering* [22]. A pioneer algorithm to optimize model (1.3)-(1.7) is COP-Kmeans [23]. It modifies the assigning step of k-means to only accept assignments for which no pairwise constraint is violated. With a good demonstrated performance, COP-Kmeans popularized the benefits of pairwise constraints, and it is still a baseline algorithm for semi-supervised clustering studies [24, 25].

Another way of incorporating pairwise supervision is through *distance metric learning*, a technique to automatically learn how to appropriately measure similarities and/or distances in a given data distribution. Accordingly, in the semi-supervised clustering paradigm with pairwise constraints, distance learning methods are designed to find transformations to the data features, so as to move pairs of data objects involved in must-link constraints closer to each other, while moving pairs of data objects involved in cannot-link constraints away from each other, in the metric space under consideration. Thus, distance metric learning can also mitigate the effect of adopting a lesser appropriated clustering model which, alone, is unable to capture the underlying clustering structure of the data.

1.3 Research Objectives

The semi-supervised clustering field has significantly evolved and attracted much attention in recent years [26]. Within the past two decades, numerous works have prompted novel approaches to address the problem, proposing new mechanisms for acquiring the side information, then incorporating it to the clustering pipeline (Chapter 2 presents a extensive literature review on the topic). Despite the achieved success, it has been observed that some clustering models have yet to be studied under the semi-supervised learning paradigm. Furthermore, most of the effort is focused on developing new algorithms and strategies to profit from the pairwise constraints, but several difficulties may arise when working with them, including the presence of inaccurate or erroneous constraints, which can deteriorate the clustering performance. Moreover, application users often have no tools at their disposal to understand the role of constraints and to perceive how the clustering solution is being altered after incorporating this supervision. In fact, domain experts do not have a way of assessing whether their supervision is actually helpful in retrieving better clustering solutions.

Given the above, within the scope of this thesis, the objective is to discuss and propose analytical techniques and algorithms that can be used to enhance the performance of semi-

supervised clustering models and algorithms. More specifically, this is conceived around three main contributions:

1. **Proposing a semi-supervised clustering algorithm for the *k*-medoids model.** The k-medoids model [27] is a classical representative-based model, whose objective is to minimize the sum of dissimilarities between data points and their associated *medoid* (i.e., cluster representative). The main feature of the k-medoids model is that the cluster representatives are selected from within the dataset. Indeed, a great advantage of k-medoids is its breadth of applicability due to its flexibility when defining the dissimilarity matrix. The k-medoids model can be used to cluster metric data, as well as more general data with notions of similarity/dissimilarity. Moreover, k-medoids is well known for its excellent classification rates and robustness to outliers [28].

   Motivated by those properties, and after identifying that the k-medoids model had not been previously examined under the semi-supervised setting, we propose a method for exploring this model. As our first contribution, we demonstrated that k-medoids is able find high-quality clustering solutions when pairwise constraints are available. In addition, we discovered that its clustering accuracy is particularly competitive when compared to more traditional semi-supervised clustering models, such as those based on MSSC.

2. **Assessing the quality of pairwise constraints in semi-supervised clustering.** Although it is reasonable to assume that the clustering performance should improve when a clustering algorithm is assisted by expert knowledge, the presence of inaccurate or conflicting pairwise constraints has been shown to degrade it [2, 29]. 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 inaccuracy of constraints can occur due to human judgment errors ultimately impacting the clustering quality [30], methods that can help users identify which constraints are likely to be wrong are helpful in improving accuracy [31]. On top of that, little is known about which are the most useful and helpful constraints to guide a semi-supervised clustering algorithm, mainly due to the unsupervised nature of the clustering techniques. Hence, users are often left with the sole option of using all available constraints, which can be seen as a "hit or miss" scenario [1].

In our second contribution, we propose a framework to address this important issue of identifying erroneous pairwise constraints. The introduced tool aids domain experts by suggesting the constraints that are likely to harm the clustering process and, thus, should be removed or revised. To this end, we established a quantitative measure of the impact caused by the pairwise constraints. This evaluation is obtained after exploiting the dual variables of a Lagrangian relaxation of the semi-supervised clustering problem.

3. **Enhancing clustering-based distance metric learning with pairwise constraints.** As our third contribution, we identified three crucial issues faced by distance metric learning algorithms for clustering, and proposed to mitigate them by exploring valuable dual information discovered during the development of the previous objective.

   We noticed that the notion of dissimilarity used as foundation for applying metric learning is crucial and has a significant impact on the resulting clustering solution. In other words, distance learning methods depend on the original distance metric under consideration. As an example, one may choose to use pairwise constraints to enhance initial dissimilarities computed from Euclidean distances. However, the enhancement this yields might not compensate for the fact that a more appropriate metric exists for clustering the data. Unfortunately, defining an appropriate distance metric is highly problem-specific, and the users of a clustering algorithm have no mechanism to support their choice. To solve this shortcoming, we propose a *metric fitting score* for measuring how suitable a metric is to the target clustering task. The main contribution of this score is to provide domain experts with a tool to automatically identify the metric that most closely matches their domain knowledge and beliefs.

   Another potential drawback concerning distance metric learning methods for clustering occurs when the feature transformations are applied regardless of whether the geometric properties of the dataset are preserved or not. With that in mind, we proposed a technique to help clustering-based distance metric algorithms to find the least impactful data transformations that one can apply to provoke minimum alteration to the original space, while still learning a more suitable metric space for clustering the data. To achieve this goal, we established an order in which the constraints must be processed such that we could profit from the supervision knowledge, as well as obtaining a trustworthy representation of the rescaled data.

   Finally, we addressed the problem of analyzing whether it is worthwhile to process all pairwise constraints when establishing the transformations that need to be applied to the data. The reasoning behind this research question is the known fact that constraints could degrade the clustering performance [2, 29]. In that regard, we propose a mecha-

nism for ranking pairwise constraints based on their associated optimal dual variables. This rank is formed according to estimations of the impact caused by the constraint in the clustering criterion. We have demonstrated that integrating this ranking strategy with the recently proposed deep learning framework of [1] aids in selecting the most useful constraints for clustering the data.

1.4 Thesis outline

The remaining of this document is organized as follows. Chapter 2 presents a critical literature review of the most relevant studies in semi-supervised clustering. An overview of the contributions of this thesis is presented in Chapter 3. The three subsequent chapters describe our methodology for achieving our research objectives: Chapter 4 introduces a semi-supervised k-medoids model with pairwise constraints and proposes an algorithm to optimize it; Chapter 5 proposes a Lagrangian-based score for assessing the quality of the pairwise constraints; and Chapter 6 addresses issues when performing clustering-based distance metric learning with pairwise constraints. Finally, a general discussion and concluding remarks regarding the three objectives are provided in chapters 7 and 8, respectively.

CHAPTER 2 LITERATURE REVIEW

This chapter presents a review of the principal developments and methodologies in semi-supervised clustering. Although a comprehensive literature review is provided here, we note that this document is composed of self-contained chapters, and thus, specific and concise literature reviews are also provided in each chapter.

2.1 Strategies of Supervision

Side information is mostly provided in the form of pointwise or pairwise constraints. Whereas pointwise information is popular in classification contexts, pairwise information can be commonly found in all types of clustering scenarios. Pairwise constraints are also useful for defining cluster-level restrictions. For example, Davidson and Ravi [32] demonstrated that it is possible to define a minimum split δ restriction by creating a must-link constraint to every pair (o_i, o_j) if the distance $d(o_i, o_j)$ is smaller or equal to δ. Analogously, a clustering maximum diameter γ can be imposed by adding a cannot-link constraint for every pair of data points (o_i, o_j) if the distance $d(o_i, o_j)$ is greater or equal to γ [33, 26]. In addition, the notion of density can be imposed on the clusters by means of a disjunction of must-link constraints [32, 26]. For instance, a density ϵ-constraint can be formulated by specifying that, for each data point $o_i \in O$, at least one other point o_j in its ϵ-neighborhood (i.e., the subsets of points whose distances to o_i are less than or equal to ϵ) is assigned to the same cluster as that of o_i. Thus, at least one must-link constraint between o_i and the data points within its ϵ-neighborhood must be respected.

Alternatively, supervision can also be acquired by directly requesting the expert's *feedback* [34, 35, 36]. Hence, instead of requiring side information a priori, this type of supervision iteratively presents the clustering result to an expert, who can indicate if the given clustering has any undesirable characteristics. This feedback is then used to bias the algorithm towards a more meaningful clustering and to eliminate wrong assumptions from the clustering model. The feedback information can be presented in different forms. For example, one could inform pointwise or pairwise constraints such as "this data point does not belong to this cluster"; "move this data point to that cluster"; or "these two data points should be (or should not be) in the same cluster." Alternatively, the expert can inquire about cluster-level related instructions such as the cluster's cardinality, shape or density. Once the feedback is given, the dataset is re-clustered to incorporate all constraints required by the expert, and the clustering result is presented again to the specialist. The process is repeated until

the expert is in accordance with the obtained clustering description. Feedback supervision relies on the fact that, in general, domain experts have an appropriate criterion by which to evaluate the clustering, so that a significant clustering improvement can be obtained with their intervention. However, since this is an online technique that requires a few interactions with the domain expert, it might be expensive or impractical in some scenarios. Some examples of applications where feedback supervision was successfully applied are the problem of spam calls detection [34], where the feedback was used to indicate whether a call should be considered a spam or not; the problem of creating hierarchical classes of documents based on their content [35], where the expert could specify if documents are in the wrong category, and select the most informative features that would lead to the desired clusters [36].

Another way of declaring side information is by means of *triplet constraints* [37]. These constraints are used to indicate relative distance between sets of three data points. For example, the triplet (A, B, C) specifies that the data point A is closer to B than A is to C, which implies that A is more likely to share the cluster membership with the data point B than with C. A main motivation for the use of triplets constraints is that clustering models may fail to interpret pairwise supervision [4]. For instance, two data points with a cannot-link constraint could be assigned to wrong clusters in order to satisfy that constraint. The usefulness of triplet constraints is demonstrated by Ienco and Pensa [4], who propose to incorporate triplet constraints during the learning steps of the Deep Embedded Clustering (Deep Embedded Clustering (DEC)) algorithm [38]. The DEC method consists of first learning an embedded representation of the data points using an autoenconde network, and then executing a self-training routine to learn a clustering partition. To accomplish this, DEC uses the embedded points to define a *soft* membership distribution (points can be partially assigned to more than one cluster), and approximating it to a target distribution that resembles a *hard* clustering membership (points are assigned to exactly one cluster) using the Kullback-Leibler divergence [39] as loss function. Accordingly, Ienco and Pensa [4] introduce a triplet loss term into the network loss function that accounts for the violation of the constraints. The reported results demonstrated this method's great ability to retrieve high-quality clustering solutions.

Motivated by the latest achievements with deep learning methods, more semi-supervised clustering works started using this technique [40, 41, 1]. In the work conducted by Zhang et al. [1], the DEC framework is modified to incorporate penalties for violating diverse types of constraints, including pairwise, triplet and cluster-cardinality constraints. The proposed algorithm demonstrated great improvement in clustering accuracy, as well as robustness to work with different types of supervision.

A major challenge in semi-supervised clustering is to identify the most useful constraints

while minimizing the domain expert's effort in providing them [42], which can be particularly problematic for large problems. Numerous works [43, 44, 45, 46, 47] have handled this matter by exploring the concept of *active learning*, a paradigm to formulate queries to an *oracle* (typically a user or other source of information) for obtaining the side information. For pairwise constrained clustering, active learning methods can extend the supervision by analyzing the underlying structure of the data and querying the oracle if a cannot-link or must-link constraint should be added for a given pair of data points. For example, the well-known PCKmeans method [43] 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 then judges if the dissimilarity between the queried pair of data points is sufficient to impose a new cannot-link constraint. In the work of Mallapragada et al. [44], 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. Xiong et al. [45] proposed to use 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 extend these neighborhoods by selecting informative points and querying the oracle about their relationship with their neighbors. Although active learning is an excellent approach to efficiently enlarge the supervision and lead the algorithm to a correct course [46], it is important to note that it usually assumes the presence of an able oracle to provide side information, and the amount of allowed queries is limited.

Active learning has also motivated another relevant semi-supervised strand called *constraints selection*, which is the topic concerned with the informative benefit behind pairwise constraints. For instance, studies have analyzed how to select pairwise information such that the margin between clusters (separation) is maximized [48]. Okabe and Yamada [49] demonstrated how multiple clustering results can be combined to learn the constraints priorities. More theoretically, Davidson [50] studied methods to count the number of feasible clustering solutions, as well as other techniques to measure the difficulty of satisfying pairwise constraints in a constrained solution space.

As a matter of fact, the accuracy of pairwise supervision is a topic of concern in semi-supervised clustering [22]. Regardless of whether it originated from the domain expert or was generated by an active learning method, the inclusion of background information may not always lead to the desired clustering solution. To circumvent this difficulty, Davidson et al. [2] proposed two measures to evaluate the *informativeness* and *coherence* of a constraint set. Informativeness aims to capture the incremental effect of adding constraints to a clustering solution. Specifically, informativeness is operationalized as the fraction of pairwise constraints that are violated once added to a clustering solution obtained without

any constraint. The higher the proportion of violated constraints, the more informative 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, composed of one must-link and one cannot-link, with an overlapping segment when they are projected onto each other. Thus, the constraint set with the highest proportion of null projections is considered the most coherent set. For both measures, the idea is that constraint sets with the higher informativeness and coherence should improve the clustering solution. A following study presented by Wagstaff [42] has found partial support for this hypothesis, suggesting that more properties related to the utility of pairwise supervision should be further developed. Furthermore, a subsequent empirical study conducted by Ares et al. [30] investigated the effects of erroneous constraints in semi-supervised clustering algorithms. The reported results demonstrated that all tested algorithms drastically decreased their solution accuracy, evidencing the necessity of robust procedures to identify non-accurate constraints.

2.2 Incorporating Supervision

Traditionally, semi-supervised methods incorporate the supervision under two main approaches [22]: (i) methods that restrict the solution space in order to respect the constraints imposed by the side information and (ii) methods based on distance metric learning, that are designed to learn how to rescale the data points according to the background information provided. The following subsections present the principal works developed using these two methodologies.

2.2.1 Restricting the Solution Space

The primary idea of methods under this category is to eliminate infeasible solutions regarding the informed supervision. The already discussed COP-Kmeans method is one that falls into this category as it optimizes model (1.3)-(1.7). A well-known caveat of this algorithm is that it may fail to find a feasible clustering solution depending on the order in which points involved in cannot-link constraints are processed. This problem is later treated by Rutayisire et al. [24], where a mitigating solution is proposed by establishing an order to process those points, so that at least one feasible clustering partition can be found. However, a more common mechanism for handling feasibility matters with pairwise constraints is to replace the hard constraints (1.5) and (1.6) by soft constraints, such that penalty terms are introduced for violating them. A classical algorithm for soft-constrained clustering is MPCK-means [51], where the clustering criterion is modified to include a cost when the constraints are not satisfied. MPCK-means, an abbreviation for "Metric learning and Pairwise-Constrained K-

means", became a popular semi-supervised algorithm because it combines metric learning and solution space restriction to obtain more reliable clustering solutions. Other relevant restricted representative-based models are proposed in [52, 53, 54], which address the problem of clustering with size constraints, and by Yang et al. [25], where a MapReduce version of COP-Kmeans is introduced for working with big data and distributed systems.

The restricting solution space methodology was also intensively studied in conjunction with density-based clustering models using both pairwise constraints [55, 56] and pointwise information [57, 58]. Density-based clustering intends to discover dense regions within the data, where each dense region constitutes a cluster. This approach is a great option to identify arbitrarily shaped clusters and handle noisy data [15]. The work of Lelis and Sander [57] introduced the SSDBSCAN algorithm, a semi-supervised version of the well-known density-based algorithm, DBSCAN [59]. It explores labeled data to automatically estimate the density properties of each cluster in the dataset. In the work conducted by Vu and Do [58], the MC-SSDBS algorithm extends SSDBSCAN to also incorporate pairwise constraints in an active learning framework. DBSCAN is also modified by Ruiz et al. [55], where pairwise constraints are used to merge or divide clusters. A study by Yang et al. [56] explores pairwise supervision to address a particular issue with density-based clustering that is to assume a global level of density within the data, failing to identify clusters that have a different granularity level.

Moreover, semi-supervised clustering has been investigated in several other clustering approaches. Some examples are *probabilistic (fuzzy) models* [60, 61, 62, 63], in which each data point may have a nonzero assignment probability to two or more clusters (soft clustering); *nonnegative matrix factorization models* [64, 65], that are designed to benefit from dimensionality reduction to project the data into a latent space more amenable to clustering; *spectral models* [66, 67, 68, 69], which define clustering as a graph partitioning problem, where the data points are the nodes and the edges are weighted according to the pairwise similarities, thus aiming to find the minimum-weight normalized cut of the resulting graph.

Another attractive method used to incorporate supervision is *ensemble* models. Ensemble clustering studies how to combine different clustering models into a single more robust one. The idea is that no single model or criterion truly captures the optimal clustering [15]. Therefore, as in the semi-supervised learning setting, one of the main motivations behind ensemble clustering is to address the absence of supervision that clustering faces, leading to the existence of several criteria and alternative solutions. Consequently, combining both ensemble and semi-supervised learning methods is expected to lead to high-quality clustering solutions. Concisely, an ensemble clustering model acquires various clustering solutions through different models or different runs of the same clustering algorithm. Then, it selects

the potentially best solutions, combining them by means of a consensus function. In fact, the manner in which the solutions are selected and combined is usually the principal difference between ensemble models. For example, when using a soft constrained objective function, the selected clustering solutions are the ones that yield the fewest amount of violated pairwise constraints [70, 71]. In the sequel, a fuzzy membership function is applied for each data point, resulting in a weighted assignment clustering. Finally, a consensus phase using a normalized cut approach (similar to spectral clustering) is employed to obtain the final clustering. A different methodology was developed by Iqbal et al. [72], where the proposed framework uses classical semi-supervised clustering algorithms to yield different clustering solutions.

2.2.2 Distance Metric Learning for Clustering

Distance metric learning is the field of machine learning that aims to improve the performance of dissimilarity-based techniques by automatically learning distances from the data. For clustering, metric learning is typically performed with the support of side information provided by domain experts. Accordingly, with pairwise supervision, the constraints are used for finding data transformations to increase the distance between the points associated with a cannot-link constraint, and to reduce the distance between points associated with a must-link constraint.

This task is often performed by assuming that dissimilarities are expressed as Mahalanobis distances [73]. Given a dataset $O=\{o_1, \ldots, o_n\}$ of n points in the s-dimensional space, the objective of Malahanobis distance learning is to determine a positive semi-definite $d \times d$ matrix $\mathcal{A}$, such that the Mahalanobis distance between two points o_i and o_j is increased if a constraint imposes that they must belong to the same cluster, and is decreased if the points are to be in different clusters. The Mahalanobis distance between points o_i and o_j in $\mathbb{R}^d$ is defined as:

$$d_{\mathcal{A}}(o_i, o_j) = \sqrt{(o_i - o_j)^T \mathcal{A}(o_i - o_j)}. \tag{2.1}$$

The learned matrix $\mathcal{A}$ can be used to transform the dataset O into a set O' by defining $o'_i = \mathcal{A}^{1/2} o_i$ for $i = 1, \ldots, n$ (where $\mathcal{A}^{1/2}$ is the unique matrix $\mathcal{B}$ that is positive semidefinite and such that $\mathcal{B}\mathcal{B} = \mathcal{B}^T\mathcal{B} = \mathcal{A}$). Then, standard Euclidean distances can be used on O' by classical data mining algorithms. The seminal work exploring this idea was done by Xing et al. [74].

The previously introduced MPCK-means model [51] is a good example of a distance metric learning algorithm that learns a form of Mahalanobis distance. However, instead of computing a single parameter $\mathcal{A}$ for learning the transformation, it computes an individual matrix

for each cluster, i.e., there is a matrix $\mathcal{A}^c$ for each cluster $c \in \{1 \dots k\}$. Thereby, the method can obtain local transformations that allows clusters to have different shapes.

In addition to learning Mahalanobis distance, some works focused on learning nonlinear transformations that can be achieved by learning Bregman distances [75, 76] and kernel functions [77, 78, 79], both being more suitable approaches for handling high-dimensional data. In the work of Wu et al. [76], the objective is to learn a function $\varphi : \mathbb{R}^d \to \mathbb{R}$, for the Bregman distance defined as follows:

$$d_\varphi(o_i, o_j) = \varphi(o_i) - \varphi(o_j) - (o_i - o_j)^T \nabla \varphi(o_j), \tag{2.2}$$

This generalization allows to further retrieve other distances. For example, the Mahalanobis distance is obtained if $\varphi(o) = \frac{1}{2} o^T M o$. The authors propose to use a symmetric version of (2.2), where learning φ can be seen as learning an infinite number of local Mahalanobis distances.

Finally, some works [51, 80, 81] studied a hybrid approach, where both the solution space restriction and the distance metric learning methodologies are considered. In fact, combining both strategies seems to often lead to improvements in clustering performance [4].

2.3 Variable Neighborhood Search

Without loss of generality, let us consider the following global optimization problem

$$\min\{f(\mathbf{x}) \mid \mathbf{x} \in X\}, \tag{2.3}$$

where $f(\mathbf{x})$ is the objective function to be minimized over the solution space defined in X. Thus, each $\mathbf{x} \in X$ constitutes a feasible solution for the problem. Finding an optimal solution of (2.3) then implies determining $\mathbf{x}^*$ such that

$$f(\mathbf{x}^*) \leq f(\mathbf{x}) \ \forall \mathbf{x} \in X. \tag{2.4}$$

However, commonly used clustering models (e.g., MSSC) are formulated as combinatorial optimization problems, and thus, determining the optimal solution is often an NP-hard problem. For instance, MSSC is NP-hard even for two clusters in a general Euclidean space [82]. Therefore, from an application point of view, exactly solving (2.3) is often impracticable considering that it may require a large amount of time to find the optimal solution.

As a consequence, to optimize those clustering problems, one usually resorts to heuristic

methods, which is a more efficiency-driven approach. Among them, Variable Neighborhood Search (VNS) [83] is a stochastic procedure designed to find optimal or near-optimal solutions for global optimization problems. It offers a framework for helping heuristics to escape from a local optimum, which categorizes it as a metaheuristic algorithm. VNS has been successfully applied to numerous clustering problems [84, 85, 86, 87, 88, 89].

A local minimum $\mathbf{x}^l$ of (2.3) is such that

$$f(\mathbf{x}^l) \leq f(\mathbf{x}),\ \forall\ \mathbf{x} \in N(\mathbf{x}^l), \tag{2.5}$$

where $N(\mathbf{x}^l)$ denotes the neighborhood of $\mathbf{x}$. The main idea of the VNS framework is that it systematically operates to improve a solution by exploring multiple neighborhoods. Specifically, for a given solution $\mathbf{x} \in X$, it requires the definition of t_{max} neighborhood structures, i.e., $N_1(\mathbf{x}), \ldots, N_{t_{max}}(\mathbf{x})$, where each neighborhood $N_i(\mathbf{x})$ constitutes a subset of the solution space X, and can be defined in many different ways. For example, one could specify that the i-th neighborhood of $\mathbf{x}$, $N_i(\mathbf{x})$, is composed of all solutions that can be obtained after switching the clustering membership of i pairs of data points. Therefore, the search is performed by exploring increasingly wider neighborhoods once a local minimum is reached.

Note that the neighborhoods are defined with respect to the best solution found so far during the search. Let $\mathbf{x}_b$ be this best solution. The VNS works by the following procedure. First, in each iteration of the VNS, a random solution $\mathbf{x}'$ is obtained from the current neighborhood $N_t(\mathbf{x}_b)$. Then, a local descent procedure is applied for improving $\mathbf{x}'$. Let $\mathbf{x}''$ be the new local optimal solution obtained. In the case where the cost of $\mathbf{x}''$ is worse than the cost of $\mathbf{x}_b$, it is discarded and the algorithm selects a new neighbor solution $\mathbf{x}'$ from a more distant neighborhood of $\mathbf{x}_b$, i.e., $N_{t+1}(\mathbf{x}_b)$. Otherwise, $\mathbf{x}_b$ is updated with $\mathbf{x}''$, and the algorithm resumes in the closest neighborhood of the new best solution, i.e., $N_1(\mathbf{x}_b)$. Whenever the farthest neighborhood $N_v(\mathbf{x}_b)$ is attained, the VNS restarts from the closest neighborhood until a stopping condition is met. The pseudo-code of these steps is presented in Algorithm 1.

The VNS framework is established over three main principles: (i) a local minimum solution with respect to one neighborhood is not necessarily a local minimum with respect to another neighborhood; (ii) for many problems, local minima with respect to one or several neighborhoods are relatively close to each other; and (iii) the global minimum is a local minimum for all possible neighborhood definitions. In fact, this is why VNS favors the exploration of the closest neighborhoods, so that these three principles are explored in a coordinated way.

Algorithm 1 Basic steps of VNS

Input: An initial solution $\mathbf{x}_b$
A set of neighborhoods N_t for $t = 1, \cdots, t_{max}$.

```
while no stopping condition is met do
  t ← 1
  repeat
    Select a random neighbor x' in N_t(x_b)
    Apply a local search algorithm from x' to obtain x''
    if the cost of x'' is better than the cost of x_b then
      x_b ← x''
      t ← 1
    else
      t ← t + 1
    end if
  until t = t_max
end while
```

2.4 Lagrangian Relaxation Optimization

Several decades ago, it was computationally observed that hard combinatorial optimization problems can be viewed as easy problems that were complicated by a relatively small set of constraints [90]. Dualizing those constraints generally produces a problem that is easy to solve, and for which the optimal value is a lower bound (for a minimization problem) of the original problem. In other words, this procedure *relaxes* difficult constraints and transfers them to the objective function, adding penalty terms, also known as *dual variables* or *Lagrangian multipliers*. These penalties refer to the cost that needs to be paid if their associated constraints are violated. This idea, which became popular after the work of Held and Karp [91, 92], is known as *Lagrangian relaxation*.

To demonstrate how it works, let us consider the following combinatorial optimization problem composed of m inequality and r equality constraints, known as the *primal* problem:

$$Z = \min_{\mathbf{x}} f(\mathbf{x}) \tag{2.6}$$

subject to

$$h_i(\mathbf{x}) \leq 0 \ \forall \ i = 1, \ldots, m \tag{2.7}$$

$$g_j(\mathbf{x}) = 0 \ \forall \ j = 1, \ldots, r \tag{2.8}$$

As an attempt at making the problem easier, one may decide to relax the m inequality constraints by introducing penalties $\boldsymbol{\lambda} \in \mathbb{R}^m$, resulting in the following Lagrangian relaxation

problem:

$$\mathcal{L}(\boldsymbol{\lambda}) = \min_{\mathbf{x}} \; f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i h_i(\mathbf{x}) \tag{2.9}$$

$$\text{subject to}$$

$$g_j(\mathbf{x}) = 0 \; \forall \; j = 1, \ldots, r \tag{2.10}$$

We note that $\mathcal{L}(\boldsymbol{\lambda})$ constitutes a lower bound for Z for any $\boldsymbol{\lambda} \geq 0$. The proof of that relies on the fact that, if $\mathbf{x}^*$ is the optimal solution of (2.6)-(2.8), then we have $g_j(\mathbf{x}^*) = 0$ for $j = 1, \ldots, r$ and $h_i(\mathbf{x}^*) \leq 0$ for $i = 1, \ldots, m$. Hence, $\mathcal{L}(\boldsymbol{\lambda}) \leq f(\mathbf{x}^*) + \sum_{i=1}^{m} h_i(\mathbf{x}^*) \leq f(\mathbf{x}^*) = Z$

Therefore, the largest lower bound one can obtain correspond to the optimal solution of the *dual* problem given by

$$\mathcal{L}_D(\boldsymbol{\lambda}) = \max_{\boldsymbol{\lambda} \geq 0} \; \mathcal{L}(\boldsymbol{\lambda}) \tag{2.11}$$

From the duality theory (see e.g. [93]), if the original problem (2.6)-(2.8) is convex and there exists at least one solution $\bar{\mathbf{x}}$ such that $h_i(\bar{\mathbf{x}}) \leq 0$ and $g_j(\bar{\mathbf{x}}) = 0$ for all $i = 1, \ldots, m$ and $j = 1, \ldots, r$, then the strong duality holds and the optimal solution for the primal, Z^*, is equal to the optimal solution of the dual.

So as for solving the dual problem, if we assume, e.g., that the solution $\mathbf{x} \in \mathbb{R}^n$ is composed of n Boolean variables, then the solution space $X = \{\mathbf{x} \mid h_i(\mathbf{x}) \leq 0 \; \forall i \in \{1, \ldots, m\}$ and $g_j(\mathbf{x}) = 0 \; \forall j \in \{1, \ldots, r\}\}$ of the original problem (2.6)-(2.8) is finite. Thus, we can formulate the Lagrangian dual as a problem with many constraints:

$$\mathcal{L}_D(\boldsymbol{\lambda}) = \max_{\boldsymbol{\lambda} \geq 0} w \tag{2.12}$$

$$\text{subject to}$$

$$w \leq f(\mathbf{x}') + \sum_{i=1}^{m} \lambda_i h_i(\mathbf{x}') \; \forall \; \mathbf{x}' \in X, \tag{2.13}$$

which can be solved by means of a row generation method [94, 95]. Although the dual formulation is always a convex optimization problem ((2.11) is a concave maximization problem) [96], it is generally non-differentiable at an optimal point. As a consequence, a sub-gradient optimization method [97, 98] is typically applied to find optimal or near-optimal solutions for the dual problem. In this method, to minimize a function $J : U \to \mathbb{R}$, the domain variables

are iteratively updated by setting

$$q \leftarrow q - \alpha_t \mathfrak{s}(q), \tag{2.14}$$

where $q \in U$ and $\mathfrak{s}(q)$ is any subgradient of $J(q)$, i.e., any vector that satisfies the inequality $J(y) \geq J(q) + \mathfrak{s}^T(y - q)$ for all $y \in U$. The step size for the t-th iteration is defined by α_t, which commonly has a diminishing behavior. Thus, at each iteration of the sub-gradient method, we take a step in the direction of a negative sub-gradient. We highlight that when J is differentiable, the only possible choice for $\mathfrak{s}(q)$ is $\nabla J(q)$, and the sub-gradient method then reduces to the gradient method. Moreover, since this procedure is not guaranteed to always maintain a descent trajectory, it is common to keep track of the best solution found so far.

Algorithm 2 presents a pseudo-code of the sub-gradient method for optimizing the dual problem of the Lagrangian relaxation (2.11). It starts by defining some initial values for the dual variables. A common practice when solving a Lagrangian relaxation is to assign value zero to the variables. Then, the algorithm begins its main loop by determining a lower bound solution of the primal problem, (2.6)-(2.8), using the fixed values of the Lagrangian multipliers. In other words, this step aims to solve the (supposedly) easier version of the original problem after relaxing the hard constraints. If the lower bound obtained is the best one found so far, the variable storing the best solution is updated. Finally, after updating the step size for the current iteration t, the dual variables are updated using the sub-gradient. This process is repeated until the algorithm converges, meaning that the step proposed by the sub-gradient is sufficiently small, or a stopping condition is met (e.g., maximum number of iterations). The convergence proof for the sub-gradient algorithm is discussed in [98] for different definitions of the step size, including the one with a diminishing behavior as presented in Algorithm 2.

Algorithm 2 Sub-gradient method for optimizing the dual problem

Initialize the dual variables $\boldsymbol{\lambda}$, e.g., setting them to zero.
$t \leftarrow 1$
repeat
  Using the current values of the dual variables, determine a lower bound solution $\underline{\mathbf{x}}$ of cost $\underline{Z}$ by means of the Lagrangian relaxation (2.9)-(2.10).
  if $\underline{Z}$ is the largest lower bound ever found **then**
    $\mathbf{x}_{best} \leftarrow \underline{\mathbf{x}}$
  end if
  Update the step size, e.g., $\alpha_t = \frac{1}{\sqrt{t}}$
  $\boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda} - \alpha_t \mathfrak{s}(\underline{\mathbf{x}})$
  $t \leftarrow t + 1$
until the algorithm converges or a stopping condition is met
return $\mathbf{x}_{best}$

CHAPTER 3 ORGANIZATION OF THE THESIS

In this chapter, we explain how the subsequent chapters propose new methodologies for achieving the research objectives outlined in Chapter 1. This thesis is presented in the form of articles, and the three works which constitute it are provided in the chronological order of their submissions in chapters 4, 5 and 6.

Chapter 4 investigates the use of a well-known clustering model, k-medoids, under the semi-supervised learning paradigm. Whereas most of the works in the semi-supervised clustering literature explore the MSSC model, popularly known as the k-means model, we demonstrate that the k-medoids model is able to retrieve high-quality clustering solutions using pairwise supervision. Moreover, the introduced semi-supervised model inherits the advantages of the k-medoids model, such as its flexibility for working with metric and non-metric data, and its robustness to outliers. For optimizing the proposed constraint-based k-medoids model, an efficient Variable Neighborhood Search heuristic is developed. The article is concluded after demonstrating that the proposed algorithm is able to obtain optimal or near-optimal solutions for the optimization problem under consideration, hence reinforcing k-medoids as a good alternative for performing clustering with pairwise supervision.

The article presented in Chapter 5 is motivated by one of the most critical issues on using pairwise supervision in semi-supervised clustering: the presence of erroneous constraints. As the accuracy of constraints imposed on the algorithm ultimately impacts clustering accuracy, the presence of erroneous constraints can actually harm the clustering task. However, due to the lack of mechanisms to evaluate a clustering outcome, little is known about which constraints are harmful. Therefore, users of clustering algorithms are thus left with the sole option of using all the available constraints, which corresponds to a hit-or-miss strategy. To address this problem, we proposed a quantitative score for assessing the quality of pairwise constraints within a clustering model. This score is defined after computing the estimated impact that the constraints have on the clustering objective used, following the idea that a highly impactful constraint may be indicative of an erroneous constraint. To obtain such information, we formulated a Lagrangian relaxation problem where the pairwise constraints are transferred to the clustering objective function by adding penalty terms, namely dual variables, from which the estimated impact is computed. Computational experiments demonstrated that the proposed methodology can greatly aid domain experts by suggesting which constraints are likely to be erroneous, thus requiring review.

The article provided in Chapter 6 is motivated by the discoveries of the previous article

about the valuable information one can acquire from dual variables. In this last research work, we proposed to explore dual information to improve clustering algorithms that use pairwise constraints to perform distance metric learning. We observed that there are difficulties associated with this process, including: (i) determining an appropriated notion of dissimilarity for establishing the transformations; (ii) learning a new metric that maintains as much as possible the geometrical properties of the data; and (iii) selecting the most beneficial constraints for distance metric learning. Then, we developed tools, based on the dual information, that are able to help the applications users to mitigate those issues, and we examined the effectiveness of our methodology by means of numerous experiments with real and synthetic data.

CHAPTER 4 ARTICLE 1: ON THE *K*-MEDOIDS MODEL FOR SEMI-SUPERVISED CLUSTERING

Authors: Rodrigo Randel, Daniel Aloise, Nenad Mladenović and Pierre Hansen
Published at the 6th International Conference on Variable Neighborhood Search (ICVNS 2018). Part of the Lecture Notes in Computer Science.[1]

Abstract. Clustering is an automated and powerful technique for data analysis. It aims to divide a given set of data points into clusters which are homogeneous and/or well separated. A major challenge with clustering is to define an appropriate clustering criterion that can express a good separation of data into homogeneous groups such that the obtained clustering solution is meaningful and useful to the user. To circumvent this issue, it is suggested that the domain expert could provide background information about the dataset, which can be incorporated by a clustering algorithm in order to improve the solution. Performing clustering under this assumption is known as semi-supervised clustering. This work explores semi-supervised clustering through the k-medoids model. Results obtained by a Variable Neighborhood Search (VNS) heuristic show that the k-medoids model presents classification accuracy compared to the traditional k-means approach. Furthermore, the model demonstrates high flexibility and performance by combining kernel projections with pairwise constraints.

Keywords. *k-medoids, semi-supervised clustering, variable neighborhood search*

4.1 Introduction

In unsupervised machine learning, no information is known in advance about the input data. In this learning category, the objective is usually to provide the best description of the input data by looking at the similarities/dissimilarities between its elements. Clustering is one of the main unsupervised machine learning techniques. It addresses the following general problem: given a set of data objects $O = \{o_1, \ldots, o_n\}$, find subsets, namely *clusters*, which are homogeneous and/or well separated [13]. Homogeneity means that objects in the same cluster must be similar and separation means that objects in different clusters must differ one from another. The dissimilarity (or similarity) d_{ij} between a pair of objects (o_i, o_j) is usually computed as a function of the objects' attributes, such that d values (usually) satisfy: (i) $d_{ij} = d_{ji} \geq 0$, and (ii) $d_{ii} = 0$. Note that dissimilarities do not need to satisfy triangle

[1] Available at [99]

inequalities, i.e., to be distances.

Despite its concise definition, the clustering problem can have significant variations, depending on the specific model used and the type of data to be clustered. The clustering criterion used plays a crucial role in the clustering obtained. For example, the homogeneity of a particular cluster can be expressed by its *diameter* defined as the maximum dissimilarity between two objects within the same cluster, while the separation of a cluster can be expressed by the *split* or the minimum dissimilarity between an object inside the cluster and another outside.

When considering dissimilarity measures, the definitions above yield two families of clustering criteria: those to be *maximized* for separation and those to be *minimized* for homogeneity. In general, these criteria are expressed in the form of thresholds, min-sum or max-sum for a set of clusters. Thus, for instance, the diameter minimization problem corresponds to minimizing for a set of clusters the maximum diameter found among them, while in the split maximization, one seeks to maximize the minimum split found in the clustering partition. The clustering criterion used is also determinant to the computational complexity of the associated clustering problem. For example, split maximization is polynomially solvable in time $O(n^2)$, while diameter minimization is NP-hard already in the plane for more than two clusters [100].

In order to overcome this difficulty and improve the result of the data clustering, it has been suggested that the domain expert could provide, whenever possible, auxiliary information regarding the data distribution, thus leading to better clustering solutions more in accordance to his knowledge, beliefs, and expectations. The clustering process driven by this side information is called *Semi-Supervised Clustering* (Semi-Supervised Clustering (SSC)). SSC has become an essential tool in data mining due to the continuous increase in the volume of generated data [12].

The most common types of side information are pairwise constraints such as *must-link* and *cannot-link* [101]. A *must-link* constraint between two objects implies that they must be assigned to the same cluster, whereas a *cannot-link* constraint that they must be allocated in different clusters. In this paper, we make an in-depth analysis of the use of the k-medoids model for the SSC problem. We also propose a new Variable Neighborhood Search (VNS) [83] algorithm that uses a location-allocation heuristic and takes into consideration pairwise constraints.

The paper is organized as follows. The next section presents the related works to this research. Section 4.3 describes the k-medoids model for the SSC problem. Section 4.4 describes the two-stage local descent algorithm proposed, and in Section 4.5 a VNS algorithm is presented for optimizing the described model. Computational experiments that demonstrate the

effectiveness of our methodology in a set of benchmark data sets are reported in Section 4.6. Finally, the conclusions are presented in Section 4.7.

4.2 Related Works

Algorithms that make use of constraints as *must-link* and *cannot-link* in clustering became widely studied and developed after the COP-Kmeans algorithm of Wagstaff and Cardie's work [23]. The algorithm is based on modifying the unsupervised original k-means algorithm by adding a routine to prevent an object from changing cluster if any of the *must-link* or *cannot-link* constraints are violated.

The model optimized by COP-Kmeans consider that objects $o_i \in O$ correspond to points p_i of a s-dimensional Euclidean space, for $i = 1, \ldots, n$. The objective is to find k clusters such that the sum of squared Euclidean distances from each point to the centroid of the cluster to which it belongs is minimized while respecting a set of pairwise constraints. The set $\mathcal{ML}$ is formed by the pairs of points (p_i, p_j) such that p_i and p_j must be clustered together, whereas the set $\mathcal{CL}$ contains the pair of points (p_i, p_j) such that p_i and p_j must be assigned to different clusters.

The *semi-supervised minimum sum-of-squares clustering* (SSMSSC) model is mathematically expressed by:

$$\min_{x,y} \quad \sum_{i=1}^{n}\sum_{j=1}^{k} x_{ij}\|p_i - y_j\|^2 \tag{4.1}$$

subject to

$$\sum_{j=1}^{k} x_{ij} = 1, \quad \forall i = 1, ..., n \tag{4.2}$$

$$x_{ij} - x_{wj} = 0, \quad \forall (p_i, p_w) \in \mathcal{ML}, \quad \forall j = 1, ..., k \tag{4.3}$$

$$x_{ij} + x_{wj} \leq 1, \quad \forall (p_i, p_w) \in \mathcal{CL}, \quad \forall j = 1, ..., k \tag{4.4}$$

$$x_{ij} \in \{0, 1\}, \quad \forall i = 1, ..., n; \quad \forall j = 1, ..., k. \tag{4.5}$$

The binary decision variables x_{ij} express the assignment of point p_i to the cluster j whose centroid is located at $y_j \in \mathbb{R}^s$. Constraints (4.2) guarantee that each data point is assigned to exactly one cluster. Constraints (4.3) refer to the must-link constraints, and constraints (4.4) to the cannot-link ones.

The simplicity and pioneering of COP-Kmeans have made it a basic algorithm for many later

works. Some examples are: semi-supervised clustering using combinatorial Markov random fields [102]; adaptive kernel method [103]; clustering by probabilistic constraints [104]; and density-based clustering [55].

A relevant work involving clustering under pairwise constraints was conducted by Xia [105]. The global optimization method proposed in that work is an adaptation of the Tuy's cutting planes method [106]. The algorithm is proved to obtain optimal solutions in exponential time in the worst case, and hence, it cannot be used for practical purpose for larger data mining tasks. Xia [105] reported a series of experiments where the algorithm is halted before convergence. The obtained clustering results were superior to other algorithms based on COP-Kmeans.

Restricting the solution space through the explict use of pairwise constraints is not the only possible approach for SSC. Many works have been published to propose mechanisms using distance metric learning to explore these side information. Among them, a well-known algorithm is the *Semi-Supervised-Kernel-kmeans* [107] that enhances the similarity matrix obtained from the application of a kernel function by adding a term that brings closer together *must-link* objects while driving away *cannot-link* objects. The algorithm defines a similarity matrix $\mathbf{S} = \mathcal{K} + W + \sigma I$, where $\mathcal{K}$ is a kernel matrix, W is the matrix responsible to include the pairwise constraints into the distance metric, and σ is the term that multiplies an identity matrix I to ensure that $\mathbf{S}$ is semi-definite positive. The kernel-k-means algorithm [108] is then executed over $\mathbf{S}$ in an unsupervised manner (see [107] for details).

4.3 Proposed Model

Another classical representative-based clustering model is the k-medoids whose objective is to partition the points into exactly k clusters so that the sum of distances between each point and the central object (i.e., the *medoid*) of their respective cluster is minimized.

The input of the k-medoids model is a distance matrix, D, with each entry d_{ij} providing the dissimilarity between points p_i and p_j. It can be mathematically formulated in its semi-supervised version as:

$$\min \quad \sum_{i=1}^{n}\sum_{j=1}^{n} x_{ij} d_{ij} \tag{4.6}$$

subject to

$$\sum_{j=1}^{n} x_{ij} = 1, \quad \forall i = 1, ..., n \tag{4.7}$$

$$x_{ij} - x_{wj} = 0 \quad \forall (p_i, p_w) \in \mathcal{ML}, \quad \forall j = 1, ..., n \tag{4.8}$$

$$x_{ij} + x_{wj} \leq 1 \quad \forall (p_i, p_w) \in \mathcal{CL}, \quad \forall j = 1, ..., n \tag{4.9}$$

$$x_{ij} \leq y_j \quad \forall i = 1, \ldots, n, \forall j = 1, ..., n \tag{4.10}$$

$$\sum_{j=1}^{n} y_j = k \tag{4.11}$$

$$x_{ij} \in \{0, 1\} \quad \forall i = 1, \ldots, n, \forall j = 1, \ldots, n, \tag{4.12}$$

$$y_j \in \{0, 1\} \quad \forall j = 1, \ldots, n, \tag{4.13}$$

where y_j is equal to 1 if p_j is selected as the medoid of cluster j, and 0 otherwise. Constraints (4.10) assure that points can only be assigned to selected medoids, and constraint (4.11) defines that k medoids must be selected. The resulting model (4.6)-(4.13) is named thereafter the **Semi-Supervised K-Medoids Problem** (SSKMP).

The possibility of defining the matrix D allows the objective function of the model to be flexible to use different measures to express the dissimilarities between points and medoids. The k-medoids model can be used to cluster metric data, as well as more generic data with notions of similarity/dissimilarity. For this reason, one of the main features of k-medoids is its vast list of applications [109].

When comparing the k-means model with the k-medoids model, Steinley [110] listed three important advantages in using the latter for clustering:

1. Although both models work with a center-based approach, the k-means model defines the central element as the centroid of the cluster, while in the k-medoids this element is taken directly from the data set. This feature allows, for example, to identify which is the most representative element of each cluster.

2. The k-medoids, in its formal definition, usually consider the Euclidean distance to measure the dissimilarity between points and medoids, instead of the quadratic one considered in k-means. As a consequence, the k-medoids is generally more robust to outliers and noise present in the data [111].

3. While k-means only uses quadratic distance and may need to constantly recompute the distances between points and centroids every time centroids are updated, the k-medoids run over any distance matrix, even those for which there exist triangle inequality violations and which are not symmetric.

4.4 Local descent algorithm for SSKMP

Several heuristics methods have already been proposed to solve the original k-medoids problem. A very popular one is the *interchange* heuristic introduced in [112]. This local descent method searches, in each iteration, for the best pair of medoids (one to be inserted in the current solution, and another to be removed) that leads to the best-improving solution if swapped. If such pair exists, the swap is performed, and the procedure is repeated. Otherwise, the algorithm stops and the best solution found during this descent path is returned. An efficient implementation of this procedure, called *fast-interchange*, was proposed by Whitaker [113]. However, this method was not widely used (possibly due to an error in the article) until Hansen and Mladenović [114] corrected it and successfully applied it as a subroutine of a VNS heuristic. After, Resende and Werneck [115] proposed an even more efficient implementation by replacing one of the data structures present in the implementation of Whitaker [113] with two new data structures. Although the implementation suggested in [115] has the same worst-case complexity, $O(n^2)$, it is significantly faster and, to the best of the authors' knowledge, is the best implementation for the heuristic *interchange* already published.

In this paper, the method proposed in [115] is used as local descent procedure for our algorithm in order to refine a given SSKMP solution, but with a slight modification to ensure that pairwise constraints are respected.

4.4.1 Handling must-link constraints

The following strategy is proposed to respect *must-link* constraints:

If must-link constraints connect a set of points, they can all be merged into a single point, which is enough to represent them all.

This assumption relies on the fact that all these points need to be together in the final partition, and aggregating them is just an efficient shortcut for assigning them to the same cluster repeatedly.

Figure 4.1 illustrates this process on a set of *must-link* constraints given by $\mathcal{ML} = \{(p_1, p_2), (p_4, p_6), (p_2, p_6)\}$. It is possible to replace the set $\mathcal{ML}$ by an equivalent set $\mathcal{ML}' = \{(p_1, p_2), (p_1, p_4), (p_1, p_6)\}$, with p_1 as the root point for all other linked points p_2, p_4 and p_6 (Figure 4.1b). This aggregation creates a so-called *super-point* and is showed in Figure 4.1c where all points involved in that *must-link* constraint are all represented by the super-point p_1. Note that the super-point could have been aggregated over p_2, p_4 or p_6 instead of p_1 without prejudice.

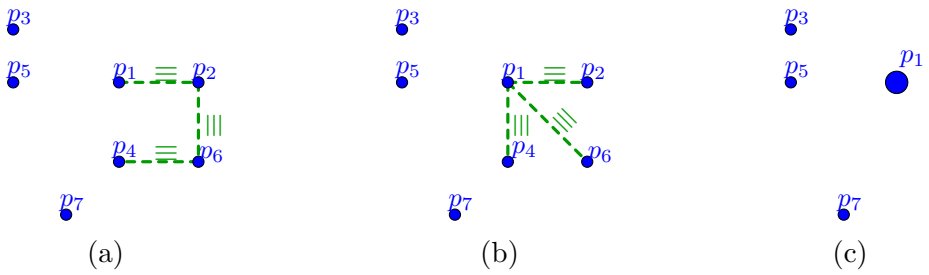


Figure 4.1 Illustration of a *super-point* aggregation.

However, since the points involved in *must-link* constraints can be aggregated and viewed as a single point, then it is also necessary to update the dissimilarity d_{ij} of a super-point p_i, and all medoids $j = 1, \ldots, n$, as the sum of dissimilarities of all points that compose it. Let $H(p_i) = \{p_h \in P \mid (p_i, p_h) \in \mathcal{ML}\}$ be the set of points that are part of the super-point p_i. The cost $d_{ij} = d_{ji}$, for each $j = 1, \ldots, n$ is then calculated as the sum of dissimilarities considering all aggregated points, i.e.,

$$d_{ij} = \sum_{h:p_h \in H(p_i)} d_{hj} \quad j = 1, \ldots, n \tag{4.14}$$

For the example in Figure 4.1, d_{13} is updated as: $d_{13} = d_{13} + d_{23} + d_{43} + d_{63}$.

The super-point aggregation is a quick step that can be entirely performed during the pre-processing stage of the algorithm. It also helps to reduce the dimension of the original data set once the points are merged. Consequently, the more *must-link* constraints are provided by the expert, the best is the performance of our algorithmic approach.

4.4.2 Handling cannot-link constraints

Once all *must-link* constraints are respected, the local descent algorithm only concerns violated *cannot-link* constraints. A solution is said to be infeasible if there exists any pair $(p_i, p_w) \in \mathcal{CL}$ such that p_i and p_w are assigned to the same cluster. In order to avoid that, the algorithm is divided into two stages:

1. **Stage 1.** In this first stage, the *cannot-link* constraints are temporarily neglected, and the local descent algorithm proceeds to improve the current best solution.

2. **Stage 2.** For each new improved solution found in stage 1, there is a chance of this solution be infeasible, so the algorithm invokes a routine able to restore its feasibility (concerning the *cannot-link* constraints).

In summary, the approach of our local descent algorithm is to allow an efficient search to be executed in the direction of the best possible solution (regardless of the *cannot-link* constraints), whereas the solutions obtained during the descent search path are turned into feasible solutions. Thus, the algorithm relies upon the possibility of restoring the feasibility of solutions generated in the first stage of the algorithm. The key point of our strategy is to guide the search exclusively by the gradient of the objective function, disregarding the *cannot-link* constraints.

Let $\mathbf{s}$ be a solution for the problem with its k selected medoids, i.e., $\mathbf{s} = \{j | y_j = 1\}$. Let us denote $X(i) = \{j | x_{ij} = 1\}$ the cluster of point p_i. We also define the set $E(i) = \{h | (p_i, p_h) \in \mathcal{CL}\}$ as the set of points that cannot be clustered together with p_i, and $B(i) = \{j \in \mathbf{s} | \exists h \in E(i), X(h) = j\}$ as the set of clusters in $\mathbf{s}$ that are *blocked* to p_i since it contains at least one point from $E(i)$.

The feasibility routine is presented in Algorithm 3. It is called whenever a new infeasible solution $\mathbf{s}$ is obtained by the algorithm. Let $\phi_1(i) \in \mathbf{s}$ be the closest medoid in $\mathbf{s}$ from point p_i. Remark that after stage 1, since *cannot-link* constraints are not considered, every point p_i, for $i = 1, \ldots, n$, is assigned to its closest medoid, i.e., $X(i) = \phi_1(i)$.

Algorithm 3 Restore feasibility function

```
1:  R ← ∅
2:  for i = 1, ..., n do
3:     if E(i) ≠ ∅ then
4:        R ← R ∪ {i}
5:     end if
6:  end for
7:  repeat
8:     shuffle(R)
9:     for all i ∈ R do
10:       if X(i) ∈ B(i) or X(i) ≠ φ1(i) then
11:          Assign p_i to the closest medoid j ∈ s such that j ∉ B(i)
12:       end if
13:    end for
14: until no assignment is made
```

The restore function works as follows: first, between lines 1-6, a new set R is built to contain all points p_i that are involved in *cannot-link* constraints (i.e., $E(i) \neq \emptyset$). The loop of lines 7-14 proceeds by removing the cannot-link violations. The order in which the points are examined determines the solution obtained or even if the method is able to restore feasibility. Therefore, set R is shuffled at the beginning of that loop at line 8. Then, the algorithm iterates in the loop of lines 9-13 searching for assignments that can make the solution feasible

(condition $X(i) \in B(i)$) or that can improve its cost (condition $X(i) \neq \phi_1(i)$). The rationale behind the second condition is that p_i might have been allocated to a farther cluster in a previous iteration of the restoration routine because its closest medoid was not available for assignment due to a cannot-link constraint. Note that algorithm 3 needs to keep that B updated. This is performed every time after a point p_i is assigned from a medoid p_ℓ to medoid p_j, blocking this medoid for each point in $h \in E(i)$, and maybe removing ℓ from their sets $B(h)$, depending on the presence of any other point in $E(h)$ assigned to medoid p_ℓ.

Algorithm 3 is assured of finishing although a feasible solution is not guaranteed. Indeed, the decision problem of whether a clustering problem is feasible given a set $\mathcal{CL}$ of cannot-link constraints is NP-complete [32]. In that case, the obtained solution is simply discarded.

4.5 Variable Neighborhood Search for SSKMP

Variable Neighborhood Search (VNS) metaheuristic [116] has been successfully applied to many clustering problems (e.g. [89, 87, 86, 88]). The neighborhood structure adopted in our VNS algorithm is based on *swapping* selected medoids of a solution $\mathbf{s}$ by others non-selected medoids outside $\mathbf{s}$. In this sense, v_{max} neighborhoods are defined, where the v-th neighborhood of $\mathbf{s}$, $N_v(\mathbf{s})$, contains all solutions obtained after replacing v medoids $j \in \mathbf{s}$ with others v not-selected medoids $l \notin \mathbf{s}$.

The Algorithm 4 presents the complete framework of our VNS algorithm. It starts by pre-processing the *must-link* constraints via the *super-point* concept (lines 1). Following that, the algorithm constructs an initial feasible solution (line 2) obtained in a series of three steps: (i) an initial solution $\mathbf{s}_b$ is built by randomly selecting k initial medoids and assigning each point to its closest medoid; (ii) the restore feasibility function is applied for $\mathbf{s}_b$; (iii) if $\mathbf{s}_b$ is still infeasible, the algorithm proceeds and replace $\mathbf{s}_b$ by the first feasible solution found during the VNS. We assume that the problem is always feasible, i.e., the sets $\mathcal{ML}$ and $\mathcal{CL}$ allows to obtain a feasible solution for the SSC under consideration. The algorithm considers that infeasible solutions have infinite cost.

Next, the algorithm starts the VNS block (loop 3-15) that chooses a random neighbor solution (line 6) and applies the two-stage local descent method described in Section 4.4 to possibly improve it (line 10). However, notice that after a random solution $\mathbf{s}^r$ is chosen in the neighborhood $N_v(\mathbf{s})$ of our VNS algorithm, an allocation step must follow to re-assign the points that were allocated to the replaced medoids (removed medoids of $\mathbf{s}$) to their new closest medoid. However, this process does not take into consideration the *cannot-link* constraints, and then, $\mathbf{s}^r$ might be infeasible. To overcome this situation, we also invoke the

restore function for $\mathbf{s}^r$ before proceeding to the local search procedure (line 8). If the best feasible solution found in the descent path has a better cost than $\mathbf{s}_b$, then it is stored in $\mathbf{s}_b$ (line 12). The algorithm repeats this process until a defined stopping criterion is met.

Algorithm 4 VNS for SSC k-medoids

1: Apply the *super-point* concept, merging points interconnected by *must-link* constraints into super-points;
2: Find an initial feasible solution $\mathbf{S}_b$;
3: **repeat**
4: $v \leftarrow 1$;
5: **repeat**
6: Choose a random neighbor solution $\mathbf{S}^r \in N_v(\mathbf{s})$;
7: **if** $\mathbf{S}^r$ is infeasible **then**
8: Call the restore feasibility function for $\mathbf{S}^r$.
9: **end if**
10: Apply the local descent method from $\mathbf{S}^r$, obtaining a local minimum S_f
11: **if** cost of $\mathbf{S}_b >$ cost of $\mathbf{S}_f$ **then**
12: $\mathbf{S}_b \leftarrow \mathbf{S}_f$; $v \leftarrow 1$;
13: **end if**
14: $v \leftarrow v + 1$;
15: **until** $v = v_{max}$
16: **until** a stopping criterion is met

4.6 Experiments

This work explores the results from three different perspectives. First, model SSKMP is analyzed concerning its accuracy performance when compared with the traditional SSC model, SSMSSC. Next, the VNS performance is tested using a set of benchmark datasets for SSC problem. Third, the flexibility of SSKMP is explored in combination with distance metric learning.

Computational experiments were performed on an Intel i7-6700 CPU with a 3.4GHz clock and 16 Gigabytes of RAM. The algorithms were implemented in C++ and compiled by gcc 6.3.

4.6.1 Model accuracy

First of all, it is essential to keep in mind that it is impossible to determine whether a model is better than another with respect to all possible data sets (see Kleinberg's impossibility theorem [117]). The SSMSSC and SSKMP are comparable models given that (i) both are

representative-based; and (ii) the data sets used in the experiments are considered as points in the Euclidean space. It was decided to compare the models regarding accuracy using the *Adjusted Rand Index* (ARI) [118], which can measure how close the clustering result is to the ground-truth classification obtained in the UCI repository [119]. The ARI is designed to yield values close to 0 for random cluster assigning, regardless of the number of clusters and data objects, and exactly 1 for identical clustering partitions.

We first compare the models using the ARI results reported by Xia [105]. As done in her work, we ran the VNS algorithm 100 times and reported the average ARI value. We also defined the stop criterion as the average CPU time used by Xia's algorithm. In all experiments we used the v_{max} parameter equal to 10. Table 4.1 presents the results of these two models for 12 benchmark data sets. For each of them, column n indicates the number of points and k the number of clusters. In the following, we present results for two configurations of $\mathcal{ML}$ and $\mathcal{CL}$ used in [105]. The first two columns refer to the number of *must-link* and *cannot-link* constraints, and the last two refer to the ARI index values obtained by each model concerning the ground-truth partition.

Table 4.1 Datasets configurations and ARI results for SSMSSC and SSKMP

			Configuration 1				Configuration 2			
Instance	n	k	$\lvert\mathcal{ML}\rvert$	$\lvert\mathcal{CL}\rvert$	**ssmssc**	**sskmp**	$\lvert\mathcal{ML}\rvert$	$\lvert\mathcal{CL}\rvert$	**ssmssc**	**sskmp**
Soybean	47	4	4	24	0.55	**0.60**	8	4	0.62	0.62
Protein	116	6	18	12	**0.31**	0.25	26	18	**0.32**	0.25
Iris	150	3	12	12	0.74	**0.75**	16	8	0.75	**0.76**
Wine	178	3	44	26	0.44	**0.45**	72	44	0.45	0.45
Ionosphere	351	2	52	36	0.16	0.16	122	64	0.14	**0.15**
Control	600	6	60	30	**0.54**	0.50	90	60	**0.53**	0.51
Balance	625	3	156	94	**0.32**	0.24	218	126	**0.43**	0.25
Yeast	1484	10	296	178	0.16	0.16	520	296	0.17	0.17
Optical	3823	10	496	306	**0.70**	0.68	689	420	**0.71**	0.69
Statlog	4435	6	444	222	0.53	0.53	666	444	**0.54**	0.53
Page	5473	5	548	274	0.01	**0.03**	1024	820	0.01	**0.03**
Magic	19020	2	1902	952	0.05	**0.18**	2854	1902	0.04	**0.16**

We note from Table 4.1 that both models present quite similar results and comparable clustering performances. For the 24 tests cases, each model had nine times each the best ARI, and for six data sets, they had the same ARI value. Moreover, even when the ARI indices were not equal, the difference in values was marginal.

Table 4.2 Performance results for VNS and CPLEX.

Instance	Configuration	f_{opt}	t_{opt}	vns	$\overline{\texttt{vns}}$	$\overline{t_{vns}}$	restore
Soybean	1	1.138047e+02	0.12	**0%**	**0%**	0.00	11%
	2	1.156629e+02	0.16	**0%**	**0%**	0.00	10%
Protein	1	1.269331e+03	2.46	**0%**	**0%**	0.01	10%
	2	1.262633e+03	0.93	**0%**	**0%**	0.00	15%
Iris	1	9.835843e+01	8.85	**0%**	**0%**	0.01	7%
	2	9.962796e+01	7.94	**0%**	**0%**	0.00	6%
Wine	1	1.749303e+04	10.07	**0%**	**0%**	0.01	7%
	2	1.907746e+04	17.16	**0%**	**0%**	7.08	7%
Ionosphere	1	8.172423e+02	61.44	**0%**	**0%**	5.13	9%
	2	8.384550e+02	53.58	**0%**	0.02%	65.04	9%
Control	1	2.693438e+04	135.20	**0%**	**0%**	0.13	5%
	2	2.693937e+04	124.34	**0%**	**0%**	0.12	5%
Balance	1	1.466425e+03	881.94	**0%**	0.002%	110.42	4%
	2	1.471803e+03	816.61	**0%**	**0%**	91.51	4%
Yeast	1	2.523202e+02	166622.71	**0%**	**0%**	97.78	3%
	2	2.605097e+02	42557.74	**0%**	0.004%	124.44	3%

4.6.2 VNS Performance

This section is dedicated to evaluating the VNS performance for optimizing the SSKMP model. To obtain the optimal solution for the tested datasets, we used the solver CPLEX 12.6. This restricted our sample in this experiment because CPLEX was not able to solve data sets `Optical`, `Statlog`, `Page` and `Magic` in a reasonable amount of time (less than 50 hours).

Table 4.2 shows the results of our computational experiments. For each configuration, we executed the algorithm 10 times using 300 seconds as time limit. Columns f_{opt} and t_{opt} provide optimal solution values and the time needed by CPLEX to obtain it, respectively. The column `vns` reports the gap between the optimal solution and the best solution found by the VNS from the 10 distinct executions. In the sequel, columns $\overline{\texttt{vns}}$ and $\overline{t_{vns}}$ report the average values for the same 10 execution of the algorithm. The column `restore` presents the average percentage of time required by the restore feasibility function during the execution.

Firstly, we justify the importance of having a heuristic approach to the problem since the time to optimally solve it increases exponentially as the number of points scales (t_{opt}). On the other hand, for all the 16 test cases, the VNS was able to find the optimal solutions using

much less time. For the test where CPLEX took the longest time to solve, 46h for `Yeast` configuration 1, the VNS only needed, on average, 98 seconds to obtain a solution with the same cost. Furthermore, only in three scenarios, the VNS was not able to obtain the best solution in all ten executions, but still, the gaps are tiny.

From the results reported in column `restore`, we verify that the restore feasibility function does not require much computational time (7% on average) for the instances used in [105]. Besides, the amount of time is reduced for larger datasets, which is expected as the local descent procedure starts to demand more computational resources to perform the search.

4.6.3 Model flexibility

One of the main advantages of using SSKMP is the ability to work with a general dissimilarity matrix D as input. This feature not only allows the model to work with many different metric systems but also provides great flexibility to define a clustering criterion. For example, it is possible to use the *distance metric learning* technique without a single modification with our algorithm. Take for instance the *Semi-Supervised-Kernel-Kmeans* (SS-Kernel-k-means) algorithm [107], which defines the similarity matrix $\mathbf{S} = \mathcal{K} + W + \sigma I$, aggregating the kernel matrix $\mathcal{K}$ and constraints matrix W (metric learning). Then, we can easily transform $\mathbf{s}$ into a dissimilarity matrix D (e.g. subtract each entry by the maximum element in $\mathbf{s}$) and use it as input for SSKMP. Furthermore, having the distance metric modification in the input does not preclude the use of pairwise constraints in combination, which has already been proven to be a good approach [51].

Consider the synthetic data set **Two Circles** showed in Figure 4.2, which presents 200 points in the Euclidean plane, with 100 points in each class. This dataset has an inner circle and a surrounding outer circle. Figure 4.3 presents the ARI results for our proposed VNS and the SS-Kernel-k-means algorithm using the two circles instance. Both algorithms were executed 100 times starting from a random initial solution, and the average ARI was reported. As suggested in [107], we used an exponential kernel ($\exp(-|\|x-y\|^2/2\sigma)$ for SS-Kernel-k-means to separate the two classes in the mapped space linearly. We also included the algorithm `vns+` which combines the distance metric learning and solution space restriction due to pairwise constraints into the model optimized by our proposed VNS. The time limit used for both VNS algorithms was the average time needed by SS-Kernel-k-means to finish one execution.

We note from Figure 4.3 that the VNS algorithms based on model SSKMP outperformed the typical kernel approach, both reaching the maximum ARI value. We highlight that the VNS algorithm improved its accuracy performance by adding the distance metric learning

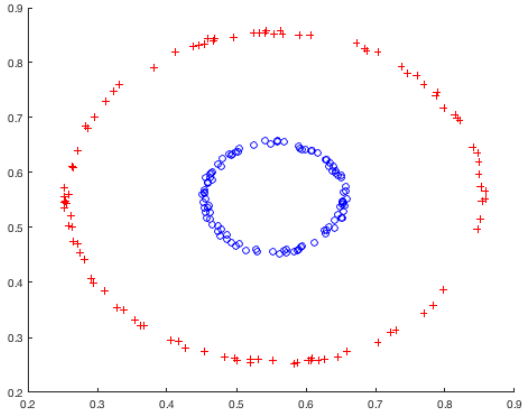


Figure 4.2 Two Circles synthetic data set.

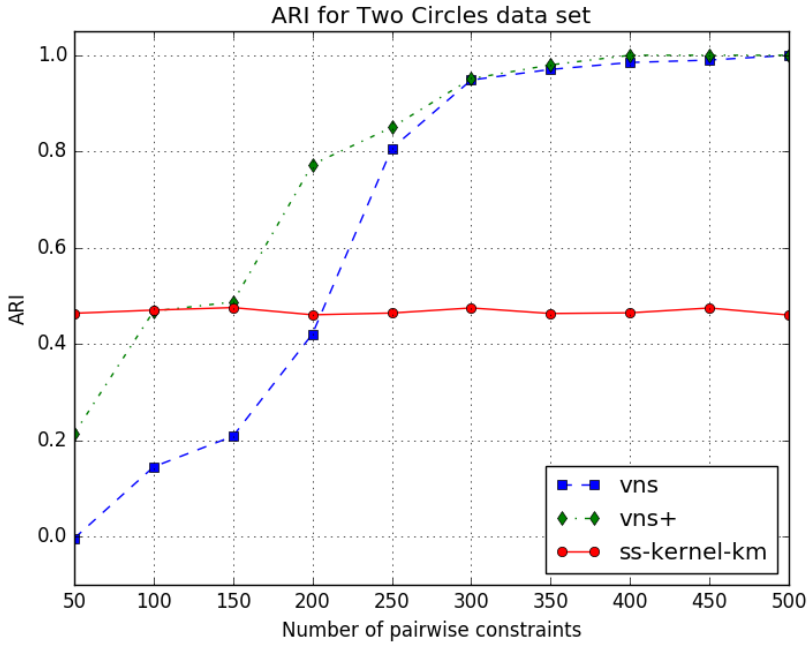


Figure 4.3 ARI performance for Two Circles data set.

mechanism, reaching the maximum ARI value with 20% fewer constraints than the VNS algorithm that uses only the pairwise constraints. We also observed that the SS-Kernel-k-means algorithm was not able to improve ARI as the number of pairwise constraints increased. We believe that the kernel-based algorithm is more sensitive to initialization besides not being able to escape from local optima. In contrast, the VNS was proved robust, making powerful use of a priori information.

4.7 Conclusion

This paper proposed a VNS heuristic for assessing the performance of the k-medoids model for semi-supervised clustering. Experiments showed that the new model had similar classification performance when compared with the traditional k-means model. The VNS algorithm was validated in a series of comparative experiments against CPLEX, presenting solutions very close to the optimal ones (never exceeding 0.02% in average) using much less CPU time. Moreover, the flexibility of the k-medoids model was tested regarding the addition of a dissimilarity matrix generated by a kernel function with distance metric learning. The VNS that combined the kernel trick with the explicit use of pairwise constraints presented the best accuracy performance among the algorithms compared.

CHAPTER 5 ARTICLE 2: A LAGRAGIAN-BASED SCORE FOR ASSESSING THE QUALITY OF PAIRWISE CONSTRAINTS IN SEMI-SUPERVISED CLUSTERING

Authors: Rodrigo Randel, Daniel Aloise, Simon J. Blanchard and Alain Hertz
Published on *Data Mining and Knowledge Discovery* journal, 2021 [1]

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 and Lagrangian duality.*

5.1 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[2]. 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

[1] Available at [120]

[2] 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.

(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 [5, 31]. 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 $O = \{o_1, \ldots, o_n\}$ 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 [13]. Among clustering methods, *partitioning* focuses on splitting O into k clusters ($P_k = \{C_1, C_2, \ldots, C_k\}$) such that:

1. $C_j \neq \emptyset$ for all $j = 1, \ldots, k$,
2. $C_i \cap C_j = \emptyset$ for all $1 \leq i < j \leq k$, and
3. $\bigcup_{j=1}^{k} C_j = O$,

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. 14]:

$$\min\{f(P) : P \in \mathcal{P}(O, k)\}. \tag{5.1}$$

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 [31] 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 [15], for instance, by using pre-determined labels for the initialization of an existing unsupervised clustering algorithm like k-means [19]. 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 [121].

Moreover, such information that experts have to provide is common to many types of applications. Basu et al. [60] 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. [122] provide an example of how managers may have prior knowledge to impose constraints into Bayesian mix-

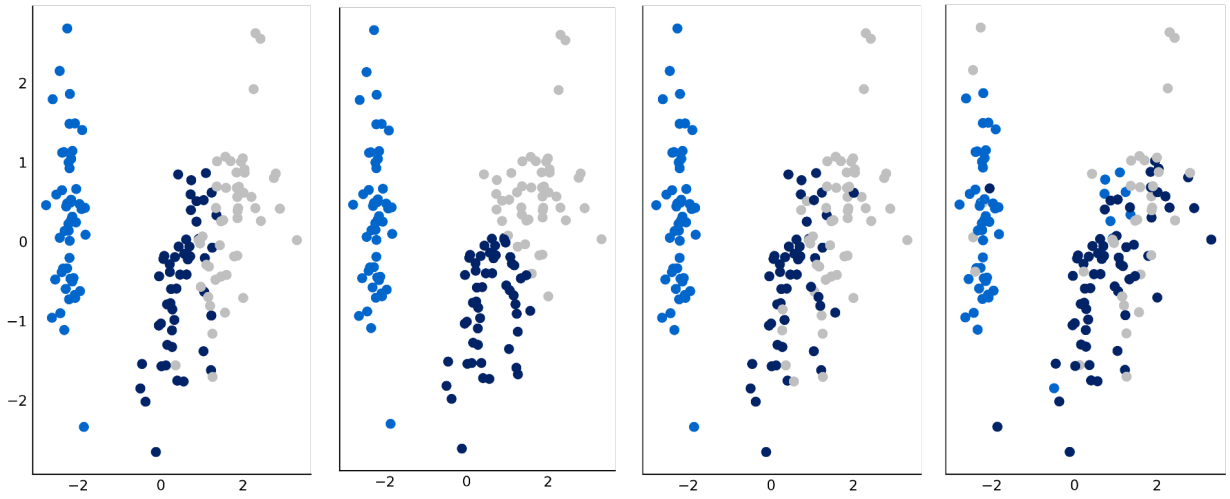
ture models to render solutions that are eventually actionable by businesses. Nonetheless, working with pairwise constraints is typically more complex than incorporating pointwise information, and the problem of whether it is possible to satisfy a given set of cannot-link constraints with k clusters is NP-complete [32].

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 [2, 29]. 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 [30], 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 [31].

To illustrate the consequences of having inaccurate constraints, we show in Figure 5.1 clustering solutions from the two principal components of an application to the Iris dataset [123]. Figure 5.1(a) illustrates the ground-truth partition, whereas Figure 5.1(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 5.1(c) illustrates the partition obtained by using the popular COP-Kmeans algorithm [23] 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 5.1(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 5.2 provides an overview of prior research regarding the difficulty of substantiating whether a constraint set is informative. Then, section 5.3 presents the proposed impact score, and section 5.4 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

Figure 5.1 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).

5.2 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 [42]. 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 [43] 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 the work conducted by Mallapragada et al. [44], 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 the work of Xiong et al. [45], 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 works of Mallapragada et al. [44] and Xiong et al. [45], 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 [46].

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. [2] 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 5.2 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 [42] has found partial support for this hypothesis, suggesting that more properties related with the utility of pairwise constraints should be further developed.

Informativeness and coherence are not the only measures available to evaluate the helpfulness of constraints. For instance, Davidson [50] 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

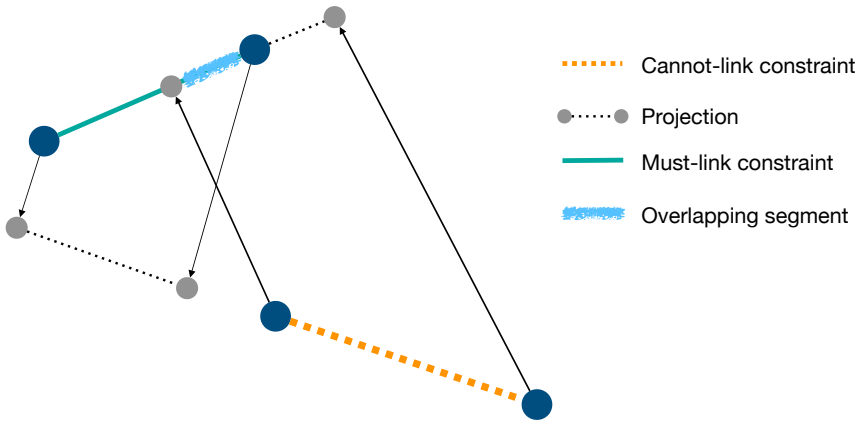


Figure 5.2 Illustration of Coherence measure proposed by Davidson et al. [2]: projection of must-link and cannot-link constraint vectors onto each other.

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.

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

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

$$Z = \min_{X} \quad f(x) \tag{5.2}$$

subject to

$$x_i^c + x_j^c \leq 1 \qquad \forall (o_i, o_j) \in \mathcal{CL}, \quad \forall c = 1, \ldots, k \tag{5.3}$$

$$x_i^c - x_j^c = 0 \qquad \forall (o_i, o_j) \in \mathcal{ML}, \quad \forall c = 1, \ldots, k \tag{5.4}$$

$$x_i^c \in \{0, 1\} \qquad \forall i = 1, ..., n; \quad \forall c = 1, \ldots, k \tag{5.5}$$

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 (5.3) and (5.4) 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 (5.3) and (5.4) are satisfied with equality, we can replace them by the following equivalent constraints where ϵ[3] is any real number in $]0, 1[$:

$$x_i^c + x_j^c \leq 1 + \epsilon \qquad \forall (o_i, o_j) \in \mathcal{CL}, \quad \forall c = 1, \ldots, k \tag{5.3'}$$

$$x_i^c - x_j^c \leq \epsilon \qquad \forall (o_i, o_j) \in \mathcal{ML}, \quad \forall c = 1, \ldots, k \tag{5.4'}$$

$$x_j^c - x_i^c \leq \epsilon \qquad \forall (o_i, o_j) \in \mathcal{ML}, \quad \forall c = 1, \ldots, k. \tag{5.4"}$$

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 $O(n^2)$ [100], diameter minimization is NP-hard for more than two clusters [124]. For example, from Huygen's theorem [125], MSSC is expressed within (5.2)-(5.5) by:

$$f(x) = \sum_{c=1}^{k} \frac{\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \|o_i - o_j\|^2 x_i^c x_j^c}{\sum_{i=1}^{n} x_i^c}, \tag{5.6}$$

which is a non-convex quadratic function, making (5.2)-(5.5) NP-hard even for two clusters in general Euclidean dimension [82]. For the k-medoids model (see e.g. Kaufman and Rousseeuw [27]), f can be expressed by:

$$f(x) = \sum_{c=1}^{k} \sum_{i=1}^{n} \sum_{j=1}^{n} \|o_i - o_j\| x_i^c y_j^c, \tag{5.7}$$

[3]The relevance of ϵ is best understood during the optimization of the dual problem expressed in (5.11). In that occasion, we must prevent the dual variables from being able to assume any value if the constraints are satisfied, as it will not affect the cost of the optimization function.

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:

$$x_i^c \leq y_i^c \quad \forall i, j = 1, \ldots, n \tag{5.8}$$

$$\sum_{i}^{n} y_i^c = k. \tag{5.9}$$

The k-medoids model is also NP-hard [126]. Algorithms for finding the optimal solution of the problem for large data sets are presented by Avella et al. [127] and García et al. [128], while efficient heuristics are proposed by Hansen et al. [129] and Resende and Werneck [115].

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 (5.3'), (5.4'), and (5.4"). Specifically, the Lagrangian function is defined as follows:

$$\begin{aligned} L(\eta, \lambda, \gamma) = \min_X \Bigg(f(x) &+ \sum_{(o_i,o_j)\in \mathcal{CL}} \sum_{c=1}^{k} \eta_{ij}^c (1 + \epsilon - x_i^c - x_j^c) \\ &+ \sum_{(o_i,o_j)\in \mathcal{ML}} \sum_{c=1}^{k} \lambda_{ij}^c (\epsilon + x_i^c - x_j^c) \\ &+ \sum_{(o_i,o_j)\in \mathcal{ML}} \sum_{c=1}^{k} \gamma_{ij}^c (\epsilon + x_j^c - x_i^c) \Bigg) \end{aligned} \tag{5.10}$$

and the dual of the integer program (5.2)-(5.5) can be expressed as follows:

$$L_D = \max_{\eta,\lambda,\gamma\leq 0} L(\eta, \lambda, \gamma) \tag{5.11}$$

where η, λ and γ correspond to its dual variables. The weak duality theorem (see e.g. Bertsimas and Tsitsiklis [93]) asserts that L_D is the best lower bound for the optimal value Z of the integer program (5.2)-(5.5).

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 (5.3) 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 $x_i^c \neq x_j^c$.

5.3.1 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 [130, 131].

To illustrate, consider any cannot-link constraint $(o_u, o_v) \in \mathcal{CL}$. Assume that the constraints (5.3') imposing $x_u^c + x_v^c \leq 1 + \epsilon$ for all $c \in \{1 \dots k\}$ are replaced by the following constraints:

$$x_u^c + x_v^c \leq 1 + \epsilon + b \qquad \forall c = 1, \dots, k \tag{5.12}$$

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:

$$L_{uv}^{\mathcal{CL}}(\eta, \lambda, \gamma, b) = L(\eta, \lambda, \gamma) + \sum_{c=1}^{k} b\eta_{uv}^c \tag{5.13}$$

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

$$\frac{\partial L_{uv}^{\mathcal{CL}}(\eta, \lambda, \gamma, b)}{\partial b} = \sum_{c=1}^{k} \eta_{uv}^c. \tag{5.14}$$

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 (5.4') and (5.4") for objects o_u and o_v. The Lagrangian function $L_{uv}^{\mathcal{ML}}(\eta, \lambda, \gamma, b)$ becomes:

$$L_{uv}^{\mathcal{ML}}(\eta, \lambda, \gamma, b) = L(\eta, \lambda, \gamma) + \sum_{c=1}^{k} b(\lambda_{uv}^c + \gamma_{uv}^c) \tag{5.15}$$

and the approximated effect on Z^b of deactivating the must-link constraint between data

points o_u and o_v is given by:

$$\frac{\partial L_{uv}^{\mathcal{ML}}(\eta, \lambda, \gamma, b)}{\partial b} = \sum_{c=1}^{k} (\lambda_{uv}^{c} + \gamma_{uv}^{c}). \tag{5.16}$$

Negative values for the partial derivatives (5.14) and (5.16) 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 observations, we propose the following impact score $\mathcal{I}_{uv}$ for a pairwise constraint associated with objects o_u and o_v:

$$\mathcal{I}_{uv} = \begin{cases} \sum_{c=1}^{k} \eta_{uv}^{c} & \text{if } (o_u, o_v) \in \mathcal{CL} \\ \sum_{c=1}^{k} (\lambda_{uv}^{c} + \gamma_{uv}^{c}) & \text{if } (o_u, o_v) \in \mathcal{ML}. \end{cases} \tag{5.17}$$

In the next section, we discuss how to solve the dual problem (5.11) to calculate the impact score (5.17).

5.3.2 Solving the dual problem

The *sub-gradient optimization algorithm* [97, 132] is a widely used technique for optimizing non-differentiable optimization problems such as (5.11). To minimize a function $g : U \subset \mathbb{R} \to \mathbb{R}$, the domain variables are iteratively updated by setting

$$w \leftarrow w + \alpha_{\ell} \mathfrak{s}(w), \tag{5.18}$$

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 5 describes the steps of the sub-gradient method for solving (5.11). The algorithm begins by defining initial values for the Lagrangian multipliers η_{uv}^{c}, λ_{uv}^{c} and γ_{uv}^{c}. As a common practice when working with Lagrangian relaxation, we initialize these penalty terms with zero [90], which means that we impose no prior cost on the objective function. We next make the reasonable assumption that there are solutions that satisfy constraints (5.3)-(5.5) 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

Algorithm 5 Subgradient method for optimizing the dual problem (5.11)

Initialize variables $\eta^c_{uv}, \lambda^c_{uv}$, and γ^c_{uv} to 0.
Set the upper bound $\bar{Z}^*$ on Z equal to the value of the best available feasible solution.
for all $\ell = 1$ to m **do**

Lower bounding step.
Use current values of the dual variables and equation (5.10) 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 (5.3)-(5.5) 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**
$\eta^c_{uv} \leftarrow \eta^c_{uv} + \alpha_\ell \frac{(\bar{Z}^* - \underline{Z})}{\sum_{(i,j)\in\mathcal{CL}} \sum_{c'=1}^{k} (1+\epsilon-\underline{x}_i^{c'}-\underline{x}_j^{c'})^2} (1 + \epsilon - \underline{x}^c_u - \underline{x}^c_v)$.
end for
for all $(o_u, o_v) \in \mathcal{ML}$ and all $c \in \{1, \ldots, k\}$ **do**
$\lambda^c_{uv} \leftarrow \lambda^c_{uv} + \alpha_\ell \frac{(\bar{Z}^* - \underline{Z})}{\sum_{(i,j)\in\mathcal{ML}} \sum_{c'=1}^{k} (\epsilon+\underline{x}_i^{c'}-\underline{x}_j^{c'})^2} (\epsilon + \underline{x}^c_u - \underline{x}^c_v)$
$\gamma^c_{uv} \leftarrow \gamma^c_{uv} + \alpha_\ell \frac{(\bar{Z}^* - \underline{Z})}{\sum_{(i,j)\in\mathcal{ML}} \sum_{c'=1}^{k} (\epsilon+\underline{x}_j^{c'}-\underline{x}_i^{c'})^2} (\epsilon + \underline{x}^c_v - \underline{x}^c_u)$.
end for
end for

the first step, a lower bound for (5.2)-(5.5) is obtained by solving model (5.10) 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 (5.2)-(5.5). This routine can be as simple as the procedure described in Algorithm 6. 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 (5.3)-(5.5). If a situation arises for which it is difficult to recover feasibility, we can stop Algorithm 5 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 6 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 (5.10) to optimality might be NP-hard for a wide variety of clustering criteria. Thus, for the lower bounding step of Algorithm 5, one likely must resort to heuristics or valid relaxations to find good approximations.

5.4 Computational Experiments

To evaluate the usefulness of the impact score defined in (5.17), 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`.

5.4.1 Experiments with synthetic data

The first experiment follows the fractional factorial experimental design similar to that used by Blanchard et al. [133] and Éverton Santi et al. [88]. 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 5.1: 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 5.1 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 5 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 [27]. To accelerate the lower bounding step, we opt for relaxing the integrality constraints (5.5) by $x_i^c \in [0, 1]$ for all $i = 1, \ldots, n$ and $c = 1, \ldots, k$, and equation (5.10) is then solved using CPLEX 12.8. Algorithm 6 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 (5.17), 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 5.3 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.

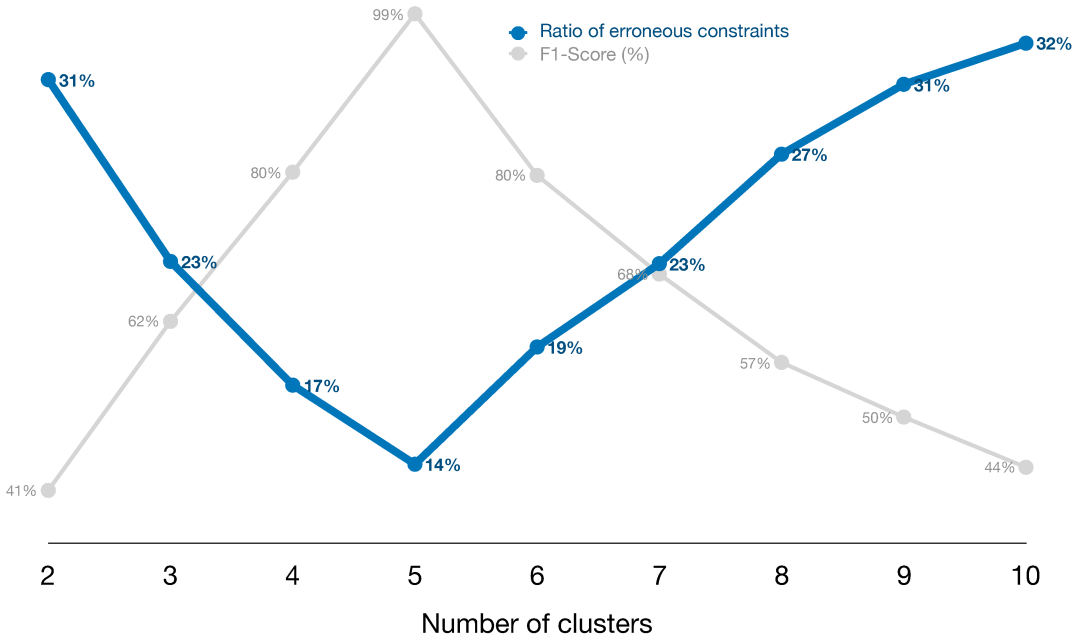


Figure 5.3 Fraction of constraints predicted as erroneous and F1-Score obtained by our impact score as a function of the number of clusters. A small proportion of constraints predicted as erroneous suggests the appropriated number of clusters.

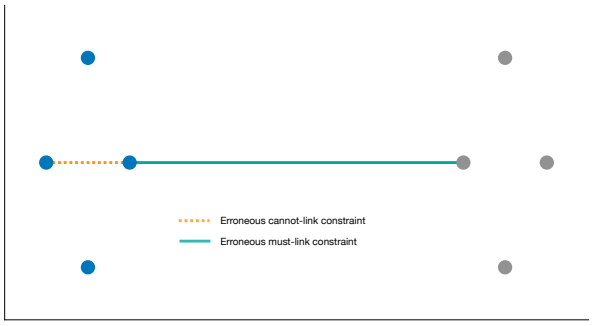
5.4.2 Comparison with optimistic and pessimistic naïve approaches

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 (5.2)-(5.5) 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.

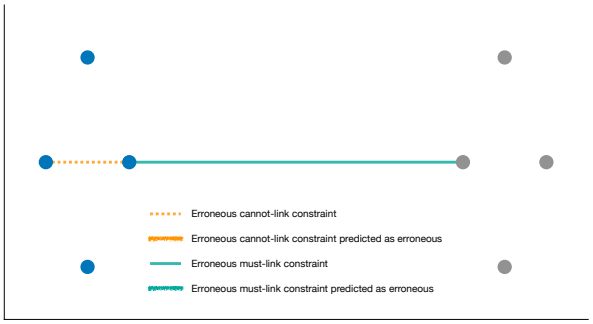
With this approach, even if a constraint is erroneous, removing it from the constraint set may have no impact on the solution cost because the clustering solution can be tied up by other constraints (i.e., assignments will not change). To illustrate, Figure 5.4(a) shows one erroneous must-link constraint and one erroneous cannot-link constraint. The optimal

MSSC partition is shown in Figure 5.4(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 5.4(c)). For comparison, the execution of the proposed Lagrangian-based method correctly predicts both constraints as erroneous (Figure 5.4(d)), and the optimal clustering solution produced by MSSC can thus be retrieved.

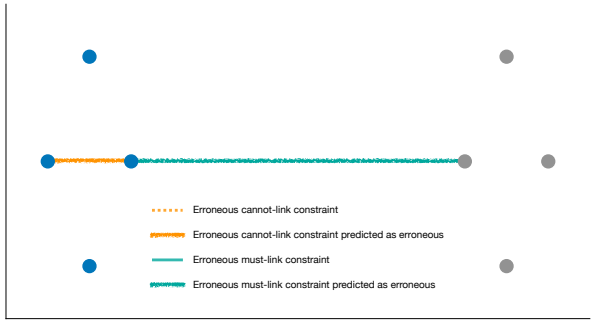


(a) Example where the optimistic approach fails.

(b) Optimal clustering using the MSSC model.



(c) Predictions using the optimistic approach.

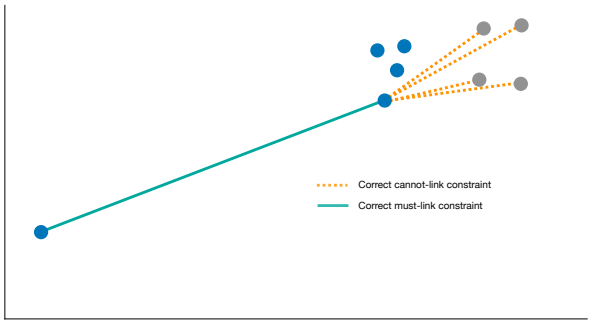


(d) Predictions using the Lagrangian-based impact score.

Figure 5.4 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.

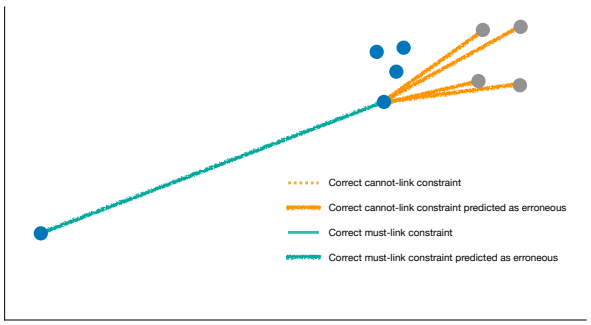
The pessimistic approach. The pessimistic approach begins by assuming that all constraints are erroneous. It begins by defining the base cost Z_B by solving the integer program without any pairwise constraint. Then, for every $(o_u, o_v) \in \mathcal{C}$, the integer program is solved again

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.

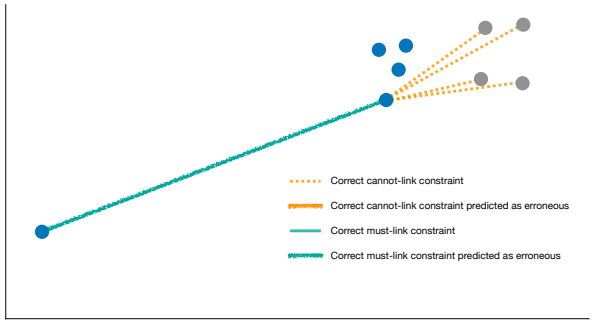


(a) Configuration where the pessimistic approach fails.

(b) Optimal clustering using the MSSC model.



(c) Predictions using the pessimistic approach.



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

Figure 5.5 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 5.5(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 5.5(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 5.5(c), the pessimistic approach yields five false negatives given that the five correct constraints are pre-

dicted as erroneous. However, as shown in Figure 5.5(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 5.5(a). Besides, it does not experience tied solutions as the one illustrated in Figure 5.4(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 5.6 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 5.7, 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

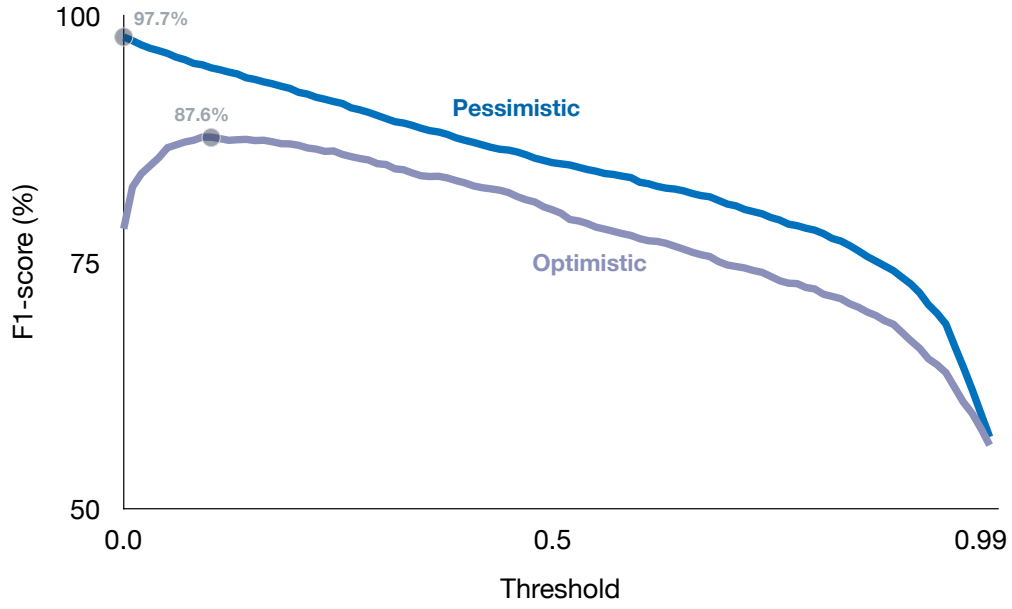


Figure 5.6 F1-score obtained by the baseline approaches when predicting erroneous constraints as a function of the threshold (τ). The latter is used for filtering slightly negative scores.

Z_{uv} for the two baseline approaches were obtained with the Variable Neighborhood Search (VNS) designed by [99] 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. R. Costa et al. [89], Hansen et al. [129]).

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 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.

5.4.3 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 subgradient algorithm converges, i.e., verify if the relaxed model (5.10) can approximate the

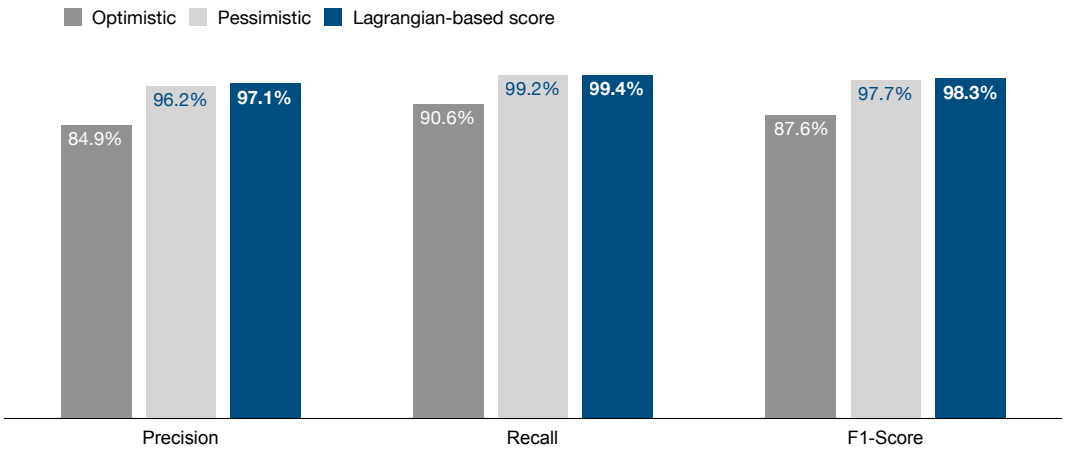


Figure 5.7 Performance comparison between the two baseline approaches and the Lagrangian-based method.

original problem (5.2)-(5.5); (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 5.2, that are available at the UCI Machine Learning Repository [134]. 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 5.2 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 (5.10) 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:

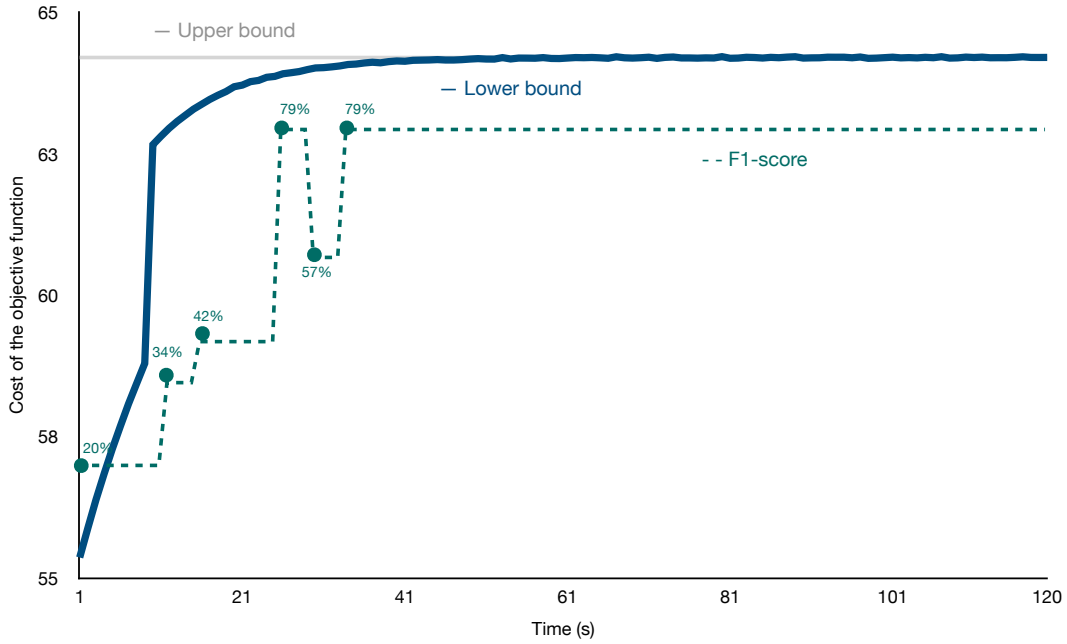
$$\sum_{c=1}^{k} \frac{\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \|o_i - o_j\|^2 x_i^c x_j^c}{\sum_{i=1}^{n} x_i^c} + \sum_{(o_i,o_j)\in\mathcal{CL}} \sum_{c=1}^{k} \eta_{ij}^c (1 + \epsilon - x_i^c - x_j^c) + \sum_{(o_i,o_j)\in\mathcal{ML}} \sum_{c=1}^{k} \lambda_{ij}^c (\epsilon + x_i^c - x_j^c) + \sum_{(o_i,o_j)\in\mathcal{ML}}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

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 5.3 presents the time allocated to each dataset. For comparison, the baseline methods decribed in Section 5.4.2 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.

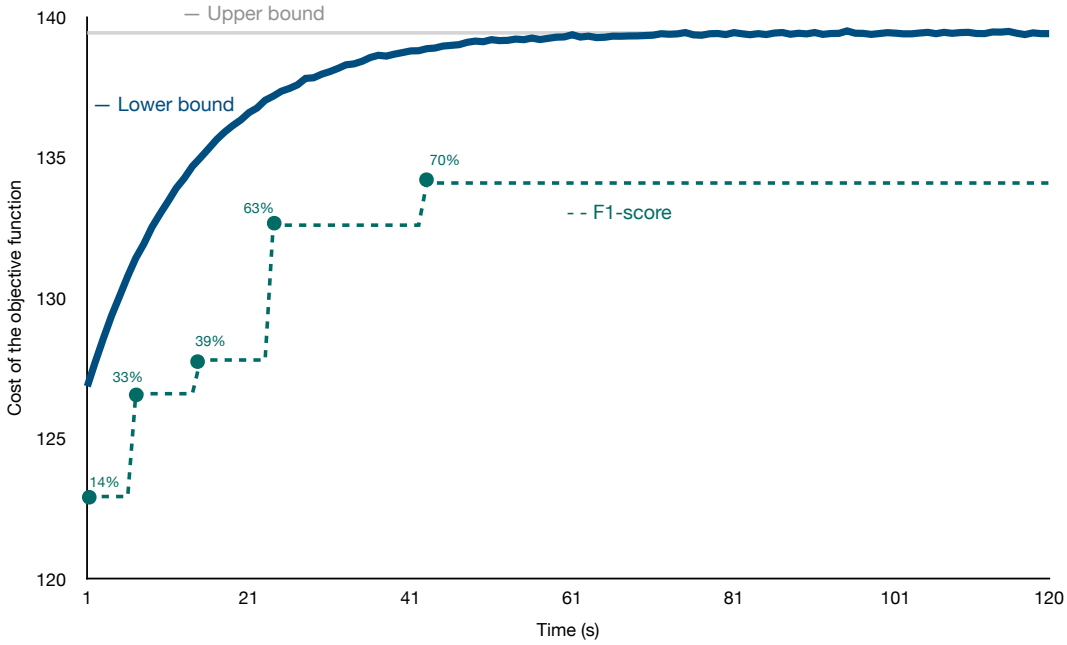
Table 5.3 Results for the selected benchmark datasets.

			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

The results are summarized in Table 5.3 where we report the final dual gaps at the end of



(a) Iris



(b) Wine

Figure 5.8 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.

the subgradient algorithm, as well as the F1-scores of our 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 5.8 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 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.

5.4.4 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 5.2)

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

wise constraints to review, we will use the standard Adjusted Rand Index (ARI) [118], 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 (5.2)-(5.5) 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 5.9 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 5.2, Davidson et al. [2] propose to evaluate the quality of a constraint set by using a coherence measure. We also show in Figure 5.9 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

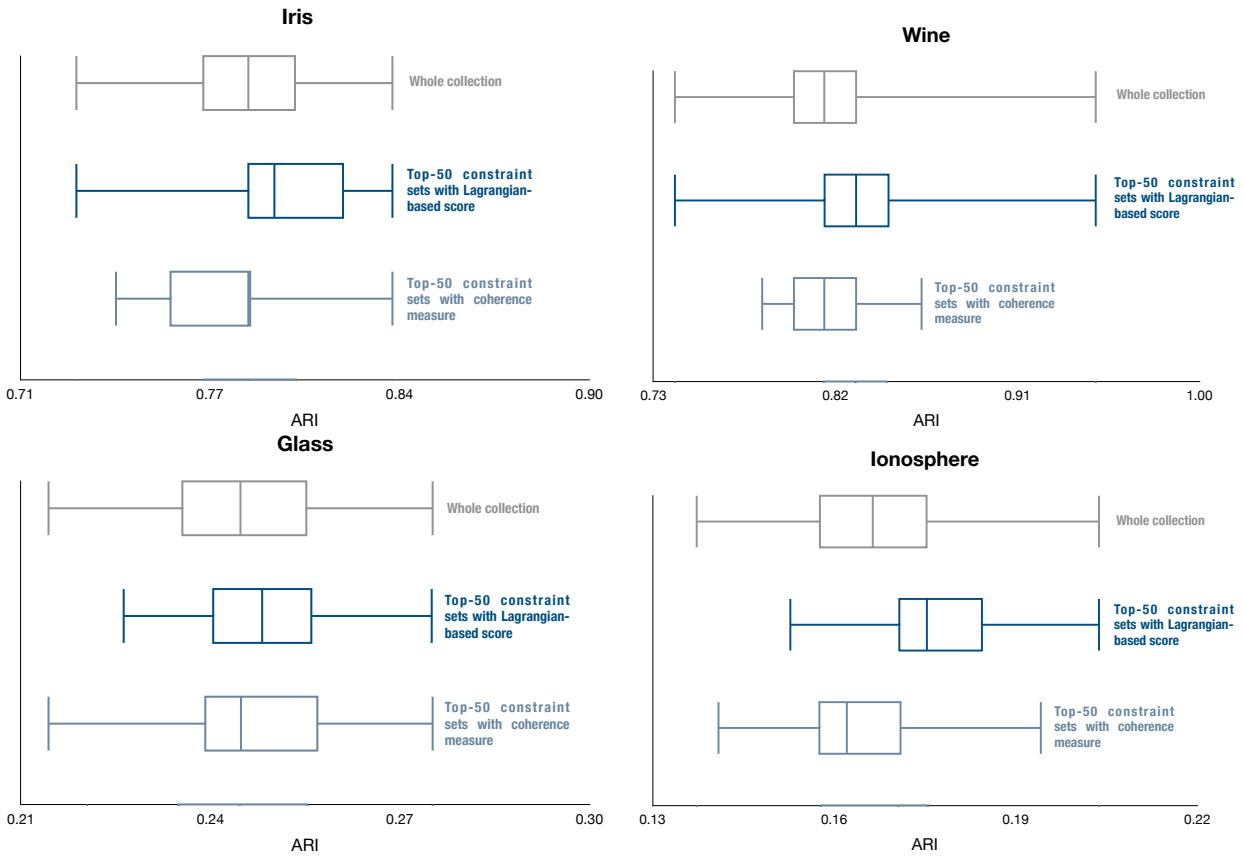


Figure 5.9 Comparison of the ARIs for the whole collection of 100 constraint sets of 25 **correct** constraints, and for the top 50 constraint sets selected by the impact score and Davidson's coherence measure.

discarded by an expert after review. Figure 5.10 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 5.4 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.

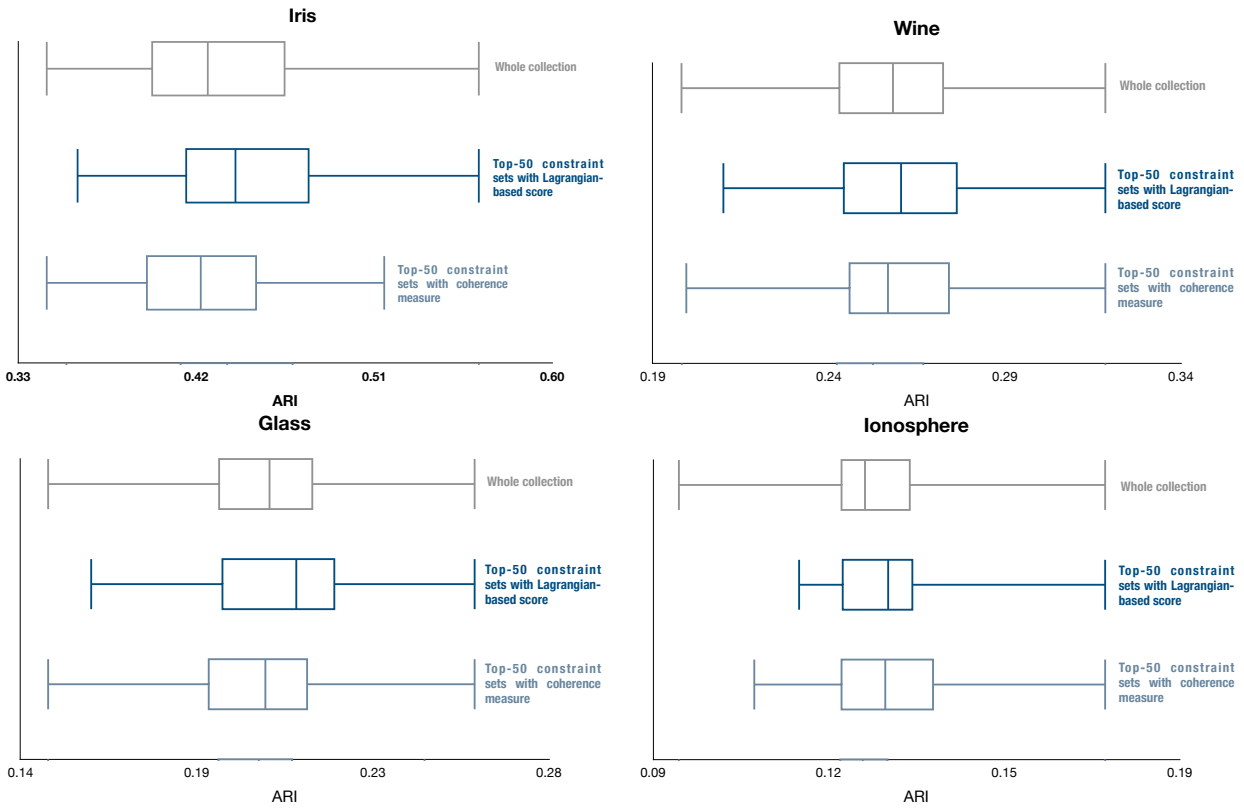


Figure 5.10 Comparison of the ARIs for the whole collection of 100 constraint sets of 25 **erroneous** constraints, and for the top 50 constraint sets selected by the impact score and Davidson's coherence measure.

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.

5.5 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,

Table 5.4 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%

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 appropriate 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* [135, 136], 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 [137, 121]. 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.

CHAPTER 6 ARTICLE 3: EXPLORING DUAL INFORMATION IN DISTANCE METRIC LEARNING FOR CLUSTERING

Authors: Rodrigo Randel, Daniel Aloise and Alain Hertz
Submitted to *Data Mining and Knowledge Discovery* journal, 2021 [1]

Abstract. Distance metric learning algorithms aim to appropriately measure similarities and distances between data points. In the context of clustering, metric learning is typically applied with the assist of side information provided by experts, most commonly expressed in the form of cannot-link and must-link constraints. In this setting, distance metric learning algorithms move closer pairs of data points involved in must-link constraints, while pairs of points involved in cannot-link constraints are moved away from each other. For these algorithms to be effective, it is important to use a distance metric that matches the expert knowledge, beliefs, and expectations, and the transformations made to stick to the side information should preserve geometrical properties of the dataset. Also, it is interesting to filter the constraints provided by the experts to keep only the most useful and reject those that can harm the clustering process. To address these issues, we propose to exploit the dual information associated with the pairwise constraints of the semi-supervised clustering problem. Experiments clearly show that distance metric learning algorithms benefit from integrating this dual information.

Keywords. *clustering, distance metric learning, pairwise constraints, Lagrangian relaxation, duality theory.*

6.1 Introduction

A large collection of data mining techniques strongly rely on the use of similarity or dissimilarity measures among data objects to successfully perform their associated tasks [15]. Consequently, defining a metric capable of recovering the underlying data structure is a critical component for achieving good results in many pattern recognition problems [138]. The effects of an inappropriate dissimilarity metric are notably severe under the unsupervised learning paradigm, mainly due to the lack of background information to evaluate classification performance [74]. This is the case, for example, for the *clustering* problem, one of the most popular data mining tasks, which is aimed to identify hidden homogeneous subgroups, named clusters, from the data [13].

[1] Available at `http://arxiv.org/abs/2105.12703`

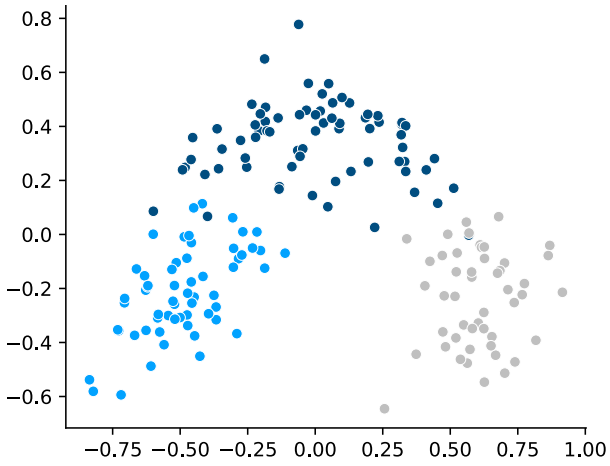
Whereas defining a good fitting dissimilarity metric is highly problem-specific [139], the user of a clustering model has often no evidence or external information to assess the quality of the dissimilarity metric adopted nor of the clusters obtained. To mitigate this difficulty, *distance metric learning* techniques emerged as a mechanism to automatically learn how to appropriately measure similarities and distances through the use of external knowledge about the data [140]. In clustering, distance learning frequently occurs under the semi-supervised paradigm, where domain experts are allowed to provide additional side information regarding the data distribution. Therefore, semi-supervised clustering methods aim to find solutions that are more in line with the expert knowledge, beliefs, and expectations.

Typically, side information is formulated by means of pairwise constraints, for which the user provides information regarding the relationship between a pair of data objects. In this sense, a *must-link* constraint informs that two data points are similar and, therefore, must be assigned to the same cluster. Likewise, a *cannot-link* constraint ensures that a pair of points must be assigned to different clusters. Pairwise constraints arise naturally in many applications, e.g., image retrieval [20], and, in many circumstances, are more practical to be obtained than class-labels [21].

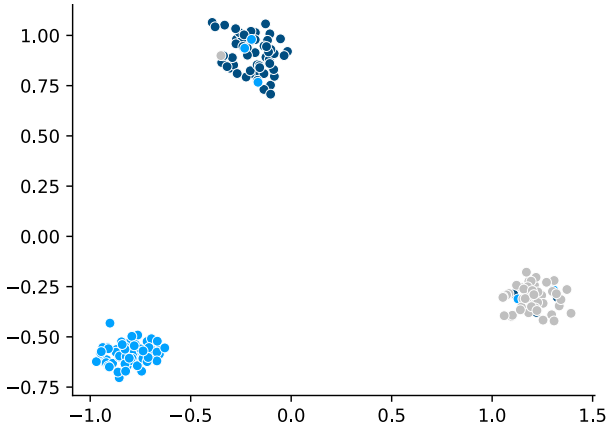
Accordingly, under the semi-supervised clustering paradigm with pairwise constraints, distance learning methods are designed to find transformations to the data features, so as to bring closer pairs of data objects involved in must-link constraints and to move away pairs of data objects involved in cannot-link constraints for the metric space under consideration. Thus, distance metric learning can mitigate the effect of adopting a less appropriate clustering model to capture the underlying structure of the data.

Establishing a re-featured dataset through distance metric learning has its own challenges. For example, the choice of the metric to be used initially in order to define the notion of dissimilarity is crucial and has a significant impact on the resulting clustering solution. For illustration, an example is shown in Figure 6.1, where clustering solutions for the popular `Wine` dataset [134] are obtained with the distance metric learning algorithm MPCK-Means [51], using two different metrics and 20 randomly generated pairwise constraints. The data points are displayed with the two principal components. We observe that the Euclidean distance is a more suitable metric for finding the ground-truth partition than the Cosine distance, even when the same set of pairwise constraints is used. This example illustrates the need to first find a suitable metric so that distance metric learning algorithms can actually benefit from the background knowledge.

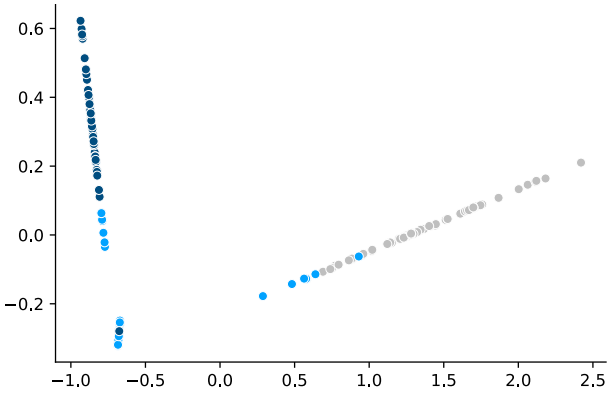
Another issue with distance metric learning methods is that the transformations that are applied to the data points do not necessarily preserve the geometric properties of the dataset.



(a) Ground-truth partition



(b) Output of MPCK-Means based on Euclidean distances



(c) Output of MPCK-means based on Cosine distances

Figure 6.1 Illustration of the use of MPCK-Means on the `Wine` dataset, with two metrics and 20 pairwise constraints.

This can be problematic since these transformations are based on pairwise constraints collected from different sources than the ones that generated the dataset. This occurs because domain experts provide pairwise constraints based on their beliefs and expectations, and these may not be consistent with the distribution of the data. Such a defect is particularly apparent when the pairwise constraints fail to deal well with critical regions of the space (e.g., overlapping of classes) [141]. Therefore, a method capable of keeping the modified space as close as possible to the original one is of great interest. This not only makes the interpretation of the clustering more reliable, but it also contributes to a more accurate view of the transformed dataset when represented using the learned metric.

It has been shown that some constraints can negatively affect the clustering quality [2].

Nevertheless, little is known about which constraints are best able to guide a semi-supervised clustering algorithm. Users of clustering algorithms are thus left with the sole option of using all the available constraints, which corresponds to a hit-or-miss game [1]. A mechanism to assess the importance of pairwise constraints, to keep only those that have a positive effect, would therefore be of great benefit to users.

This research proposes to address the above issues by defining a methodology for measuring and analyzing the effects of pairwise constraints in a clustering model. To this end, we resort to a Lagrangian formulation of the clustering problem, where pairwise constraints are relaxed and transferred to the clustering objective function. This process introduces Lagrange multipliers, called *dual variables*, which indicate the price to be paid by the model in order to violate a constraint. In other words, by associating each pairwise constraint with a dual variable, we can measure the impact of each constraint in the clustering solution. This idea was introduced in our previous work [120] with the objective of identifying the presence of erroneous/contradictory pairwise constraints. Motivated by the wealth of information that can be acquired from dual variables, we propose in this paper to further exploit this knowledge by developing tools to help manage the problems described above in distance metric learning for clustering.

The rest of the paper is organized as follows. In Section 6.2, we summarize prior research on distance metric learning for clustering, with supervision based on pairwise constraints. We then explain in Section 6.3 how to generate and exploit dual information obtained from pairwise constraints within clustering optimization problems. The next three sections illustrate the use of the proposed tools: we show in Section 6.4 how to identify the most appropriate metric in clustering models; in Section 6.5, we explain how to design distance metric learning methods so that the transformations that are applied to the data points preserve as much as possible the geometric properties of the dataset; we describe in Section 6.6 a procedure that filters the most useful constraints and demonstrate how it can be integrated with a deep learning framework recently proposed in the literature. Concluding remarks are given in the last section.

6.2 Related works

In this paper, we focus on distance metric learning algorithms for clustering, with supervision based on pairwise constraints. The literature on metric learning is however much broader, and the reader is referred to Bellet et al. [140] for a thorough review on the subject.

The distance metric learning task is often performed by assuming that dissimilarities are

expressed as Mahalanobis distances [73]. This distance has generated a lot of interest due to its flexibility and nice interpretation in terms of a linear projection [140, 79]. Given a dataset O=$\{o_1, \ldots, o_n\}$ of n points in the d-dimensional space, the objective of the Mahalanobis distance learning is to determine a positive semi-definite $d \times d$ matrix $\mathcal{A}$, such that the Mahalanobis distance between two points o_i and o_j is decreased if a constraint imposes that they must belong to the same cluster, and is increased if the points are to be in different clusters. The Mahalanobis distance between points o_i and o_j in $\mathbb{R}^d$ is defined as:

$$d_{\mathcal{A}}(o_i, o_j) = \sqrt{(o_i - o_j)^T \mathcal{A}(o_i - o_j)}. \tag{6.1}$$

The learned matrix $\mathcal{A}$ can be used to transform the dataset O into a set O' by defining $o'_i = \mathcal{A}^{1/2} o_i$ for $i = 1, \ldots, n$ (where $\mathcal{A}^{1/2}$ is the unique matrix $\mathcal{B}$ that is positive semidefinite and such that $\mathcal{B}\mathcal{B} = \mathcal{B}^T\mathcal{B} = \mathcal{A}$). The standard Euclidean distance can then be used on O' by classical data mining algorithms. The seminal work exploring this idea was done by [74]. Formally, given a set $\mathcal{ML}$ of must-link constraints and a set $\mathcal{CL}$ of cannot-link constraints, the *Mahalanobis distance metric learning problem* is expressed as follows, where $\mathbb{S}$ is the set of positive semidefinite $d \times d$ matrices :

$$\begin{aligned} &\max_{\mathcal{A} \in \mathbb{S}} \sum_{(o_i, o_j) \in \mathcal{CL}} d_{\mathcal{A}}(o_i, o_j) \\ &\text{subject to} \sum_{(o_i, o_j) \in \mathcal{ML}} d_{\mathcal{A}}(o_i, o_j) \leq c \end{aligned} \tag{6.2}$$

In words, the sum of the distances between dissimilar points is maximized, while for similar points, this sum must be less than a constant c which is typically set equal to 1. One could also choose to minimize the sum of the distances between similar points, while keeping the sum of distances between dissimilar points greater than c. Problem (6.2) can be solved to optimality using a projected gradient descent algorithm. However, if $\mathcal{A}$ is a full matrix, the time complexity of determining the eigenvalues of $\mathcal{A}$ is $O(d^2)$, which is computationally too expensive for high-dimensional data [51, 76]. To reduce this complexity, one can impose that $\mathcal{A}$ be a diagonal matrix, which is equivalent to performing a feature weighting on the dataset.

A classical distance metric learning technique for clustering with pairwise constraints is the MPCK-Means algorithm [51]. Instead of computing a single matrix $\mathcal{A}$, MPCK-Means determines one matrix per cluster. The method can thus define local transformations which allow the clusters to have different shapes. This feature can be particularly useful to enforce specific shapes. For example, squared Euclidean distances used by the classical k-means heuristic are suitable for spheroidal clusters, but fail to represent dissimilarities when the clusters have

other shapes.

We should also mention that the learning of nonlinear shapes can be achieved by using Bregman distances [75, 76] and kernel functions [77, 78, 79], both being more suitable than Mahalanobis distances for high-dimensional data, or to work with datasets with nonlinear structures.

Methods based on deep learning have also been developed [40, 41, 1]. For example, [1] recently proposed to use the Deep Embedded Clustering (DEC) algorithm [38] followed by a training procedure that incorporates the violation of secondary constraints in the loss function. The DEC method consists of first learning an embedded representation of the data points using an autoencoder network, and then executing a self-training routine to learn a clustering partition. To accomplish that, DEC uses the embedded points to define a *soft* membership distribution (points can be partially assigned to more than one cluster), and then approximates it to a target distribution that resembles a *hard* clustering membership (where data points are assigned to exactly one cluster) using the Kullback-Leibler divergence [39] loss function. The work of Zhang et al. [1], extends the loss function by adding terms that accounts for diverse types of supervision, including pairwise constraints. It has been shown that the proposed algorithm outperforms previous semi-supervised clustering algorithms in terms of clustering accuracy.

6.3 Dual information from the pairwise constraints

A k-partition of a set is a grouping of its elements into k disjoint non-empty subsets called clusters. Given a set O of n data points and an integer k, we write $\mathcal{P}(O,k)$ for the set of all k-partitions of O. Let $f : \mathcal{P}(O,k) \to \mathbb{R}$, be a function, usually called *clustering criterion*, that assigns a value to every k-partition in $\mathcal{P}(O,k)$. The clustering problem is to determine a k-partition $P \in \mathcal{P}(O,k)$ with minimum (or maximum) value $f(P)$.

A widely used clustering criterion is the minimum sum of squared Euclidean distances from each data point to the centroid of its cluster, or minimum sum-of-squares clustering (MSSC) for short, which expresses both homogeneity and separation [131]. Let $\mathcal{ML}$ and $\mathcal{CL}$ be the sets of pairs (o_i, o_j) of data points involved in must-link and cannot-link constraints, respectively, where $(o_i, o_j) \in \mathcal{ML} \cup \mathcal{CL}$ only if $i < j$. The semi-supervised version of the clustering problem with MSSC can be expressed by means of the following optimization problem:

$$\min \quad \sum_{i=1}^{n} \sum_{c=1}^{k} x_i^c \|o_i - y_c\|^2 \tag{6.3}$$

$$\text{subject to} \quad \sum_{c=1}^{k} x_i^c = 1, \quad \forall i = 1 \dots n \tag{6.4}$$

$$x_i^c + x_j^c \leq 1 \quad \forall (o_i, o_j) \in \mathcal{CL}, \forall c = 1, \dots, k \tag{6.5}$$

$$x_i^c - x_j^c = 0 \quad \forall (o_i, o_j) \in \mathcal{ML}, \ \forall c = 1, \dots, k \tag{6.6}$$

$$x_i^c \in \{0, 1\} \quad \forall i = 1, ..., n, \ \forall c = 1, \dots, k. \tag{6.7}$$

The feasible solutions to this problem correspond to partitions $\{C_1, \dots, C_k\} \in \mathcal{P}(O, k)$ where every binary decision variable x_i^c indicates whether data point o_i is assigned to cluster C_c, and y_c is the centroid of cluster C_c. Constraints (6.4) ensure that each data point is assigned to exactly one cluster. The cannot-link constraints are expressed by equations (6.5) and the must-link constraints by equations (6.6). To avoid situations where these constraints are satisfied with equality, we can replace them by the following equivalent constraints where ϵ is any real number in $]0, 1[$:

$$x_i^c + x_j^c \leq 1 + \epsilon \qquad \forall (o_i, o_j) \in \mathcal{CL}, \quad \forall c = 1, \dots, k \tag{6.5$'$}$$

$$x_i^c - x_j^c \leq \epsilon \qquad \forall (o_i, o_j) \in \mathcal{ML}, \quad \forall c = 1, \dots, k \tag{6.6$'$}$$

$$x_j^c - x_i^c \leq \epsilon \qquad \forall (o_i, o_j) \in \mathcal{ML}, \quad \forall c = 1, \dots, k. \tag{6.6$''$}$$

We aim to investigate how the pairwise constraints affect the clustering objective as expressed in (6.3). We believe that the cost of complying with a constraint may provide information on its importance. We thus attempt to answer the following questions:

- what are the most difficult constraints to satisfy?
- is the adopted metric in agreement with the imposed constraints?
- what are the most useful constraints for obtaining a low cost partition?

This constitutes a form of sensitivity analysis [142] whose goal is to measure the impact of modifications of the input variables on the result of a clustering model. To obtain such information, we have recourse to the Lagrangian duality theory, as explained in [120]

The Lagrangian function for the optimization problem (6.3)-(6.7) can be obtained by introducing dual variables $\eta_{ij}^c, \lambda_{ij}^c$ and γ_{ij}^c to penalize violations of inequality constraints (6.5$'$), (6.6$'$) and (6.6$''$). Specifically, the Lagrangian function $L(\eta, \lambda, \gamma)$ is defined as follows:

$$L(\eta, \lambda, \gamma) = \min \quad \sum_{i=1}^{n} \sum_{c=1}^{k} x_i^c \|o_i - y_c\|^2$$

$$+ \sum_{(o_i,o_j)\in\mathcal{CL}} \sum_{c=1}^{k} \eta_{ij}^{c}(1+\epsilon-x_i^c-x_j^c)$$

$$+ \sum_{(o_i,o_j)\in\mathcal{ML}} \sum_{c=1}^{k} \lambda_{ij}^{c}(\epsilon+x_i^c-x_j^c)$$

$$+ \sum_{(o_i,o_j)\in\mathcal{ML}} \sum_{c=1}^{k} \gamma_{ij}^{c}(\epsilon+x_j^c-x_i^c) \tag{6.8}$$

$$\text{subject to} \quad \sum_{c=1}^{k} x_i^c = 1, \quad \forall i = 1\ldots n \tag{6.9}$$

$$x_i^c \in \{0,1\} \quad \forall i = 1,...,n; \quad \forall c = 1,\ldots,k \tag{6.10}$$

and the dual of the integer program (6.3)-(6.7) can be expressed as follows:

$$L_D = \max_{\eta,\lambda,\gamma\leq 0} L(\eta,\lambda,\gamma). \tag{6.11}$$

The dual variables $\eta_{ij}^c, \lambda_{ij}^c$ and γ_{ij}^c provide an estimation of the cost paid by the clustering optimization model for respecting the associated constraints. If a pairwise constraint is inherently satisfied by (6.3)-(6.7), its associated dual variable should be zero. On the other hand, if a constraint is necessary to obtain an optimal solution to (6.3)-(6.7), a penalty must be paid for its violation, which implies that the associated dual variable should be strictly negative.

A sub-gradient optimization algorithm can be used to solve (6.11) and thus determine optimal values of the dual variables. The weak duality theorem [93, 90] asserts that L_D is the best possible lower bound on the optimal value of (6.3)-(6.7). Consequently, the optimal values of the dual variables are reliable estimates of the price to be paid for complying with the constraints associated with them. For more details, the reader is referred to [120].

Formulation (6.3)-(6.7) is based on the MSSC clustering criterion. Neverthe- less, the proposed methodology is not limited to this particular objective. In fact, the strategy is flexible to be used in conjunction with several other clustering criteria and models represented in mathematical programming language [13]. To illustrate that, we use the k-medoids model [27]. More precisely, let D be a matrix, where each entry d_{ij} indicates the dissimilarity between data points o_i and o_j. The medoid of a cluster is defined as the data point in the cluster whose average dissimilarity to all the data points in the cluster is minimal. The k-medoids problem is to determine a partition in $\mathcal{P}(O,k)$ that minimizes the sum of the distances of the data points to the medoid of their cluster. The semi-supervised version of this clustering

problem can be expressed by means of the following optimization problem:

$$\min \quad \sum_{i=1}^{n}\sum_{c=1}^{n} x_i^c d_{ic} \tag{6.12}$$

$$\text{subject to} \quad \sum_{c=1}^{n} x_i^c = 1, \qquad \forall i = 1, ..., n \tag{6.13}$$

$$x_i^c \leq y_c \qquad \forall i = 1, \ldots, n, \forall c = 1, ..., n \tag{6.14}$$

$$\sum_{c=1}^{n} y_c = k \tag{6.15}$$

$$x_i^c + x_j^c \leq 1 \qquad \forall (o_i, o_j) \in \mathcal{CL}, \forall c = 1, \ldots, n \tag{6.16}$$

$$x_i^c - x_j^c = 0 \qquad \forall (o_i, o_j) \in \mathcal{ML}, \forall c = 1, \ldots, n \tag{6.17}$$

$$x_i^c \in \{0, 1\} \qquad \forall i = 1, \ldots, n, \forall c = 1, \ldots, n, \tag{6.18}$$

$$y_c \in \{0, 1\} \qquad \forall c = 1, \ldots, n, \tag{6.19}$$

where every binary variable y_c indicates whether the data point o_c is one of the k medoids, and every binary variable x_i^c indicates whether data point o_i is assigned to a cluster having o_c as medoid. The Lagrangian function $L(\eta, \lambda, \gamma)$ associated with this optimization problem is then defined as in the previous case by replacing constraints (6.16) and (6.17) by penalty terms in the objective function and the same sub-gradient optimization technique as in [120] can be used to determine $L_D = \max_{\eta,\lambda,\gamma \leq 0} L(\eta, \lambda, \gamma)$.

6.4 Identifying an appropriate dissimilarity measure

Although the choice of the most appropriate dissimilarity metric is problem-specific, users of an algorithm or a clustering model often have very little information to support their choice which can have a significant impact on the analysis of the quality of a solution. For example, the homogeneity of a cluster depends on the similarity measure between the objects which is defined using the chosen metric.

Distance learning methods are also highly dependent on the chosen distance metric. For example, the most commonly used dissimilarity measure for quantitative data is the Euclidean distance [76]. It assumes that each data feature is independent and equally important, which is not necessarily the case for the considered dataset. An alternative is to weight the relative importance of the data features. For instance, in the weighted Euclidean Distance-Based approach [143], data features are weighted according to their discriminating effect. These weights are generally set based on user preferences and interpretation, which is a subjective task prone to error. One can also choose to use secondary information such as pairwise

constraints to improve the initial dissimilarities calculated from Euclidean distances, but this might not compensate for the fact that a more suitable metric exists for the underlying clustering optimization model.

To tackle this issue, we propose to use a *fitness score* which measures the adequacy of a metric with a given clustering model. This score is therefore a tool that helps experts to identify the metric that most closely matches their knowledge and beliefs. The fitness score $F(m)$ of a metric m is defined as

$$F(m) = \sum_{(o_i,o_j)\in\mathcal{CL}} \sum_{c=1}^{k} \delta(\eta_{ij}^c = 0) + \sum_{(o_i,o_j)\in\mathcal{ML}} \sum_{c=1}^{k} \delta(\lambda_{ij}^c = \gamma_{ij}^c = 0), \tag{6.20}$$

where $\eta_{ij}^c, \lambda_{ij}^c$ and γ_{ij}^c are the optimal values of the dual variables resulting from the optimization of L_D in (6.11), and $\delta(e) = 1$ if expression e is true. The metric with the highest fitness score should be preferred for the given clustering model, as more pairwise constraints are inherently respected.

In the following subsections, we validate the ability of the fitness score to detect the most adequate metric for a clustering problem. Experiments are conducted on synthetic and real-world datasets.

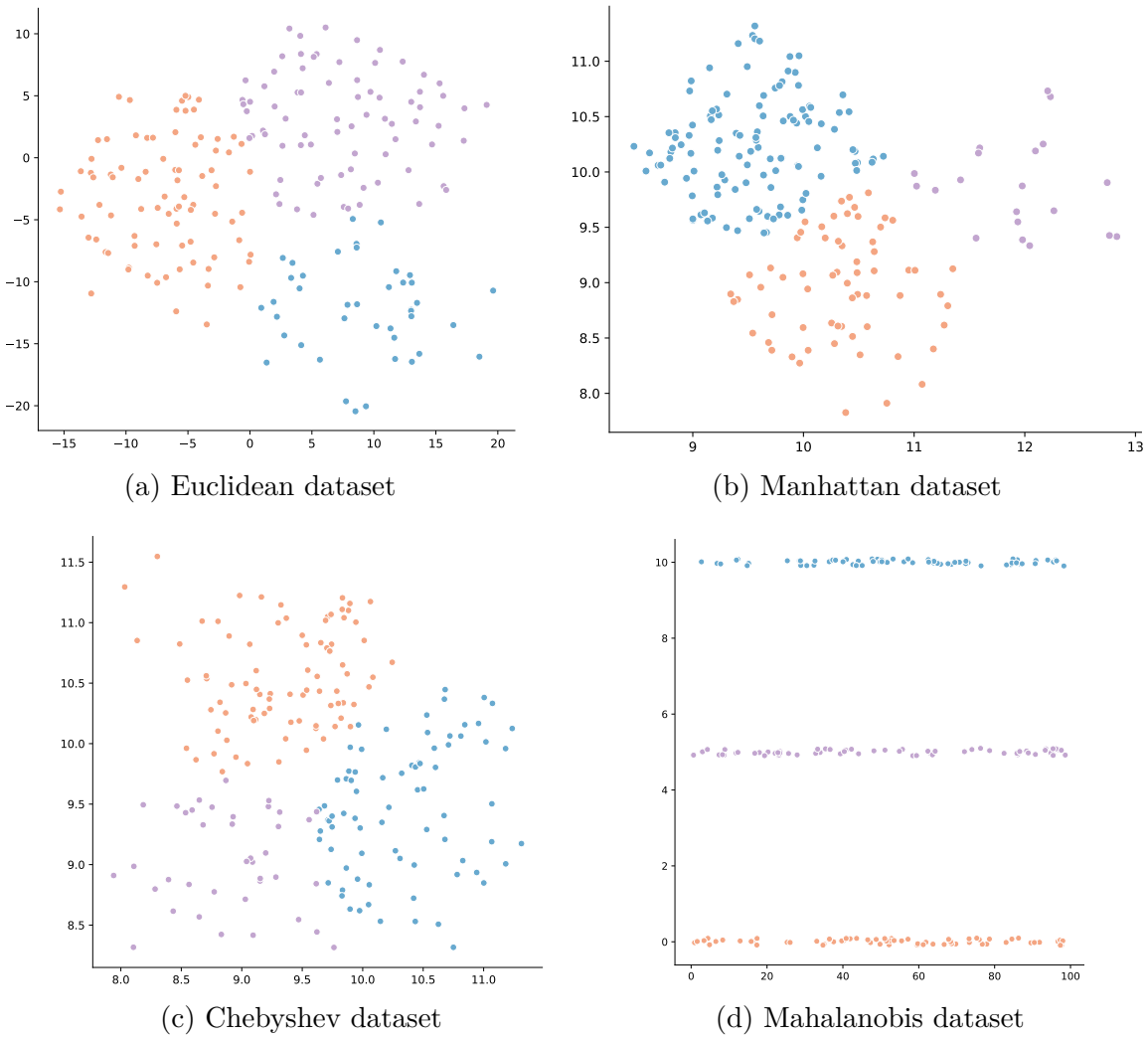
6.4.1 Validation of the fitness score for synthetic datasets

In order to assess the usefulness of the proposed fitness score, we selected four of the most commonly used distance metrics in clustering problems, namely the Euclidean, Manhattan, Chebyshev, and Mahalanobis distances.

For each of the four metrics, we generated a two-dimensional dataset with 200 points and three balanced clusters C_1, C_2, C_3. This is how we proceeded.

- For the Euclidean, Manhattan and Chebyshev distances, we first generated 1000 pairs of points with coordinates drawn from a normal distribution $\mathcal{N}(10, 1)$, and determined the maximum distance Δ (using the chosen metric) between the two points of a pair. We then repeatedly generated three points c_1, c_2, c_3 with coordinates drawn from a normal distribution $\mathcal{N}(10, 1)$, until $\min\{d(c_1, c_2), d(c_1, c_3), d(c_2, c_3)\} \geq \Delta/3$, where d is the considered distance metric. The three resulting points were defined as the centers of the three clusters. Starting from C_1=C_2=C_3=$\emptyset$, we then iteratively generated points p with coordinates drawn from a normal distribution $\mathcal{N}(10, 1)$, determined the index i such that $d(p, c_i) = \min\{d(p, c_1), d(p, c_2), d(p, c_3)\}$, and added p to C_i if $|C_i|$<67. The process stopped when $|C_1|+|C_2|+|C_3|$=200.

- For the Mahalanobis distance, we have defined c_1=(50, 0), c_2=(50, 5) and c_3=(50, 10) as centers of the three clusters. The 200 data points were then generated so that every x coordinate is a random number chosen from the uniform distribution on the interval [0,100], and the y coordinate of every data point in C_i ($i = 1, 2, 3$) is a random number generated from the uniform distribution on the interval $[5(i-1)-0.1, 5(i-1)+0.1]$.



(a) Euclidean dataset

(b) Manhattan dataset

(c) Chebyshev dataset

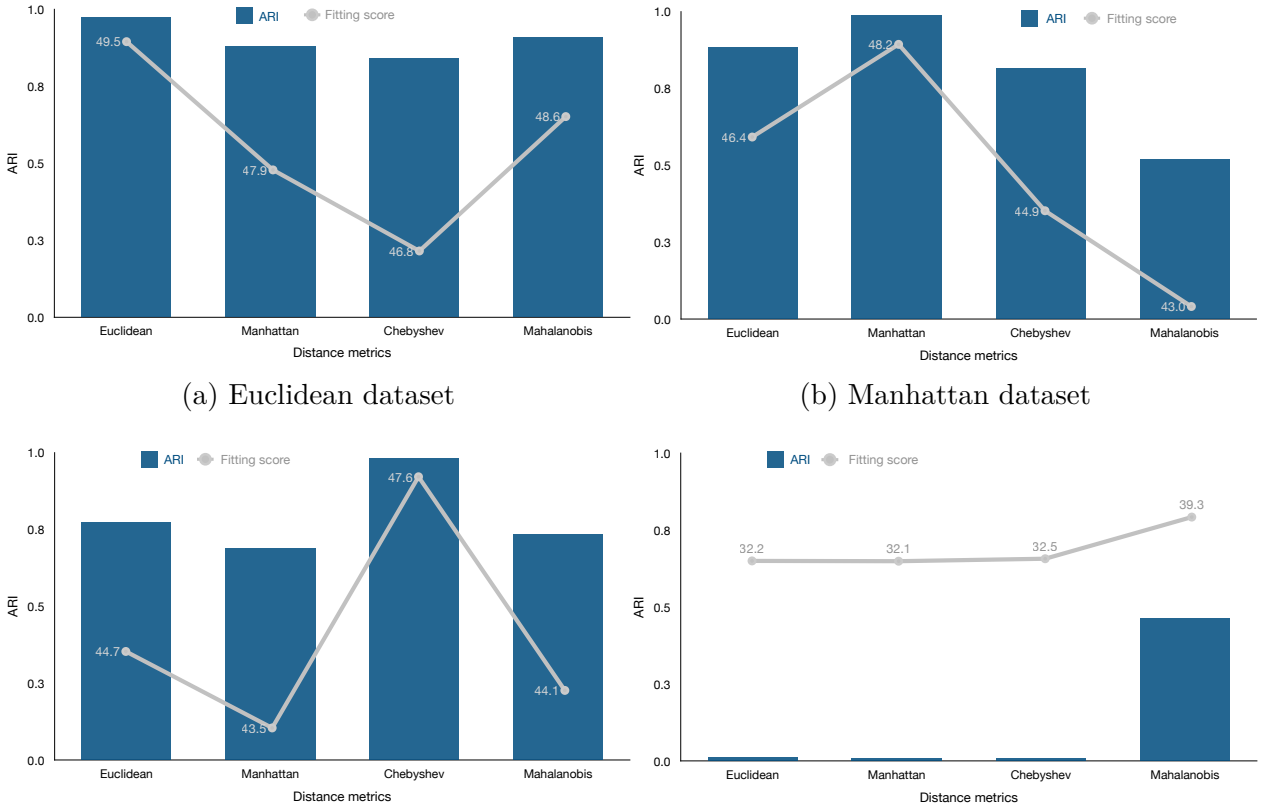
(d) Mahalanobis dataset

Figure 6.2 Four synthetic datasets for four distance metrics, with their ground-truth partitions.

The four generated datasets and their ground-truth partitions are shown on Figure 6.2. Pairwise constraints were obtained using the *boosting aggregating framework* [144]. For each of the 4 datasets, we generated 500 sets $E_1, \ldots, E_{500}$ of constraints, using the ground-truth partition. Each set E_i contains n_i constraints, with n_i chosen randomly according to a uniform distribution in $\{1, \ldots, 100\}$.

We take advantage of the flexibility of the k-medoids model to solve the unsupervised clustering problem associated with each of the four datasets and each of the four metrics. More precisely, we set d_{ij} equal to the distance between o_i and o_j, using the considered metrics, and use CPLEX 12.8 to solve model (6.12)-(6.19) without constraints (6.16) and (6.17). The clustering accuracy of the resulting solutions is obtained using the Adjusted Random Index (ARI) [118], which measures the similarity between the obtained clustering solutions and the ground-truth partition.

For each dataset, each set E_i of pairwise constraints, and each metric m, we ran the sub-gradient algorithm of [120] to solve (6.11) and so produce optimal dual variables which give a fitness score as defined in (6.20). A time limit of 10 seconds has been allocated to each execution of the sub-gradient algorithm. We report in Figure 6.3, for each dataset and each of the four metrics, the average fitness score calculated over the 500 sets of constraints. We also indicate the ARI values of the solutions produced by the unsupervised clustering algorithm based on the k-medoids model.



(a) Euclidean dataset

(b) Manhattan dataset

(c) Chebyshev dataset

(d) Mahalanobis dataset

Figure 6.3 Fitness scores and ARI for synthetic datasets.

We observe that the fitness score is able to identify the most suitable metric for all datasets.

The fluctuation of the fitness score agrees with that of the ARI and the best score is always obtained with the appropriate metric. Although the experiments were carried out on synthetic data, the results demonstrate that the optimal dual values of (6.11) are useful for exploiting the information provided by the pairwise constraints, in order to suggest the metric which seems to adhere the most to the data.

6.4.2 Validation of the fitness score for five real-world datasets

We now consider five real-world datasets, the first four being available on the UCI repository [134], and the fifth being described in Rodrigues et al. [145]. Their characteristics are summarized in Table 6.1.

Table 6.1 Real-data Applications for Evaluating the Score.

Dataset	Samples	Classes	Features	Type
`Iris`	150	3	4	Quantitative
`Wine`	178	3	13	Quantitative
`Control`	600	6	60	Quantitative; time series
`Twenty newsgroup`	600	4	-	Text
`Eclipse bug report`	460	5	-	Discrete sequence; text

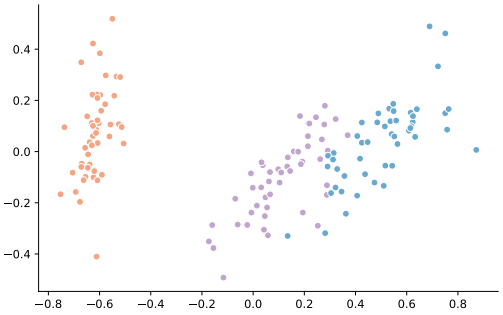
Quantitative datasets

For the quantitative datasets `Iris` and `Wine`, we used the same distance metrics as those carried out for the synthetic datasets of Section 6.4.1, that is, Euclidean, Manhattan, Chebyshev, and Mahalanobis.

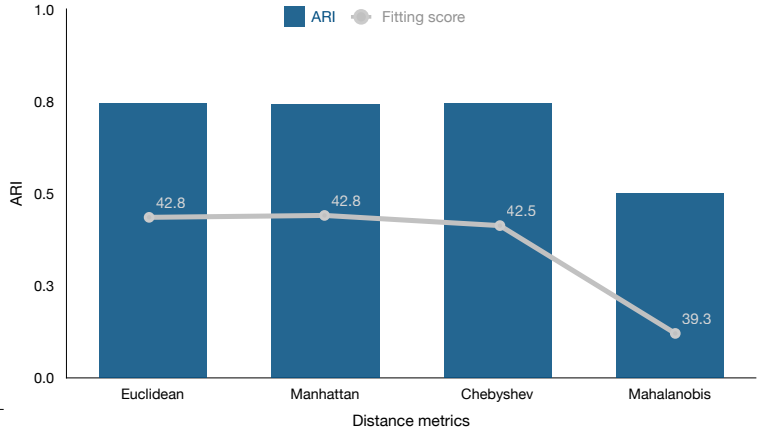
For the `Control` dataset, each data point represents a time series composed of 60 values. Each time series was decomposed into six segments of 10 consecutive values, and we have set $o_i = (o_i^1, \ldots, o_i^6)$, where o_i^k $(k = 1, \ldots, 6)$ are the segments of o_i. The distance between two data points o_i and o_j was then defined as $\sum_{k=1}^{6} d(o_i^k, o_j^k)$, where d is the considered metric.

For each dataset, we generated 500 constraints sets from the available ground-truth partitions, using the *boosting aggregating framework* [144], as was done for the synthetic dataset. The time limit for the sub-gradient algorithm has been set at 10 seconds for the `Iris` and `Wine` instances, and at 30 seconds for the `Control` dataset. The results are shown in Figure 6.4, alongside the representation of the datasets with the two principal components and their ground-truth partitions.

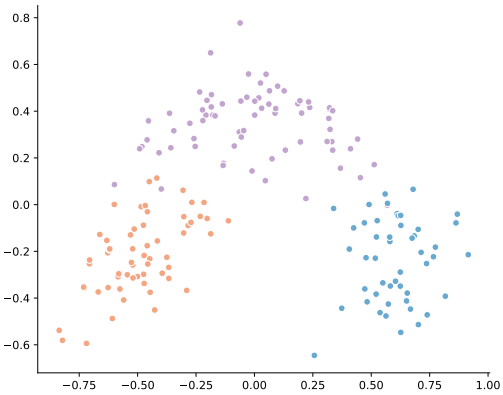
We observe that the L_p norm-based distances (i.e., the Euclidean, Manhattan and Chebyshev



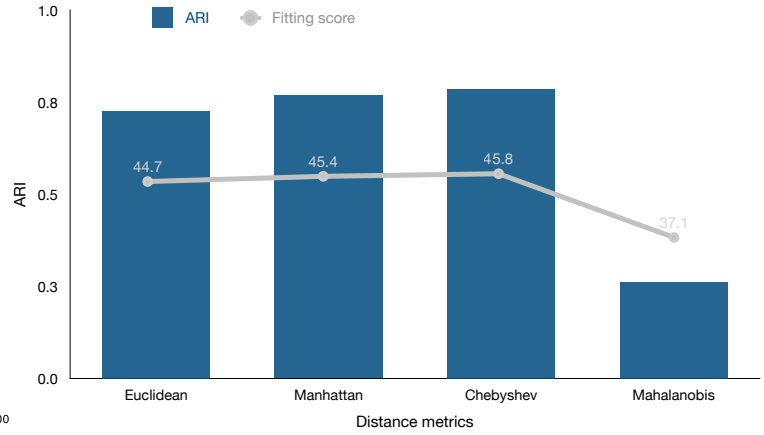
(a) `Iris` dataset



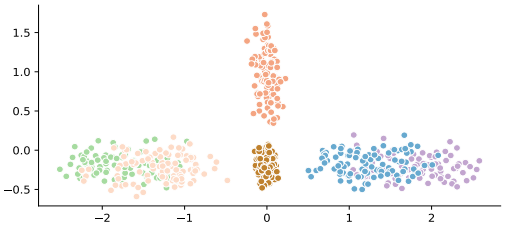
(b) Results for `Iris`



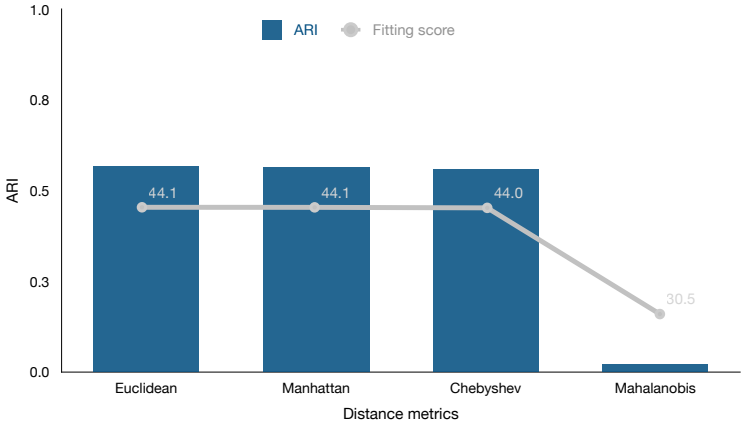
(c) `Wine` dataset



(d) Results for `Wine`



(e) `Control` dataset



(f) Results for `Control`

Figure 6.4 Ground-truth partition, fitness score and ARI for three quantitative datasets.

distances) give similar results, both in terms of clustering accuracy and fitness score. This is not surprising given the popularity of these types of metrics for quantitative data clustering.

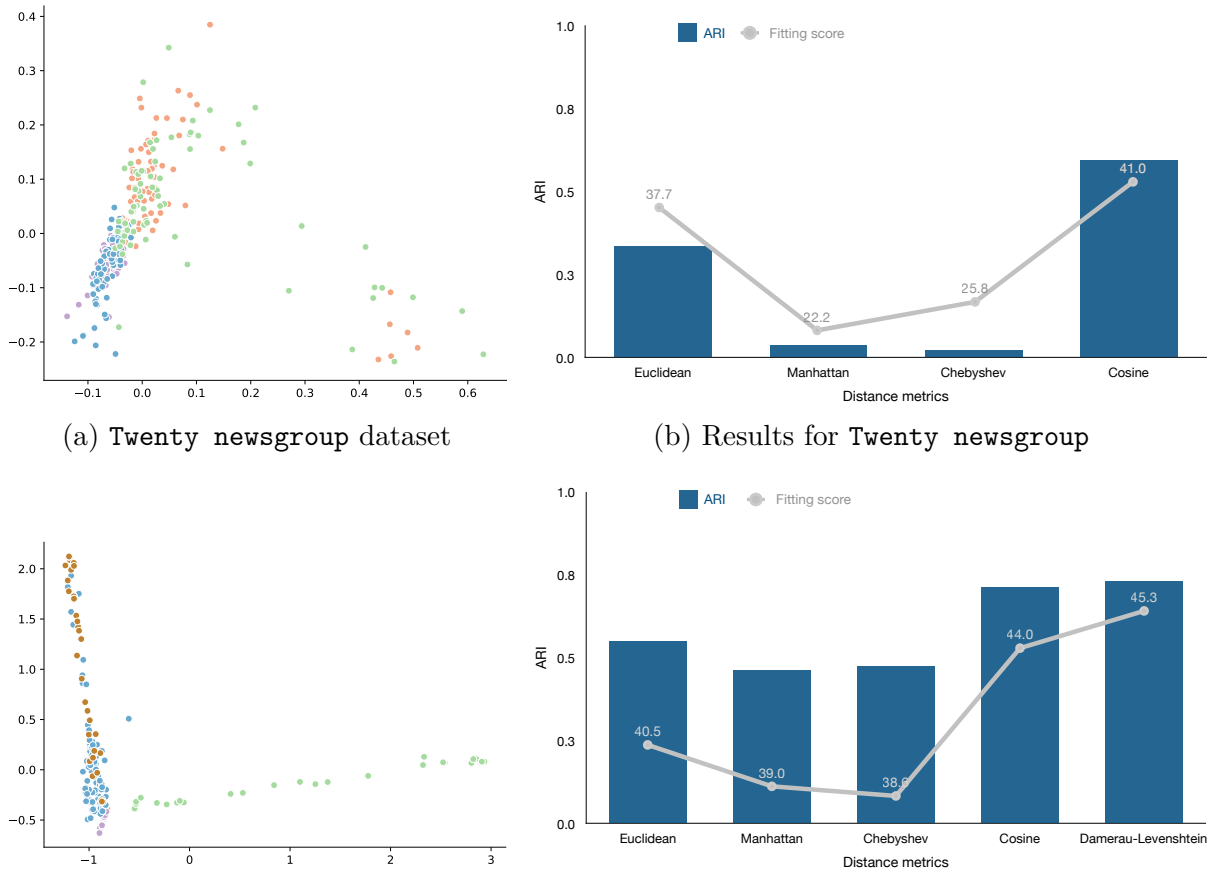
Textual data

The datasets `Twenty newsgroup` and `Eclipse bug report` are composed of text samples. As typically done in document classification tasks, the texts were first converted to quantitative data through the use of a bag of words. For both datasets, we defined 10,000 features, which are originated from the most frequent terms in the dataset vocabulary. The direct use of L_p norm-based distances on raw textual data is not recommended due to data sparsity and high-dimensionality. Besides, they are prone to be inconsistent if not normalized because the distance between two long documents is very often larger than that between two short documents even if the two long documents are similar, and the short documents are unrelated [15]. In contrast, the *cosine distance* works by computing the angle between data samples, and is a popular choice for varied length-sized document classification.

In addition, instead of working with the raw frequency of the terms, we enhanced the features by calculating the *term frequency–inverse document frequency* (Term Frequency–Inverse Document Frequency (TF-IDF)) [e.g. 146] to use global statistical measures to improve the dissimilarity computation. TF-IDF is based on the principle that documents matching with respect to rare terms are more likely to be similar than documents sharing many common terms (e.g. *the*, *like*). In our experiments, we considered every data point as a vector in $\mathbb{R}^{10,000}$, where each coordinate indicates the TF-IDF of the corresponding term.

The *Damerau–Levenshtein* distance is commonly used to establish the dissimilarity between discrete sequences. It measures the minimum number of operations necessary to change one sequence into another. Since the `Eclipse bug report` dataset is structured as a discrete sequence of stack trace functions, we decided to include this other distance in our tests. A time limit of 30 seconds has been allocated to each execution of the sub-gradient algorithm.

The results are reported in Figure 6.5. We observe that our fitness score was able to identify the Cosine distance as an appropriate metric for clustering text documents. In addition, the Damerau–Levenshtein distance has given the highest fitness score for the `Eclipse bug report` instance. This demonstrates once again that the proposed fitness score is a valid tool to suggest a metric that adheres the most to the data.



(a) `Twenty newsgroup` dataset

(b) Results for `Twenty newsgroup`

(c) `Eclipse bug report` dataset

(d) Results for `Eclipse bug report`

Figure 6.5 Ground-truth partition, fitness score and ARI for two text datasets.

6.5 Maintaining geometrical properties of the dataset

In this section, we demonstrate that dual information can be of great interest when experts are not entirely sure of the pairwise constraints they provide. In such circumstances, one may wish to perform distance metric learning more carefully, preserving as much as possible the characteristics of the original data distribution while incorporating the domain knowledge embedded in the pairwise constraints.

With that in mind, we propose to use dual information to establish an order in which pairwise constraints should be processed by metric learning algorithms. This sorting step aims to help these algorithms find the least impactful data transformations that can be applied, that is, those transformations that require minimal modifications to the original data. Thus, our strategy aims not only to integrate the information contained in the pairwise constraints, but also to maintain a reliable representation of the data.

With MSSC as underlying clustering model, we designed a very simple distance metric learning algorithm that iteratively selects a violated constraint, and then performs data transformations necessary to satisfy it. The method exploits the dual information in order to rank the pairwise constraints according to their estimated impact on the clustering objective function (6.3).

Algorithm 7 describes the steps of the proposed method. The algorithm begins by running an unsupervised clustering algorithm that generates an initial solution X to the clustering problem (6.3)-(6.7), without constraints (6.5) and (6.6) (Step 3). We then run the subgradient algorithm to solve (6.11) and thus produce optimal dual values $\eta_{ij}^c, \lambda_{ij}^c$ and γ_{ij}^c (Step 4). An impact score $\mathcal{I}_{ij}$ is then calculated for each pairwise constraint $(o_i, o_j) \in \mathcal{CL} \cup \mathcal{ML}$, this score being defined as the sum of the values of the dual variables which are associated with the constraint (Step 5). Next, we determine the pair $(o_i, o_j) \in \mathcal{CL} \cup \mathcal{ML}$ of maximum impact score $\mathcal{I}_{ij}$ whose associated constraint is violated by X (Step 10). The reason behind this idea of maximizing the impact score is that the higher the value of a dual variable (i.e. the less negative it is), the lower will be the gain in the objective function if the associated constraint is relaxed. Thus, the data transformation required to satisfy such a constraint will presumably be small. We then run Algorithm 8 that moves o_i and/or o_j in order to try to satisfy constraint (o_i, o_j) (Step 11). Details on how this is done are given below. This process is repeated until X satisfies all constraints.

Note that the pairwise constraints can be conflicting and Algorithm 7 would therefore never stop. In addition, the algorithm can cycle. Indeed, suppose that $k = 2$ and that there are only two cannot-link constraints (o_1, o_2) and (o_1, o_3). If o_1 and o_2 belong to C_1 while o_3 belongs to C_2, the algorithm can possibly move o_1 towards C_2 to satisfy constraint (o_1, o_2). But (o_1, o_3) will then probably be violated, and the algorithm can possibly move o_1 back towards C_1, and this cycle can be repeated indefinitely. To avoid such situations, when moving a data point $o_i \in C_j$ to a new position, we add the pair (i, j) in an initially empty list $\mathcal{L}$ (Step 12), and we then forbid moving o_i back to C_j. More precisely, suppose constraint $(o_i, o_j) \in \mathcal{ML}$ is violated by X. At Step 6, we define M_{ij} as the set of indices c of clusters such that the move of o_i and o_j towards C_c is not forbidden (i.e., their move does not belong to $\mathcal{L}$). Similarly, for a violated constraint $(o_i, o_j) \in \mathcal{CL}$, we define M_{ij} at Step 7 as the set of indices c of clusters such that C_c does not contain o_i and o_j (i.e., $x_i^c{=}x_j^c{=}0$), and the move of at least one of the two data points towards cluster C_c does not belong to $\mathcal{L}$. Let $\mathcal{M}$ be equal to the union of the sets M_{ij} over all $(o_i, o_j) \in \mathcal{ML} \cup \mathcal{CL}$ (Step 8). The algorithm stops (Step 14) when $\mathcal{M}{=}\emptyset$, which means that X satisfies all the constraints, or all moves which would satisfy a violated constraint are in $\mathcal{L}$. The algorithm is finite since a data point o_i is moved towards a cluster C_c only if $(i, c){\notin}\mathcal{L}$, while $\mathcal{L}$ increases at each iteration.

Algorithm 7 Using dual information for satifying all pairwise constraints

1: Set $\mathcal{L} \leftarrow \emptyset$.
2: **repeat**
3: Run an unsupervised MSSC algorithm for solving model (6.3)-(6.7) without constraints (6.5) and (6.6) and let $X = (x_i^c)$ be the resulting solution.
4: Run the sub-gradient algorithm to solve (6.11) and thus produce optimal dual values $\eta_{ij}^c, \lambda_{ij}^c$ and γ_{ij}^c.
5: For all $(o_i, o_j) \in \mathcal{CL} \cup \mathcal{ML}$ do $\mathcal{I}_{ij} \leftarrow \begin{cases} \sum_{c=1}^{k} \eta_{ij}^c & \text{if } (o_i, o_j) \in \mathcal{CL} \\ \sum_{c=1}^{k} (\lambda_{ij}^c + \lambda_{ij}'^c) & \text{if } (o_i, o_j) \in \mathcal{ML}. \end{cases}$
6: For all $(o_i, o_j) \in \mathcal{ML}$ set $M_{ij} \leftarrow \{c \in \{1, \ldots, k\} \mid \{(i,c),(j,c)\} \cap \mathcal{L} = \emptyset\}$ if X violates (o_i, o_j), and $M_{ij} \leftarrow \emptyset$ otherwise.
7: For all $(o_i, o_j) \in \mathcal{CL}$ set $M_{ij} \leftarrow \{c \in \{1, \ldots, k\} \mid x_i^c = x_j^c = 0 \text{ and } \{(i,c),(j,c)\} \nsubseteq \mathcal{L}\}$ if X violates (o_i, o_j), and $M_{ij} \leftarrow \emptyset$ otherwise.
8: Set $\mathcal{M}$ equal to the union of the sets M_{ij} over all $(o_i, o_j) \in \mathcal{ML} \cup \mathcal{CL}$.
9: **if** $\mathcal{M} \neq \emptyset$ **then**
10: Determine the pair (i, j) that maximizes $\mathcal{I}_{ij}$ among those with $M_{ij} \neq \emptyset$.
11: Run Algorithm 8 to modify the coordinates of o_i and/or o_j.
12: Let c_i and c_j be the indices of the clusters such that $x_i^{c_i} = x_j^{c_j} = 1$.
if the coordinates of o_i have been changed at Step 11 **then** add (i, c_i) to $\mathcal{L}$.
if the coordinates of o_j have been changed at Step 11 **then** add (j, c_j) to $\mathcal{L}$.
13: **end if**
14: **until** $\mathcal{M} = \emptyset$.

We now explain how Algorithm 8 moves o_i and/or o_j in an attempt to satisfy a violated constraint $(o_i, o_j) \in \mathcal{ML} \cup \mathcal{CL}$. Let us first define more precisely what we mean by moving a point $o_i \in C_c$ towards a new cluster C_{c^*}. Such a move is obtained by modifying the coordinates of o_i as follows. Let ℓ_1 be the line containing o_i and parallel to $\overrightarrow{y_c, y_{c^*}}$, let ℓ_2 be the perpendicular bisector of the segment that connects y_c with y_{c^*}, and let p be the point at the intersection of ℓ_1 and ℓ_2 : point o_i is moved to the new location p' such that $\overrightarrow{o_i p'} = \frac{101}{100}\overrightarrow{o_i p}$, which means that o_i is then slightly closer to y_{c^*} than to y_c. Figure 6.6 illustrates such movement.

As mentioned in Section 6.3, the value $f(X)$ of a solution X with MSSC as clustering criterion is

$$\sum_{i=1}^{n} \sum_{c=1}^{k} x_i^c \|o_i - y_c\|^2$$

where y_c is the centroid of cluster C_c. As shown in [86], if a new solution X' is obtained from X by moving a data point $o_i \in C_c$ to a new cluster C_{c^*} with $c \neq c^*$ (i.e., $x_i^c = 1$ and $x_i^{c^*} = 0$

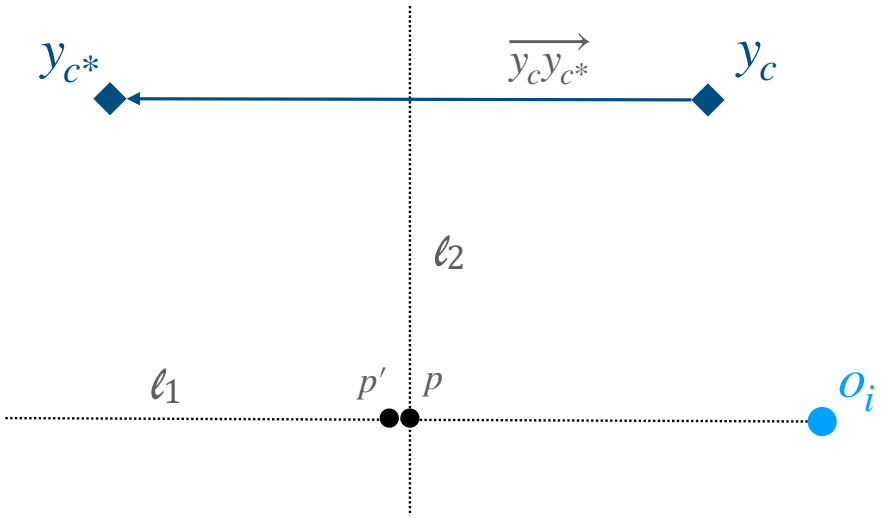


Figure 6.6 Illustration associated to moving a point o_i towards a new cluster c^*

are replaced by $x_i^c = 0$ and $x_i^{c^*} = 1$), then the variation $\Delta(i, c, c^*) = f(X') - f(X)$ is equal to the following expression:

$$\Delta(i, c, c^*) = \frac{\sum_{i=1}^n x_i^{c^*}}{\sum_{i=1}^n x_i^{c^*} + 1} \|o_i - y_{c^*}\|^2 - \frac{\sum_{i=1}^n x_i^c}{\sum_{i=1}^n x_i^c - 1} \|o_i - y_c\|^2.$$

Since we do not want to move o_i towards C_{c^*} if $(i, c^*) \in \mathcal{L}$, and o_i does not move if $c = c^*$, we consider $\delta(o_i, c, c^*)$ defined as follows:

$$\delta(i, c, c^*) = \begin{cases} \Delta(i, c, c^*) & \text{if } (i, c^*) \notin \mathcal{L} \text{ and } c \neq c^* \\ \infty & \text{if } (i, c^*) \in \mathcal{L} \text{ and } c \neq c^* \\ 0 & \text{if } c = c^* \end{cases}$$

We are now ready to explain how Algorithm 8 works. Let (o_i, o_j) be a violated constraint $\in \mathcal{CL}$ and assume that o_i and o_j currently belong both to cluster C_c. Algorithm 8 determines the index $u \in \{i, j\}$ and the cluster index $c^* \neq c$ in M_{ij} with minimum value $\delta(u, c, c^*)$, and then moves o_u towards C_{c^*}, as explained above.

Similarly, let (o_i, o_j) be a violated constraint $\in \mathcal{ML}$ and let c_i and c_j be the indices of the clusters that contain o_i and o_j, respectively. Algorithm 8 determines the cluster index $c^* \in M_{ij}$ with minimum value $\delta(i, c_i, c^*) + \delta(j, c_j, c^*)$. If $o_i \notin C_{c^*}$ then o_i is moved towards C_{c^*}. Also, if $o_j \notin C_{c^*}$, then o_j is moved towards C_{c^*}.

We now empirically demonstrate that Algorithm 7 produces high-quality clustering solutions

Algorithm 8 Moving data points of violated constraints

Input: a violated constraint $(o_i, o_j) \in \mathcal{ML} \cup \mathcal{CL}$.

1: **if** $(o_i, o_j) \in \mathcal{CL}$ **then**
2: Determine $u \in \{i, j\}$ and the cluster index $c^* \in M_{ij}$ with minimum value $\delta(u, c, c^*)$.
3: Move o_u towards C_{c^*}.
4: **else**
5: Let c_i and c_j be the indices of the clusters that contain o_i and o_j, respectively.
6: Determine the cluster index $c^* \in M_{ij}$ with minimum value $\delta(i, c_i, c^*) + \delta(j, c_j, c^*)$.
7: **If** $o_i \notin C_{c^*}$ **then** move o_i towards C_{c^*}.
8: **If** $o_j \notin C_{c^*}$ **then** move o_j towards C_{c^*}.
9: **end if**

while maintaining geometrical properties of the data as much as possible. For this purpose, we compare it to a baseline procedure that iteratively selects, at random, one violated constraint, and then performs the move defined in Algorithm 8.

The algorithms are tested on three real-world datasets, namely the `Iris` dataset mentioned in Table 6.1, the `Seed` dataset that contains 210 samples with seven features, and the `Optical Digits` dataset composed of 3823 handwritten digits images of 8×8 pixels. These datasets were selected because (i) their ground-truth partitions are sufficiently different from the solutions obtained by optimizing the corresponding unsupervised MSSC model, and (ii) clustering them using the unsupervised MSSC model is still effective in terms of the obtained ARI indices, thereby reducing the number of necessary pairwise constraints to get close to the ground-truth partitions.

To get the solution X in step 3 of Algorithm 7, we executed 100 times the algorithm k-means, which is the most popular heuristic for solving unsupervised MSSC clustering problems, and we then set X equal to the solution with minimum MSSC value.

For each dataset, we generated 20 constraint sets. All these sets are composed of pairwise constraints, randomly chosen among those that are violated by the initial solution X. For the `Iris` instance, these sets contain 15 constraints, while their number is 20 and 400 for the `Seed` and `Optical` datasets, respectively.

At the end of each iteration, we computed the ARI of the current solution X as well as the cumulative Euclidean distance traveled by the transformed data points, these moves resulting from running Algorithm 8. The results produced by Algorithm 7 and by the baseline method are reported in Figure 6.7. The line and the band refer to the mean and standard deviation with respect to the 20 runs of each method.

Our first observation is that for each set of m constraints ($m = 15, 20$ or 400), each execution

of Algorithm 7 only required m iterations to satisfy them all, which means that each move resulting from a run of Algorithm 8 allowed to satisfy the violated constraint given in input.

Next, for all datasets, we observe an upward progression of the ARI both for the proposed algorithm that exploits the dual information and for the baseline method. This behavior asserts the effectiveness of the designed distance metric learning method to leverage information from the provided pairwise constraints. More interestingly, we highlight that Algorithm 7 improves the clustering quality faster (except for `Seed`) and with less significant transformations of the original dataset when compared to the baseline algorithm that does not exploit dual information.

6.6 Filtering useful pairwise constraints

Semi-supervised clustering techniques use constraints to guide an algorithm towards better clustering solutions. However, some constraints can have a negative effect on the clustering task [2], and it is not necessarily easy to identify which constraints are useful or harmful to the clustering process by basing this identification on a solution produced by an algorithm.

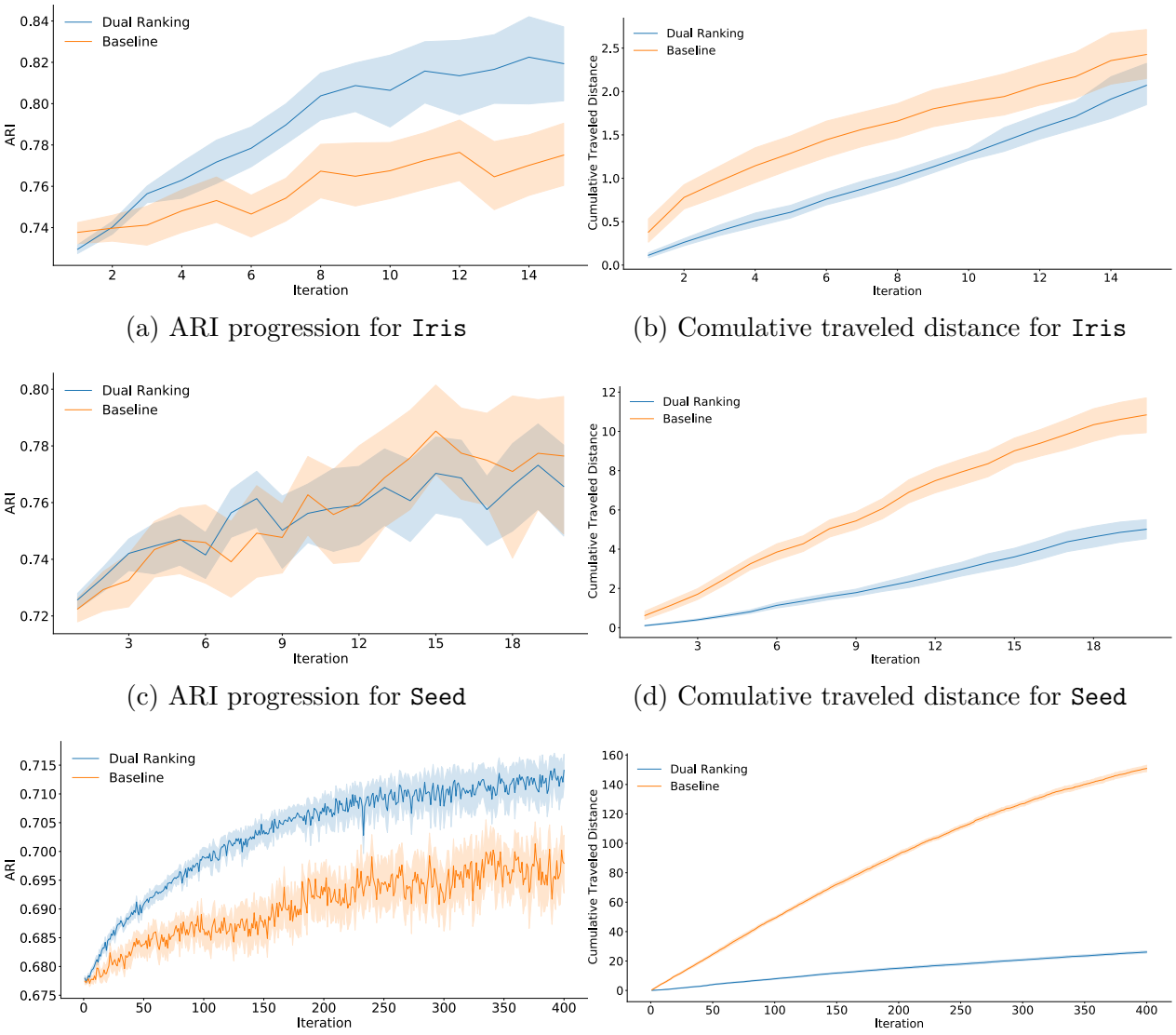
We propose to use the information associated with the dual variables to rank the pairwise constraints according to the likelihood that they contribute favorably to the semi-supervised clustering process. We then only use the best ranked constraints, this selection aiming to reduce the number of constraints to consider without compromising the quality of the solution produced. As in the previous section, the pairwise constraints $(o_i, o_j) \in \mathcal{ML} \cup \mathcal{CL}$ are ordered according to their associated impact $\mathcal{I}_{ij}$.

A particular family of methods that can greatly benefit from this ranking approach are those designed to handle massive datasets, where the time required to process the data itself and the set of constraints is crucial for their use in the practice. We demonstrate the effectiveness of our approach by integrating it into the semi-supervised deep learning framework recently proposed by Zhang et al. [1]. Their algorithm can be summarized as follows.

The algorithm first uses a deep neural network to determine an embedding from the data space to a lower-dimensional feature space Z. The algorithm then computes soft and hard allocation values q_{ic} and p_{ic} in $[0, 1]$ which, roughly speaking, can be interpreted as the probability of assigning sample o_i to cluster C_c. In the sequel, it defines a clustering loss function ℓ_C which is defined as

$$\ell_C = \sum_{i=1}^{n} \sum_{c=1}^{k} p_i^c \log \frac{p_i^c}{q_i^c},$$

as well as a loss $\ell_{\mathcal{ML}}$ for the must-link constraints, and a loss $\ell_{\mathcal{CL}}$ for the cannot-link con-



(a) ARI progression for `Iris`

(b) Comulative traveled distance for `Iris`

(c) ARI progression for `Seed`

(d) Comulative traveled distance for `Seed`

(e) ARI progression for `Optical`

(f) Comulative traveled distance for `Optical`

Figure 6.7 ARI progression and cumulative Euclidian traveled distance for Algorithm 7 and the baseline method applied to three real-world instances.

straints, where $\ell_{\mathcal{ML}}$ and $\ell_{\mathcal{CL}}$ are defined as

$$\ell_{\mathcal{ML}} = \sum_{(o_i,o_j)\in\mathcal{ML}} \log \sum_{c=1}^{k} q_i^c q_j^c \quad \text{and} \quad \ell_{\mathcal{CL}} = \sum_{(o_i,o_j)\in\mathcal{CL}} \log(1 - \sum_{c=1}^{k} q_i^c q_j^c).$$

The three loss functions ℓ_C, ℓ_{ML} and ℓ_{CL} are then used to update the parameters of the neural network and to obtain a new mapping of the data points in Z. This process is repeated at most m times, where m is a parameter, but may stop earlier if the ratio of changed cluster

assignments between two consecutive iterations is smaller than 0.001, each data point o_i being assigned to the cluster c that maximizes q_i^c. For more details, the reader is referred to [1].

Instead of defining ℓ_{ML} and ℓ_{CL} by considering all the available pairwise constraints, we propose to use only a subset. More precisely, we run the sub-gradient algorithm to determine optimal dual values $\eta_{ij}^c, \lambda_{ij}^c$ and γ_{ij}^c for the clustering problem based on the embedding of the points in Z. As in the previous section, these dual values are used to define the impact score $\mathcal{I}_{ij}$ of each constraint $(o_i, o_j) \in \mathcal{ML} \cup \mathcal{CL}$, where $\mathcal{I}_{ij}$ is defined as the sum of the values of the dual variables associated with that constraint. Let Ω be the subset of pairwise constraints having a strictly negative impact : we eliminate the $\lfloor \alpha \rfloor |\Omega|$ constraints of Ω which have the smallest impact score, where $\alpha \in [0, 1]$ is a parameter, and we denote by Ω^α the set of remaining constraints. By defining $\mathcal{ML}^\alpha = \mathcal{ML} \cap \Omega^\alpha$ and $\mathcal{CL}^\alpha = \mathcal{CL} \cap \Omega^\alpha$, we obtain the following new loss functions $\ell_{\mathcal{ML}}^\alpha$ and $\ell_{\mathcal{CL}}^\alpha$ given by :

$$\ell_{\mathcal{ML}}^\alpha = \sum_{(o_i,o_j)\in\mathcal{ML}^\alpha} \log \sum_{c=1}^{k} q_i^c q_j^c \quad \text{and} \quad \ell_{\mathcal{CL}}^\alpha = \sum_{(o_i,o_j)\in\mathcal{CL}^\alpha} \log(1 - \sum_{c=1}^{k} q_i^c q_j^c).$$

The neural network parameters are updated by using ℓ_{ML}^α and ℓ_{CL}^α instead of ℓ_{ML} and ℓ_{CL}. Algorithm 9 describes the main steps of this integration of our dual approach in the algorithm of [1].

Algorithm 9 is compared with the original algorithm of Zhang et al. [1] on the same three datasets they used for their experiment: `MNIST` which contains 60,000 handwritten digits of 28×28 pixel size which have to be grouped into $k = 10$ classes, `Fashion-MNIST` which contains 60,000 grayscale 28×28 images and $k = 10$ classes, and `Reuters-10K` which contains 10,000 document texts with 2,000 features and $k = 10$ classes. For each dataset, we generated 20 sets of pairwise constraints randomly chosen from the ground-truth partition. The number of constraints in each set is equal to 10% of the size of the dataset, which amounts to 6,000 for `MNIST` and `Fashion-MNIST`, and 1,000 for `Reuters-10K`. We ran Algorithm 9 for at most $m = 500$ iterations, with $\alpha = 0.95, 0, 9, 0.8, 0.7, 0.6$ and 0.0. Note that running our algorithm with α=0.0 is not totally equivalent to applying the original algorithm of Zhang et al. [1] since we do not consider pairwise constraints (o_i, o_j) with impact score $\mathcal{I}_{ij} = 0$.

We indicate in Table 6.2 the average ARI (mean ± standard deviation) over the 20 constraint sets for each value of α and for the three datasets. CPU times (in seconds) and the number of iterations required to reach convergence are shown in Table 6.3 . We observe that our strategy that uses the dual information for filtering the constraints improves the unfiltered version of Zhang et al. [1] for all three datasets. The best ARI is obtained by removing 70 to 80% of the

Algorithm 9 Integration of the dual approach in the deep learning framework of [1]

1: Train the neural network to obtain an embedding of the data points in a lower dimensional feature space Z. Initialize the iteration counter *iter* to zero.
2: **repeat**
3: Compute the soft and hard allocation values q_i^c and p_i^c as well as the loss ℓ_C.
4: Run the sub-gradient algorithm to produce optimal dual values $\eta_{ij}^c, \lambda_{ij}^c$ and γ_{ij}^c for the clustering problem based on the current embedding of the data points in Z.
5: For all $(o_i, o_j) \in \mathcal{CL} \cup \mathcal{ML}$ do $\mathcal{I}_{ij} \leftarrow \begin{cases} \sum_{c=1}^{k} \eta_{ij}^c & \text{if } (o_i, o_j) \in \mathcal{CL} \\ \sum_{c=1}^{k} (\lambda_{ij}^c + \lambda_{ij}'^c) & \text{if } (o_i, o_j) \in \mathcal{ML}. \end{cases}$
6: Set $\Omega \leftarrow \{(o_i, o_j) \in \mathcal{ML} \cup \mathcal{CL} \mid \mathcal{I}_{ij} < 0\}$, sort these constraints in increasing order of their impact score $\mathcal{I}_{ij}$, and remove the $\lfloor \alpha \rfloor |\Omega|$ top-ranked constraints of Ω to obtain Ω^α.
7: Set $\mathcal{ML}^\alpha \leftarrow \mathcal{ML} \cap \Omega^\alpha$, $\mathcal{CL}^\alpha \leftarrow \mathcal{CL} \cap \Omega^\alpha$, and compute the losses $\ell_{ML}^\alpha$and ℓ_{CL}^α.
8: Set *iter* $\leftarrow$ *iter*+1, update the neural network parameters by using ℓ_C, ℓ_{CL}^α and ℓ_{ML}^α, and get a new embedding of the data points in Z.
9: **until** *iter*=m or the ratio of changed cluster assignments between two consecutive iterations is below 10^{-3}, each o_i being assigned to the cluster c that maximizes q_i^c.

constraints with the lowest impact score. Note also that in addition to improving the ARI, our filtering approach does not significantly modify the CPU times and the number of iterations necessary to reach convergence. More precisely, a reduction is obtained, both in terms of time and number of iterations, for the datasets `MNIST` and `Fashion-MNIST` when using our filtering algorithm with $\alpha = 70\%$. However, such gains are not observed for the smallest dataset `Reuters-10k`, where the unfiltered algorithm converges faster than Algorithm 9.

Table 6.2 Mean ARI and standard deviations for Algorithm 9 and for the unfiltered original algorithm of Zhang et al. [1].

	Datasets		
α	MNIST	Fashion-MNIST	Reuters-10K
0.95	0.9370 ± 0.0136	0.4175 ± 0.0244	0.5992 ± 0.0140
0.90	0.9389 ± 0.0083	0.4115 ± 0.0193	0.6006 ± 0.0157
0.80	0.9338 ± 0.0132	$\mathbf{0.4237 \pm 0.0250}$	$\mathbf{0.6009 \pm 0.0153}$
0.70	$\mathbf{0.9427 \pm 0.0077}$	0.4132 ± 0.0127	0.5949 ± 0.0154
0.60	0.9343 ± 0.0244	0.4164 ± 0.0101	0.5843 ± 0.0153
0.00	0.9343 ± 0.0161	0.4012 ± 0.0167	0.5830 ± 0.0120
Unfiltered	0.9314 ± 0.0040	0.3916 ± 0.0163	0.5995 ± 0.0108

Table 6.3 CPU times (in seconds) and number of iterations to reach convergence for Algorithm 9 and for the unfiltered original algorithm of Zhang et al. [1].

	MNIST		Fashion-MNIST		Reuters-10K	
α	CPU times	iterations	CPU times	iterations	CPU times	iterations
0.95	171.7 ± 22.6	126.5 ± 16.4	646.4 ± 19.5	499.3 ± 2.6	33.4 ± 5.0	67.1 ± 18.2
0.90	152.5 ± 17.9	127.7 ± 15.4	658.7 ± 05.1	497.1 ± 8.8	34.2 ± 3.8	61.8 ± 14.8
0.80	161.0 ± 33.1	134.9 ± 29.0	673.4 ± 23.9	465.3 ± 37.1	30.0 ± 1.9	50.1 ± 08.1
0.70	$\mathbf{142.8 \pm 23.8}$	$\mathbf{120.8 \pm 20.5}$	$\mathbf{587.5 \pm 48.7}$	$\mathbf{463.2 \pm 53.5}$	29.3 ± 2.0	44.8 ± 06.3
0.60	164.3 ± 33.4	128.2 ± 26.2	591.0 ± 18.1	497.9 ± 09.1	28.8 ± 1.4	34.4 ± 06.2
0.00	183.6 ± 22.9	137.4 ± 20.2	623.8 ± 40.3	488.3 ± 26.5	26.6 ± 1.2	28.6 ± 06.3
Unfiltered	187.2 ± 27.5	149.7 ± 14.8	775.3 ± 27.4	500.0 ± 00.0	$\mathbf{13.2 \pm 0.4}$	$\mathbf{10.7 \pm 01.9}$

6.7 Conclusion

Our goal was to explore how dual information can improve distance metric learning algorithms for clustering problems with must-link and cannot-link constraints. We have shown that the optimal dual values associated to these constraints in a clustering problem are an effective tool to estimate the impact of violating a constraint. These dual values are obtained by running a sub-gradient algorithm on a Lagrangian relaxation of the original clustering problem, where pairwise constraints are replaced by penalty terms in the objective function. We have illustrated this benefit for the following three tasks.

1. **Identification of an appropriate dissimilarity measure**. We have defined a fitness score based on optimal dual values for recommending to experts a dissimilarity measure that best agrees with their beliefs and expectations.

2. **Preserving geometrical properties of the dataset**. Distance metric learning algorithms aim to move closer pairs of data points involved in must-link constraints, and to move pairs of points involved in cannot-link constraints away from each other. These transformations are often applied without worrying too much about their magnitude. We have shown how the use of dual information makes it possible to determine transformations with little impact on the original space. Our methodology therefore offers a tool to data analysts to allow them to integrate their knowledge of the domain while controlling the way in which the data is modified.

3. **Filtering useful pairwise constraints**. While it is known that some pairwise constraints can have a negative effect on the clustering task, the identification of these harmful constraints is not an easy task. We have defined an impact score, based on

optimal dual values, which helps to filter the constraints that appear to be the most useful. We have demonstrated the benefit of this filter by integrating it into the recently proposed deep learning framework of Zhang et al. [1].

CHAPTER 7 GENERAL DISCUSSION

In this thesis, we studied and introduced novel analytical techniques and optimization methods to enhance constraint-based semi-supervised clustering. Our main contributions are summarized in Section 7.1. Finally, Section 7.2 discusses the limitations of the proposed methods and suggest potential research avenues to be exploited.

7.1 Summary of Works

Motivated by the lack of studies exploring different clustering models under a semi-supervised paradigm, Chapter 4 introduces a pairwise-constrained k-medoids model. To optimize it, we proposed a two-phased Variable Neighborhood Search heuristic, and demonstrated, through a series of experiments, that it is efficient to obtain optimal or near-optimal clustering solutions for the proposed problem. In addition, the work corroborated the k-medoids as a competitive clustering model, able to retrieve high-quality partitions.

In Chapter 5, we addressed the crucial issue of clustering under the presence of erroneous pairwise constraints. This manly occurs because side information, which are generally provided by domains experts, are prone to human misjudgments. Nonetheless, application users are often left with the sole option of using all informed constraints. To address this issue, we proposed a Lagrangian-based score for assessing the quality of pairwise semi-supervision within clustering. This quantitative assessment is computed from dual variables expressed in a Lagrangian relaxation, which measures the estimated impact of pairwise constraints to the clustering criterion under optimization. As a result, we provided a useful tool to help domain experts to identify which pairwise constraints should be reviewed. The effectiveness of our method is examined by means of several experiments on synthetic and real data.

Finally, Chapter 6 describes procedures for enhancing distance metric learning for clustering. The work leverages the dual optimal information obtained after solving the Lagrangian relaxation problem defined in Chapter 5. Considering that, the study focused on three main shortcomings that distance metric learning methods for clustering may face:

- *Adopting an inappropriate notion of dissimilarity for clustering the data.* Distance metric learning algorithms may handicap themselves by learning data transformations upon inappropriate notions of dissimilarities. However, no mechanism is available to identify if a given metric is suitable to retrieve good data description. To circumvent this difficulty, we proposed a *metric fitting score* for recommending to experts a dissimilarity

measure that best agrees with their beliefs and expectations. This score uses the dual optimal values for measuring the amount of pairwise constraints that do not agree with the clustering criterion under optimization.

- *Drastically modifying the geometric properties of the data.* This situation occurs when distance metric learning algorithms apply transformations to data features regardless of how impactful they are to the geometric properties of the original space. We addressed this issue by establishing a dual-based order in which pairwise constraints should be processed by distance metric learning algorithms. We demonstrated that sorting constraints according to their estimated impact on the clustering objective function allows distance learning algorithms to apply transformations that yield minimal modifications to the original data. The reason behind this idea is that the less impactful a constraint is, the lower will be the gain in the objective function if the constraint is relaxed. Thus, the data transformation required to satisfy such a constraint is presumably small. In addition, we proposed a baseline metric learning algorithm that incorporates pairwise supervision, and profits from this dual-ordering strategy to maintain a reliable representation of the data.

- *Learning transformations from less helpful constraints.* Motivated by the known fact that some pairwise constraints may harm the clustering task, we proposed to explore dual optimal information for selecting the potentially most helpful constraints for distance metric learning. Accordingly, dual optimal values are used to rank the pairwise constraints according to their likelihood of positively contributing to the semi-supervised clustering process itself. To test our methodology, we integrated the dual-ranking strategy with a recently proposed deep learning framework for semi-supervised learning, and demonstrated that the clustering accuracy is improved from such approach.

7.2 Limitations and Future Research

We have demonstrated throughout experiments that our approach provides valuable information regarding the usefulness of pairwise clustering constraints. Ultimately, the quality of this information depends on how much time the proposed methods are allowed to run in order to refine that information. Besides, the quality of clustering solutions obtained by means of our proposed analytical tools and algorithms are arguably connected to the ability of the chosen clustering model to recover the underlying structure of the data. Therefore, our results should yield higher clustering accuracy and be more reliable if appropriated clustering

models are adopted. Finally, it is assumed that constraint sets are always feasible, which might be false especially if many constraints are provided by several heterogeneous experts.

As a matter of fact, among future research avenues, we intend to study techniques that can handle infeasible sets of pairwise constraints. Therefore, a method for identifying which constraints could be discarded without compromising the quality of supervision is of great interest for application users. That would permit a wider acquisition of side information from different sources or methods. From the results obtained with the analytical tools proposed in Chapter 5 and 6, we believe that our methodology has the potential to address this issue as well. Accordingly, we could investigate the use of Lagrangian-based scores for identifying the subset of constraints that could be disregarded in order to produce a useful and feasible constraint set. For instance, this can be done by analyzing the chromatic number of the *constraint graph*. In this graph, vertices correspond to the data points and the edges connect points associated with cannot-link constraints. We note that the chromatic number of this graph corresponds to the minimum number of clusters required to satisfy all pairwise cannot-link constraints. To achieve clustering feasibility for a given number of clusters, dual values can inform which are the best constraints to be pruned out of the constraint graph in order to reduce its chromatic number. Thus, we aim to identify pairwise constraints that, if eliminated, lead to an informative and *feasible* constraint set.

In addition, the introduced techniques could be investigated to strengthen the constraint acquisition of active learning algorithms for pairwise constrained clustering. Generally speaking, those methods work by identifying pairs of data points that are potentially informative if their relationship is known, and query an oracle to know if a must-link or cannot-link constraint can be included for them. Thereof, we believe that the oracle criterion for recommending a constraint could also take into consideration the impact estimation measured by dual variable values.

Finally, it would be interesting to extend our studies for incorporating other types of supervision, such as pointwise information and triplet constraints, or even cluster-level constraints such as maximum cluster cardinality, minimum separation between clusters (split) and maximum diameter within clusters. Indeed, generally speaking, our Lagrangian methodology for exploring dual information can be extended to any clustering problem whose objective function and constraints (supervision) can be expressed by means of mathematical programming formulations.

CHAPTER 8 CONCLUSION

The semi-supervised learning paradigm proposes mechanisms to perform data analysis tasks in presence of both labeled and unlabeled data. It has attracted much interest in recent years as the amount of collected data rapidly increases, while ground-truth information remains scarce and expensive to acquire. Thus, semi-supervised learning is indeed an important branch of machine learning that seeks to learn from both labeled and unlabeled data.

Clustering, a fundamental unsupervised data mining task, can benefit from semi-supervision to retrieve group descriptions that are more in accordance with the background knowledge of domain experts. In this thesis, we proposed optimization methods to enhance the clustering process with pairwise constraints, the most common manner of expressing side-information.

In our first contribution, we proposed a k-medoids model for semi-supervised clustering with pairwise constraints. Although the k-medoids is a well-known clustering model, it had not yet been studied in a constraint-based environment. To optimize the proposed problem, we devised a Variable Neighborhood Search algorithm and demonstrated it as an efficient procedure to find optima or near-optima solutions.

In the second contribution, we studied and addressed the issue of incorporating erroneous constraints, which negatively affects the quality of semi-supervised clustering. To this end, we proposed a Lagrangian-based procedure for assessing the quality of pairwise supervision. Accordingly, we exploit dual information associated with the introduced Lagrangian relaxation, which provides an estimated impact of pairwise constraints in the clustering model. Finally, using this valuable information, we developed a tool for quantifying the likely accuracy of pairwise constraints. This offers a method for domain experts to identify which pairwise constraints are degrading the clustering solution, and therefore, should be removed or revised. Such a technique is of great interest since information concerning the usefulness of pairwise constraints help application users to select the most helpful pairwise constraints, and thus, retrieving better data descriptions.

In our last research work, we employed the methodology that exploits dual information to distance metric learning algorithms for clustering. Specifically, we proposed effective tools and algorithms to help domain experts: (i) choosing the notion of dissimilarity that agrees the most with their beliefs and expectation; (ii) preventing drastic modifications of the geometrical properties of the data; and (iii) preventing distance metric learning algorithms from using the most impactful constraints when learning a new dissimilarity. Therefore, the developed techniques can enhance the quality of clustering solutions produced by constraint-

based distance metric learning algorithms.

REFERENCES

[1] H. Zhang, T. Zhan, S. Basu, and I. Davidson, "A framework for deep constrained clustering," *Data Mining and Knowledge Discovery*, vol. 35, no. 2, pp. 593–620, 2021.

[2] I. Davidson, K. L. Wagstaff, and S. Basu, "Measuring constraint-set utility for partitional clustering algorithms," in *Knowledge Discovery in Databases: PKDD 2006*, J. Fürnkranz, T. Scheffer, and M. Spiliopoulou, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 115–126.

[3] D. P. Kingma, D. J. Rezende, S. Mohamed, and M. Welling, "Semi-supervised learning with deep generative models," 2014.

[4] D. Ienco and R. G. Pensa, "Deep triplet-driven semi-supervised embedding clustering," in *Discovery Science*, P. Kralj Novak, T. Šmuc, and S. Džeroski, Eds. Cham: Springer International Publishing, 2019, pp. 220–234.

[5] X. Zhu, A. B. Goldberg, R. Brachman, and T. Dietterich, *Introduction to Semi-Supervised Learning*. Morgan and Claypool Publishers, 2009.

[6] H. Scudder, "Probability of error of some adaptive pattern-recognition machines," *IEEE Transactions on Information Theory*, vol. 11, no. 3, pp. 363–371, 1965.

[7] S. Fralick, "Learning to recognize patterns without a teacher," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 57–64, 1967.

[8] A. Agrawala, "Learning with a probabilistic teacher," *IEEE Transactions on Information Theory*, vol. 16, no. 4, pp. 373–379, 1970.

[9] A. Cholaquidis, R. Fraiman, and M. Sued, "On semi-supervised learning," 2019.

[10] X. Zhou and M. Belkin, "Chapter 22 - semi-supervised learning," in *Academic Press Library in Signal Processing: Volume 1*, ser. Academic Press Library in Signal Processing, P. S. Diniz, J. A. Suykens, R. Chellappa, and S. Theodoridis, Eds. Elsevier, 2014, vol. 1, pp. 1239–1269. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B978012396502800022X

[11] J. E. van Engelen and H. H. Hoos, "A survey on semi-supervised learning," *Machine Learning*, vol. 109, no. 2, pp. 373–440, 2020.

[12] C. C. Aggarwal and C. K. Reddy, *Data Clustering: Algorithms and Applications*, 1st ed. Chapman & Hall/CRC, 2013.

[13] P. Hansen and B. Jaumard, "Cluster analysis and mathematical programming," *Mathematical Programming*, vol. 79, no. 1-3, pp. 191–215, 1997.

[14] I. T. Christou, "Coordination of cluster ensembles via exact methods," *IEEE Trans Pattern Anal Mach Intell*, vol. 33, no. 2, pp. 279–93, 2011.

[15] C. C. Aggarwal, *Data Mining*. Springer, 2015.

[16] S. Lloyd, "Least squares quantization in pcm," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[17] H. Späth, *Cluster analysis algorithms for data reduction and classification of objects*, ser. Computers and their applications. E. Horwood, 1980. [Online]. Available: https://books.google.ca/books?id=4ofgAAAAMAAJ

[18] M. Peikari, S. Salama, S. Nofech-Mozes, and A. L. Martel, "A cluster-then-label semi-supervised learning approach for pathology image classification," *Scientific Reports*, vol. 8, no. 1, p. 7193, 2018.

[19] S. Basu, A. Banerjee, and R. J. Mooney, "Semi-supervised clustering by seeding," in *Proceedings of the Nineteenth International Conference on Machine Learning*, ser. ICML '02. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2002, Conference Proceedings, pp. 27–34. [Online]. Available: http://dl.acm.org/citation.cfm?id=645531.656012

[20] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall, "Learning a mahalanobis metric from equivalence constraints," *Journal of Machine Learning Research*, vol. 6, no. 32, pp. 937–965, 2005. [Online]. Available: http://jmlr.org/papers/v6/bar-hillel05a.html

[21] D. Zhang, S. Chen, and Z.-H. Zhou, "Constraint score: A new filter method for feature selection with pairwise constraints," *Pattern Recognition*, vol. 41, no. 5, pp. 1440–1451, 2008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320307004505

[22] A. Jain, R. Jin, and R. Chitta, "Semi-supervised clustering," in *Handbook of Cluster Analysis*. Chapman & Hall/CRC, 2014, ch. 20, pp. 443–468.

[23] K. Wagstaff, C. Cardie, S. Rogers, and S. Schrödl, "Constrained K-Means Clustering with Background Knowledge," in *Proceedings of the Eighteenth International Conference on Machine Learning*, ser. ICML '01, vol. 1. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, pp. 577–584.

[24] T. Rutayisire, Y. Yang, C. Lin, and J. Zhang, "A modified cop-kmeans algorithm based on sequenced cannot-link set," in *Rough Sets and Knowledge Technology*, J. Yao, S. Ramanna, G. Wang, and Z. Suraj, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 217–225.

[25] Y. Yang, T. Rutayisire, C. Lin, T. Li, and F. Teng, "An improved cop-kmeans clustering for solving constraint violation based on mapreduce framework," *Fundamenta Informaticae*, vol. 126, no. 4, pp. 301–318, 2013.

[26] K.-C. Duong, C. Vrain *et al.*, "Constrained clustering by constraint programming," *Artificial Intelligence*, 2015.

[27] L. Kaufman and P. J. Rousseeuw, *Finding groups in data: an introduction to cluster analysis*. John Wiley & Sons, 2009, vol. 344.

[28] N. Mladenović, J. Brimberg, P. Hansen, and J. A. Moreno-Pérez, "The p-median problem: A survey of metaheuristic approaches," *EJOR*, vol. 179, no. 3, pp. 927 – 939, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0377221706000750

[29] I. Davidson and S. S. Ravi, "Identifying and generating easy sets of constraints for clustering," in *Proceedings of the 21st national conference on Artificial intelligence - Volume 1*. 1597593: AAAI Press, 2006, Conference Proceedings, pp. 336–341.

[30] M. E. Ares, J. Parapar, and Álvaro Barreiro, "An experimental study of constrained clustering effectiveness in presence of erroneous constraints," *Information Processing & Management*, vol. 48, no. 3, pp. 537 – 551, 2012, soft Approaches to IA on the Web. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0306457311000914

[31] J. Anil, J. Rong, and C. Radha, *Semi-Supervised Clustering*. CRC Press, 2015, book section Semi-Supervised Clustering.

[32] I. Davidson and S. S. Ravi, "Clustering with constraints: Feasibility issues and the k-means algorithm." in *Proceedings of the 2005 SIAM International Conference on Data*

Mining, SDM 2005, vol. 5, SIAM. Society for Industrial and Applied Mathematics, apr 2005, pp. 201–211.

[33] I. Davidson, S. S. Ravi, and L. Shamis, "A SAT-based framework for efficient constrained clustering," in *Proceedings of the 2010 SIAM International Conference on Data Mining*. Society for Industrial and Applied Mathematics, apr 2010.

[34] Y. S. Wu, S. Bagchi, N. Singh, and R. Wita, "Spam detection in voice-over-ip calls through semi-supervised clustering," in *2009 IEEE/IFIP International Conference on Dependable Systems Networks*, June 2009, pp. 307–316.

[35] D. Cohn, R. Caruana, and A. McCallum, "Semi-supervised clustering with user feedback," *Constrained Clustering: Advances in Algorithms, Theory, and Applications*, vol. 4, no. 1, pp. 17–32, 2003.

[36] Y. Hu, E. E. Milios, and J. Blustein, "Interactive feature selection for document clustering," in *Proceedings of the 2011 ACM Symposium on Applied Computing*, ser. SAC '11. New York, NY, USA: ACM, 2011, pp. 1143–1150.

[37] N. Kumar, K. Kummamuru, and D. Paranjpe, "Semi-supervised clustering with metric learning using relative comparisons," in *Fifth IEEE International Conference on Data Mining (ICDM'05)*, 2005, pp. 4 pp.–.

[38] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," 2016.

[39] S. Kullback and R. A. Leibler, "On information and sufficiency," *Ann. Math. Statist.*, vol. 22, no. 1, pp. 79–86, 03 1951.

[40] Y.-C. Hsu and Z. Kira, "Neural network-based clustering using pairwise constraints," 2015.

[41] S. Fogel, H. Averbuch-Elor, D. Cohen-Or, and J. Goldberger, "Clustering-Driven Deep Embedding With Pairwise Constraints," *IEEE Computer Graphics and Applications*, vol. 39, no. 4, pp. 16–27, 2019.

[42] K. L. Wagstaff, "Value, cost, and sharing: Open issues in constrained clustering," in *Knowledge Discovery in Inductive Databases: 5th International Workshop, KDID 2006 Berlin, Germany, September 18, 2006 Revised Selected and Invited Papers*, S. Džeroski and J. Struyf, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 1–10. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-75549-4_1

[43] S. Basu, A. Banerjee, and R. J. Mooney, "Active semi-supervision for pairwise constrained clustering," in *Proceedings of the 2004 SIAM International Conference on Data Mining*, vol. 4, SIAM. Society for Industrial and Applied Mathematics, 2004, pp. 333–344.

[44] P. K. Mallapragada, R. Jin, and A. K. Jain, "Active query selection for semi-supervised clustering," in *2008 19th International Conference on Pattern Recognition*. IEEE, Dec 2008, pp. 1–4.

[45] S. Xiong, J. Azimi, and X. Z. Fern, "Active learning of constraints for semi-supervised clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 1, pp. 43–54, Jan 2014.

[46] C. Xiong, D. M. Johnson, and J. J. Corso, "Active clustering with model-based uncertainty reduction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 1, pp. 5–17, Jan 2017.

[47] V.-V. Vu, N. Labroche, and B. Bouchon-Meunier, "Improving constrained clustering with active query selection," *Pattern Recognition*, vol. 45, no. 4, pp. 1749–1758, 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320311004407

[48] F. Wang, S. Chen, C. Zhang, and T. Li, "Semi-supervised metric learning by maximizing constraint margin," in *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, ser. CIKM '08. New York, NY, USA: ACM, 2008, pp. 1457–1458.

[49] M. Okabe and S. Yamada, "Clustering by learning constraints priorities," in *2012 IEEE 12th International Conference on Data Mining(ICDM)*, vol. 00. IEEE, 12 2012, Conference Proceedings, pp. 1050–1055. [Online]. Available: doi.ieeecomputersociety.org/10.1109/ICDM.2012.150

[50] I. Davidson, "Two approaches to understanding when constraints help clustering," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '12. New York, NY, USA: ACM, 2012, pp. 1312–1320.

[51] M. Bilenko, S. Basu, and R. J. Mooney, "Integrating constraints and metric learning in semi-supervised clustering," *Proceedings, Twenty-First International Conference on Machine Learning, ICML 2004*, pp. 81–88, 2004.

[52] S. Zhang, H. S. Wong, and D. Xie, "Semi-supervised clustering with pairwise and size constraints," in *2014 International Joint Conference on Neural Networks (IJCNN)*, July 2014, pp. 2450–2457.

[53] D. Vallejo-Huanga, P. Morillo, and C. Ferri, "Semi-supervised clustering algorithms for grouping scientific articles," *Procedia Computer Science*, vol. 108, pp. 325 – 334, 2017, international Conference on Computational Science, ICCS 2017, 12-14 June 2017, Zurich, Switzerland. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050917307998

[54] M. N. Haouas, D. Aloise, and G. Pesant, "An exact cp approach for the cardinality-constrained euclidean minimum sum-of-squares clustering problem," in *Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, E. Hebrard and N. Musliu, Eds. Cham: Springer International Publishing, 2020, pp. 256–272.

[55] C. Ruiz, M. Spiliopoulou, and E. Menasalvas, "Density-based semi-supervised clustering," *Data Mining and Knowledge Discovery*, vol. 21, no. 3, pp. 345–370, Nov. 2010.

[56] Y. Yang, Z. Li, W. Wang, and D. Tao, "An adaptive semi-supervised clustering approach via multiple density-based information," *Neurocomputing*, vol. 257, pp. 193 – 205, 2017, machine Learning and Signal Processing for Big Multimedia Analysis. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231217301418

[57] L. Lelis and J. Sander, "Semi-supervised density-based clustering," in *2009 Ninth IEEE International Conference on Data Mining*, Dec 2009, pp. 842–847.

[58] V. V. Vu and H. Q. Do, "Density-based clustering with side information and active learning," in *2017 9th International Conference on Knowledge and Systems Engineering (KSE)*, Oct 2017, pp. 166–171.

[59] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, ser. KDD'96. Portland, Oregon: AAAI Press, 1996, pp. 226–231. [Online]. Available: http://dl.acm.org/citation.cfm?id=3001460.3001507

[60] S. Basu, M. Bilenko, A. Banerjee, and R. J. Mooney, "Probabilistic semi-supervised clustering with constraints," in *Semi-Supervised Learning*, O. Chapelle, B. Sch"olkopf,

and A. Zien, Eds. Cambridge, MA: MIT Press, 2006, pp. 71–98. [Online]. Available: http://www.cs.utexas.edu/users/ai-lab/?basu:bkchapter06

[61] N. Grira, M. Crucianu, and N. Boujemaa, "Active semi-supervised fuzzy clustering," *Pattern Recogn.*, vol. 41, no. 5, pp. 1834–1844, May 2008.

[62] A. Arshad, S. Riaz, L. Jiao, and A. Murthy, "A semi-supervised deep fuzzy c-mean clustering for two classes classification," in *2017 IEEE 3rd Information Technology and Mechatronics Engineering Conference (ITOEC)*, Oct 2017, pp. 365–370.

[63] C. Li, Y. Tan, D. Wang, and P. Ma, "Research on 3d face recognition method in cloud environment based on semi supervised clustering algorithm," *Multimedia Tools and Applications*, vol. 76, no. 16, pp. 17 055–17 073, Aug 2017.

[64] H. Lee, J. Yoo, and S. Choi, "Semi-supervised nonnegative matrix factorization," *IEEE Signal Processing Letters*, vol. 17, no. 1, pp. 4–7, Jan 2010.

[65] W. Wu, Y. Jia, S. Kwong, and J. Hou, "Pairwise constraint propagation-induced symmetric nonnegative matrix factorization," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–14, 2018.

[66] L. Jiao, F. Shang, F. Wang, and Y. Liu, "Fast semi-supervised clustering with enhanced spectral embedding," *Pattern Recognition*, vol. 45, no. 12, pp. 4358 – 4369, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0031320312002373

[67] N. Voiron, A. Benoit, A. Filip, P. Lambert, and B. Ionescu, "Semi-supervised spectral clustering with automatic propagation of pairwise constraints," in *2015 13th International Workshop on Content-Based Multimedia Indexing (CBMI)*, June 2015, pp. 1–6.

[68] L. Li, S. Wang, S. Xu, and Y. Yang, "Constrained spectral clustering using nyström method," *Procedia Computer Science*, vol. 129, pp. 9 – 15, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050918302552

[69] Y. Jia, S. Kwong, and J. Hou, "Semi-supervised spectral clustering with structured sparsity regularization," *IEEE Signal Processing Letters*, vol. 25, no. 3, pp. 403–407, March 2018.

[70] Z. Yu, P. Luo, J. You, H. S. Wong, H. Leung, S. Wu, J. Zhang, and G. Han, "Incremental semi-supervised clustering ensemble for high dimensional data clustering,"

IEEE Transactions on Knowledge and Data Engineering, vol. 28, no. 3, pp. 701–714, March 2016.

[71] Z. Yu, P. Luo, J. Liu, J. J. You, H. S. Wong, G. Han, and J. Zhang, "Semi-supervised ensemble clustering based on selected constraint projection," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–1, 2018.

[72] A. M. Iqbal, A. Moh'd, and Z. A. Khan, "Semi-supervised clustering ensemble by voting," *CoRR*, vol. abs/1208.4138, 2012.

[73] P. C. Mahalanobis, "On the generalized distance in statistics," *Proceedings of the National Institute of Sciences (Calcutta)*, vol. 2, pp. 49–55, 1936.

[74] E. P. Xing, M. I. Jordan, S. J. Russell, and A. Y. Ng, "Distance Metric Learning with Application to Clustering with Side-Information," *Advances in Neural Information Processing Systems 15*, pp. 521–528, 2003. [Online]. Available: http://papers.nips.cc/paper/2164-distance-metric-learning-with-application-to-clustering-with-side-information.pdf

[75] W. Lei, R. Jin, S. C. Hoit, J. Zhu, and N. Yu, "Learning bregman distance functions and its application for semi-supervised clustering," in *Advances in Neural Information Processing Systems 22 - Proceedings of the 2009 Conference*, 2009, pp. 2089–2097.

[76] L. Wu, S. C. Hoi, R. Jin, J. Zhu, and N. Yu, "Learning bregman distance functions for semi-supervised clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 3, pp. 478–491, March 2012.

[77] M. S. Baghshah and S. B. Shouraki, "Semi-supervised metric learning using pairwise constraints," in *IJCAI International Joint Conference on Artificial Intelligence*, 2009, pp. 1217–1222.

[78] W. Kalintha, S. Ono, M. Numao, and K. I. Fukui, "Kernelized evolutionary distance metric learning for semi-supervised clustering," in *31st AAAI Conference on Artificial Intelligence, AAAI 2017*, vol. 3, no. 2, 2017, pp. 4945–4946.

[79] B. Nguyen and B. D. Baets, "Kernel-Based Distance Metric Learning for Supervised k -Means Clustering," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 10, pp. 3084–3095, 2019.

[80] S. Basu, M. Bilenko, and R. J. Mooney, "A probabilistic framework for semi-supervised clustering," in *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2004)*, Seattle, WA, August 2004, pp. 59–68. [Online]. Available: http://www.cs.utexas.edu/users/ai-lab/?basu:kdd04

[81] B. M. Nogueira, Y. K. Benevides Tomas, and R. M. Marcacini, "Integrating distance metric learning and cluster-level constraints in semi-supervised clustering," in *2017 International Joint Conference on Neural Networks (IJCNN)*, 2017, pp. 4118–4125.

[82] D. Aloise, A. Deshpande, P. Hansen, and P. Popat, "Np-hardness of euclidean sum-of-squares clustering," *Machine Learning*, vol. 75, no. 2, pp. 245–248, 2009.

[83] P. Hansen and N. Mladenović, "Variable neighborhood search: Principles and applications," *EJOR*, vol. 130, no. 3, pp. 449–467, 2001.

[84] D. Aloise, G. Caporossi, P. Hansen, L. Liberti, S. Perron, and M. Ruiz, "Modularity maximization in networks by variable neighborhood search." *Graph Partitioning and Graph Clustering*, vol. 588, pp. 113–125, 2012.

[85] N. Belacel, P. Hansen, and N. Mladenovic, "Fuzzy j-means: a new heuristic for fuzzy clustering," *Pattern Recognition*, vol. 35, no. 10, pp. 2193–2200, 2002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320301001935

[86] P. Hansen and N. Mladenović, "J-means: a new local search heuristic for minimum sum of squares clustering," *Pattern Recognition*, vol. 34, no. 2, pp. 405–413, 2001. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320399002162

[87] P. Hansen, M. Ruiz, and D. Aloise, "A vns heuristic for escaping local extrema entrapment in normalized cut clustering," *Pattern Recognition*, vol. 45, no. 12, pp. 4337–4345, 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320312002099

[88] Éverton Santi, D. Aloise, and S. J. Blanchard, "A model for clustering data from heterogeneous dissimilarities," *European Journal of Operational Research*, vol. 253, no. 3, pp. 659–672, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377221716301618

[89] L. R. Costa, D. Aloise, and N. Mladenović, "Less is more: basic variable neighborhood search heuristic for balanced minimum sum-of-squares clustering," *Information Sciences*, vol. 415, pp. 247–253, 2017.

[90] M. L. Fisher, "The lagrangian relaxation method for solving integer programming problems," *Management Science*, vol. 27, no. 1, pp. 1–18, 1981.

[91] M. Held and R. M. Karp, "The traveling-salesman problem and minimum spanning trees," *Operations Research*, vol. 18, no. 6, pp. 1138–1162, 1970.

[92] ——, "The traveling-salesman problem and minimum spanning trees: Part ii," *Mathematical Programming*, vol. 1, no. 1, pp. 6–25, 1971.

[93] D. Bertsimas and J. N. Tsitsiklis, *Introduction to Linear Optimization*, 1st ed. Athena Scientific, 1997, vol. 6.

[94] A. H. Land and A. G. Doig, "An automatic method of solving discrete programming problems," *Econometrica*, vol. 28, no. 3, pp. 497–520, 1960. [Online]. Available: http://www.jstor.org/stable/1910129

[95] H. Marchand, A. Martin, R. Weismantel, and L. Wolsey, "Cutting planes in integer and mixed integer programming," *Discrete Applied Mathematics*, vol. 123, no. 1, pp. 397–446, 2002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0166218X01003481

[96] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[97] N. Z. Shor, *Minimization Methods for Non-Differentiable Functions*. Springer-Verlag, 1985, vol. 3, no. September 2013. [Online]. Available: http://eprints.utas.edu.au/4774/{%}5Cnhttp://link.springer.com/10.1007/978-3-642-82118-9

[98] S. Boyd, L. Xiao, and A. Mutapcic, "Subgradient methods," *lecture notes of EE392o, Stanford University, Autumn Quarter*, vol. 2004, pp. 2004–2005, 2003.

[99] R. Randel, D. Aloise, N. Mladenović, and P. Hansen, "On the k-medoids model for semi-supervised clustering," in *Variable Neighborhood Search*, A. Sifaleras, S. Salhi, and J. Brimberg, Eds. Cham: Springer International Publishing, 2019, pp. 13–27.

[100] M. Delattre and P. Hansen, "Bicriterion cluster analysis," *IEEE TPAMI*, vol. 4, no. 4, pp. 277–291, July 1980.

[101] S. Basu, I. Davidson, and K. Wagstaff, *Constrained Clustering: Advances in Algorithms, Theory, and Applications*, 1st ed. Chapman & Hall/CRC, 2008.

[102] R. Bekkerman and M. Sahami, "Semi-supervised clustering using combinatorial MRFs," in *ICML*, 2006.

[103] B. Yan and C. Domeniconi, "An adaptive kernel method for semi-supervised clustering," in *ECML*. Springer, 2006, pp. 521–532.

[104] M. H. C. Law, A. Topchy, and A. K. Jain, "Model-based Clustering with Probabilistic Constraints," in *SIAM-SDM*, 2005, pp. 641–645.

[105] Y. Xia, "A global optimization method for semi-supervised clustering," *Data Mining and Knowledge Discovery*, vol. 18, no. 2, pp. 214–256, 2009.

[106] H. Tuy, "Concave programming under linear constraints," *Soviet Math*, vol. 5, pp. 1437–1440, 1964.

[107] B. Kulis, S. Basu, I. Dhillon, and R. Mooney, "Semi-supervised graph clustering: a kernel approach," *Machine learning*, vol. 74, no. 1, pp. 1–22, 2009.

[108] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Comp.*, vol. 10, no. 5, pp. 1299–1319, 1998.

[109] N. Christofides, *Graph Theory: An Algorithmic Approach (Computer Science and Applied Mathematics)*. Orlando, FL, USA: Academic Press, Inc., 1975.

[110] D. Steinley, *Handbook of cluster analysis*, ser. Chapman & Hall/CRC Handbooks of Modern Statistical Methods. Taylor & Francis, 2015, ch. K-medoids and Other Criteria for Crisp Clustering. [Online]. Available: https://books.google.ca/books?id=OJ-JtgAACAAJ

[111] L. Kaufman and P. J. Rousseeuw, "Partitioning around medoids (program PAM)," *Finding groups in data: An introduction to cluster analysis*, pp. 68–125, 1990.

[112] M. B. Teitz and P. Bart, "Heuristic methods for estimating the generalized vertex median of a weighted graph," *Oper. Res.*, vol. 16, no. 5, pp. 955–961, Oct. 1968.

[113] R. Whitaker, "A fast algorithm for the greedy interchange for large-scale clustering and median location problems," *INFOR*, vol. 21, no. 2, pp. 95–108, 1983.

[114] P. Hansen and N. Mladenović, "Variable neighborhood search for the p-median," *Location Science*, vol. 5, no. 4, pp. 207–226, 1997. [Online]. Available: https://www.scopus.com/inward/record.uri$?$eid=2-s2.0-0031390388&partnerID=40&md5=6dd33143c56608d7604fe2b12bc3de72

[115] M. G. C. Resende and R. F. Werneck, "A fast swap-based local search procedure for location problems," *Annals of Operations Research*, vol. 150, no. 1, pp. 205–230, 2007.

[116] N. Mladenović and P. Hansen, "Variable neighborhood search," *Computers & Operations Research*, vol. 24, no. 11, pp. 1097–1100, 1997. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0305054897000312

[117] J. Kleinberg, "An impossibility theorem for clustering," *Advances in neural information processing systems*, pp. 463–470, 2003.

[118] L. Hubert and P. Arabie, "Comparing partitions," *Journal of Classification*, vol. 2, no. 1, pp. 193–218, dec 1985. [Online]. Available: https://doi.org/10.1007/BF01908075

[119] M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml

[120] R. Randel, D. Aloise, S. J. Blanchard, and A. Hertz, "A lagrangian-based score for assessing the quality of pairwise constraints in semi-supervised clustering," *Data Mining and Knowledge Discovery*, 2021. [Online]. Available: https://doi.org/10.1007/s10618-021-00794-0

[121] V. Grossi, A. Romei, and F. Turini, "Survey on using constraints in data mining," *Data Mining and Knowledge Discovery*, vol. 31, no. 2, pp. 424–464, 2017.

[122] S. Kim, S. J. Blanchard, W. S. DeSarbo, and D. K. Fong, "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*, vol. 50, no. 5, pp. 664–673, 2013.

[123] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[124] P. Brucker, "On the complexity of clustering problems," in *Optimization and Operations Research*, R. Henn, B. Korte, and W. Oettli, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1978, pp. 45–54.

[125] A. W. F. Edwards and L. L. Cavalli-Sforza, "A method for cluster analysis," *Biometrics*, vol. 21, no. 2, pp. 362–375, 1965. [Online]. Available: http://www.jstor.org/stable/2528096

[126] O. Kariv and S. L. Hakimi, "An algorithmic approach to network location problems. ii: The p-medians," *SIAM Journal on Applied Mathematics*, vol. 37, no. 3, pp. 539–560, 1979.

[127] P. Avella, A. Sassano, and I. Vasil'ev, "Computational study of large-scale p-median problems," *Mathematical Programming*, vol. 109, no. 1, pp. 89–114, 2007.

[128] S. García, M. Labbé, and A. Marín, "Solving large p-median problems with a radius formulation," *INFORMS Journal on Computing*, vol. 23, no. 4, pp. 546–556, 2011.

[129] P. Hansen, J. Brimberg, D. Urošević, and N. Mladenović, "Solving large p-median clustering problems by primal–dual variable neighborhood search," *Data Mining and Knowledge Discovery*, vol. 19, no. 3, pp. 351–375, 2009.

[130] Y. Kochetov and D. Ivanenko, *Computationally Difficult Instances for the Uncapacitated Facility Location Problem*. Boston, MA: Springer US, 2005, pp. 351–367. [Online]. Available: https://doi.org/10.1007/0-387-25383-1_16

[131] D. Aloise, P. Hansen, and L. Liberti, "An improved column generation algorithm for minimum sum-of-squares clustering," *Mathematical Programming*, vol. 131, no. 1, pp. 195–220, Feb 2012.

[132] M. Held, P. Wolfe, and H. P. Crowder, "Validation of subgradient optimization," *Mathematical Programming*, vol. 6, no. 1, pp. 62–88, dec 1974.

[133] S. J. Blanchard, D. Aloise, and W. S. DeSarbo, "The heterogeneous p-median problem for categorization based clustering," *Psychometrika*, vol. 77, no. 4, pp. 741–762, sep 2012.

[134] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[135] J. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York, NY: Plenum Press, 1981.

[136] D. N. Pinheiro, D. Aloise, and S. J. Blanchard, "Convex fuzzy k-medoids clustering," *Fuzzy Sets and Systems*, vol. 389, pp. 66–92, 2020.

[137] R. J. Campello, D. Moulavi, A. Zimek, and J. Sander, "A framework for semi-supervised and unsupervised optimal extraction of clusters from hierarchies," *Data Mining and Knowledge Discovery*, vol. 27, no. 3, pp. 344–371, 2013.

[138] H. Chang and D. Y. Yeung, "Locally linear metric adaptation with application to semi-supervised clustering and image retrieval," *Pattern Recognition*, vol. 39, no. 7, pp. 1253–1264, 2006.

[139] S. Xiang, F. Nie, and C. Zhang, "Learning a Mahalanobis distance metric for data clustering and classification," *Pattern Recognition*, vol. 41, no. 12, pp. 3600–3612, 2008.

[140] A. Bellet, A. Habrard, and M. Sebban, "Metric learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 9, no. 1, pp. 1–151, 2015.

[141] H. Le Capitaine, "Constraint selection in metric learning," *Knowledge-Based Systems*, vol. 146, pp. 91–103, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705118300418

[142] C. Pichery, "Sensitivity analysis," in *Encyclopedia of Toxicology (Third Edition)*, third edition ed., P. Wexler, Ed. Oxford: Academic Press, 2014, pp. 236 – 237. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780123864543004310

[143] R. V. Rao, *A novel weighted Euclidean distance-based approach.* London: Springer London, 2013, pp. 159–191.

[144] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[145] I. Rodrigues, D. Aloise, and E. R. Fernandes, "An effective sequence alignment method for duplicate crash report detection," in *4th International Workshop on Machine Learning Techniques for Software Quality Evolution (MaLTeSQuE2020)*, Nov. 2020.

[146] C. Manning and H. Schutze, *Foundations of statistical natural language processing.* MIT press, 1999.