Data Clustering Validation using Constraints

João M. M. Duarte[1,2], Ana L. N. Fred[1] and F. Jorge F. Duarte[2]
[1]*Instituto de Telecomunicações, Instituto Superior Técnico, Lisboa, Portugal,*
[2]*GECAD - Knowledge Engineering and Decision-Support Research Center,*
Institute of Engineering, Polytechnic of Porto, Porto, Portugal

Keywords: Clustering Validation, Constrained Data Clustering.

Abstract: Much attention is being given to the incorporation of constraints into data clustering, mainly expressed in the form of must-link and cannot-link constraints between pairs of domain objects. However, its inclusion in the important clustering validation process was so far disregarded. In this work, we integrate the use of constraints in clustering validation. We propose three approaches to accomplish it: produce a weighted validity score considering a traditional validity index and the constraint satisfaction ratio; learn a new distance function or feature space representation which better suits the constraints, and use it with a validation index; and a combination of the previous. Experimental results in 14 synthetic and real data sets have shown that including the information provided by the constraints increases the performance of the clustering validation process in selecting the best number of clusters.

1 INTRODUCTION

Data clustering aims at discovering structure in data, i.e. the natural grouping(s) of a set of domain objects, such that similar domain objects are assigned to the same cluster and objects that are dissimilar are grouped in different clusters (Jain, 2010). Data clustering is used in several important applications, such as image segmentation, data summarization, grouping customers for marketing purposes, organizing documents into a hierarchy of topics and subtopics or to the study the genome data. Let $\mathcal{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$ be a set of n domain objects. The goal of a clustering algorithm is to divide a data set $\mathcal{X}$ into K clusters, producing a data partition P, $P = \{C_1, \cdots, C_K\}$ where C_k represents an individual cluster.

There are situations where domain knowledge exists for a particular application that may not be characterized as data features. Also, a data analyst may want to express some preferences or conditions for the data clustering. Constrained data clustering algorithms (Basu et al., 2008; Wagstaff, 2002; Basu, 2005; Davidson and Ravi, 2005; Wang and Davidson, 2010) use *a priori* knowledge about the data, mapped in the form of constraints, to produce more useful solutions.

Distinct data partitions for the same data set may be obtained using different clustering algorithms, parameters and/or algorithm initializations. This arises the problem of evaluating the quality of a given data partition or selecting the best data partition among a set of possible ones. To address this problem, many clustering validity techniques have been proposed (Arbelaitz et al., 2013). These can be organized into two categories: internal validity indices, which use the information contained in the data set and in a data partition; and external validity indices, that compare a data partition with another partition believed to be the correct one. The former should be used in real applications where the underlying data structure needs to be discovered while the latter can be used to assess the performance of clustering algorithms using benchmark data sets.

So far, the constraints are not being used in clustering validation. This is an important flaw, especially if the clustering solution was obtained from a constrained clustering algorithm. In this paper we introduce the use of constraints in the clustering validation process. We address the problem in three ways: using both the constraint satisfaction ratio score and a validity index score to produce a weighted validity score; learning a distance function or a feature space representation that transforms the original feature space accordingly to the set of constraints and use it in the clustering validity process; and a combination of these approaches.

The rest of the paper is organized as follows. In section 2, constrained data clustering is briefly introduced and the algorithms used in the experiments are

M. N. Duarte J., L. N. Fred A. and F. Duarte F..
Data Clustering Validation using Constraints.
DOI: 10.5220/0004543800170027
In *Proceedings of the International Conference on Knowledge Discovery and Information Retrieval and the International Conference on Knowledge Management and Information Sharing* (KDIR-2013), pages 17-27
ISBN: 978-989-8565-75-4

presented. Section 3 describes how we address the constrained clustering validity problem. Experimental results are discussed in section 4, and conclusions are presented in section 5.

2 DATA CLUSTERING WITH CONSTRAINTS

Traditional clustering algorithms fail to deal with peculiarities that may exist in certain data clustering problems: in clustering with obstacles (Tung et al., 2000) the distance between domain objects may not follow a straight line (e.g. the walking distance between houses in opposite margins of a river must consider passing trough a bridge); in other situations one may want to shape the characteristics of the clusters, such as the minimum and maximum number of objects that a cluster must contain (Ge et al., 2007); sometimes the labels of a subset of the data are known and using that information would improve the clustering solution (Basu, 2005); also, relations between pairs of objects, such as must-link and cannot-link constraints, may be available from an expert or can be generated for a specific problem (Wagstaff, 2002).

In our work, we will focus in the relations between pairs of clusters. The reason is that many of the other types of clustering constraints may be converted into must-link and cannot-link constraints. The set of must-link constraints, $\mathcal{R}_=$, contains all the pairs of objects $(\mathbf{x}_i, \mathbf{x}_j)$ that should belong to the same cluster, while all the pairs of objects that should not be grouped in the same cluster are added to the cannot-link set $\mathcal{R}_{\neq}$.

2.1 Constrained Data Clustering Algorithms

In this subsection, we introduce two pairwise-constrained clustering algorithms that are used in the experiments presented in section 4.

The first one, the Pairwise-Constrained K-Means (PCKM) (Basu et al., 2004), is a partitive algorithm based on the well-know k-means algorithm (MacQueen, 1967), which minimizes the distances between each cluster mean vector $\overline{\mathbf{x}}_k$ and the corresponding domain objects $\mathbf{x}_i \in C_k$. The objective of PCKM consists of minimizing the k-means objective and the violation of constraints, simultaneously. The PCKM associates a weight w_{ij} to each constraint between a pair of objects $\mathbf{x}_i, \mathbf{x}_j$. These weights are used as penalization costs every time a constraint is not satisfied in the current data partition. The PCKM objective-function is defined as

$$J = \frac{1}{2}\sum_{\mathbf{x}_i \in X} ||\mathbf{x}_i - \overline{\mathbf{x}}_{P_i}||^2 + \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{R}_=} w_{ij} I(P_i \neq P_j) \quad (1)$$
$$+ \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{R}_{\neq}} w_{ij} I(P_i = P_j),$$

where P_i corresponds to the label of the cluster attributed to $\mathbf{x}_i$, and $\overline{\mathbf{x}}_{P_i}$ is the mean vector of the cluster to which $\mathbf{x}_i$ belongs. $I(\cdot)$ returns 1 if the argument is true, and returns 0 otherwise.

The other clustering algorithm is the Constrained Average-Link (CAL) (Duarte et al., 2012) which is based on the agglomerative hierarchical clustering algorithm Average-Link (Sokal and Michener, 1958). The algorithm works as follows. It starts with n clusters, one for each domain object $\mathbf{x}_i$. Then, at each step, the two closest clusters, according to a distance measure between clusters, are merged. The process iterates until some stopping criteria is met (e.g. a predefined number of clusters is reached) or all objects belong to same cluster. The distance between clusters measures the average distance between all pairs of objects belonging to different clusters plus a penalization for each constraint that is not satisfied. This distance is defined as

$$d(C_k, C_l) = \frac{\sum_{i=1}^{|C_k|}\sum_{j=1}^{|C_l|} \mathrm{dist}(\mathbf{x}_i,\mathbf{x}_j) - \mathrm{I}_=(\mathbf{x}_i,\mathbf{x}_j) + \mathrm{I}_{\neq}(\mathbf{x}_i,\mathbf{x}_j)}{|C_k||C_l|}, \quad (2)$$

where $\mathrm{I}_a(\mathbf{x}_i,\mathbf{x}_j) = \mathrm{p}$ if $(\mathbf{x}_i,\mathbf{x}_j) \in \mathcal{R}_a$ and 0 otherwise. $\mathrm{p} \geq 0$ is a user parameter that influences the "softness" of the constraints. In our experiments we defined p as the maximum distance between objects in a data set.

2.2 Acquiring Pairwise Constraints

Pairwise constraints for data clustering may be obtained by several ways. For instance, in image segmentation, pixels may be regarded as domain objects and must-link constraints may be generated between a object $\mathbf{x}_i$ and all its direct neighbors (Wagstaff, 2002). Must-link constrains may also be generated between a object $\mathbf{x}_i$ and all the other objects within a certain distance, while cannot-link constraints may be created if the distance between $\mathbf{x}_i$ and other objects exceed a threshold value (Davidson and Ravi, 2005). In web page categorization, an expert may indicate if two web pages are similar or dissimilar (Cohn et al., 2003). In some real world applications, the labels from a subset of the data are known and can be used to derive pairwise constraints (Basu, 2005).

In our work we used two different schemes for acquiring must-link and/or cannot-link constraints. The first one is the Random Acquisition of Constraints (RAC) and consists of randomly selecting two objects $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{X}$, iteratively, and ask the user (or some oracle) if both objects should be assigned to the same group. If the answer is “Yes”, a must-link constraint is added to the set of must-link constraints, $\mathcal{R}_{=} = \mathcal{R}_{=} \cup \{(\mathbf{x}_i, \mathbf{x}_j)\}$. If the answer is “No” a cannot-link constraint is added to the set of cannot-link constraints $\mathcal{R}_{\neq} = \mathcal{R}_{\neq} \cup \{(\mathbf{x}_i, \mathbf{x}_j)\}$. If the user cannot decide, the pair of objects is marked as *already tested*. Questions involving pairs of objects that are already in a constraint set or marked as *already tested* cannot be performed. The process repeats until a predefined number of constraints is achieved. Another way to acquire pairwise constraints is the Random Acquisition of Labels (RAL), which is based on randomly selecting a subset of the data set objects and ask the user/oracle for the corresponding cluster labels. After having a labeling for the subset of the data, for each pair $(\mathbf{x}_i, \mathbf{x}_j)$ in that subset we add a must-link constraint to $\mathcal{R}_{=}$ if the labels of both objects are the same. If the labels are different, we add a cannot-link constraint between $\mathbf{x}_i$ and $\mathbf{x}_j$ to $\mathcal{R}_{\neq}$. Notice that the RAL method produces several more constraints than the RAC for the same number of questions to the user/oracle.

3 CLUSTERING VALIDATION

In this section we will propose how constraints can be used to define internal validity criteria.

3.1 Clustering Validation using Constraints

A clustering algorithm takes as input the representation of a given data set and produces a data partition according to its clustering criterion and parameters. Traditionally, a clustering validity index uses the data partition and the original data representation to produce a quantitative indicator, or score, of the quality of the data partition, as illustrated in figure 1. Two examples of internal validity indices will be described in subsection 3.2.

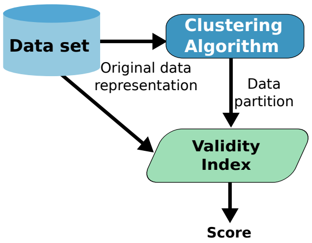


Figure 1: Traditional clustering validation.

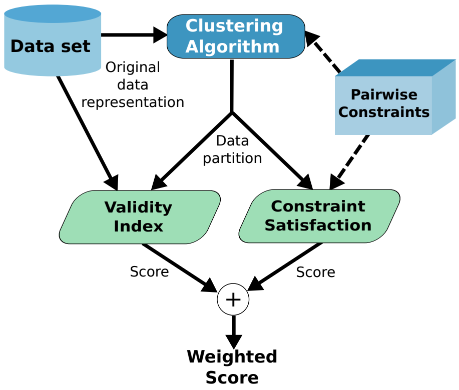


Figure 2: Clustering validation with weighted score.

With the emergence of constrained data clustering this process is no longer suitable, since the constraints are not considered in the validation phase and these contain important information for assessing the quality of data partitions. To include constraints in the validation process, we present three scenarios: weighting the scores of a validation index with the ratio of the constraint satisfaction; learning a distance measure or data representation that considers both the data feature attributes and the constraints and outputs it to a validation index; and the combination of the previous two.

Figure 2 depicts the simple weighting approach. A clustering algorithm uses as input both the data set representation and the constraints. Let us assume the constraints are in the form of must-link and cannot-link constraints. As before, a validity index evaluates the resulting partition using the original data representation and produces a score. In addition, a constraint satisfaction ratio is computed from the set of constraints and the data partition. The constraint satisfaction ratio is defined as

$$CS(P) = \frac{\sum\limits_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{R}_{=}} I(P_i = P_j) + \sum\limits_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{R}_{\neq}} I(P_i \neq P_j)}{|\mathcal{R}_{=}| + |\mathcal{R}_{\neq}|} \tag{3}$$

where $I(\cdot)$ returns 1 if the argument is true, and returns 0 otherwise. Then a weighted score is computed as

$$WS(P) = (1-\alpha)NormIndex(P) + \alpha CS(P) \tag{4}$$

where $0 \leq \alpha \leq 1$, and $NormIndex(P)$ represents the normalization of the validity index with values in the $[0,1]$ interval, and assuming that values close to 1 indicate good data partitions while values close to 0 indicate bad data partitions. Thus, the weighted score is $[0,1]$ bounded in the same way.

Another approach consists of learning a distance metric or feature space representation that simultaneously reflects both the original feature space and

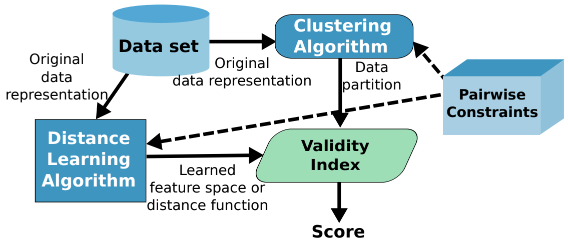

Data set
Clustering Algorithm
Original data representation
Original data representation
Data partition
Pairwise Constraints
Distance Learning Algorithm
Validity Index
Learned feature space or distance function
Score

Figure 3: Clustering validation with distance learning.

the constraints. An example of a distance learning method will be presented in 3.3 and will be used in our experiments. When learning a new metric, the idea is determining a positive semi-definite matrix $\mathbf{M}$ that satisfies the properties of a metric, and then use it to parameterize the Mahalanobis distance:

$$\texttt{dist}_{\mathbf{M}} = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{M} (\mathbf{x}_i - \mathbf{x}_j)}. \qquad (5)$$

In this case, existing implementations of validity indices should be modified to take into account the Mahalanobis distance. When learning a new feature space, that inconvenience does not exist because after the new representation is learned it can be simply fed to the clustering validation index. Figure 3 illustrates this approach. The clustering algorithm produces a data partition using the original data and the constraints. Then, a distance learning algorithm learns a new metric represented by $\mathbf{M}$ or a new feature space representation using as input both the original data features and the constraints, and outputs it to the validity index finally evaluate the data partition.

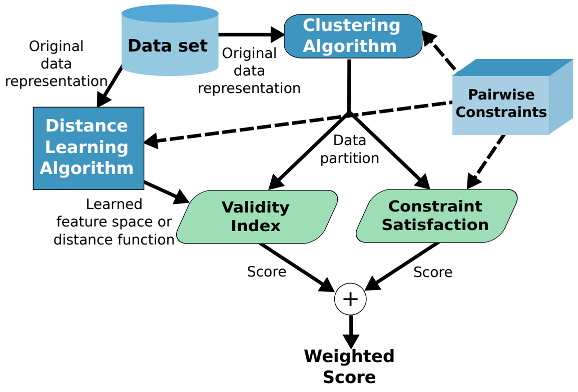

Data set
Clustering Algorithm
Original data representation
Original data representation
Pairwise Constraints
Distance Learning Algorithm
Data partition
Learned feature space or distance function
Validity Index
Constraint Satisfaction
Score
Score
+
Weighted Score

Figure 4: Clustering validation with distance learning and weighted score.

The last approach for constrained clustering validations is simply the combination of the previous two approaches, as can be seen in figure 4. The data partition produced by the clustering algorithm is scored by both the constraint satisfaction ratio and the validity index. In this case, the validity index (that should be normalized to the $[0,1]$ interval) will use the learned feature space or distance function to assess the data partition, just as explained before. Finally, the weighted score is computed as defined in equation 4.

3.2 Validity Indices

We used two well-known internal validity indices to assess the incorporation of constraints into the clustering validation process. These will be described next.

Let a_i denote the average distance between $\mathbf{x}_i \in C_l$ and the other objects in the same cluster, and b_i the minimum average distance between $\mathbf{x}_i$ and all objects grouped in another cluster:

$$a_i = \frac{1}{|C_l| - 1} \sum_{\substack{\mathbf{x}_j \in C_l \\ j \neq i}} \texttt{dist}(\mathbf{x}_i, \mathbf{x}_j), \qquad (6)$$

$$b_i = \min_{k \neq l} \frac{1}{|C_k|} \sum_{\mathbf{x}_j \in C_k} \texttt{dist}(\mathbf{x}_i, \mathbf{x}_j), \mathbf{x}_i \in C_l. \qquad (7)$$

$\texttt{dist}(\mathbf{x}_i, \mathbf{x}_j)$ may by any distance function between two objects. The silhouette width is defined for each object $\mathbf{x}_i$

$$s_i = \frac{b_i - a_i}{\max\{a_i, b_i\}}, \qquad (8)$$

and indicates how well $\mathbf{x}_i$ is adjusted to its cluster when compared to other clusters. A value close to 1 means that $\mathbf{x}_i$ has been assigned to the appropriate cluster, a value close to 0 suggests $\mathbf{x}_i$ could also have been assigned to the nearest cluster, and a value close to -1 indicates that $\mathbf{x}_i$ was incorrectly assigned. The Silhouette index (Rousseeuw, 1987), $S(P)$, is given by the average silhouette width computed over all objects in the data set:

$$S(P) = \frac{1}{n} \sum_{i=1}^{n} s_i. \qquad (9)$$

The Hubert's Statistic (Hubert and Arabie, 1985) measures the correlation between a $n \times n$ co-membership matrix, $\mathbf{U}$, that represents the data partition P, and a $n \times n$ distance matrix, $\mathbf{D}$, which stores the distances between all pairs of objects. The co-membership $\mathbf{U} = [U_{ij}]$ is built by setting each entry U_{ij} to 1 if both $\mathbf{x}_i$ and $\mathbf{x}_j$ were assigned to the same cluster ($P_i = P_j$), or to 0 otherwise. Each entry of the distance matrix $\mathbf{D} = [D_{ij}]$ consists of the distance between $\mathbf{x}_i$ and $\mathbf{x}_j$, i.e., $D_{ij} = \texttt{dist}(\mathbf{x}_i, \mathbf{x}_j)$, where $\texttt{dist}$ can be any distance function. The Hubert's Statistic is defined as

$$H(P) = \frac{2}{n(n-1)} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} U_{ij} D_{ij} \qquad (10)$$

considering the matrices to be symmetric. High $H(P)$ values indicate good data partitions. However, $H(P)$ values increases with the number of clusters. A normalized version of Hubert's Statistic prevents this bias and is defined as

$$NH(P) = \frac{2}{n(n-1)} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{(U_{ij} - \mu_U)(D_{ij} - \mu_D)}{\sigma_U \sigma_D} \qquad (11)$$

where μ_U, μ_D, σ_U and σ_D are the means and standard deviations of $\mathbf{U}$ and $\mathbf{D}$, respectively.

3.3 Distance Learning using Pairwise Constraints

The Discriminant Component Analysis (DCA) (Hoi et al., 2006) was used in our work to learn new distance metrics that simultaneously represent both the original data and the clustering preferences, expressed as must-link and cannot-link constraints. The DCA builds a set of chunklets $\mathcal{Q} = \{Q_1, \cdots, Q_q\}$, i.e. groups of objects connected by must-link constraints, and a set of discriminative chunklets $\mathcal{S} = \{S_1, \cdots, S_q\}$, one for each chunklet Q_i. Each element of the discriminative chunklet S_i indicates the chunklets that have at least one cannot-link constraint connecting a object in Q_i. Then DCA learns a data transformation which minimizes the variance between objects in the same chunklet Q_i and maximizes the variance between discriminative data chunklets S_i. The covariance matrices, $\mathbf{C}_b$ and $\mathbf{C}_w$, store the total variance between objects in each $S_i \in \mathcal{S}$ and the total variance within objects in the same chunklets $\forall Q_i \in \mathcal{Q}$. These matrices are computed as:

$$\mathbf{C}_b = \frac{1}{\sum_{i=1}^{q} |S_i|} \sum_{i=1}^{q} \sum_{i \in S_j} (\mathbf{m}_j - \mathbf{m}_i)(\mathbf{m}_j - \mathbf{m}_i)^\top, \qquad (12)$$

$$\mathbf{C}_w = \frac{1}{q} \sum_{j=1}^{q} \frac{1}{|Q_j|} \sum_{\mathbf{x}_i \in Q_j} (\mathbf{x}_i - \mathbf{m}_j)(\mathbf{x}_i - \mathbf{m}_j)^\top, \qquad (13)$$

where $\mathbf{m}_j$ is the mean vector of Q_j. The optimal transformation matrix $\mathbf{A}$ is obtained by optimizing the following objective-function:

$$J(\mathbf{A}) = \arg\max_{\mathbf{A}} \frac{|\mathbf{A}^\top \mathbf{C}_b \mathbf{A}|}{|\mathbf{A}^\top \mathbf{C}_w \mathbf{A}|}. \qquad (14)$$

The learned Mahalanobis matrix $\mathbf{M}$ is computed as $\mathbf{M} = \mathbf{A}^\top \mathbf{A}$. The algorithm we used to optimize equation 14 is described in (Hoi et al., 2006).

4 EXPERIMENTAL RESULTS

In this section we show how the different validity indices perform in the selection of the number of clusters to partition the data for a given clustering algorithm.

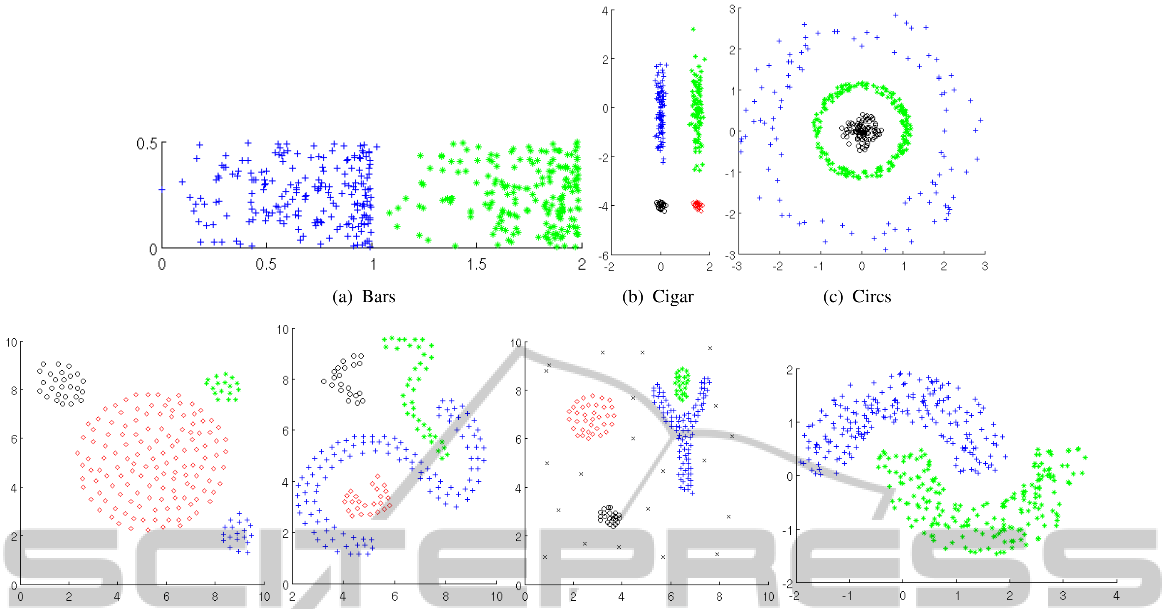
In our experiments, 7 synthetic data sets (shown in figure 5) and 7 real data sets taken from the UCI ML repository (Bache and Lichman, 2013) were used to assess the performance of the validity indices. The number of domain objects (n), number of dimensions (d), number of *natural* clusters (K^0) and the distribution of objects per cluster for each data set are presented in table 1. A brief description of each real data set is given next.

Table 1: Overview of the data sets.

Data sets	n	d	K^0	Cluster Distribution
Bars	400	2	2	2×200
Cigar	250	2	4	$2 \times 100 + 2 \times 25$
Circs	400	2	3	$2 \times 100 + 200$
D1	200	2	4	$19 + 17 + 26 + 138$
D2	200	2	4	$116 + 39 + 21 + 24$
D3	200	2	5	$98 + 23 + 23 + 35 + 21$
Half Rings	400	2	2	2×200
Wine	178	13	3	$59 + 71 + 48$
Yeast Cell	384	17	5	$67 + 135 + 75 + 52 + 55$
Optdigits	1000	64	10	10×100
Iris	150	4	3	3×50
House Votes	232	16	2	$124 + 108$
Breast Cancer	683	9	2	$444 + 239$

The Iris data set consists of 50 objects from each of three species of Iris flowers (setosa, virginica and versicolor) characterized by four features. One of the clusters is well separated from the other two overlapping clusters. The Breast Cancer data set is composed of 683 domain objects characterized by nine features and divided into two clusters: benign and malignant. The Yeast Cell data set consists of 384 objects described by 17 attributes, split into five clusters concerning five phases of the cell cycle. There are two versions of this data set: the first one, called Log Yeast, uses the logarithm of the expression level; the other, called Std Yeast, is a "standardized" version of the same data set, with mean 0 and variance 1. The Optdigits is a subset of Handwritten Digits data set containing only the first 100 objects of each digit, from a total of 3823 domain objects characterized by 64 attributes. The House Votes data set is composed of two clusters of votes for each of the U.S. House of Representatives Congressmen on the 16 key votes identified by the Congressional Quarterly Almanac. From a total of 435 (267 democrats and 168 republicans) only the objects without missing values were considered, resulting in 232 objects (125 democrats and 107 republicans). The Wine data set consists of the results of a chemical analysis of wines grown in the same region in Italy, divided into three clusters with 59, 71 and 48 objects, described by 13 features.

In order to test the proposed approaches for con-



(a) Bars (b) Cigar (c) Circs

(d) D1 (e) D2 (f) D3 (g) Half Rings

Figure 5: Synthetic data sets.

strained clustering validation, two constraint sets were built for each data set: one using the RAL process considering the labels of 10% of the objects; and the other using the RAC process producing $0.5n$ random pairwise constraints, where n is the number of objects in a data set. We use ground-truth information to identify the labels of the objects and to answer whether or not two objects belong to the same cluster. Therefore, an "Yes"/"No" answer is always obtained in RAC process. Then, for each constraint set, the PCKM and CAL algorithms produced 14 data partition each, one for each number of clusters k in the set $\{2,\cdots,15\}$. Finally, we applied the original Silhouette index and the Normalized Hubert's statistic, and our constrained validation approaches using these validation indices to all the data partitions. Both the Silhouette index and the Normalized Hubert's statistic return a score in the $[-1,1]$ range. To normalize the score range to the $[0,1]$ for the weighting approaches, we simply compute it as $NormScore = \frac{1}{2}(Score+1)$. The α parameter for the constraint satisfaction ratio was defined as $\frac{1}{2}$. We used the Consistency index (*CI*) (Fred, 2001), an external validity index, to evaluate the quality of the data partitions. Thereby, we can assess the performance of the internal validity indices by comparing its scores with the ones obtained by *CI*. The Consistency index measures the fraction of shared objects in matching clusters of a given partition (P) and the *real* data partition (P^0) obtained from ground-truth information. The Consistency index is computed as

$$CI(P,P^0) = \frac{1}{n} \sum_{k=1}^{\min\{K,K^0\}} |C_k \cap C_k^0| \qquad (15)$$

where K is the number of clusters of the partition that is being evaluated, K^0 is the true number of clusters, and it is assumed the clusters of P and P^0 have been permuted in a way that the cluster C_k matches with the real cluster C_k^0.

Tables 2 to 5 present the consistency index values for the data partitions selected (corresponding to the selection of K) by each clustering validation index using CAL and PCKM clustering algorithms with the constraint sets built by the RAL and RAC processes. The first column of each table indicates the benchmark data set, the second and third columns show the *CI* values for the partitions selected by the traditional Silhouette (S) and Normalized Hubert's statistic (NH), and the forth and fifth columns show the *CI* values for the partitions selected by our weighted score approach using also the Silhouette and Normalized Hubert's statistic indices (S+CS and NH+CS) with the original metric/feature space. Columns 6 to 9 present the analogous results for the proposed distance learning approach with and without weighting the constraint satisfaction ratio. The last column indicates the consistency index for the best partition produced for each data set, according to the ground-truth information. The last line shows the number of times that each validity measure selected the best data partition (best K).

Table 2 shows the results for the CAL clustering algorithm using the RAL process. By comparing all the results that use the Silhouette index, it can be seen that the traditional Silhouette index selected the best data partition in 7 out of the 14 data sets, the same number obtained the corresponding weighting approach despite the selected partitions are not the same. The results obtained by learning a distance function were slightly better. With and without weighting scores, the indices picked the best partitions 8 times. By doing the same analysis for the Normalized Hubert's statistic results, we noticed that the traditional approach only identifies the best partition in 5 out of the 14 data sets. The weighted score approach with the original distance metric choose the best partition 6 times. By using a learned distance metric the best partition is selected 8 times and by combining it with the constraint satisfaction the best partitions are picked in 9 out 14 data sets.

The results for the PCKM clustering algorithm using the RAL process are shown in Table 3. The worst results were again achieved by the traditional validity approach, having the best partitions been selected only 4 times by both the Silhouette index and the Normalized Hubert's statistic. The results for the weighted score approach are somewhat better. Both indices selected 6 times the best partition. By learning a new distance metric the results are considerable better. The Silhouette index selects the best partition 8 and 6 times with and without using the score weighting, respectively. The Normalized Hubert's statistic identifies the best data partitions 9 out of the 14 data sets both with and without the score weighting.

Table 4 shows the results for the CAL clustering algorithm using the RAC process. The traditional Silhouette index picked the best partitions 8 times and with the score weighting 10 times, both with the original and learned metrics. For the simple distance learning approach, the best partitions were selected in 8 out of 14 data sets. The Normalized Hubert's statistic results were not so good by selecting only 3, 4, 8 and 6 times the best partition for the traditional, score weighting, distance learning and distance learning plus score weighting approaches, respectively. Nonetheless, the results obtained using constraints are better, especially when the distance metric was learned.

The results for the PCKM algorithm using the RAC process are presented in table 5. The traditional Silhouette index determined the best data partitions 8 times and the corresponding weighted score approach identified the same best partitions plus another one. The simple distance learning approach selected the best partition in 9 out of the 14 data sets and combining it with the weighted score approach decreases the number of identified best partitions by one. The traditional Normalized Hubert's statistic only selects the best partition 5 times and the score weighting approach in 8 times. The simple metric learning approaches picks the best partition also 8 times and, again, the weighted score with distance learning approach diminishes the number of identified best partitions by one. We may conclude from the previous results that the incorporation of constraints clearly increases the performance of the clustering validation process. By simple weighting a validity index score with the constraint satisfaction ratio the results were better. Also, it seems that learning a new metric based on the pairwise constraints leads to even better results.

Table 6 indicates the number of times that each validity measure (by line) achieved better/worse/equal results than the other validity measures (by column). The Silhouette with the score weighting approach obtained 14 times better results than the traditional Silhouette and 9 times worse. The Silhouette with distance learning achieved 10 times better partitions and only 6 times worse. The metric learning combined with the score weighting achieved 16 better results and 10 worse. By performing the same analysis for the Normalized Hubert's statistic, the weighted score approach was better than the traditional one 15 times and 9 worse. The distance learning approach obtained better results 21 times and 7 worse. The combination of metric learning and score weighting obtained 22 improvements and only 12 results worst. These results evidence again that constrained clustering validation outperforms the traditional validation approach especially when using distance learning.

Figure 6 shows the plots of the consistency index values and the constraint satisfaction ratio obtained for all partitions produced in our experiments versus each internal validation index, distinguished by clustering algorithm and constraint acquisition method, and table 7 presents the correlation between the internal validation indices and the consistency index. It can be seen that very different consistency values may be achieved for partitions with all constraints satisfied (figure 6c). This indicates that the constraint satisfaction ratio alone is not a good indicator of the quality of the partitions which is corroborated by the low correlation with the consistency index. We can also conclude that the validation approaches that use distance learning have higher correlation with the consistency index, which is another indication that these are a good option for clustering validation.

Table 2: Consistency index values for the data partitions selected by each clustering validation index using the CAL algorithm and the constraint sets generated by the RAL process.

Data sets	**Original Distance**				**Learned Distance**				**Best**
	S	**NH**	**S + CS**	**NH + CS**	**S**	**NH**	**S + CS**	**NH + CS**	**Partition**
Bars	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	1.000
Cigar	0.500	0.504	0.688	**0.896**	0.872	**0.896**	0.872	**0.896**	0.896
Circs	**0.583**	**0.583**	0.430	0.430	**0.583**	0.545	0.430	0.430	0.583
D1	0.820	0.875	0.410	0.875	0.820	**0.915**	**0.915**	**0.915**	0.915
D2	0.475	0.475	0.395	**0.600**	0.535	0.535	0.535	0.495	0.600
D3	0.640	**0.795**	0.655	0.655	**0.795**	**0.795**	0.655	0.655	0.795
Half Rings	**0.930**	**0.930**	**0.930**	**0.930**	**0.930**	**0.930**	**0.930**	**0.930**	0.930
Wine	**0.685**	0.584	0.663	0.663	**0.685**	0.584	0.663	0.663	0.685
Std Yeast	0.659	0.680	**0.698**	0.680	0.659	0.656	**0.698**	**0.698**	0.698
Optical	**0.862**	0.837	**0.862**	0.837	**0.862**	**0.862**	**0.862**	**0.862**	0.862
Log Yeast	0.349	0.315	0.315	0.315	0.349	0.315	0.349	0.315	0.378
Iris	0.667	0.667	**0.920**	**0.920**	0.667	0.667	**0.920**	**0.920**	0.920
House Votes	**0.888**	**0.888**	**0.888**	**0.888**	**0.888**	**0.888**	**0.888**	**0.888**	0.888
Breast Cancer	**0.950**	0.753	**0.950**	0.753	**0.950**	**0.950**	**0.950**	**0.950**	0.950
#Best Partitions	7	5	7	6	8	8	8	9	

Table 3: Consistency index values for the data partitions selected by each clustering validation index using the PCKM algorithm and the constraint sets generated by the RAL process.

Data sets	**Original Distance**				**Learned Distance**				**Best**
	S	**NH**	**S + CS**	**NH + CS**	**S**	**NH**	**S + CS**	**NH + CS**	**Partition**
Bars	**0.995**	**0.995**	**0.995**	**0.995**	**0.995**	**0.995**	**0.995**	**0.995**	0.995
Cigar	0.468	0.468	0.552	**0.876**	**0.876**	**0.876**	**0.876**	**0.876**	0.876
Circs	0.465	0.468	0.465	0.468	0.465	0.468	0.465	0.468	0.543
D1	**0.595**	**0.595**	**0.595**	**0.595**	0.535	**0.595**	0.535	**0.595**	0.595
D2	0.495	0.495	0.495	0.495	0.485	0.485	0.485	0.485	0.620
D3	0.630	**0.665**	0.630	0.630	**0.665**	**0.665**	**0.665**	0.630	0.665
Half Rings	0.665	0.665	0.665	0.665	0.665	0.665	0.665	0.665	0.718
Wine	0.669	0.669	**0.691**	**0.691**	0.669	0.669	**0.691**	**0.691**	0.691
Std Yeast	0.589	0.763	0.763	0.763	0.651	**0.773**	**0.773**	**0.773**	0.773
Optical	0.833	0.877	0.833	0.877	**0.885**	**0.885**	**0.885**	**0.885**	0.885
Log Yeast	0.320	0.266	0.323	0.326	0.320	**0.367**	0.326	0.359	0.367
Iris	0.667	0.667	**0.913**	**0.913**	0.667	0.667	0.667	**0.913**	0.913
House Votes	**0.901**	**0.901**	**0.901**	**0.901**	**0.901**	**0.901**	**0.901**	**0.901**	0.901
Breast Cancer	**0.963**	0.766	**0.963**	0.766	**0.963**	**0.963**	**0.963**	**0.963**	0.963
#Best Partitions	4	4	6	6	6	9	8	9	

Table 4: Consistency index values for the data partitions selected by each clustering validation index using the CAL algorithm and the constraint sets generated by the RAC process.

Data sets	**Original Distance**				**Learned Distance**				**Best**
	S	**NH**	**S + CS**	**NH + CS**	**S**	**NH**	**S + CS**	**NH + CS**	**Partition**
Bars	**0.993**	**0.993**	**0.993**	**0.993**	**0.993**	**0.993**	**0.993**	**0.993**	0.993
Cigar	**0.880**	**0.880**	**0.880**	**0.880**	**0.880**	**0.880**	**0.880**	**0.880**	0.880
Circs	0.495	0.460	0.460	0.460	**0.523**	0.460	0.460	0.460	0.523
D1	**0.580**	**0.580**	**0.580**	**0.580**	**0.580**	**0.580**	**0.580**	**0.580**	0.580
D2	0.510	0.510	0.510	0.405	0.500	0.405	0.500	0.405	0.610
D3	0.570	0.570	0.570	0.570	**0.620**	0.615	0.570	0.550	0.620
Half Rings	**0.840**	0.675	**0.840**	0.675	**0.840**	0.675	**0.840**	0.675	0.840
Wine	0.663	0.663	0.663	**0.713**	0.663	**0.713**	0.663	**0.713**	0.713
Std Yeast	**0.667**	**0.667**	**0.667**	**0.667**	**0.667**	**0.667**	**0.667**	**0.667**	0.667
Optical	0.755	0.711	0.750	0.750	0.755	**0.782**	0.755	0.750	0.782
Log Yeast	0.297	0.328	**0.336**	**0.336**	0.297	0.315	0.234	0.315	0.336
Iris	**0.927**	**0.927**	**0.927**	**0.927**	0.633	**0.927**	**0.927**	**0.927**	0.927
House Votes	**0.940**	0.780	**0.940**	**0.940**	**0.940**	**0.940**	**0.940**	**0.940**	0.940
Breast Cancer	**0.963**	0.851	**0.963**	0.851	**0.963**	0.851	**0.963**	0.851	0.963
#Best Partitions	8	3	10	4	8	8	10	6	

Table 5: Consistency index values for the data partitions selected by each clustering validation index using the PCKM algorithm and the constraint sets generated by the RAC process.

Data sets	Original Distance				Learned Distance				Best Partition
	S	NH	S + CS	NH + CS	S	NH	S + CS	NH + CS	
Bars	**0.993**	**0.993**	**0.993**	**0.993**	**0.993**	**0.993**	**0.993**	**0.993**	0.993
Cigar	**0.880**	**0.880**	**0.880**	**0.880**	**0.880**	**0.880**	**0.880**	**0.880**	0.880
Circs	0.495	0.460	0.460	0.460	**0.523**	0.460	0.460	0.460	0.523
D1	**0.580**	**0.580**	**0.580**	**0.580**	**0.580**	**0.580**	**0.580**	**0.580**	0.580
D2	0.510	0.510	0.510	0.405	0.500	0.405	0.500	0.405	0.610
D3	0.570	0.570	0.570	0.570	**0.620**	0.615	0.570	0.550	0.620
Half Rings	**0.840**	0.675	**0.840**	0.675	**0.840**	0.675	**0.840**	0.675	0.840
Wine	0.663	0.663	0.663	**0.713**	0.663	**0.713**	0.663	**0.713**	0.713
Std Yeast	**0.667**	**0.667**	**0.667**	**0.667**	**0.667**	**0.667**	**0.667**	**0.667**	0.667
Optical	0.755	0.711	0.750	0.750	0.755	**0.782**	0.755	0.750	0.782
Log Yeast	0.297	0.328	**0.336**	**0.336**	0.297	0.315	0.234	0.315	0.336
Iris	**0.927**	**0.927**	**0.927**	**0.927**	0.633	**0.927**	**0.927**	**0.927**	0.927
House Votes	**0.940**	0.780	**0.940**	**0.940**	**0.940**	**0.940**	**0.940**	**0.940**	0.940
Breast Cancer	**0.963**	0.851	**0.963**	0.851	**0.963**	0.851	**0.963**	0.851	0.963
#Best Partitions	8	5	9	8	9	8	8	7	

Table 6: Number of times that each validity measure achieved better/worse/equal results than the other validity measures.

Validity Index		Original Distance				Learned Distance			
		S	NH	S + CS	NH + CS	S	NH	S + CS	NH + CS
Original Distance	**S**	-	18/11/27	9/14/33	20/19/17	6/10/40	12/18/26	10/16/30	17/19/20
	NH	11/18/27	-	8/25/23	9/15/32	13/24/19	7/21/28	9/28/19	12/22/22
	S+CS	14/9/33	25/8/23	-	16/8/32	16/14/26	15/16/25	6/10/40	13/9/34
	NH+CS	19/20/17	15/9/32	8/16/32	-	20/22/14	11/18/27	9/21/26	4/10/42
Learned Distance	**S**	10/6/40	24/13/19	14/16/26	22/20/14	-	10/13/33	7/11/38	17/18/21
	NH	18/12/26	21/7/28	16/15/25	18/11/27	13/10/33	-	12/12/32	11/8/37
	S+CS	16/10/30	28/9/19	10/6/40	21/9/26	11/7/38	12/12/32	-	14/8/34
	NH+CS	19/17/20	22/12/22	9/13/34	10/4/42	18/17/21	8/11/37	8/14/34	-

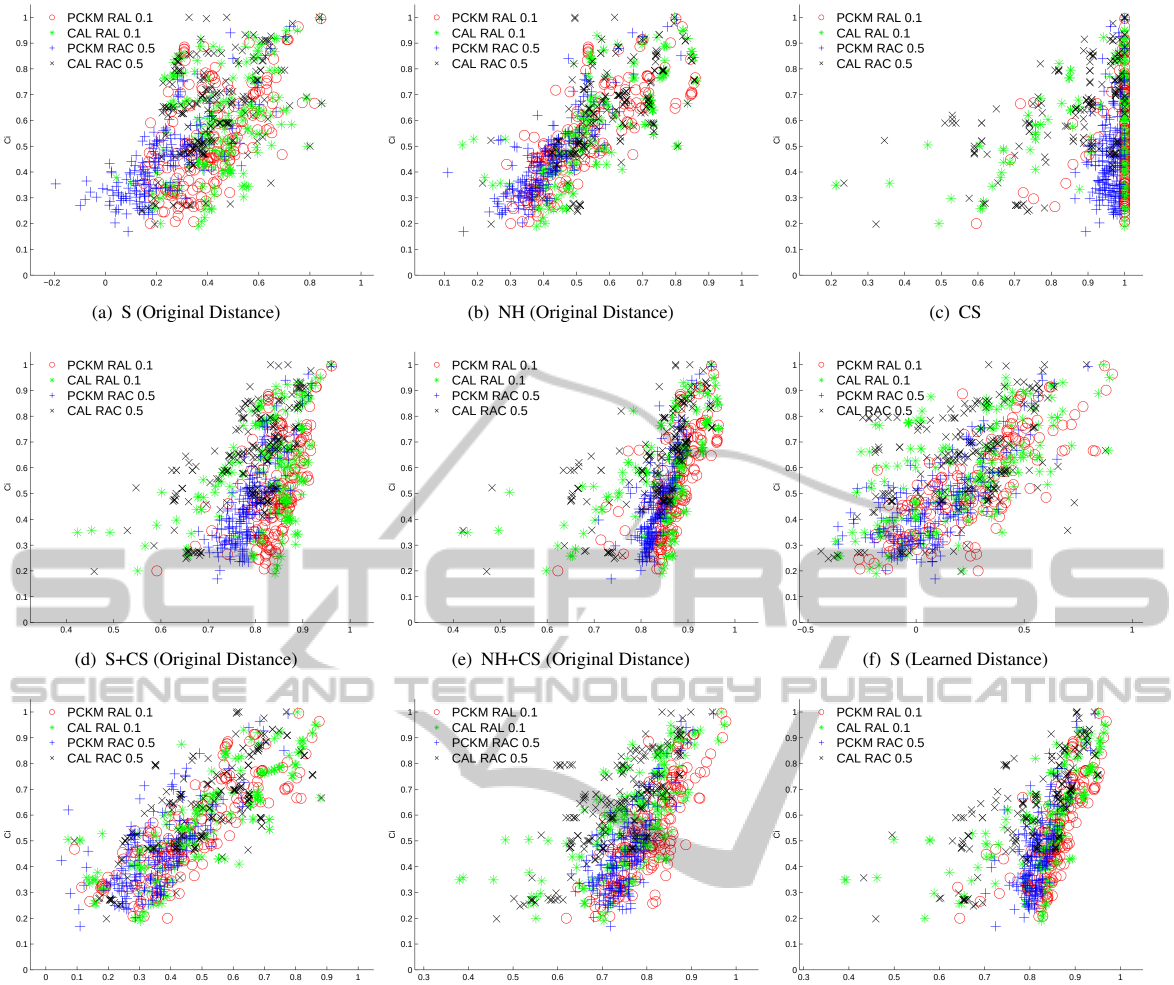
Table 7: Correlation between the internal validation indices and the consistency index.

	Validity Index	RAL 0.1		RAC 0.5		ALL
		PCKM	CAL	PCKM	CAL	
	CS	0.127	0.302	0.238	0.540	0.164
Original Distance	**S**	0.536	0.346	0.773	0.455	0.568
	NH	0.800	0.726	**0.840**	0.642	0.780
	S+CS	0.497	0.407	0.786	0.665	0.472
	NH+CS	0.684	0.547	0.786	0.661	0.542
Learned Distance	**S**	0.788	0.614	0.711	0.486	0.647
	NH	**0.828**	**0.818**	0.701	0.728	**0.797**
	S+CS	0.816	0.671	0.753	**0.747**	0.613
	NH+CS	0.742	0.634	0.694	0.717	0.576

5 CONCLUSIONS AND FUTURE WORK

We proposed the incorporation of constraints to the clustering validation in three ways: using both the scores of a validity index and the constraint satisfaction ratio to produce a weighted validity score; learning a distance function or feature space representation which considers the original feature space and the set of constraints, and use it in the clustering validity process; and combining the previous approaches.

Experimental results have shown that including constraints increases the performance of the clustering validation, especially if a new distance metric is learned. In the future, we want to extend our study to include other distance learning algorithms and validity indices, as well as evaluating data partitions produced by other algorithms.



(a) S (Original Distance) (b) NH (Original Distance) (c) CS

(d) S+CS (Original Distance) (e) NH+CS (Original Distance) (f) S (Learned Distance)

(g) NH (Learned Distance) (h) S+CS (Learned Distance) (i) NH+CS (Learned Distance)

Figure 6: Consistency index versus each internal validation index.

ACKNOWLEDGEMENTS

This work is supported by FCT “Fundação para a Ciência e a Tecnologia” under the project “LearningS” - PTDC/EEI-SII/2312/2012.

REFERENCES

Arbelaitz, O., Gurrutxaga, I., Muguerza, J., Prez, J. M., and Perona, I. (2013). An extensive comparative study of cluster validity indices. *Pattern Recognition*, 46(1):243 – 256.

Bache, K. and Lichman, M. (2013). UCI machine learning repository.

Basu, S. (2005). *Semi-supervised clustering: probabilistic models, algorithms and experiments*. PhD thesis, Austin, TX, USA. Supervisor-Mooney, Raymond J.

Basu, S., Banjeree, A., Mooney, E., Banerjee, A., and Mooney, R. J. (2004). Active semi-supervision for pairwise constrained clustering. In *In Proceedings of the 2004 SIAM International Conference on Data Mining (SDM-04*, pages 333–344.

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

Cohn, D., Caruana, R., and McCallum, A. (2003). Semi-supervised clustering with user feedback.

Davidson, I. and Ravi, S. (2005). Clustering with constraints feasibility issues and the k-means algorithm. In *2005 SIAM International Conference on Data Mining (SDM’05)*, pages 138–149, Newport Beach,CA.

Duarte, J. M. M., Fred, A. L. N., and Duarte, F. J. F. (2012). Evidence accumulation clustering using pairwise constraints. In Fred, A. L. N. and Filipe, J., editors, *KDIR 2012 - Proceedings of the International Conference on Knowledge Discovery and Information Retrieval,* *Barcelona, Spain, 4 - 7 October, 2012*, pages 293–299. SciTePress.

Fred, A. L. N. (2001). Finding consistent clusters in data partitions. In *Proceedings of the Second International Workshop on Multiple Classifier Systems*, MCS '01, pages 309–318, London, UK. Springer-Verlag.

Ge, R., Ester, M., Jin, W., and Davidson, I. (2007). Constraint-driven clustering. In *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 320–329, New York, NY, USA. ACM.

Hoi, S., Liu, W., Lyu, M., and Ma, W.-Y. (2006). Learning distance metrics with contextual constraints for image retrieval. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2072–2078.

Hubert, L. and Arabie, P. (1985). Comparing partitions. *Journal of Classification*, 2:193–218.

Jain, A. K. (2010). Data clustering: 50 years beyond k-means. *Pattern Recogn. Lett.*, 31(8):651–666.

MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In Cam, L. M. L. and Neyman, J., editors, *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press.

Rousseeuw, P. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65.

Sokal, R. R. and Michener, C. D. (1958). A statistical method for evaluating systematic relationships. *University of Kansas Scientific Bulletin*, 28:1409–1438.

Tung, A. K. H., Hou, J., and Han, J. (2000). Coe: Clustering with obstacles entities. a preliminary study. In *PADKK '00: Proceedings of the 4th Pacific-Asia Conference on Knowledge Discovery and Data Mining, Current Issues and New Applications*, pages 165–168, London, UK. Springer-Verlag.

Wagstaff, K. L. (2002). *Intelligent clustering with instance-level constraints*. PhD thesis, Ithaca, NY, USA. Chair-Claire Cardie.

Wang, X. and Davidson, I. (2010). Flexible constrained spectral clustering. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '10, pages 563–572, New York, NY, USA. ACM.