Handbook of Cluster Analysis (provisional top level file)

C. Hennig, M. Meila, F. Murtagh, R. Rocci (eds.)

August 19, 2014

Chapter 1

Semi-supervised Clustering

Anil Jain

Rong Jin

Radha Chitta

Department of Computer Science and Engineering

Michigan State University

East Lansing, MI, USA

Abstract

Clustering is an unsupervised learning problem whose objective is to find a partition of the given data. However, a major challenge in clustering is to define an appropriate objective function in order to to find an *optimal* partition that is useful to the user. To facilitate data clustering, it has been suggested that the user provide some supplementary information about the data (eg. pairwise relationships between few data points), which when incorporated in the clustering process, could lead to a better data partition. Semi-supervised clustering algorithms attempt to improve clustering performance by utilizing this supplementary information. In this chapter, we present

an overview of semi-supervised clustering techniques and describe some prominent algorithms in the literature. We also present several applications of semi-supervised clustering.

1.1 Introduction

Clustering is an inherently ill-posed problem due to its unsupervised nature. Consider a grouping of a subset of face images from the CMU Face data set [51][1], shown in Figure 1.1. These images have been clustered based on facial expression in Figure 1.1(a), and on the basis of presence of sunglasses in Figure 1.1(b). Both these partitions are equally valid, illustrating that the given data can be partitioned in many ways depending on user's intent and goal.

Most clustering algorithms seek a data partition that minimizes an objective function defined in terms of the data points and cluster labels. It is often the case that multiple partitions of the same data are equally good in terms of this objective function, making it difficult to determine the optimal data partition. Consider the two-dimensional data shown in Figure 1.2. If an algorithm such as K-means [40] is used to cluster the points into two clusters, then both the partitions shown in Figures 1.2(c) and 1.2(d) are equally good in terms of the *sum-of-squared-error* criterion[2]. Hence, additional information (constraints) is required to resolve this ambiguity and determine the partition sought by the user.

In many clustering applications, the user is able to provide some *side-information* besides the vector representation of data points (or the pairwise similarity between the data points). This side-information can be used to tune the clustering algorithm towards finding the data partition sought by the user. Semi-supervised clustering deals with mechanisms to obtain and incorporate this side-information in the clustering process to attain better clustering performance. Formally, given a data set $\mathcal{D} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ containing n points, side-information π, and a clustering algorithm $\mathcal{A}$, the objective of semi-supervised clustering is to augment $\mathcal{A}$ with π, and partition $\mathcal{D}$ into K clusters $\{\mathcal{C}_1, \ldots, \mathcal{C}_K\}$. It is expected that the resulting partition of $\mathcal{D}$ will be better than the partition obtained from $\mathcal{A}$ in the absence

[1] This data set is available in the UCI repository [52].

[2] Sum-of-squared-error is defined as the sum of the square of the distances between each point and its corresponding cluster center.

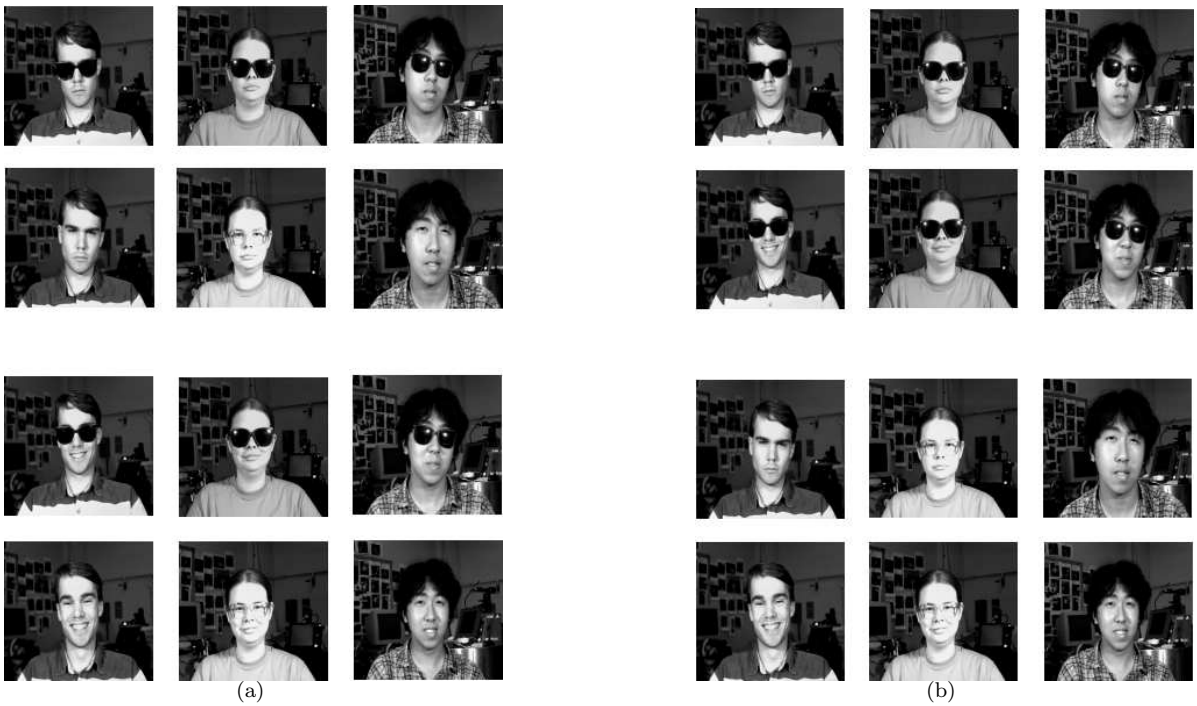


Figure 1.1: A clustering of a subset of face images from the CMU face data set [51]. It is possible to cluster these images in many ways, all of them equally valid. Faces have been clustered into two clusters based on the facial expression of the subjects in (a), and on the basis of whether or not the subjects are wearing sunglasses in (b). Without additional information from the user, it is not possible to determine which one is the correct or preferred partition.

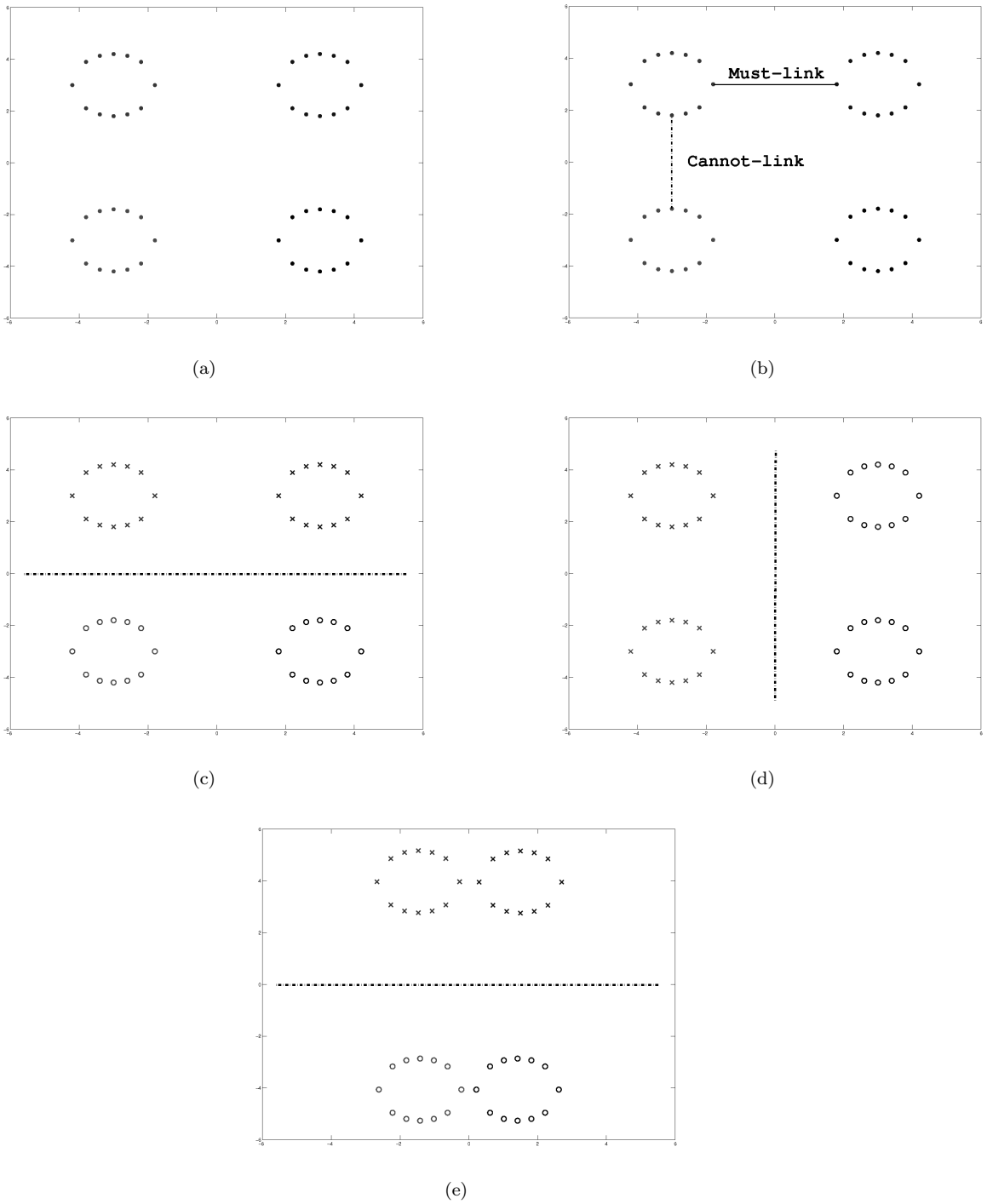


Figure 1.2: User-specified pairwise constraints. It is possible to partition the data points in (a) into two clusters in the following two ways: horizontally, as shown in (c), and vertically as shown in (d). While both of them yield the same sum-of-squared-error value, it is not possible to determine the desired partition, without additional information. Given the two pairwise constraints shown in (b), we can ascertain that the correct partition is the one in (c). A distance metric learning technique [77] for incorporating the pairwise constraints is illustrated in (e).

of side-information. In this chapter, two measures of clustering performance are employed: F-measure and Normalized Mutual Information. These are *external* measures adapted from information retrieval, which compare the data partition obtained from the clustering algorithm with the true class labels:

F-measure: A pair of points $(\mathbf{x}_i, \mathbf{x}_j)$ are said to be correctly paired if (i) they have the same cluster label, and they have the same class label (represented as true positive TP), or (ii) they have different cluster labels and different class labels (represented as true negative TN). Otherwise they are said to be paired incorrectly: False positive FP if they are fall in the same cluster but belong to different classes, and false negative FN if they are assigned to different clusters but belong to the same class. F-measure is defined as

$$\frac{2\,\#TP^2}{\#TP(2\,\#TP + \#FN + \#FP)},$$

where # represents the number of the corresponding quantity.

Normalized Mutual Information (NMI): Let n_i^c represent the number of data points that have been assigned to the i^{th} cluster ($1 \leq i \leq K$), n_j^p the number of data points from the j^{th} class ($1 \leq j \leq P$), and $n_{i,j}^{c,p}$ the number of data points from class j that have been assigned to the i^{th} cluster. NMI is defined as

$$\frac{\sum_{i=1}^{K}\sum_{j=1}^{P} n_{i,j}^{c,p} \log\left(n\frac{n_{i,j}^{c,p}}{n_i^c n_j^p}\right)}{\sqrt{\left(\sum_{i=1}^{K} n_i^c \log\frac{n_i^c}{n}\right)\left(\sum_{j=1}^{P} n_j^p \log\frac{n_j^p}{n}\right)}}.$$

There are two main issues that need to be addressed in semi-supervised clustering: (a) How is the side-information obtained and specified?, and (b) How can the side-information be used to improve the clustering performance? These questions are addressed in Sections 1.1.1 and 1.1.2.

1.1.1 Acquisition and expression of side-information

The most common form of expressing the side-information is pairwise must-link and cannot-link constraints [10]. A *must-link* constraint between two data points $\mathbf{x}_a$ and $\mathbf{x}_b$, denoted by $\mathcal{ML}(\mathbf{x}_a, \mathbf{x}_b)$, implies that the points $\mathbf{x}_a$ and $\mathbf{x}_b$ must be assigned the same cluster label. A *cannot-link* constraint, denoted by $\mathcal{CL}(\mathbf{x}_a, \mathbf{x}_b)$, implies that the points $\mathbf{x}_a$ and $\mathbf{x}_b$ should be assigned to different clusters. Figure 1.2(b) illustrates two such constraints applied to partition the data shown in Figure 1.2(a). Given the pairwise must-link and cannot-link constraints, we can bias the clustering algorithm towards obtaining the desired partition. Figure 1.3 shows the improvement in the clustering performance, measured in terms of the F-measure [50], as a result of providing pairwise constraints, on six benchmark data sets. All the semi-supervised clustering algorithms considered in [13], viz.: supervised-means, PCK-means, MK-means and MPCK-means, perform better than the K-means clustering algorithm, when provided with a sufficient number of pairwise constraints [13].

Pairwise constraints for data clustering occur naturally in many application domains. In applications involving graph clustering, such as social network analysis, the given edges in the graph indicate the pairwise relationships. Some protein data sets[3] contain information about co-occurring proteins, which can be viewed as must-link constraints during clustering [43]. In image segmentation, neighboring pixels are likely to be a part of the same homogeneous region in an image, whereas pixels which are far from each other tend to belong to different regions [47]. This fact can be used to generate pairwise constraints. Pairwise constraints can be derived from the domain knowledge and other external sources as well [36, 56]. For example, Wikipedia [2] was used to identify semantic relationships between documents in [36].

Although the side-information is, in general, expected to improve the clustering performance, inaccurate or conflicting pairwise constraints may actually degrade the clustering performance [21, 22, 23, 24]. For example, consider the scenario shown in Figure 1.4, where point pairs $(\mathbf{x}_1, \mathbf{x}_2)$ and $(\mathbf{x}_3, \mathbf{x}_4)$ are involved in must-link constraints, but $\mathbf{x}_2$ and its neighbors, and $\mathbf{x}_3$ and its neighbors are related by cannot-link constraints. These constraints lead

[3]The Database of Interacting Proteins [73].

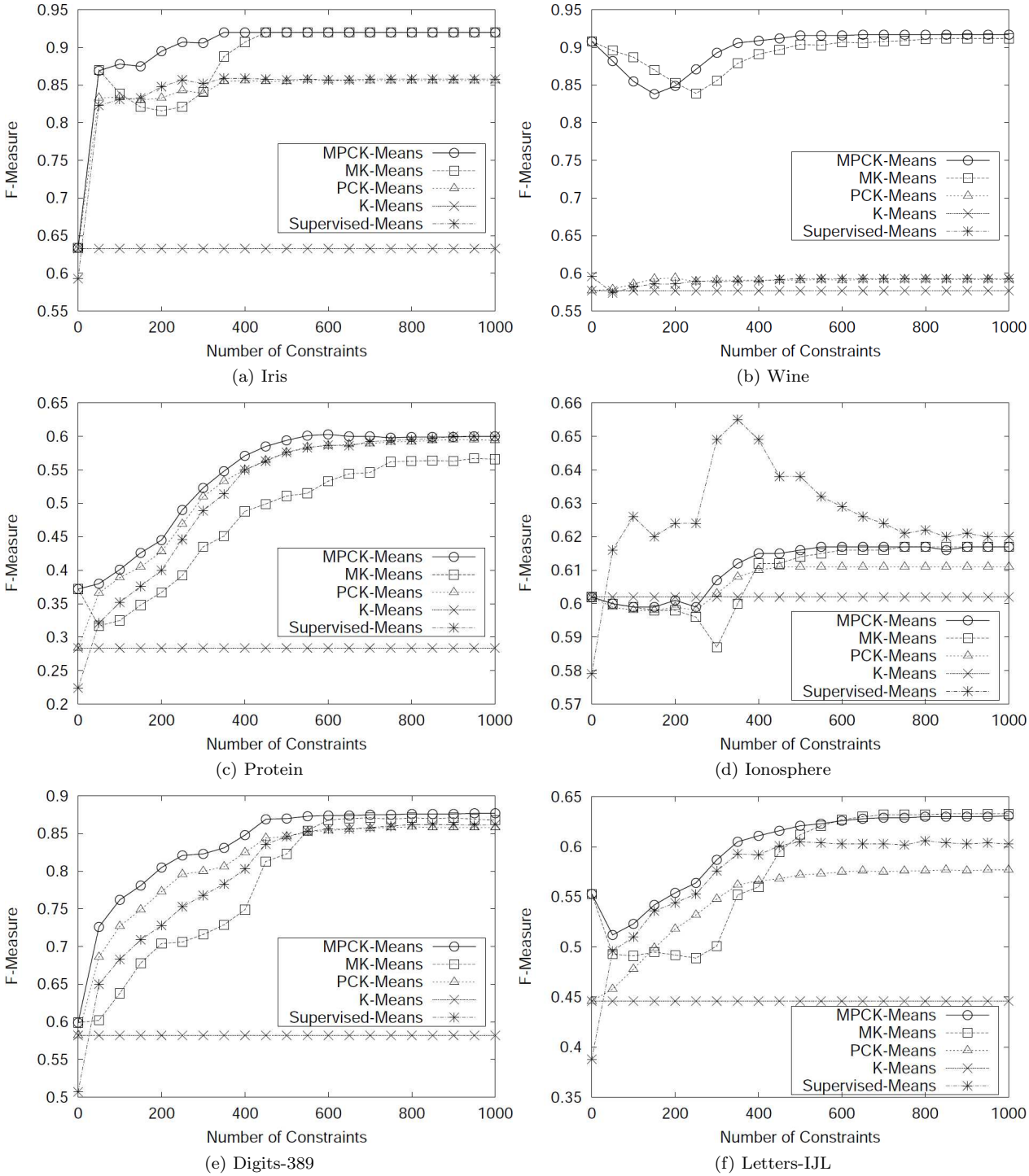


Figure 1.3: Improvement in clustering performance of the K-means algorithm using pairwise constraints on six benchmark data sets. Figures reproduced from Bilenko et al. [13].

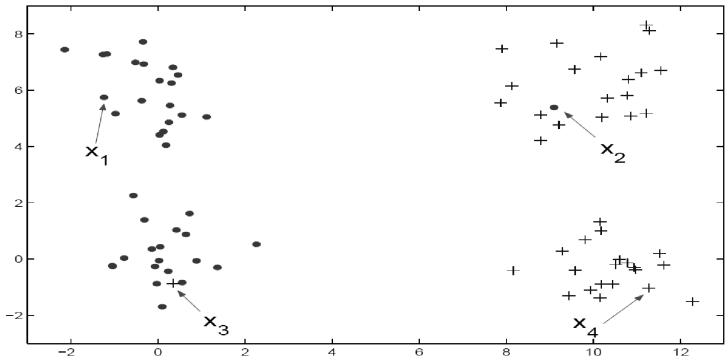


Figure 1.4: Pairwise constraints can lead to counter-intuitive solutions. Points pairs $(\mathbf{x}_1, \mathbf{x}_2)$ and $(\mathbf{x}_3, \mathbf{x}_4)$ are related by a must-link constraints. $\mathbf{x}_2$ and its neighbors, and $\mathbf{x}_3$ and its neighbors are related by cannot-link constraints. This yields a counter-intuitive clustering solution, where points $\mathbf{x}_2$ and $\mathbf{x}_3$ are assigned cluster labels that are different from those of their neighbors. Figure from [47].

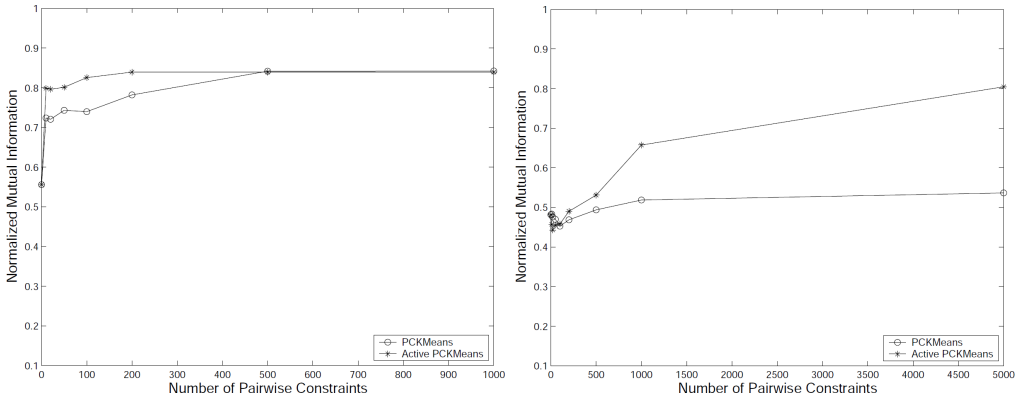
to a counter-intuitive clustering solution, where points $\mathbf{x}_2$ and $\mathbf{x}_3$ are assigned cluster labels that are different from those of their neighbors. This behavior can also be observed while clustering real data sets, as shown in Figure 1.3. Some semi-supervised clustering algorithms exhibit a small dip in the performance, for some number of constraints. In other words, the performance of semi-supervised clustering algorithms is not guaranteed to increase monotonically with the number of constraints. For example, in Figure 1.3(d), the accuracy of the MK-means algorithm on the Ionosphere data set is lower than that of the unsupervised K-means algorithm when the number of constraints is less than 400. This suggests that the nature of the constraints, not simply the number of constraints, is crucial for attaining performance gain from semi-supervised clustering.

In order to identify the most informative pairwise constraints, some studies [8, 38, 70] have focused on active learning [61], originally developed for semi-supervised classification [18]. Semi-supervised classification techniques based on active learning assume the presence of an oracle which can supply the class labels for points selected from a large pool of unlabeled points. Starting with a small set of labeled points, the oracle is iteratively queried for the labels of the points most useful for determining the classification model. The key idea behind active learning based semi-supervised clustering is to find the constraints that would be violated if the clustering algorithm was executed without supervision [3, 79]. Most active clustering techniques assume an oracle can answer queries involving pairwise constraints

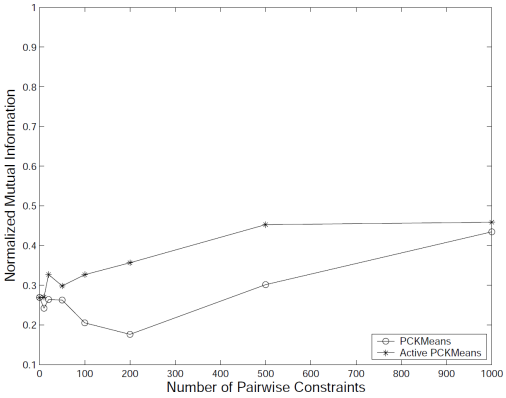
among data points. These algorithms differ in the manner in which the queries are made. For instance, the active PCK-means algorithm [8] aims at identifying the individual cluster boundaries. It first identifies pairs of data points which are farthest from each other, and queries the oracle until a pre-defined number of cannot-link constraints are obtained. It then queries the relationship between the points involved in the cannot-link constraints and their nearest neighbors to obtain must-link constraints. The active spectral clustering technique [70], on the other hand, iteratively refines the data partition by querying the pairwise relationship between the data points which leads to the largest change in the current partition towards the desired partition. Figure 1.5 compares the PCK-means algorithm, which assumes pairwise constraints are available *a priori*, with the active PCK-means algorithm on three data sets. The active PCK-means algorithm achieves better accuracy, measured in terms of the Normalized Mutual Information [45] with respect to the true cluster membership, with fewer number of constraints, demonstrating that active clustering is able to identify the most informative constraints [8].

Given that the most common form of specifying side-information to clustering algorithms is pairwise constraints, semi-supervised clustering is also referred to as **constrained clustering** to distinguish between semi-supervised clustering and general semi-supervised learning [14]. In its most popular form, semi-supervised learning involves using a large number of unlabeled examples along with the labeled training set to improve the learning efficiency. Figure 1.6 illustrates the spectrum of learning methodologies as we transition from supervised learning to unsupervised learning.

Besides pairwise constraints, other forms of side-information and constraints to obtain the desired clusters have also been studied in the literature. Class labels for a subset of the data set to be clustered can be used as the side-information [7, 25, 41]. In applications such as document clustering and multimedia retrieval, class labels are easily obtained through crowdsourcing [6] tools such as the Amazon Mechanical turk [1]. Class labels of subsets of data can be used to assign temporary labels to the remaining unlabeled data points by learning a classifier. These labels can then be employed to constrain the solution search space [25, 28]. The class labels can also be employed to initialize the clusters in addition to restricting the possible partitions [7].

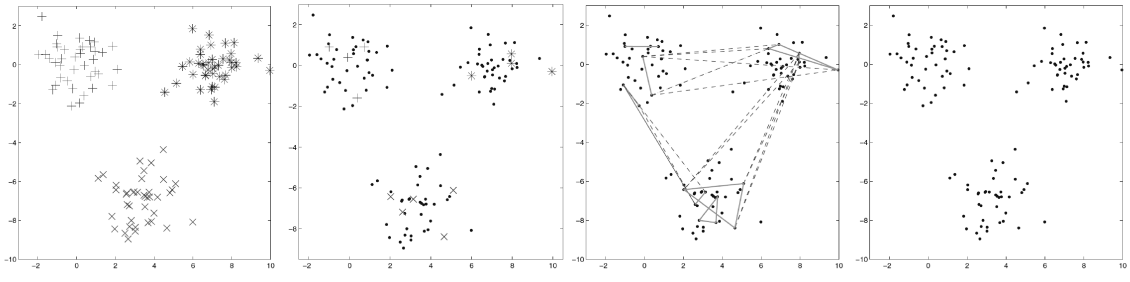


(a) Iris (b) Newsgroups



(c) Classic3

Figure 1.5: Better clustering results are achieved by employing active learning to obtain the most informative pairwise constraints. Figures reproduced from [8].



(a) Supervised (b) Partially labelled (c) Partially constrained (d) Unsupervised

Figure 1.6: Spectrum between supervised and unsupervised learning. Dots correspond to points without label information. Points with labels are denoted by pluses, asterisks and crosses, each representing a different class. In (c), the must-link and cannot-link constraints are denoted by solid and dashed lines, respectively. Figure from [46].

Triplet constraints deal with the relative distance between sets of three data points, and are less restrictive than pairwise constraints. A triplet constraint $(\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c)$ indicates that $\mathbf{x}_a$ is closer to $\mathbf{x}_b$ than $\mathbf{x}_a$ is to $\mathbf{x}_c$, which in turn, implies that $\mathbf{x}_a$ is more likely to form a cluster with $\mathbf{x}_b$ than with $\mathbf{x}_c$. These constraints are used to estimate the similarity between data points and thereby enhance the clustering performance [12, 37, 44]. The SSSVaD algorithm [44] uses the triplet constraints to learn the underlying dissimilarity measure. In [5, 78, 80], the triplet constraints are used to determine the order in which the agglomerative hierarchical clustering algorithm merges the clusters.

When the data set is sufficiently small in size, user feedback can also be employed to obtain the partition sought by the user [19, 30, 33, 35]. For instance, in [30], the user is iteratively presented with a possible partition of the data, and allowed to choose whether or not it is the desired partition. This feedback is used to eliminate undesired partitions that are similar to the one presented to the user. User supervision is employed to select the most informative features that lead to the desired clusters in [35].

1.1.2 Incorporation of side-information

Several mechanisms have been developed to exploit the side-information to achieve better clustering performance. They can be classified into two main categories: (i) methods based on constraining the solution space, and (ii) methods based on distance metric learning [10].

Solution space restriction

Most methods in this category deal with side-information in the form of pairwise constraints. They use the pairwise constraints to restrict the feasible data partitions when deciding the cluster assignment. The COP-Kmeans [68] and the SS-SOM [3] algorithms modify the cluster membership update phase of the K-means and the Self Organizing map algorithms respectively, to ensure that the data partitions are consistent with the given pairwise constraints. In the COP-Kmeans algorithm, the cluster centers are first initialized randomly. Each data point is then assigned to the nearest cluster center ensuring that no constraints are violated. The cluster centers are updated by finding the mean of the points assigned to the cluster, like

in the K-means algorithm. In [62], the generalized Expectation Maximization (EM) algorithm is modified such that only the mixture models that are compliant with the constraints are considered. These approaches treat the side-information as *hard constraints* and ensure that all the constraints are strictly satisfied. As mentioned before, such an approach may lead to counter-intuitive clustering solutions, as shown in Figure 1.4, and may even render the clustering problem infeasible.

A number of studies have used the side-information in the form of *soft constraints*. Instead of trying to satisfy all the constraints, the key idea behind such methods is to satisfy as many constraints as possible, and introduce a penalty term to account for constraints that cannot be satisfied [8, 11, 41, 46, 47]. In [47], the authors modified the mixture model for data clustering by redefining the data generation process through the introduction of hidden variables. In [46], a mean field approximation method was proposed to find appropriate data partition that is consistent with pairwise constraints. The pairwise constraints are enforced in the form of additional penalty terms in the objective function for clustering in spectral learning [41] and PCK-means [8].

Distance metric learning

The distance measure used to determine the dissimilarity between data points is crucial to the clustering process. Semi-supervised clustering methods which fall under this category attempt to find and apply a transformation to the data such that (a) the data points in must-link constraints are separated by small distances, and (b) data point in cannot-link constraints are separated by larger distances [77]. The distance between any two data points $\mathbf{x}_a$ and $\mathbf{x}_b$ is expressed as

$$d_M(\mathbf{x}_a, \mathbf{x}_b) = \|\mathbf{x}_a - \mathbf{x}_b\|_M^2 = (\mathbf{x}_a - \mathbf{x}_b)^\top M(\mathbf{x}_a - \mathbf{x}_b),$$

where M is the distance metric cast as a positive semi-definite matrix.

An example of distance metric learning is illustrated in Figure 1.2(e). The data points which satisfy must-link constraints move closer to each other, and the data points that satisfy cannot-link constraints move farther away. After learning this distance metric, any conventional clustering algorithm such as K-means or spectral clustering can be applied to

the resulting similarity matrix.

Techniques for distance metric learning have been studied extensively in the semi-supervised and unsupervised learning literature [77]. *Local* distance metric learning techniques only focus on constraints in local regions, and are typically used in semi-supervised classification [4, 32]. On the other hand, *global* distance metric learning methods consider all the pairwise constraints simultaneously [17, 31, 49, 74, 76]. For example, a convex optimization problem that minimizes the distance between points that are related by must-link constraints, and maximizes the distance between points that are related by cannot-link constraints, is solved to find the optimal metric in [74]. Techniques to learn non-linear distance metrics such as Bregman divergence were proposed in [20, 72]. Besides pairwise constraints, triplet constraints have also been employed for distance metric learning.

The idea of distance metric learning for semi-supervised clustering has been extended to learn the kernel representing the pairwise data similarity [26, 34]. Similar to distance metric learning, the kernel similarity function is modified to accommodate the given pairwise constraints, i.e., (a) data points in must-link relationships have large similarity, and (b) data points in cannot-link relationships have small similarity. The kernel similarity is modified by incorporating the constraints in the objective function in [9, 15, 43]. Non-parametric approaches for kernel learning are proposed in [34, 65] to learn the pairwise similarity measure.

Methods that combine the two approaches to semi-supervised clustering have also been proposed [9, 13]. One example is the MPCK-means [13] which performs both solution space restriction and distance metric learning. It performs better than the methods that employ only distance metric learning, or methods that only constrain the solution space.

Table 1.1 presents a summary of the major semi-supervised clustering techniques proposed in the literature. They are classified based on the form of the available side-information and the manner in which the side-information is acquired and incorporated.

Table 1.1: Summary of prominent semi-supervised clustering techniques

Type of side-information	**Side-information acquisition**	**Side-information incorporation**	**Examples**
Pairwise constraints	Prior knowledge	Constrain the solution space	PCK-Means [8], Spectral Learning [41], COPK-Means [68], [11], [46], [47], [48], [58], [62] [71], [75]
Pairwise constraints	Prior knowledge	Distance metric learning	HMRF-Kmeans [9], MPCK-means [13], SSKKM [43], BoostCluster [49], [15], [16], [20], [34], [65], [72], [74], [76]
Pairwise constraints	Active learning	Constrain the solution space	Active PCK-Means [8], Active spectral clustering [70], [3], [36], [38], [79]
Pairwise constraints	Active learning	Distance metric learning	[31]
Class labels	Prior knowledge, Crowdsourcing	Constrain the solution space	Constrained K-means [7], [25], [28]
Triplet constraints	Prior knowledge	Distance metric learning	SSSVaD [44], [78], [80]
User feedback		Constrain the solution space	[19], [20], [30], [33], [35]

1.2 Semi-supervised clustering algorithms

We describe three semi-supervised clustering algorithms in this section. These three algorithms are representatives of various approaches that have been used for semi-supervised clustering. The semi-supervised kernel K-means (SSKKM) [43] and BoostCluster [49] algorithms are based on distance metric learning. The SSKKM algorithm modifies the pairwise similarity between the data points using the must-link and cannot-link constraints. The BoostCluster algorithm projects the data into a subspace where points related by must-links are closer to each other, than the points related by cannot-links. In both these algorithms, the pairwise constraints are assumed to be available *a priori*. The active spectral clustering algorithm [70] obtains the constraints using the active learning mechanism. It then finds the solution by restricting the solution space of the spectral clustering algorithm.

1.2.1 Semi-supervised Kernel K-means

The semi-supervised kernel K-means algorithm, abbreviated as **SSKKM**, is based on the strategy of distance metric learning. It aims to enhance the accuracy of the K-means algorithm by constructing a kernel matrix, which incorporates the given pairwise constraints.

The objective of the K-means algorithm is to minimize the sum-of-squared-error, expressed as the following optimization problem:

$$\min \sum_{k=1}^{K} \sum_{\mathbf{x}_i \in \mathcal{C}_k} \|\mathbf{x}_i - \mathbf{c}_k\|^2, \tag{1.1}$$

where $\mathbf{c}_k$ is the cluster center of $\mathcal{C}_k, k = 1, 2, \ldots, K$.

Let $w_{a,b}$ be the cost of violating the constraint between data points $\mathbf{x}_a$ and $\mathbf{x}_b$. The set of pairwise must-link and cannot-link constraints, denoted by $\mathcal{ML}$ and $\mathcal{CL}$ respectively, are embedded in the optimization problem (1.1) as follows:

$$\min \sum_{k=1}^{K} \left(\sum_{\mathbf{x}_i \in \mathcal{C}_k} \|\mathbf{x}_i - \mathbf{c}_k\|^2 - \sum_{\substack{(\mathbf{x}_p, \mathbf{x}_q) \in \mathcal{ML} \\ \mathbf{x}_p, \mathbf{x}_q \in \mathcal{C}_k}} \frac{w_{p,q}}{n_k} + \sum_{\substack{(\mathbf{x}_r, \mathbf{x}_s) \in \mathcal{CL} \\ \mathbf{x}_r, \mathbf{x}_s \in \mathcal{C}_k}} \frac{w_{r,s}}{n_k} \right)$$

which is equivalent to

$$\min \sum_{k=1}^{K} \left(\sum_{\mathbf{x}_i, \mathbf{x}_j \in \mathcal{C}_k} \frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{n_k} - \sum_{\substack{(\mathbf{x}_p, \mathbf{x}_q) \in \mathcal{ML} \\ \mathbf{x}_p, \mathbf{x}_q \in \mathcal{C}_k}} \frac{2w_{p,q}}{n_k} + \sum_{\substack{(\mathbf{x}_r, \mathbf{x}_s) \in \mathcal{CL} \\ \mathbf{x}_r, \mathbf{x}_s \in \mathcal{C}_k}} \frac{2w_{r,s}}{n_k} \right) \tag{1.2}$$

where n_k is the number of data points assigned to cluster $\mathcal{C}_k$.
Let $U = [u_{k,j}]$ be the $K \times n$ normalized cluster membership matrix defined as

$$u_{k,j} = \begin{cases} 1/\sqrt{n_k} & \text{if } \mathbf{x}_j \in \mathcal{C}_k, \\ 0 & \text{otherwise.} \end{cases}$$

Also, define the Euclidean similarity matrix $E = [E_{a,b}]$ and the constraint matrix $W = [W_{a,b}]$

as

$$E_{a,b} = \|\mathbf{x}_a - \mathbf{x}_b\|^2 = \mathbf{x}_a^\top \mathbf{x}_a + \mathbf{x}_b^\top \mathbf{x}_b - 2\mathbf{x}_a^\top \mathbf{x}_b \tag{1.3}$$

and

$$W_{a,b} = \begin{cases} w_{a,b} & \text{if there is a must-link constraint between } \mathbf{x}_a \text{ and } \mathbf{x}_b, \\ -w_{a,b} & \text{if there is a cannot-link constraint between } \mathbf{x}_a \text{ and } \mathbf{x}_b, \\ 0 & \text{otherwise.} \end{cases}$$

The problem in (1.2) can be re-written as

$$\min \operatorname{trace}\left(U\left(E - 2W\right)U^\top\right),$$

which is equivalent to the trace maximization problem

$$\max \operatorname{trace}\left(U\mathcal{K}U^\top\right), \tag{1.4}$$

where the kernel matrix is given by $\mathcal{K} = S + W$ and the entries of the similarity matrix $S = [S_{a,b}]$ are given by $S_{a,b} = \mathbf{x}_a^\top \mathbf{x}_b$.

It has been shown that the optimization problem (1.4) can be solved by applying the kernel K-means algorithm [29, 59] on the kernel matrix $\mathcal{K}$. To ensure that $\mathcal{K}$ is positive semi-definite, it is diagonal shifted using a positive parameter σ. The method is illustrated in Figure 1.7.

$S_{a,b}$ was replaced by the RBF kernel $\kappa(\mathbf{x}_a, \mathbf{x}_b) = \exp(-\lambda\|\mathbf{x}_a - \mathbf{x}_b\|^2)$ to achieve better clustering performance [42]. An adaptive scheme to estimate the kernel width λ using the pairwise constraints was proposed in [75]. This was done by scaling the penalty terms in (1.2) by the kernel distance between the data points involved in the pairwise constraints.

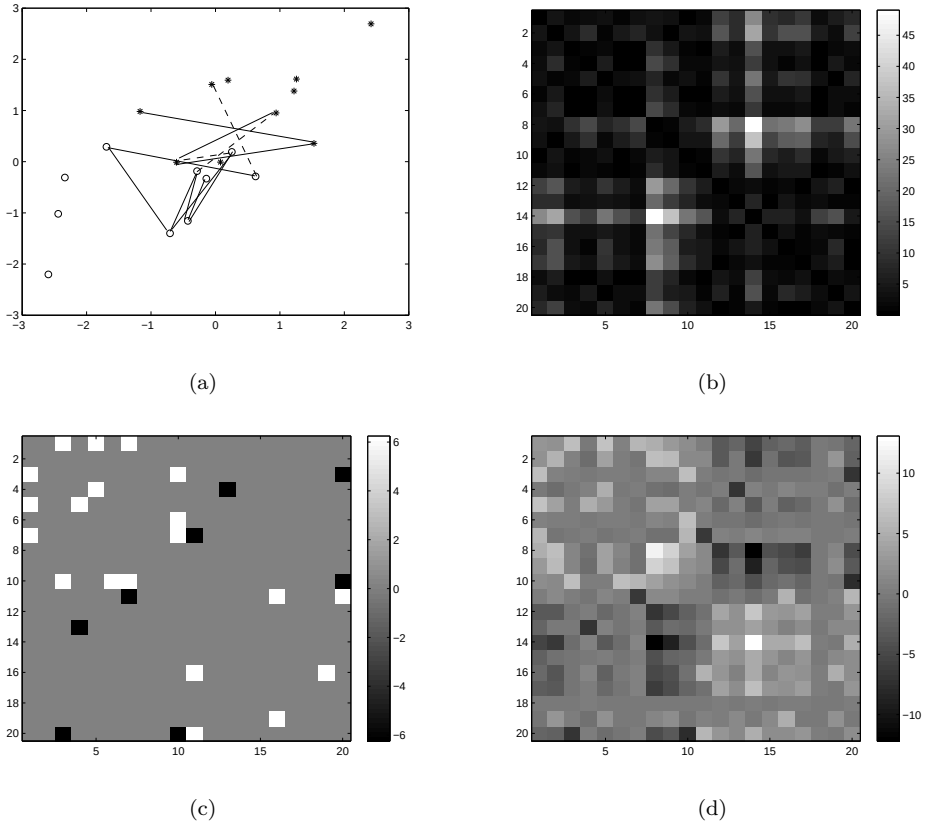


Figure 1.7: Illustration of SSKKM algorithm [43] on a toy two-dimensional example. Figure (a) shows the 2-D dataset containing 20 points, along with the must-link (solid lines) and cannot-link constraints (dashed lines). Figures (b)-(d) represent the Euclidean similarity matrix S, the constraint matrix W, and the kernel matrix $\mathcal{K} = S + W$ between the points. The points get clustered perfectly into the two groups indicated in Figure (a) on executing kernel K-means with $\mathcal{K}$ as the input, whereas the points do not get clustered as expected using the Euclidean similarity between them.

Algorithm SSKKM

Input

- Input data set $\mathcal{D} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$.
- Set of pairwise constraints $\mathcal{ML}$ and $\mathcal{CL}$.
- Constraint penalty matrix W.
- Kernel function $\kappa(\cdot, \cdot)$ to obtain the pairwise similarity matrix.
- Parameter σ for making the kernel matrix positive semi-definite.
- Number of clusters K.

1. Compute the $n \times n$ similarity matrix S using the kernel function $S_{i,j} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$.
2. Compute the kernel matrix $\mathcal{K} = S + W + \sigma I$.
3. Initialize the cluster membership of all the data points, ensuring no constraints are violated.
4. Repeat until convergence or a pre-defined maximum number of iterations is reached:
   (a) For each point $\mathbf{x}_i$ and cluster $\mathcal{C}_k$, compute distance

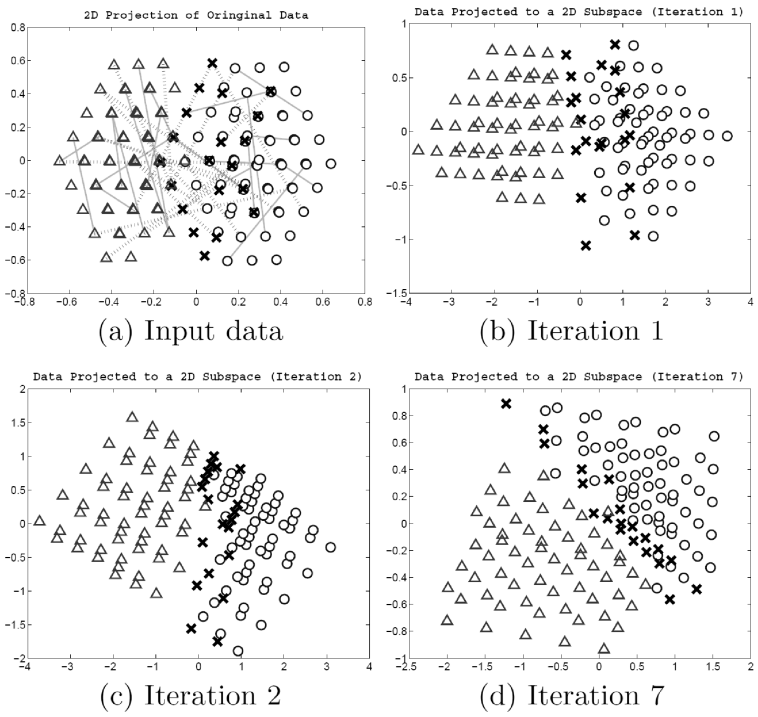
$$d(\mathbf{x}_i, \mathcal{C}_k) = \mathcal{K}_{i,i} - \frac{2\sum_{\mathbf{x}_j \in \mathcal{C}_k} \mathcal{K}_{i,j}}{n_k} + \frac{\sum_{\mathbf{x}_j, \mathbf{x}_l \in \mathcal{C}_k} \mathcal{K}_{j,l}}{n_k^2},$$

   where n_k is the number of points assigned to cluster $\mathcal{C}_k$.
   (b) Assign each point $\mathbf{x}_i$ to the cluster $\mathcal{C}_k^*$ which minimizes the distance $d(\mathbf{x}_i, \mathcal{C}_k)$, resolving ties arbitrarily.

Output Cluster memberships of the data points.

1.2.2 BoostCluster

The BoostCluster algorithm [49] follows the general boosting framework employed in data classification. Given a clustering algorithm $\mathcal{A}$, the BoostCluster algorithm iteratively modifies the input data representation, ensuring that the data points related by must-link constraints are more similar to each other than the data points related by cannot-link constraints. This idea is illustrated in Figure 1.8. A 2-dimensional representation of a subset of the *Balance scale* data set [39] from the UCI repository [52] is shown in Figure 1.8(a). The three



(a) Input data (b) Iteration 1

(c) Iteration 2 (d) Iteration 7

Figure 1.8: Illustration of BoostCluster algorithm [49] on the Balance Scale data set [39]. A 2-dimensional projection of the Balance Scale data set, obtained using PCA is shown in Figure (a). Figures (b)-(d) show the derived data representations based on the must-link and cannot-link constraints in iterations 1, 2 and 7 of the BoostCluster algorithm. Figure from [49].

clusters in this data are represented by triangles, crosses and circles. The must-link and cannot-link constraints are represented using the solid and dotted lines, respectively. The data is iteratively projected into subspaces such that the constraints are satisfied (see Figures 1.8(b)-(d)), thereby increasing the separation among the three clusters and enhancing the clustering performance.

The key steps in the BoostCluster algorithm involve (i) identifying the constraints that are not satisfied by the current partition, and (ii) projecting the data into a space where the points linked by must-link constraints are relatively closer to each other than the points linked by cannot-link constraints.

Let kernel matrix $\mathcal{K}$ denote the current similarity between the points. The objective of Boostcluster is to minimize the inconsistency between $\mathcal{K}$ and the pairwise constraints:

$$\mathcal{L}(\mathcal{K}) = \sum_{i,j=1}^{n} \sum_{a,b=1}^{n} \mathcal{ML}(\mathbf{x}_i, \mathbf{x}_j)\mathcal{CL}(\mathbf{x}_a, \mathbf{x}_b) \exp(\mathcal{K}_{a,b} - \mathcal{K}_{i,j}), \tag{1.5}$$

where $\mathcal{ML}(\mathbf{x}_i, \mathbf{x}_j) = 1$ if there is a must-link constraint between points $\mathbf{x}_i$ and $\mathbf{x}_j$ and 0 otherwise, and $\mathcal{CL}(\mathbf{x}_i, \mathbf{x}_j) = 1$ if there is a cannot-link constraint between points $\mathbf{x}_i$ and $\mathbf{x}_j$ and 0 otherwise.

Let $\Delta = [\Delta_{i,j}]$ represent the incremental similarity matrix inferred from the current partition of the data. Entry $\Delta_{i,j} = 1$ when points x_i and x_j belong to the same cluster in the current partition, and $\Delta_{i,j} = 0$ otherwise. The kernel is incrementally updated as $\mathcal{K}' = \mathcal{K} + \alpha\Delta$, where α is a weight parameter. Let matrix $T = [T_{i,j}]$, defined by

$$T_{i,j} = \frac{p_{i,j}}{\sum_{a,b=1}^{n} p_{a,b}} - \frac{q_{i,j}}{\sum_{a,b=1}^{n} q_{a,b}},$$

where $p_{i,j} = \mathcal{ML}(\mathbf{x}_i, \mathbf{x}_j) \exp(-\mathcal{K}_{i,j})$ and $q_{i,j} = \mathcal{CL}(\mathbf{x}_i, \mathbf{x}_i) \exp(\mathcal{K}_{i,j})$, represent the inconsistency between $\mathcal{K}$ and the constraints. A large positive (negative) value of the entry $T_{a,b}$ indicates that the corresponding entry in the similarity matrix $\mathcal{K}_{a,b}$ does not reflect the must-link (cannot-link) constraint between the data points $\mathbf{x}_a$ and $\mathbf{x}_b$. Using Jensen's inequality,

the loss $\mathcal{L}(\mathcal{K}')$ can be upper bounded by

$$\mathcal{L}(\mathcal{K}') \leq \mathcal{L}(\mathcal{K}) \times \left(\frac{(\exp(3\alpha) + \exp(-3\alpha) + 1) - (1 - \exp(-3\alpha)) \operatorname{trace}(T\Delta)}{3} \right)$$

In order to ensure that $\mathcal{L}(\mathcal{K}') \leq \mathcal{L}(\mathcal{K})$ in successive iterations of the algorithm, the upper bound is minimized with respect to α, which can be accomplished by maximizing the expression trace$(T\Delta)$. The incremental kernel matrix Δ is approximated as $\left(P^\top \mathcal{D}\right)^\top \left(P^\top \mathcal{D}\right)$, where P is a projection matrix that specifies the direction along which the data should be projected to obtain the new data representation. The optimal projection matrix is obtained as $P = (\sqrt{\lambda_1}\mathbf{v}_1, \sqrt{\lambda_2}\mathbf{v}_2, \cdots, \sqrt{\lambda_s}\mathbf{v}_s)$, where $\{(\lambda_i, \mathbf{v}_i)\}_{i=1}^s$ represent the top s (non-zero) eigenvalues and the corresponding eigenvectors of $\mathcal{D}T\mathcal{D}^\top$.

The BoostCluster algorithm falls into the same category of semi-supervised clustering algorithms as the SSKKM algorithm discussed in Section 1.2.1. It also modifies the similarity between the data points based on the given constraints, and hence yields similar clustering performance enhancement as the SSKKM algorithm. The advantage of BoostCluster over other semi-supervised clustering algorithms is that it can serve as a wrapper around any clustering algorithm. Hence, given the pairwise constraints, BoostCluster is able to boost the performance of any clustering algorithm. BoostCluster was used to enhance the performance of K-Means, single-link hierarchical clustering and spectral clustering algorithms on several data sets [49].

Algorithm BoostCluster

Input

- Input data set $\mathcal{D} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$.
- Clustering algorithm $\mathcal{A}$.
- Set of pairwise constraints $\mathcal{ML}$ and $\mathcal{CL}$.

  $\mathcal{ML}(\mathbf{x}_i, \mathbf{x}_j) = 1$ if there exists a must-link constraint between points $\mathbf{x}_i$ and $\mathbf{x}_j$, and 0 otherwise.

  $\mathcal{CL}(\mathbf{x}_i, \mathbf{x}_j) = 1$ if there exists a cannot-link constraint between points $\mathbf{x}_i$ and $\mathbf{x}_j$, and 0 otherwise.
- Number of principal eigenvectors s to be used for data projection.
- Number of clusters K.

1. Initialize $\mathcal{K}_{i,j} = 0 \ \forall i, j = 1, 2, \cdots, n$.
2. Repeat until all constraints are satisfied or a pre-defined maximum number of iterations is reached:
   (a) Compute the inconsistency matrix T given by

   $$T_{i,j} = \frac{p_{i,j}}{\sum_{a,b=1}^{n} p_{a,b}} - \frac{q_{i,j}}{\sum_{a,b=1}^{n} q_{a,b}},$$

   where $p_{i,j} = \mathcal{ML}(\mathbf{x}_i, \mathbf{x}_j) \exp(-\mathcal{K}_{i,j})$ and $q_{a,b} = \mathcal{CL}(\mathbf{x}_a, \mathbf{x}_b) \exp(\mathcal{K}_{a,b})$.

   (b) Construct the projection matrix $P = (\sqrt{\lambda_1}\mathbf{v}_1, \sqrt{\lambda_2}\mathbf{v}_2, \cdots, \sqrt{\lambda_s}\mathbf{v}_s)$, where $\{(\lambda_i, \mathbf{v}_i)\}_{i=1}^{s}$ represent the top s (non-zero) eigenvalues and the corresponding eigenvectors of $\mathcal{D}T\mathcal{D}^\top$.

   (c) Project the data $\mathcal{D}$ into the space spanned by the vectors in P and obtain the new representation of the data set $\widehat{\mathcal{D}} = P^\top \mathcal{D}$.

   (d) Run algorithm $\mathcal{A}$ with $\widehat{\mathcal{D}}$ as the input.

   (e) Update the similarity matrix $\mathcal{K}$ as $\mathcal{K} = \mathcal{K} + \alpha\Delta$, where

   $$\alpha = \frac{1}{2} \log \left(\frac{\sum_{i,j=1}^{n} p_{i,j}\delta(\Delta_{i,j}, 1)}{\sum_{i,j=1}^{n} p_{i,j}\delta(\Delta_{i,j}, 0)} \times \frac{\sum_{i,j=1}^{n} q_{i,j}\delta(\Delta_{i,j}, 0)}{\sum_{i,j=1}^{n} q_{i,j}\delta(\Delta_{i,j}, 1)} \right) \text{ and}$$

   $$\Delta_{i,j} = \begin{cases} 1 & \text{if } x_i \text{ and } x_j \text{ belong to the same cluster} \\ 0 & \text{otherwise} \end{cases}$$

3. Run algorithm $\mathcal{A}$ with $\mathcal{K}$ as kernel similarity matrix or with the data representation generated by the top $s+1$ eigenvectors of $\mathcal{K}$.

Output Cluster memberships of the data points.

1.2.3 Active spectral clustering

The spectral clustering algorithm [63] poses data clustering as a graph partitioning problem. The data points are represented as nodes in a graph and the pairwise similarities are represented as the weights on the edges connecting the vertices. The algorithm then finds the minimum weight normalized cut of the graph, and the resulting components of the graph form the clusters. The active spectral clustering algorithm [70] employs pairwise constraints to enhance the performance of spectral clustering. Instead of using the pairwise relationships between randomly sampled data point pairs, it employs the active learning mechanism to identify a subset of the most informative pairwise constraints.

Let S represent the $n \times n$ pairwise similarity matrix corresponding to the given set of n data points. The objective of spectral clustering, expressed as a graph *bi-partition* problem, is to find the solution to the following optimization problem:

$$\begin{aligned} \arg\min_{\mathbf{u}\in\mathbb{R}^n} \quad & \mathbf{u}^\top L\mathbf{u} \\ \text{s.t. } \mathbf{u}^\top D\mathbf{u} &= \mathbf{1}^\top D\mathbf{1} \\ \mathbf{u}^\top D\mathbf{1} &= 0, \end{aligned} \tag{1.6}$$

where $L = D - S$ is the graph Laplacian, D is the degree matrix given by

$$D_{i,j} = \begin{cases} \sum_{l=1}^{n} S_{i,l} & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases},$$

and $\mathbf{u}$ is the relaxed cluster membership vector. If the data is perfectly separable into two clusters, then $\mathbf{u} \in \{-1, 1\}^n$. Its i^{th} element $\mathbf{u}_i = 1$ if the point $\mathbf{x}_i$ belongs to the first cluster, and $\mathbf{u}_i = -1$ if it belongs to the second cluster.
The solution is given by the eigenvector associated with the second smallest eigenvalue of the normalized Laplacian $D^{-1/2}LD^{-1/2}$. A data partition containing K clusters is obtained through recursive bi-partitioning [63].

The active spectral clustering algorithm embeds the pairwise constraints in the form of a

constraint matrix W in the above optimization problem as follows:

$$\begin{aligned} \arg\min_{\mathbf{u}\in\mathbb{R}^n} \quad & \mathbf{u}^\top L\mathbf{u} \\ \text{s.t. } \mathbf{u}^\top D\mathbf{u} &= \mathbf{1}^\top D\mathbf{1} \\ \mathbf{u}^\top W\mathbf{u} &\geq \alpha, \end{aligned} \tag{1.7}$$

where α is a user-defined parameter, indicating how well the constraints in W are satisfied. A heuristic for selecting a suitable value for α is described in [71]. To obtain the constraint matrix, the active spectral clustering algorithm uses the active learning scheme. It assumes the presence of an oracle which has access to the true pairwise relationship matrix $W^* = \mathbf{u}^*\mathbf{u}^{*\top}$, where $\mathbf{u}^*$ is the desired data partition. The objective is to minimize the difference between the solution $\mathbf{u}$ obtained from the spectral clustering algorithm and $\mathbf{u}^*$ by querying the oracle for entries from W^*. The problem (1.7) is solved using the algorithm proposed in [71].

This active clustering mechanism is shown to perform better than methods which assume that the pairwise constraints are available *a priori*. Its only drawback is the high computational complexity of solving the eigenvalue problem. Fast approximate eigendecomposition techniques [64, 66] can be used to mitigate this issue.

Algorithm Active Spectral Clustering

Input

- Input data set $\mathcal{D} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$.
- Oracle which has access to the true pairwise relationships W^*.
- Number of clusters K.
- Parameter α.

1. Compute the similarity matrix $S = [d(\mathbf{x}_i, \mathbf{x}_j)]_{n \times n}$.
2. Compute the graph Laplacian $L = D - S$, where
$$D_{i,j} = \begin{cases} \sum_{l=1}^{n} S_{i,l} & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases}$$
3. Initialize the constraint matrix $W = \mathbf{0}$.
4. Repeat until convergence
   (a) Find the eigenvector $\mathbf{u}$ of $L\mathbf{u} = \lambda(W - \alpha\mathbf{I})\mathbf{u}$, which is associated with a positive eigenvalue and minimizes $\mathbf{u}^\top L\mathbf{u}$.
   (b) Find the singular vector $\bar{\mathbf{u}}$ corresponding to the largest singular value of W.
   (c) Compute the rank one approximation of W as $\bar{W} = \bar{\mathbf{u}}\bar{\mathbf{u}}^\top$.
   (d) Calculate the probability $p_{i,j}$ that the data points $\mathbf{x}_i$ and $\mathbf{x}_j$ are related by a must-link constraint, given by
$$p_{i,j} = \frac{1 + \min\{1, \max\{-1, \bar{W}_{i,j}\}\}}{2}.$$
   (e) Solve
$$(r, s) = \arg \max_{(i,j)|W_{i,j}=0} \quad p_{i,j}(\mathbf{u}_i\mathbf{u}_j^\top - 1)^2 + (1 - p_{i,j})(\mathbf{u}_i\mathbf{u}_j^\top + 1)^2.$$
   (f) Query the oracle for the entry $W^*_{r,s}$, and set $W_{r,s}$ and $W_{s,r}$ equal to $W^*_{r,s}$.

Output Cluster memberships of the data points.

1.3 Applications

Semi-supervised clustering has been successfully applied in several fields including bioinformatics, medical diagnosis, marketing, social network analysis, and web mining. Prior knowledge has been used to generate the side-information to enhance the clustering performance. Some of the applications are described below:

- **Character recognition:** Semi-supervised clustering was employed to decipher heavily degraded characters in historical typewritten documents in [56]. Due to various problems such as discoloration, aging, and disintegration of portions of documents, commercial OCR systems fail to recognize a majority of the characters. The documents were segmented down to the glyph[4] level, and each glyph was represented by a set of features such as width, height and the ratio of the number of black pixels to the number of white pixels. The glyphs were then clustered using the MPCK-means semi-supervised clustering algorithm [13]. Pairwise constraints were generated from typography related domain knowledge. For example, characters with very different aspect ratios were related by cannot-link constraints. Constraints were also obtained through pre-clustering of the glyph images. Figure 1.9 shows a plot of the clustering performance (F-measure [50]) of K-means, FineReader OCR engine and semi-supervised MPCK-means as a function of the number of constraints. The plots show that the availability of pairwise constraints improves the character recognition performance. The best performance was achieved using a set of constraints containing 731 must-link constraints and $2,091$ cannot-link constraints.

- **Image Segmentation:** Image segmentation is an important problem in computer vision. The goal is to identify homogeneous regions in an image whose pixels share similar visual patterns (eg. color and texture). Figure 1.10(b) shows a segmentation of a Mondrian image, consisting of five textured regions, that was obtained using the mean field approximation technique [46]. The segmented image in Figure 1.10(b) does not capture the true textured regions in Figure 1.10(a) very well. In order to generate the pairwise constraints, the image was divided into grids; segment labels for the grids

[4] A glyph represents a character or a symbolic figure.

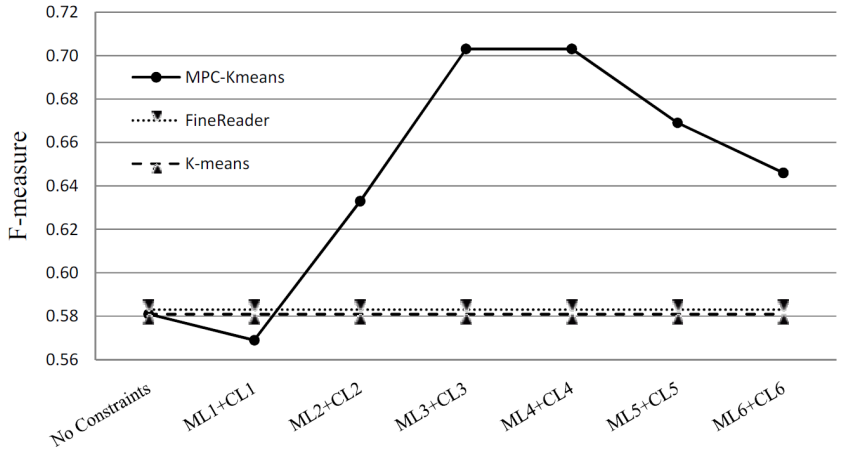


Figure 1.9: Character recognition using semi-supervised clustering [56]. ML1-ML5 and CL1-CL5 represent sets of must-link and cannot-link constraints, respectively with increasing number of constraints. Best performance is achieved using the set ML3 + CL3, containing 731 must-link constraints and 2,091 cannot-link constraints.

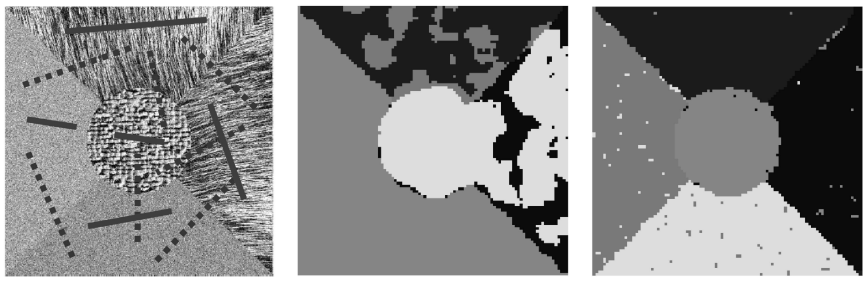


Figure 1.10: Image segmentation using semi-supervised clustering [46].

were obtained from the ground truth and converted to pairwise constraints. With 10% of the grids in pairwise constraints, the true textured regions were very well identified through semi-supervised clustering, as shown in Figure 1.10(c).

Image segmentation algorithms based on semi-supervised clustering have been employed in various applications such as medical imaging [27, 57], and remote sensing [67].

- **Document clustering:** Semi-supervised clustering has been very useful in document clustering. Semi-supervised Non-negative Matrix factorization (SS-NMF) [15] and its variants supplied with pairwise relationships between the documents were used to cluster documents in [15] and [16]. In [35], users were allowed to label discriminative features in addition to specifying pairwise constraints. This is useful in scenarios where the

user desires clusters based on specific attributes. For example, a set of articles related to sports may be organized by sport or by country. The users identified words which described the topic of the document, when presented with a subset of documents for labeling. This information was incorporated in the K-Means algorithm to obtain the desired document clusters.

- **Gene expression data analysis:** Semi-supervised clustering techniques have been popular in the domain of gene expression analysis [53, 54, 55]. In addition to pairwise constraints, interval constraints which define the spatial and temporal cluster boundaries, are employed. Genes which are known to have the same function in a biological process are related by must-link constraints. Interval constraints relate genes which are close to each other in terms of sampling time and/or spatial position in the DNA sequence. These constraints are easily attainable and aid biologists in capturing interesting patterns in the data. In [60], the yeast gene expression data was augmented with labels for a subset of the data. These labels were used to generate pairwise constraints, which were then used in the semi-supervised EM algorithm [46] to obtain better clusters.

1.4 Conclusions and open problems

Semi-supervised clustering is a useful mechanism to integrate side-information or prior knowledge in the clustering algorithm to find desired clusters in a data set. We have described different ways in which side-information can be utilized in clustering. Pairwise must-link and cannot-link constraints are the most common form of specifying the side-information. Between them, experimental results suggest that must-link constraints are more useful than cannot-link constraints [65]. The side-information can be incorporated in two ways: constraining the set of partitions that can be found by the clustering algorithm, and learning a distance metric which takes the constraints into account. Many semi-supervised algorithms have been developed and studied in the literature, and it continues to be a thriving field of study.

We have seen that the accuracy of the constraints is crucial to the semi-supervised clus-

tering performance. A major challenge in semi-supervised clustering is identifying the most useful constraints, while minimizing user effort [69]. Though active clustering alleviates this issue to some extent, it may not always lead to the desired solution. In some scenarios, only partial information may be available, and it may not be feasible to determine the pairwise relationship accurately. In [28], class labels are associated with a confidence rating and used as side-information. Similar mechanisms for assigning confidence measures to the pairwise constraints and incorporating them in the semi-supervised clustering algorithm need to be developed.

Index

References

[1] Amazon mechanical turk. https://www.mturk.com/.

[2] Wikipedia. http://www.wikipedia.org/.

[3] K. Allab and K. Benabdeslem. Constraint selection for semi-supervised topological clustering. *Machine Learning and Knowledge Discovery in Databases*, pages 28–43, 2011.

[4] C.G. Atkeson, A.W. Moore, and S. Schaal. Locally weighted learning. *Artificial Intelligence Review*, 11(1):11–73, 1997.

[5] K. Bade and A. Nürnberger. Creating a cluster hierarchy under constraints of a partially known hierarchy. In *Proceedings of the SIAM International Conference on Data Mining*, pages 13–24, 2008.

[6] G. Barbier, R. Zafarani, H. Gao, G. Fung, and H. Liu. Maximizing benefits from crowdsourced data. *Computational and Mathematical Organization Theory*, 2011.

[7] S. Basu, A. Banerjee, and R. Mooney. Semi-supervised clustering by seeding. In *Proceedings of the International Conference on Machine Learning*, pages 19–26, 2002.

[8] S. Basu, A. Banerjee, and R.J. Mooney. Active semi-supervision for pairwise constrained clustering. In *Proceedings of the SIAM International Conference on Data Mining*, pages 333–344, 2004.

[9] S. Basu, M. Bilenko, A. Banerjee, and R.J. Mooney. Probabilistic semi-supervised clustering with constraints. *Semi-supervised Learning*, pages 71–98, 2006.

[10] S. Basu, I. Davidson, and K.L. Wagstaff. *Constrained clustering: Advances in Algorithms, Theory, and Applications*. Chapman & Hall/CRC, 2009.

[11] R. Bekkerman and M. Sahami. Semi-supervised clustering using combinatorial mrfs. In *Proceedings of the ICML Workshop on Learning in Structured Output Spaces*, 2006.

[12] J. Bi, D. Wu, L. Lu, M. Liu, Y. Tao, and M. Wolf. Adaboost on low-rank psd matrices for metric learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2617–2624, 2011.

[13] M. Bilenko, S. Basu, and R.J. Mooney. Integrating constraints and metric learning in semi-supervised clustering. In *Proceedings of the International Conference on Machine Learning*, pages 81–88, 2004.

[14] O. Chapelle, B. Schölkopf, A. Zien, et al. *Semi-supervised Learning*, volume 2. MIT Press Cambridge, MA, 2006.

[15] Y. Chen, M. Rege, M. Dong, and J. Hua. Incorporating user provided constraints into document clustering. In *Proceedings of the International Conference on Data Mining*, pages 103–112, 2007.

[16] Y. Chen, L. Wang, and M. Dong. Semi-supervised document clustering with simultaneous text representation and categorization. *Machine Learning and Knowledge Discovery in Databases*, pages 211–226, 2009.

[17] H. Cheng, K.A. Hua, and K. Vu. Constrained locally weighted clustering. In *Proceedings of the VLDB Endowment*, volume 1, pages 90–101, 2008.

[18] D. Cohn, L. Atlas, and R. Ladner. Improving generalization with active learning. *Machine Learning*, 15(2):201–221, 1994.

[19] D. Cohn, R. Caruana, and A. McCallum. Semi-supervised clustering with user feedback. *Constrained Clustering: Advances in Algorithms, Theory, and Applications*, 4(1):17, 2003.

[20] David Cohn, Rich Caruana, and Andrew Mccallum. Semi-supervised clustering with user feedback. Technical report, University of Texas at Austin, 2003.

[21] I. Davidson and S. Ravi. Towards efficient and improved hierarchical clustering with instance and cluster level constraints. Technical report, University of Albany, 2005.

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

pages 138–149, 2005.
[23] I. Davidson and SS Ravi. Identifying and generating easy sets of constraints for clustering. In *Proceedings of the National Conference on Artificial Intelligence*, pages 336–341, 2006.
[24] I. Davidson, K. Wagstaff, and S. Basu. Measuring constraint-set utility for partitional clustering algorithms. In *Proceedings of the European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 115–126, 2006.
[25] A. Demiriz, K.P. Bennett, and M.J. Embrechts. Semi-supervised clustering using genetic algorithms. *Artificial neural networks in Engineering*, pages 809–814, 1999.
[26] C. Domeniconi, J. Peng, and B. Yan. Composite kernels for semi-supervised clustering. *Knowledge and Information Systems*, 28(1):99–116, 2011.
[27] Roman Filipovych, Susan M Resnick, and Christos Davatzikos. Semi-supervised cluster analysis of imaging data. *NeuroImage*, 54(3):2185–2197, 2011.
[28] J. Gao, P.N. Tan, and H. Cheng. Semi-supervised clustering with partial background information. In *Proceedings of the SIAM International Conference on Data Mining*, pages 487–491, 2006.
[29] M. Girolami. Mercer kernel-based clustering in feature space. *IEEE Transactions on Neural Networks*, 13(3):780–784, 2002.
[30] D. Gondek. Clustering with model-level constraints. In *Proceedings of the SIAM International Conference on Data Mining*, pages 126–137, 2005.
[31] M. Guillaumin, J. Verbeek, and C. Schmid. Multiple instance metric learning from automatically labeled bags of faces. In *Proceedings of the European Conference on Computer Vision*, pages 634–647, 2010.
[32] C.D.D. Gunopulos. Adaptive nearest neighbor classification using support vector machines. *Advances in Neural Information Processing Systems*, 14, 2001.
[33] M. Halkidi, D. Gunopulos, N. Kumar, M. Vazirgiannis, and C. Domeniconi. A framework for semi-supervised learning based on subjective and objective clustering criteria. In *Proceedings of the International Conference on Data Mining*, pages 637–640, 2005.
[34] S.C.H. Hoi, R. Jin, and M.R. Lyu. Learning nonparametric kernel matrices from pairwise constraints. In *Proceedings of the International Conference on Machine Learning*, pages 361–368, 2007.
[35] Y. Hu, E.E. Milios, and J. Blustein. Interactive feature selection for document clustering. In *Proceedings of the ACM Symposium on Applied Computing*, pages 1143–1150, 2011.
[36] A. Huang, D. Milne, E. Frank, and I.H. Witten. Clustering documents with active learning using wikipedia. In *Proceedings of the International Conference on Data Mining*, pages 839–844, 2008.
[37] K. Huang, R. Jin, Z. Xu, and C.L. Liu. Robust metric learning by smooth optimization. In *Proceeding of the Conferene on Uncertainity in Artificial Intelligence*, pages 244–251, 2010.
[38] R. Huang and W. Lam. Semi-supervised document clustering via active learning with pairwise constraints. In *Proceedings of the International Conference on Data Mining*, pages 517–522, 2007.
[39] Time Hume. Balance scale data set. http://archive.ics.uci.edu/ml/datasets/Balance Scale, 1994.
[40] A.K. Jain. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8):651–666, 2010.
[41] K. Kamvar, S. Sepandar, K. Klein, D. Dan, M. Manning, and C. Christopher. Spectral learning. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 2003.
[42] D.W. Kim, K.Y. Lee, D. Lee, and K.H. Lee. Evaluation of the performance of clustering algorithms in kernel-induced feature space. *Pattern Recognition*, 38(4):607–611, 2005.
[43] B. Kulis, S. Basu, I. Dhillon, and R. Mooney. Semi-supervised graph clustering: A

kernel approach. In *Proceedings of the International Conference on Machine Learning*, pages 457–464, 2005.

[44] N. Kumar and K. Kummamuru. Semi-supervised clustering with metric learning using relative comparisons. *IEEE Transactions on Knowledge and Data Engineering*, 20(4):496–503, 2008.

[45] T.O. Kvalseth. Entropy and correlation: Some comments. *IEEE Transactions on Systems, Man and Cybernetics*, 17(3):517–519, 1987.

[46] T. Lange, M.H.C. Law, A.K. Jain, and J.M. Buhmann. Learning with constrained and unlabelled data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 731–738, 2005.

[47] M.H.C. Law, A. Topchy, and A.K. Jain. Model-based clustering with probabilistic constraints. In *Proceedings of SIAM International Conference on Data Mining*, pages 641–645, 2005.

[48] T. Li, C. Ding, and M.I. Jordan. Solving consensus and semi-supervised clustering problems using nonnegative matrix factorization. In *Proceedings of the International Conference on Data Mining*, pages 577–582, 2007.

[49] Y. Liu, R. Jin, and A.K. Jain. Boostcluster: Boosting clustering by pairwise constraints. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, pages 450–459, 2007.

[50] C.D. Manning, P. Raghavan, and H. Schutze. *Introduction to Information Retrieval*, volume 1. 2008.

[51] T.M. Mitchell. Cmu face images data set. http://archive.ics.uci.edu/ml/datasets/CMU Face Images, 1999.

[52] DJ Newman, S. Hettich, CL Blake, CJ Merz, and DW Aha. Uci repository of machine learning databases. http://archive.ics.uci.edu/ml/datasets.html, 1998.

[53] R. Pensa, C. Robardet, and J.F. Boulicaut. Towards constrained co-clustering in ordered 0/1 data sets. *Foundations of Intelligent Systems*, pages 425–434, 2006.

[54] R.G. Pensa and J.F. Boulicaut. Constrained co-clustering of gene expression data. In *Proceedings SIAM International Conference on Data Mining*, pages 25–36, 2008.

[55] R.G. Pensa, C. Robardet, and J.F. Boulicaut. Constraint-driven co-clustering of 0/1 data. *Constrained Clustering: Advances in Algorithms, Theory and Applications*, pages 145–170, 2008.

[56] S. Pletschacher, J. Hu, and A. Antonacopoulos. A new framework for recognition of heavily degraded characters in historical typewritten documents based on semi-supervised clustering. In *Proceedings of the International Conference on Document Analysis and Recognition*, pages 506–510, 2009.

[57] Annemie Ribbens, Frederik Maes, Dirk Vandermeulen, and Paul Suetens. Semisupervised probabilistic clustering of brain mr images including prior clinical information. In *Proceedings of the Medical Computer Vision Workshop on Recognition Techniques and Applications in Medical Imaging*, pages 184–194. 2011.

[58] C. Ruiz, M. Spiliopoulou, and E. Menasalvas. Density-based semi-supervised clustering. *Data Mining and Knowledge Discovery*, 21(3):345–370, 2010.

[59] B. Scholkopf, A. Smola, and K.R. Muller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1314, 1996.

[60] A. Schönhuth, IG Costa, and A. Schliep. Semi-supervised clustering of yeast gene expression data. *Cooperation in Classification and Data Analysis*, pages 151–159, 2009.

[61] Burr Settles. Active learning literature survey. Computer Science Technical Report 1648, University of Wisconsin–Madison, 2009.

[62] N. Shental, A. Bar-Hillel, T. Hertz, and D. Weinshall. Computing gaussian mixture models with em using equivalence constraints. *Advances in Neural Information Processing Systems*, 16:465–472, 2004.

[63] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on*

Pattern Analysis and Machine Intelligence, 22(8):888–905, 2002.
[64] G.L.G. Sleijpen and H.A. Van der Vorst. A jacobi-davidson iteration method for linear eigenvalue problems. *SIAM Review*, pages 267–293, 2000.
[65] M. Soleymani Baghshah and S. Bagheri Shouraki. Kernel-based metric learning for semi-supervised clustering. *Neurocomputing*, 73:1352–1361, 2010.
[66] F. Tisseur and K. Meerbergen. The quadratic eigenvalue problem. *SIAM Review*, pages 235–286, 2001.
[67] Miguel Torres, Marco Moreno, Rolando Menchaca-Mendez, Rolando Quintero, and Giovanni Guzman. Semantic supervised clustering approach to classify land cover in remotely sensed images. In *Signal Processing and Multimedia*, pages 68–77. 2010.
[68] K. Wagstaff, C. Cardie, S. Rogers, and S. Schrödl. Constrained k-means clustering with background knowledge. In *Proceedings of the International Conference on Machine Learning*, pages 577–584, 2001.
[69] K.L. Wagstaff. Value, cost, and sharing: Open issues in constrained clustering. In *Proceedings of the International Conference on Knowledge Discovery in Inductive Databases*, pages 1–10, 2006.
[70] X. Wang and I. Davidson. Active spectral clustering. In *Proceedings of the International Conference on Data Mining*, pages 561–568, 2010.
[71] Xiang Wang and Ian Davidson. Flexible constrained spectral clustering. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, pages 563–572, 2010.
[72] L. Wu, S.C.H. Hoi, R. Jin, J. Zhu, and N. Yu. Learning bregman distance functions for semi-supervised clustering. *IEEE Transactions on Knowledge and Data Engineering*, pages 478–491, 2010.
[73] I. Xenarios, D.W. Rice, L. Salwinski, M.K. Baron, E.M. Marcotte, and D. Eisenberg. Dip: The database of interacting proteins. *Nucleic Acids Research*, 28(1):289–291, 2000.
[74] E.P. Xing, A.Y. Ng, M.I. Jordan, and S. Russell. Distance metric learning with application to clustering with side-information. *Advances in Neural Information Processing Systems*, 15:505–512, 2002.
[75] B. Yan and C. Domeniconi. An adaptive kernel method for semi-supervised clustering. In *Proceedings of the European Conference on Machine Learning*, pages 521–532, 2006.
[76] S. Yan, H. Wang, D. Lee, and C. Giles. Pairwise constrained clustering for sparse and high dimensional feature spaces. *Advances in Knowledge Discovery and Data Mining*, pages 620–627, 2009.
[77] L. Yang and R. Jin. Distance metric learning: A comprehensive survey. Technical report, Michigan State Universiy, 2006.
[78] H. Zhao and Z. Qi. Hierarchical agglomerative clustering with ordering constraints. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, pages 195–199, 2010.
[79] W. Zhao, Q. He, H. Ma, and Z. Shi. Effective semi-supervised document clustering via active learning with instance-level constraints. *Knowledge and Information Systems*, pages 1–19, 2011.
[80] L. Zheng and T. Li. Semi-supervised hierarchical clustering. In *Proceedings of the International Conference on Data Mining*, pages 982–991, 2011.