

A Clustering Framework Based on Subjective and Objective Validity Criteria

M. HALKIDI

Athens University of Economics and Business

D. GUNOPULOS

University of Athens

M. VAZIRGIANNIS

INRIA/FUTURS and Athens University of Economics and Business

N. KUMAR

University of California, Riverside

and

C. DOMENICONI

George Mason University

18

Clustering, as an unsupervised learning process is a challenging problem, especially in cases of high-dimensional datasets. Clustering result quality can benefit from user constraints and objective validity assessment. In this article, we propose a semisupervised framework for learning the weighted Euclidean subspace, where the best clustering can be achieved. Our approach capitalizes on: (i) user constraints; and (ii) the quality of intermediate clustering results in terms of their structural properties. The proposed framework uses the clustering algorithm and the validity measure as its parameters. We develop and discuss algorithms for learning and tuning the weights of contributing dimensions and defining the “best” clustering obtained by satisfying user constraints. Experimental results on benchmark datasets demonstrate the superiority of the proposed approach in terms of improved clustering accuracy.

The work of D. Gunopulos is supported by NSF and the work of M. Halkidi is funded by the Marie Curie Outgoing Int. Fellowship (MOIF-CT-2004-509920) from EU Commission. M. Vazirgiannis’s work is supported by the NGWeMiS Marie Curie Intra-European Fellowship (MEIF-CT-2005-011549).

Authors’ addresses: M. Halkidi (contact author), Athens University of Economics and Business, 76, Patission Str. GR10434 Athens-Greece; email: mhalk@aueb.gr; D. Gunopulos, Department of Informatics and Telecommunications, National and Kapodistrian University of Athens, Athens Greece; M. Vazirgiannis, Athens University of Economics and Business, 76, Patission Str. GR10434 Athens-Greece; N. Kumar, Department of Computer Science and Engineering, University of California at Riverside, 900 University Ave., Riverside, CA 92521; and C. Domeniconi, Department of Information and Software Engineering, George Mason University, 4400 University Drive, Fairfax, VA 22030.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or direct commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org. © 2008 ACM 1556-4681/2008/01-ART18 \$5.00 DOI 10.1145/1324172.1324176 <http://doi.acm.org/10.1145/1324172.1324176>

Categories and Subject Descriptors: H.2.8 [**Database Management**]: Database Applications—*Data mining*; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Clustering*; I.5.3 [**Pattern Recognition**]: Clustering

General Terms: Algorithms, Theory, Reliability, Experimentation

Additional Key Words and Phrases: Semisupervised learning, cluster validity, similarity measure learning, space learning, data mining

ACM Reference Format:

Halkidi, M., Gunopulos, D., Vazirgiannis, M., Kumar, N., and Domeniconi, C. 2008. A clustering framework based on subjective and objective validity criteria. *ACM Trans. Knowl. Discov. Data* 1, 4, Article 18 (January 2008), 25 pages. DOI = 10.1145/1324172.1324176 <http://doi.acm.org/10.1145/1324172.1324176>

1. INTRODUCTION

Clustering aims at providing useful information by organizing data into groups (referred to as clusters). It is applicable in many real-life applications because there is typically a large amount of unlabeled data available. Such data may contain information not previously known or which changes rapidly (e.g., genes of unknown function, dynamically changing webpages in an automatic web document classification system). On the other hand, labeled data is often limited, as well as difficult and expensive to generate because the labeling procedure requires human expertise.

However, in many cases the use of labeled data is critical for the success of the clustering process and for evaluation of the clustering accuracy. Consequently, learning approaches which use both labeled and unlabeled data have recently attracted the interest of researchers [Basu et al. 2004; Xing et al. 2002; Kulis et al. 2005].

Consider, for example, the problem of clustering different cars into segments based on a set of technical attributes. Some attributes, for example, the number of doors, are much more important than others, such as the weight of the car. A small variation (three to four doors) of the first attribute may result in a different type of car (hatchback or sedan). On the other hand, cars within the same segment may have relatively large variation in “weight” values. As a graphical example, consider the dataset presented in Figure 1(a) and assume that we aim to partition it into two clusters. A traditional clustering algorithm (K-Means) produces the partitioning into two clusters, as Figure 1(b) depicts. However, if the user knows or believes that one of the objects in B should be in the same cluster as one of those in C, or if the user knows or believes that one of the objects in A should not be in the same cluster with one of those in B, then the clustering in Figure 1(b) does not reflect the user intuition of “good” clustering. In fact, a good clustering from the user’s perspective could be the one depicted in Figure 1(d). This clustering has an additional, very desirable characteristic: It tends to conform to the similarity of the objects, assigning objects that are very similar to each other to the same cluster. Our goal is to develop algorithms that properly generalize user constraints (which are imposed on individual pairs of objects) to distance measures. Then these measures can be used to provide a “good” clustering that conforms to user intuition.

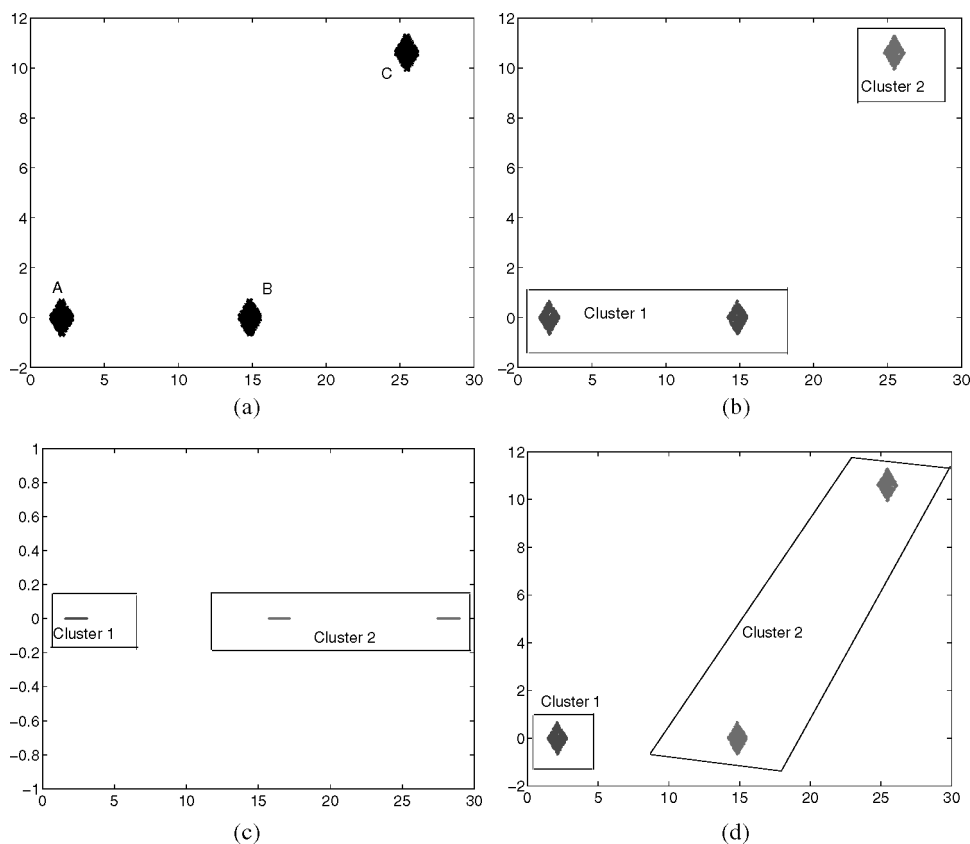


Fig. 1. (a) The original dataset; (b) clustering of the original data into two clusters using the K-Means, hierarchical (complete and average linkage) algorithms without constraints; (c) the clustering results in space defined by our approach so that the user constraints are satisfied; (d) projection of the clusters presented in Figure (c) to the original space.

Estimation of different attributes' relevance (significance) for different clusters is typically performed by feature selection techniques or subspace clustering algorithms [Aggarwal et al. 1999, 1998; Aggarwal and Yu 2000]. Automatic techniques alone are very limited in addressing this problem because the right clustering may depend on the user's perspective. For instance, the weight value might be less important when we want to partition cars based on type, but might be more relevant if other criteria (e.g., fuel consumption) are employed. Unfortunately, a clustering problem does not provide the criterion to be used. Hence, user intervention is needed in order to resolve this clustering problem.

Another approach for assessing the validity of clustering results is to evaluate structural/statistical properties of the data (i.e., density distribution, variance) to assess the validity of the clusters. Such clustering validity criteria are called *objective criteria*. Nevertheless, the presence of such structural/statistical properties does not guarantee the interestingness and usefulness of clustering results for the user [Stein et al. 2003]. Thus, the requirement for approaches

that take into account users' capability to tune the clustering process is well founded.

In this article, we present a framework for semisupervised learning that can be used with any of the known clustering algorithms. The framework allows the user to guide the clustering procedure by providing some constraints and giving his/her feedback during the clustering process.

This feedback is used to adjust weights to each attribute (data dimension), and to combine these weights into a global distance metric that satisfies both *objective* and *subjective* clustering criteria. Essentially, we map the original objects to a new metric space of the same dimensionality, and the resulting distance function, namely weighted Euclidean distance, remains a metric. This allows us to use existing clustering algorithms (e.g., K-Means [MacQueen 1967], density-based [Ester et al. 1997; Hinneburg and Keim 1998], or subspace clustering [Aggarwal et al. 1999, 1998; Aggarwal and Yu 2000]), as well as existing indexing and clustering validity techniques, without modification.

In our framework, the subjective clustering criteria are user-defined constraints in the form of pairs of data points that should belong to the same (different) cluster. Although such constraints are simple, our results show that the technique can generalize these constraints efficiently, and typically only a small number of constraints are required to achieve a satisfactory partitioning.

Several methods have been proposed in the literature [Basu et al. 2004; Cohn et al. 2003; Kulis et al. 2005; Xing et al. 2002] that learn the weights of data dimensions so that a set of user constraints is satisfied. We note, however, that different sets of weights can satisfy the given constraints, and therefore the problem arises of selecting the best weights in order to achieve the best clustering results still respect the user constraints. We use objective validity criteria to tackle this problem. Specifically, we present a hill-climbing method that optimizes the set of attribute weights. The method optimizes a *cluster-quality* criterion that reflects the objective evaluation of the defined clusters while maintaining a measure of the clusters' accuracy in relation to the user constraints.

Summarizing, to the best of our knowledge, we present the first framework for semisupervised learning that efficiently employs both objective and subjective criteria for discovering the data partitioning. An earlier version of our work appeared in Halkidi et al. [2005]. The main characteristics of the proposed approach are as follows.

- Learning a Global Set of Dimension (Attribute) Weights.* The adoption of global weights for the weighted Euclidean distance metric preserves the metric properties of the data space and the structural properties of the original dataset.
- Learning Data Dimensions' Weights Based on User Constraints and Cluster Validity Criteria.* A hill-climbing method learns the weights of the data dimensions that satisfy subjective user-specified constraints while optimizing objective cluster validity criteria.
- User Interaction.* During the learning procedure, intermediate clustering results are presented to users, who can guide the clustering procedure by providing additional feedback in the form of clustering constraints.

—*Flexibility*. The proposed framework provides a mechanism for learning the data space where the “best” clustering can be defined with respect to the user constraints. It takes as parameters: (a) the clustering algorithm; (b) the cluster validity index, which incorporates the objective cluster validity criteria; and (c) the data transformation technique.

The article is organized as follows. Section 2 presents the related work. Section 3 addresses the fundamental concepts and techniques for our approach. Then we discuss the main steps of the proposed learning algorithm in Section 3.5. Experimental results that demonstrate the accuracy and efficiency of our approach are discussed in Section 4. Finally, we conclude in Section 5.

2. RELATED WORK

Clustering is a well-known problem and has been studied extensively by researchers, since it arises in many application domains in engineering and social sciences [Berry and Linoff 1996; Fayyad et al. 1996; Jain et al. 1999].

One of the challenging issues in clustering is the selection of dimensions that are relevant to clusters. The data is highly likely to lack clustering tendency in high-dimensional space and there might be only some subspace where data can be organized into clusters. Since this problem arises in many application domains (e.g., text mining, biomedical applications), a number of techniques have recently been proposed. These approaches employ learning feature (dimension) weights and guiding the clustering process in order to partition the dataset into more meaningful clusters. The proposed techniques in Frigui and Nasraoui [2004], Jing et al. [2005], and Blansch et al. [2006] perform feature weighting and clustering in an unsupervised manner.

However, the problem of *semisupervised learning* has recently attracted significant interest among researchers [Blum and Mitchell 1998; Nigam et al. 2000]. As the term suggests, semisupervised learning is the middle road between supervised and unsupervised learning. It employs user input to guide the algorithmic process that is used to identify significant groups in a dataset.

A constrained-based clustering algorithm is COP-KMeans [Wagstaff et al. 2001], which has a heuristically motivated objective function. It takes as input a dataset X and a set of must-link and cannot-link constraints, and returns a partition of instances in X that satisfies all specified constraints. The major modification with respect to K-Means is that when updating clustering assignments, the algorithm ensures that none of the specified constraints is violated. Also, the COP-COBWEB [Wagstaff and Cardie 2000] algorithm is a constrained partitioning variant of COBWEB [Fisher and Douglas 1987].

A related model for semisupervised clustering with constraints was proposed in Segal et al. [2003]. It is applied to gene data. More specifically, the model is a unified Markov network that combines a binary Markov derived from pairwise protein interaction data and a naive Bayes Markov network modeling expression data. In recent work on distance-based semisupervised clustering, Xing et al. [2002] propose an algorithm that, given a set of similar pairs of points, learns a distance metric that satisfies these relationships. The proposed approach is based on posing metric learning as a combination of gradient descent

and iterative projection. In Bar-Hillel et al. [2003] the RCA algorithm is proposed, which uses only must-link constraints to learn a Mahalanobis distance metric. The problem of learning distance metrics is also addressed by Cohn et al. [2003], who use gradient descent to train the weighted Jensen-Shannon divergence in the context of EM clustering.

In Basu et al. [2004] a semisupervised clustering algorithm, MPCK-Means, is introduced. It incorporates both metric learning and the use of pairwise constraints in a principal manner. Also, Basu et al. [2004] introduce a framework for semisupervised clustering, which employs hidden random Markov fields (HMRFs). The approach aims at utilizing both labeled and unlabeled data in the clustering process. The authors introduced the HMRF-KMeans algorithm that performs clustering in this framework and incorporates supervision in the form of pairwise constraints in all stages of the clustering algorithm.

A new clustering algorithm based on Kernel-KMeans is proposed in Kulis et al. [2005], which aims to optimize a semisupervised objective for both vector- and graph-based inputs. Also, it uses kernel methods that enable the mapping of input data to a higher-dimensional kernel space, where we can find clusters with nonlinear boundaries in the original data space.

An approach is presented in Gao et al. [2005] that aims to incorporate the partial knowledge information into a clustering algorithm. The labeled data (background knowledge) can be specified in a different feature space than the unlabeled data. The clustering algorithm is formulated as a constrained optimization problem. Specifically, the objective function for K-Means is properly modified to incorporate the constraints due to partially labeled information.

All the aforementioned discussed approaches for clustering train the distance measure and perform clustering using K-Means, or provide modifications of the K-Means algorithm to take into account the user constraints in the clustering process.

A clustering algorithm is proposed in Yip et al. [2005] which can identify projected clusters of extremely low dimensionality. It combines object clustering and dimension selection into a single optimization problem. Thus, it aims to select the relevant dimensions that form the subspace where the clusters can efficiently be discovered. Also, the proposed algorithm can utilize domain knowledge in the form of labeled objects and labeled dimensions to improve clustering results. However, this knowledge is only used to define the initial set of clusters and is not incorporated into the whole clustering process.

Our approach is not a clustering algorithm, as it provides a mechanism for learning the dimension weights driving a clustering algorithm to partition the datasets such that the user constraints are satisfied. It integrates distance learning with the clustering process by selecting the features (dimensions) that result in the “best” partitioning of the underlying dataset, performing cluster validity and data space transformation techniques.

3. SEMISUPERVISED LEARNING FRAMEWORK

In this section, we focus on a framework for learning the data space so that the best partitioning of a dataset with respect to the user preferences is defined by

a given clustering algorithm. Given a dataset and a set of constraints (provided in the form of must- and cannot-link constraints), our approach deals with the problem of *selecting* and *weighting* the relevant dimensions according to the user's intention for clustering.

In this work, the weight dimension assignment problem is defined as a cluster quality optimization problem over the data space. The objective is to maximize a cluster quality criterion which assesses the quality of the dimensions' weighting (i.e., data projection) with respect to (w.r.t.) the definition of the good clustering. Specifically, our approach evaluates the quality of the clustering that is defined in the new space to which the data is projected, taking into account: (i) its accuracy w.r.t. the user constraints, and (ii) its validity w.r.t. widely accepted cluster validity criteria. Then the "best" clustering refers to the data partitioning that maximally satisfies these two quality criteria (objective and subjective) of "good" clustering.

In general terms, our approach for learning the dimension weights involves the following steps.

- (1) Initialization of the dimension weights based on the user constraints.
- (2) Learning the data dimension weights to satisfy both cluster validity criteria and user constraints.

The following sections discuss in detail the fundamental concepts and procedures that are performed in the context of our semisupervised learning model.

3.1 Defining the Constraints

Our semisupervised learning approach considers the model where the supervision is provided in form of: (i) *must-link* constraints that provide information about the pairs of points that must be assigned to the same cluster; and (ii) *cannot-link* constraints that inform us about the pairs of points that should belong to different clusters.

Assuming a set of points $X = \{x_1, \dots, x_n\}$, the sets of must-link constraints S and cannot-link constraints D that the user has provided for X can be formally defined as follows.

$$\begin{aligned} S &: (x_i, x_j) \in X \text{ if } x_i \text{ and } x_j \text{ are in the same cluster, and} \\ D &: (x_i, x_j) \in X \text{ if } x_i \text{ and } x_j \text{ are in different clusters.} \end{aligned}$$

3.2 Initializing Data Dimension Weights Based on User Constraints

One of the main issues in clustering is the selection of the distance/similarity measure. The choice of this measure depends on the properties and requirements of the application domain of concern. Another issue that arises in the context of semisupervised clustering is the learning of distance measures so that the user constraints are satisfied. Thus, recently, several semisupervised clustering approaches have been proposed using adaptive versions of widely used distance/similarity function. In this work, we adopt the approach proposed in Xing et al. [2002] to obtain an initial assignment of the dimensions' weights. Next, we provide a brief description of this approach.

We assume a set of points X and the sets of must-link (S) and cannot-link (D) constraints as defined in Section 3.1. The goal is to learn a distance metric between the points in X that satisfies the given constraints. In other words, considering a distance metric of the form

$$d_A(x, y) = \sqrt{(x - y)^T A (x - y)},$$

we aim to define the A matrix so that both the must-link and cannot-link constraints are satisfied. To ensure that d_A satisfies nonnegativity and triangle inequality (i.e., it is metric), we require that A is positive semidefinite. In general terms, A parametrizes a family of Mahalanobis distances over R^m . Specifically, when $A = I$, d_A gives the Euclidean distance. In our approach, we restrict A to be diagonal. This implies learning a distance metric in which different weights are assigned to different dimensions. Then the problem of learning a distance measure with respect to a set of must-link and cannot-link constraints reduces to the following optimization problem.¹

$$\min_A \sum_{(x_i, x_j) \in S} \|x_i - x_j\|_A^2 \quad (1)$$

Here, it is given that $\sum_{(x_i, x_j) \in D} \|x_i - x_j\|_A \geq 1$.

In this article we consider the case of a diagonal A . Then we can solve the original problem using Newton-Raphson to efficiently optimize the function

$$g(A) = g(A_{11}, \dots, A_{mm}) = \sum_{(x_i, x_j) \in S} \|x_i - x_j\|_A^2 - \log \left(\sum_{(x_i, x_j) \in D} \|x_i - x_j\|_A \right). \quad (2)$$

3.3 Learning the Dimension Weights

Based on the approach described in Section 3.2 we define the initial set of the data dimensions' weights. However, there are cases where different weightings of dimensions satisfy to the same degree the user constraints. Thus the issue arises as, which of these dimensions' weightings result in the best clustering of the underlying data. To address this problem, we further tune the dimensions' weights so that the best rescaling of data, in terms of the given constraints and the validity of the defined clusters, is found.

Specifically, to find a meaningful weighting of a specific set of d dimensions Dim , for a given set of must-link and cannot-link constraints, further referred to as $S\&D$, our approach performs hill climbing. The method searches for those weights that correspond to the best projection of data in the d -dimensional space according to $S\&D$. We use a clustering validity measure to assess the relevance of the dimensions' weighting (i.e., data projection) to the clustering. In the sequel, we formalize the quality measure that we adopt.

Definition 3.1 (Cluster Quality w.r.t. User Constraints). Cluster Quality evaluates a clustering C_i of a dataset in terms of its accuracy w.r.t. the user constraints (S&D) and its validity based on well-defined cluster validity

¹A detailed discussion of this problem is presented in Xing et al. [2002].

criteria. It is defined as:

$$QoC_{constr}(C_i) = w \cdot Accuracy_{S\&D}(C_i) + ClusterValidity(C_i), \quad (3)$$

where $Accuracy_{S\&D}(C_i)$ is the proportion of the S&D constraints that are satisfied in the C_i clustering and $ClusterValidity(C_i)$ is an index of the C_i cluster validity. The weight w denotes the significance of the user constraints in relation to the cluster validity criteria (objective criteria) with regard to the definition of the “best” clustering of the underlying dataset. We note that both terms (i.e., $Accuracy_{S\&D}$ and $ClusterValidity$) are in the range of $[0, 1]$. Then we set the value of w such that the violation cost of the user constraints is higher than that of the cluster validity criteria. Specifically, w is defined to be equal to the number of constraints to ensure that none of the constraints will be violated in favor of satisfying objective validity criteria. This implies that, having satisfied the user constraints, the quality criterion (used in the hill-climbing method) aims at selecting the data clustering that is also considered “good” based on the objective validity criteria (as defined by $ClusterValidity$).

Given the initial weights computed via the Newton-Raphson technique (as described in Section 3.2), using the input clustering algorithm (e.g., K-Means) we compute an initial clustering of the data in the space transformed by the weights. An iterative process is then performed on each data dimension to perform hill climbing (HC) over the function in Eq. (3). Our iterative procedure tries to compute a local optimum in the space of the weights, so that the clustering measure QoC_{constr} is optimized and the best weighting of data dimensions is defined.

Definition 3.2 (Best Weighting of Data Dimensions). Let $W = \{W_j\}_{j=1}^p$ be the set of different weightings defined for a specific set of data dimensions, and d be the number of dimensions. Each weighting $W_j = \{w_{j1}, \dots, w_{jd}\}$ reflects a projection of data to a d -dimensional space. Among the different clusterings $\{C_i\}_{i=1}^m$ defined by an algorithm for the different weightings in W , the one that maximizes QoC_{constr} , is considered to be the best partitioning of the underlying dataset and the respective $W_j(W_{best})$ defines the best weighting of dimensions in the projected data space. In other words, $W_{best} = \{W_j \in W \mid clustering(W_j) \in \{C_i\}_{i=1}^{n_c} \wedge QoC_{constr}(clustering(W_j)) = \max\{QoC_{constr}(C_i)\}\}$, where $clustering(W_j)$ denotes the clustering in $\{C_i\}_{i=1}^{n_c}$ that corresponds to the W_j weighting.

Algorithm 1 presents, at a high level, the procedure for tuning the weights of a set of dimensions under a set of user constraints. The hill-climbing procedure (step 3 in Algorithm 1) starts updating the weight of the first dimension (while the weights of other dimensions are retained as they have been currently defined) until there is no improvement in the clustering. Having defined the best weight for the first dimension, we repeat the same procedure for tuning the second dimension. The algorithm proceeds iteratively until the weights of all dimensions are tuned.

Traditional hill-climbing techniques for maximizing a function typically use the gradient of the function for updating the weights [Press et al. 1997]. In this work, however, we want to optimize a function that we can compute, but since we do not have it in a closed form we cannot compute its gradient. One approach

Algorithm 1. TuneDimWeights**Input:** the set of user constraintsX: d -dimensional datasetS: set of pairs of points with *must-link* constraint,D: set of pairs of points with *cannot-link* constraint,**Output:** Best weighting of dimensions in X1: W_{cur} = the initial weights of dimensions in X, according to S and D , using the method of Section 3.2.

$$W_{cur} = \{W_i | i = 1, \dots, d\}$$

2: Cl_{cur} = clustering of data in space defined by W_{cur} .3: **for** $i=1$ to d **do**4: **Repeat**{

$$W_{cur} = W'_{cur}$$

a. *Update* (i.e.increase or decrease) the i th dimension of W_{cur} and let W'_{cur} be the updated weighting of dimensions.

b. *Project* data to the space defined by W'_{cur} .

c. *Redefine* Cl_{cur} based on W'_{cur} .

d. *Use the quality criterion* to assign a score to W'_{cur} w.r.t. its clustering (i.e., Cl_{cur}).

5: } **Until** { W'_{cur} does not have a better score than W_{cur} }6: **End for**7: **Set** the best weighting W_{best} to be the one with the “best” score, (i.e., the weighting resulting in “best” clustering).8: **Return** (W_{best})

in such a case is to try to estimate the gradient by recomputing the function after changing each weight by a small fraction. Some recent techniques have been proposed to optimize this process [Anderson et al. 2000], but the main problem is properly defining how much to change the weights at each step. Clearly, if we change the weights by a large fraction, the local maximum can be missed. On the other hand, a small fraction is inefficient. To solve this problem, we employ the following heuristic approach: We start with a large fraction δ (0.1 in the experiments), but before we take a step we also compute the QoC_{constr} using $\delta/2$. If the change with the smaller step is significantly different than that using the larger step, we conclude that the original fraction δ is too large, and we try again after halving it.

3.4 Defining Parameters of the Semisupervised Learning Framework

The proposed framework for semisupervised learning takes as input the following parameters: (i) the cluster validity index, (ii) the clustering algorithm, (iii) the data transformation technique that can be optionally used before the learning procedure of weights is applied. Based on these parameters it aims to learn the data dimensions’ weights so that the best clustering is defined with respect to both the user constraints and the objective cluster validity criteria.

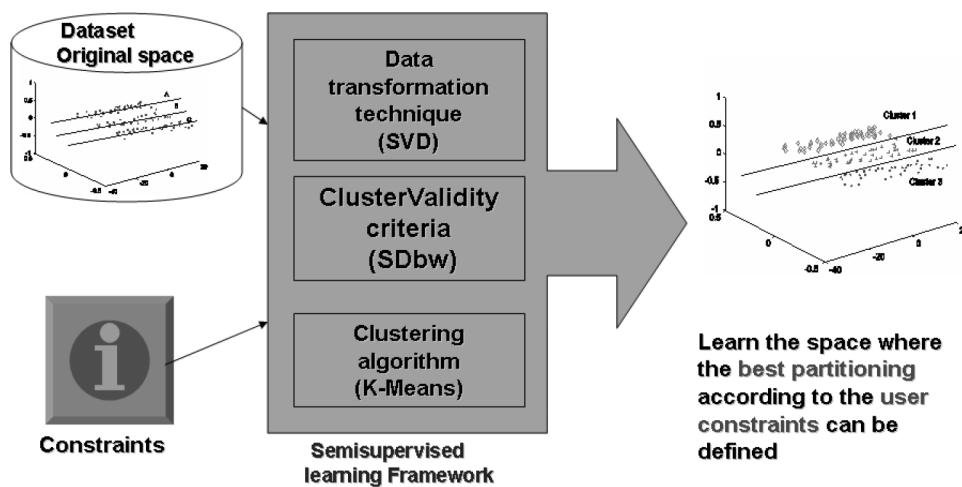


Fig. 2. The semisupervised learning framework.

The configuration of our framework is presented in Figure 2.

3.4.1 Cluster Validity Criteria. Eq. (3) suggests a general framework for incorporating both objective and subjective criteria to the quality assessment procedure of clustering results. In this work, we use the cluster validity index proposed in Halkidi and Vazirgiannis [2001] to assess the quality of the defined clusterings w.r.t. objective validity criteria. However any other cluster validity index can be used to evaluate the clustering

The notion of “good” clustering w.r.t. cluster validity criteria relies on: (i) the accuracy of the clustering w.r.t. the user constraints, namely, to what degree the clustering satisfies the S&D set; (ii) the compactness of clusters evaluated in terms of clusters’ scattering; and (iii) the separation of clusters in terms of intercluster density.

To follow, we formally define the notion of good clustering w.r.t. objective validity criteria. Based on these criteria, a cluster validity index is defined which is further adopted in the procedure of learning the dimensions’ weights. We note that our approach searches for the optimum in terms of user intuition for good clustering (subjective criteria) and the cluster validity criteria (objective criteria) that we have defined previously.

Let $X = \{x_i\}_{i=1}^N$ be a set of d -dimensional points and $C = \{C_i\}_{i=1}^p$ be a set of p different partitionings of X , corresponding to different weightings $\{W_j\}_{j=1}^p$ of data dimensions in d -space. Hence for each weighting $W_j = (w_{j1}, \dots, w_{jd})$ a “rescaling” of X to a new space is defined, that is, $X' = W_j^{1/2} \cdot X$, where X' is a column data vector. In this new space the C_j partitioning of X corresponding to W_j is defined.

Definition 3.3 (Intercluster Variance). This measures the average variance of the clusters of concern with respect to the overall variance of the data. It is

given by

$$Scat(C_i) = \frac{\frac{1}{n_c} \sum_{j=1}^{n_c} \|\sigma(v_j)\|}{\|\sigma(X')\|}, \quad (4)$$

where $C_i \in C$ is a clustering of X in the space defined by the W_i weighting (i.e., a clustering of X'), while n_c is the number of clusters in C_i , and v_j is the center of the j th cluster in C_i . Also $\sigma(v_j)$ is the variance within the cluster c_j while $\sigma(X')$ is the variance of the whole dataset.

Definition 3.4 (Intercluster Density). This evaluates the average density in the region among clusters in relation to the density of the clusters. The goal is to have the density among clusters significantly low in comparison with the density in the considered clusters. Then, considering a partitioning of the dataset $C_l \in C$ into more than one cluster (i.e., $n_c > 1$), the intercluster density is defined as

$$Dens_bw(C_l) = \frac{1}{2n_c(n_c - 1)} \sum_{i=1}^{n_c} \left(\sum_{j=i+1, i \neq j}^{n_c} \frac{density(u_{ij})}{\max\{density(v_i), density(v_j)\}} \right), \quad (5)$$

where v_i, v_j are the centers of clusters $c_i \in C_l, c_j \in C_l$, respectively, and u_{ij} is the middle point of the line segment defined by the clusters' centers v_i and v_j .

The term $density(u)$ is defined as the number of points in the neighborhood of u . In our work, the neighborhood of a data point, u , is defined to be a hypersphere with center u and radius the average standard deviation of the clusters, $stdev$. Then the density in neighborhood of a point u is defined as

$$density(u) = \sum_{l=1}^{n_{ij}} f(x_l, u), \quad (6)$$

where x_l is a point of X' that belongs to the neighborhood of u and n_{ij} is the number of points that belong to the clusters c_i and c_j , namely, $x_l \in c_i \cup c_j \subseteq X'$.

More specifically, the function $f(x, u)$ is defined as

$$f(x, u) = \begin{cases} 0 & \text{if } d(x, u) > stdev, \\ 1 & \text{otherwise} \end{cases}. \quad (7)$$

Definition 3.5 (Cluster Validity Index S_Dbw). The cluster validity index assesses the validity of clustering results based on intracluster variance and intercluster density of the defined clusters. Given a clustering of X, C_i , the S_Dbw is defined as follows.

$$S_Dbw(C_i) = Scat(C_i) + Dens_bw(C_i) \quad (8)$$

The first term of S_Dbw , that is, $Scat(C_i)$, is the average scattering within the clusters of C_i . A small value of this term is an indication of compact clusters. On the other hand, $Dens_bw(C_i)$ is the average number of points among the clusters of C_i (i.e., an indication of intercluster density) in relation to the density within clusters. A small value of $Dens_bw(C_i)$ indicates well-separated clusters. A more

detailed discussion on the definition and the properties of S_Dbw is provided in Halkidi and Vazirgiannis [2001].

We note that S_Dbw depends on the clustering C_i of the data of concern corresponding to the different weightings W_j of the data dimensions. However, it is independent of the global scaling of the data space.

Adopting S_Dbw to evaluate the validity of clustering results in terms of objective criteria, the second term of QoC_{constr} is defined as $ClusterValidity(C_i) = (1 + S_Dbw(C_i))^{-1}$.

Then the clustering quality criterion, namely Eq. (3), that our approach aims to optimize gets the following form.

$$QoC_{constr}(C_i) = w \cdot Accuracy_{S\&D}(C_i) + (1 + S_Dbw(C_i))^{-1} \quad (9)$$

The definition of $QoC_{constr}(C_i)$ indicates that both objective criteria of a good clustering (i.e., compactness and separation of clusters) are properly combined with the accuracy of clustering w.r.t. the user constraints. The first term of QoC_{constr} assesses how well the clustering results satisfy the given constraints. The second term of QoC_{constr} , is based on a cluster validity index S_Dbw which is first introduced in Halkidi and Vazirgiannis [2001]. As noted earlier, a small value of S_Dbw and hence a high value of $(1 + S_Dbw)^{-1}$ is an indication of compact and well-separated clusters. Then the partitioning that maximizes both terms of QoC_{constr} is perceived to reflect a good clustering w.r.t. the user constraints.

3.4.2 Applying Data Transformation Techniques. The SVD technique is used to map the original dataset to a new space where significant groups (clusters) can be identified. This is applied before the learning procedure. Essentially, the points are projected to the new space in such a way that the strongest linear correlations in the underlying data are maintained. However, the relative distances among points in the SVD space are preserved, as well as possible under linear projection.

Applying a data transformation technique (such as SVD) before the learning procedure, we achieve a ranking of dimensions (features) based on their relevance to the cluster definition. Those features with the highest eigenvalues are considered to be most important for clustering. Thus, SVD is used to guide the sequential feature selection and learning methods to start from for finding those dimensions (features) that are most likely to be interesting for data clustering and thus will more efficiently result in a good clustering.

3.4.3 Clustering Algorithm. As we have already noted, our semisupervised learning framework aims to define the weights of data dimensions so that, given a clustering algorithm A, it results in the best clustering that A can define based on the user constraints. Thus, it is used in conjunction with a clustering algorithm, but is independent of it. For our experimental study we have used K-Means in order to be able to compare our results to the competing approaches (all using K-Means or its modifications). Nevertheless, any other algorithm can be considered to partition a given dataset.

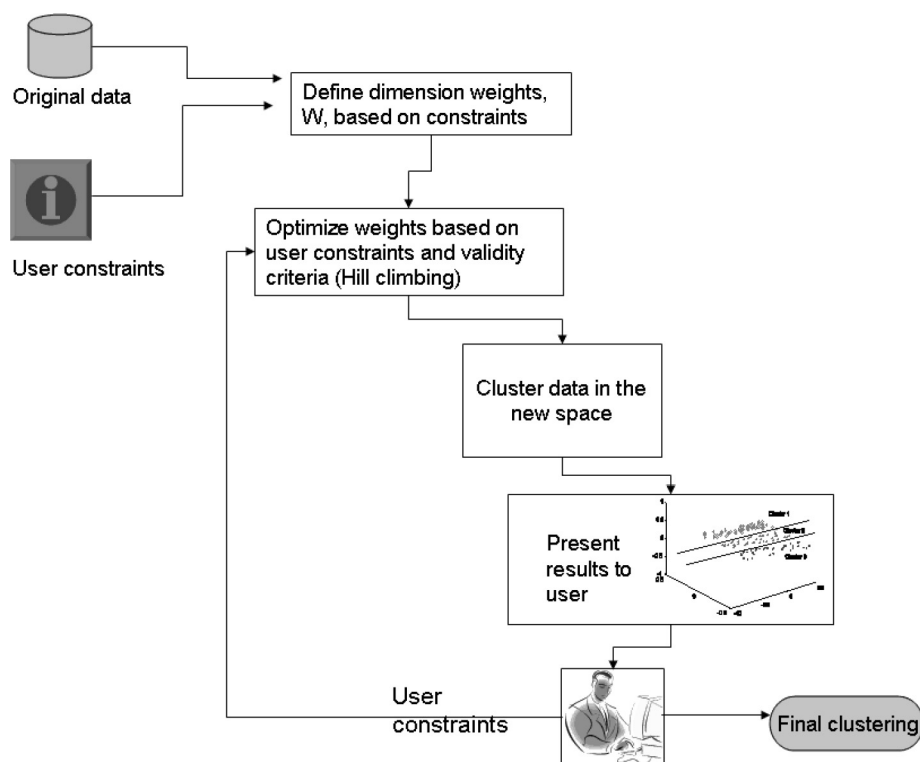


Fig. 3. The main procedures of the semisupervised learning approach.

3.5 Putting It All Together: Semisupervised Clustering

In this section we summarize our approach. Figure 3 graphically illustrates the whole procedure. Given a dataset X and a set of must-link (ML) and cannot-link (CL) constraints, our approach learns the space where the clusters are well separated based on user constraints. A preprocessing step initializes the weight of dimensions based on the data correlation and user constraints. Then we proceed with an optimization step where the weights of dimensions are further tuned. A hill-climbing method is applied to maximize a clustering quality measure in terms of objective and subjective quality criteria. As already mentioned, the proposed learning approach works in two steps which we briefly present next.

Step 1. Preprocessing Step: Initialization of Dimension Weights. Our approach uses SVD to guide the sequential learning procedure for selecting the subspace where the best clustering is defined. As discussed in Section 3.4.2, based on SVD the dimensions are ranked according to their significance for the cluster definition. Then step 2 starts to tune the weights of dimensions that are most likely to be interesting for the data clustering.

Then a visualization of the data is presented to the user, who is asked to give his/her clustering requirements in terms of must-link and cannot-link constraints. Based on the user constraints, the dimension weights are initialized

using the approach described in Section 3.2. Thus the data is projected to a new space that respects user preferences.

Step 2. Learning Dimension Weights Based on the User Constraints and Cluster Validity Criteria. Starting with the first two dimensions in the new feature space defined at step 1, we use the approach described in Section 3.3 to determine the best weighting of data dimensions. We aim to define a partitioning that fits the data as well as possible and also respects the constraints. We further tune the weights of the dimensions according to the hill-climbing method presented in Section 3.3. Once the best partitioning for the given set of dimensions has been defined, the clustering results are presented to the users, who are asked to give their feedback. If the users are not satisfied with the clustering results, we add a new dimension and the previous clustering procedure is repeated for defining the weights of the new set of dimensions in the new space. The process proceeds iteratively until the subspace that satisfies the user constraints is defined.

In general terms, our approach aims at finding the lower-dimensional space in which the original data is projected so that the best partitioning according to the user constraints is defined.

3.6 Time Complexity

Let N be the number of points in the dataset, and d be the number of dimensions. The first step of our approach refers to the projection of the data to the SVD space. Thus, the time cost is of constructing the singular value decomposition of the N -by- d data matrix, which is $O(d^2 \cdot N)$. The second step, which refers to the definition of clusters according to the user constraints, depends on the complexity of Algorithm 1, based on which the weights of dimensions in the SVD space are learned with respect to the user constraints.

Initially, the Newton-Raphson method is applied to define the initial weights of data dimensions. It is a method for efficiently solving the optimization problem of defining the weights of dimensions, given a set of constraints. Intuitively, the complexity of the Newton-Raphson method depends on the constraints. However, it is expected that it reaches an optimum in a finite number of iterations that is significantly smaller than N . Hence its complexity is estimated to be $O(N)$. The set of weights are tuned based on a hill-climbing method (HC) that relies on the optimization of the cluster-quality criterion QoC_{constr} . The complexity of the quality criterion is $O(d \cdot N)$. The tuning procedure is iterative and at each step the weights of dimensions are updated, defining a rescaling of the space into which the data is projected.

Given a clustering algorithm Alg, the respective clustering of data in the space defined by the current dimensions' weights is defined while the clustering results are evaluated based on QoC_{constr} . Though HC mainly depends on the number of constraints, it is expected to reach an optimum in a number of iterations that is smaller than the number of points. According to the preceding analysis, the complexity of Algorithm 1 is $O(d^2 \cdot N + Complexity(Alg))$. Usually $d \ll N$. Hence, the complexity of our learning approach depends on the complexity of the clustering algorithm.

4. EXPERIMENTAL EVALUATION

In this section we test our approach with a comprehensive set of experiments. In Section 4.1, we discuss the dataset we used for experimental purposes and the accuracy measure, based on which we evaluate the clustering performance of our approach. In Section 4.2, we show simple experiments which require subjective evaluation, but strongly hint at the value of our approach. In Section 4.2.1, we present a comparison of our method with both a related approach proposed in the literature and with the unsupervised clustering method. In Section 4.3, we present an experimental study of the time complexity of our approach.

4.1 Methodology and Datasets

We used MATLAB to implement our approach and we experimented with various datasets.

To show the advantage of our approach w.r.t. unsupervised learning, we used synthetic datasets generated to show the indicative cases where unsupervised clustering fails to find the clusters that correspond to the user intention. We also used datasets from the UC Irvine repository² to evaluate the effectiveness of our method with respect to a prespecified clustering method in which the same datasets were used.

Clustering accuracy. Rand statistic [Hubert and Arabie 1985] is an external cluster-validity measure. It measures the degree of correspondence between a prespecified structure (which reflects our intuition of a good clustering of the underlying dataset) and the clustering results after applying our approach to X .

Let $C = \{c_1, \dots, c_r\}$ be a clustering structure of a dataset X into r clusters and $P = \{P_1, \dots, P_s\}$ be a defined partitioning of the data. We refer to a pair of points $(x_v, x_u) \in X$ from the dataset, using the following terms.

- SS: if both points belong to the same cluster of the clustering structure C and to the same group of partition P .
- SD: if points belong to the same cluster of C and to different groups of P .
- DS: if points belong to different clusters of C and to the same group of P .
- DD: if both points belong to different clusters of C and to different groups of P .

Assuming now that a, b, c , and d are the number of SS, SD, DS , and DD pairs, respectively, then $a + b + c + d = M$, which is the maximum number of all pairs in the dataset (meaning that $M = n \cdot (n - 1)/2$, where n is the total number of points in the dataset). Now we can define the Rand statistic index to measure the degree of similarity between C and P , as follows.

$$R = (a + d)/M$$

4.2 Results and Discussion

Referring to Figure 1 again, we applied our approach to cluster a dataset based on a set of given constraints. These constraints define must-link and cannot-link

²<http://www.ics.uci.edu/mllearn/MLRepository.html>

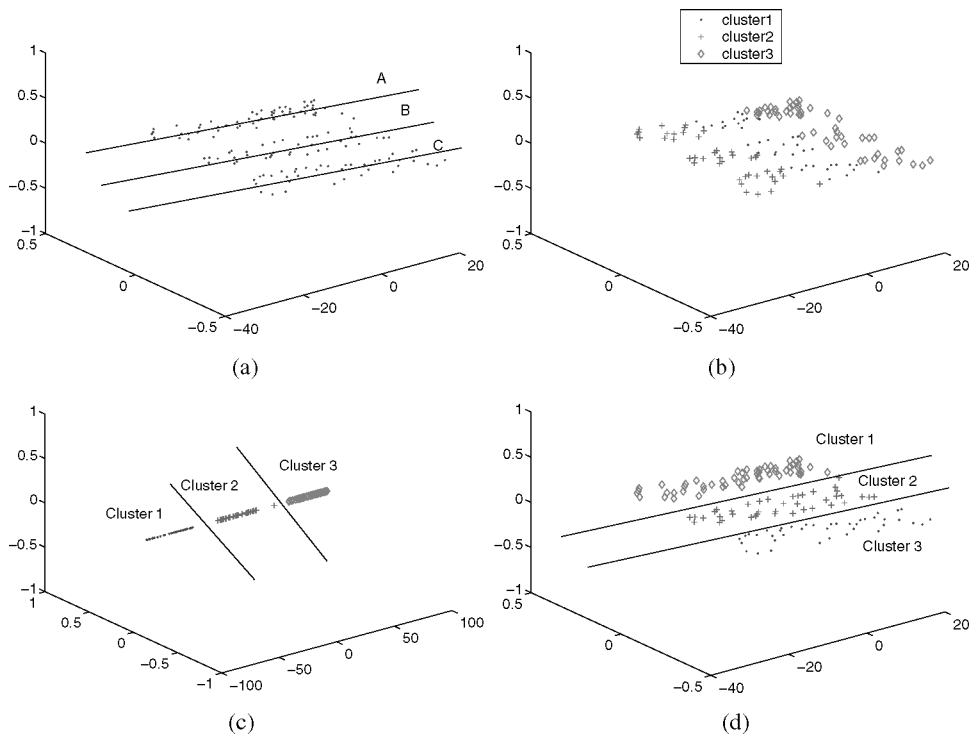


Fig. 4. A dataset containing three lines of points in 3D space: (a) original dataset. The data is distributed around the lines A, B, and C; (b) clustering of the original data using K-Means; (c) clustering of the original data in the new space using our approach; (d) projection of the clusters presented in figure (c) to the original space.

constraints between points in the dataset (e.g., must link between $x \in B$ and $y \in C$, cannot link between $x' \in A$ and $y' \in C$). Figure 1(c) shows clustering results in the transformed space using our learning approach, whereas Figure 1(d) shows the projection of clusters obtained in Figure 1(c) in the original space.

The visualization of a similar example is presented in Figure 4. One can claim that there are three groups of data as defined by the three lines A, B, and C, as Figure 4(a) depicts. We apply K-Means [MacQueen 1967] to partition it into three clusters. The result of unsupervised K-Means is presented in Figure 4(b). It is clear that K-Means is not able to identify the three clusters that the user requested. Given a set of constraints, we applied our learning approach. Figure 4(c) shows the projection of the dataset and its clustering to a new space, while Figure 4(d) demonstrates the projection of clusters in the original space.

4.2.1 Comparison to Other Approaches. We compare our approach with unsupervised K-Means clustering, and the semisupervised approaches proposed in Xing et al. [2002], Bilenko et al. [2004], and Basu et al. [2004]. These are described next.

(1) *Naive K-Means.* The K-Means algorithm using the default Euclidean metric.

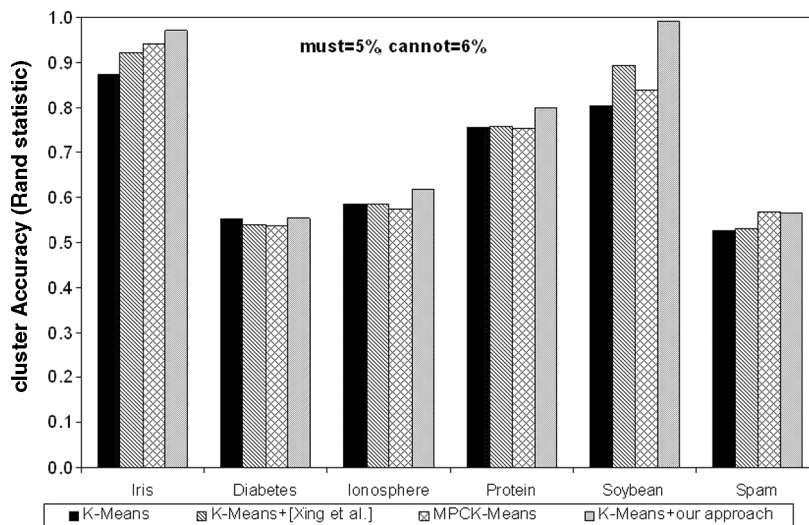
- (2) *K-Means + Xint et al.* [2002]. K-Means using the distance metric learned to satisfy a set of constraints based on Xing et al.’s approach.
- (3) *MPCK-Means*. The complete K-Means algorithm that involves both seeding and metric learning, as described in Bilenko et al. [2004] and Basu et al. [2004].
- (4) *K-Means + Our Approach*. K-Means is applied to the subspace learned by our approach to satisfy the user constraints.

We applied these methods to six datasets with X to Y dimensions from the UCI repository. For each of the datasets, we evaluate their performance using the same set of constraints. We varied the number of constraints used as the percentage of data points; for example, using 5% of data points as must-(cannot)-link constraints in a dataset of 100 points means that we used only 5 must- (cannot)-link constraints. The results are evaluated based on the external validity measure Rand statistic presented earlier. Here, the prespecified clustering structure (“true” clustering) is given by the class labels of data as provided from the UCI repository. Then the “true” clustering represents the intuition of the user for good clustering and the goal is to approximate it as accurately as possible.

As we have noted in Section 3.4.2, SVD is a step of our approach that provides a ranking of the features (dimensions) based on their relevance to the clustering process. It guides the sequential feature learning procedure starting from the features that are the most significant for clustering. In our experimental study, however, we used SVD as a preprocessing step for all approaches in order to have an equal standing comparison. We note that in this experimental study, we applied our learning approach to all the dimensions of the considered datasets. Then, in order to define the subspace that corresponds to the user preferences, we selected the weighting corresponding to the clustering that satisfies with the highest accuracy the user constraints.

Figure 5 gives a comparative overview of the results. The clustering accuracy was averaged over 10 runs using randomly selected constraints (must-link= 5% and cannot-link=6% of points). We observe that our learning approach outperforms the other three related approaches used for comparison.

To prove the robustness of these results, we used a t-test method [Hogg and Craig 1978] to evaluate the statistical significance of advantageous results. The t-test is used to determine whether two samples are likely to have originated from the same two underlying populations that have the same mean value. In our case the first sample refers to the clustering results defined by 10 different runs of the proposed approach, while the second sample contains the respective results of each of the competing approaches (naive K-Means, Xing et al.’s approach, and MPCK-Means). Table I presents the probability associated with a student’s *t*-test, with one-tailed distribution. The lower the probability, the higher our confidence that the difference between the two sample means is statistically significant. It is obvious that in almost all cases the probability that our approach gives similar results to those of the competing approaches (i.e., comes from the same underlying populations) is significantly low. For instance, Table I shows that for the *Iris* dataset, the probability that



Iris(d=4), Diabetes(d=8), Ionosphere(d=34), Protein(d=20), Soybean(d=35), Spam(d=57).

Fig. 5. Clustering accuracy on UCI datasets using SVD as a preprocessing step. The learning procedure used 5% of the data points as must-link and 6% of points as cannot-link constraints. This implies that in a dataset of 100 points, we used only 5 must-link and 6 cannot-link constraints.

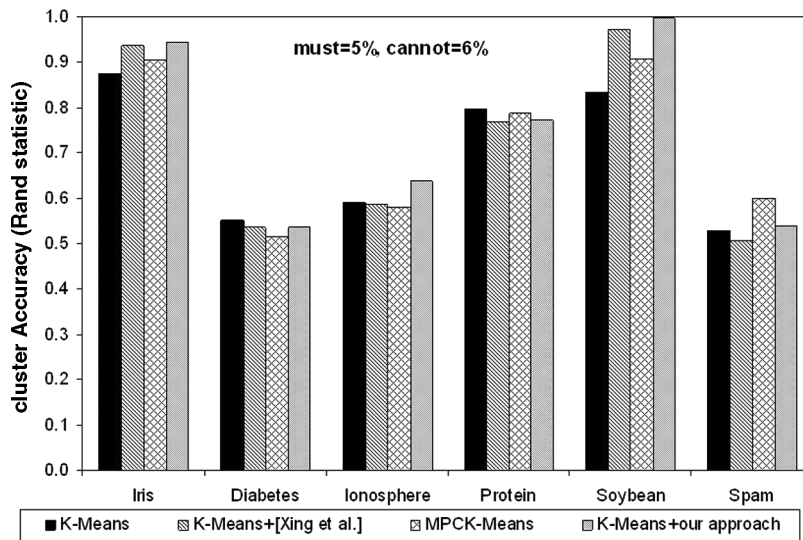
Table I. The t -Test: Our Learning Approach versus Competing Approaches

Our approach	MPCK-Means	K-Means	K-Means + [Xing et al. 2002]
Iris	0.00071274	5.46105E-12	0.008514
Diabetes	8.2091E-06	0.028850713	0.000223
Ionosphere	0.00173141	0.008099125	0.006594
Protein	6.9001E-05	3.77801E-07	0.001114
Soybean	8.9585E-13	2.23264E-12	0.007402
Spam	0.4295784	0.001485	0.012849

the results of our approach and those of MPC-KMeans are similar is about 0.0007. We note that only for the *Spam* dataset is there high probability (i.e., 0.429) that the proposed approach achieves similar clustering to that of MPCK-Means.

Generally, the improvement (as Figure 5 shows) in clustering accuracy that our approach achieves in relation to other, related approaches is statistically significant.

Contribution of the preprocessing step. We propose using a data transformation method (such as SVD) before the learning procedure. To evaluate the contribution of this preprocessing step to the results of the whole learning approach we have experimented with various datasets from the UCI repository. More specifically, we applied our approach without using SVD as the initialization step for the learning procedure. Figure 6 shows the results of our approach in comparison to the competing approaches. We observe that our approach has in better, or at least similar, results to those of the other three related approaches. Only in case of the *Spam* dataset does MPCK-Means achieve more



Iris(d=4), Diabetes(d=8), Ionosphere(d=34), Protein(d=20), Soybean(d=35), Spam(d=57).

Fig. 6. Clustering accuracy without using SVD as a preprocessing step.

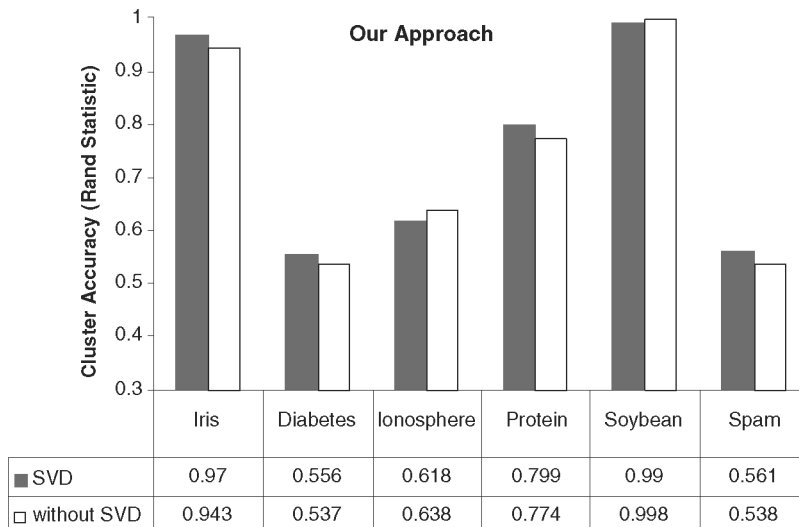


Fig. 7. Clustering accuracy when semisupervised learning is applied to both the SVD space and the original space.

accurate clustering (as defined by the given classes in the UCI repository) than our approach. However, comparing Figures 5 with 6, we can claim that SVD is a useful preprocessing step, since it efficiently contributes to the learning process.

Moreover, Figure 7 presents in comparative fashion the clustering accuracy of our approach in cases of using SVD as a preprocessing step, or applying the weight learning procedure directly to the original data. We observe that in most of the cases SVD seems to contribute to better learning of the space

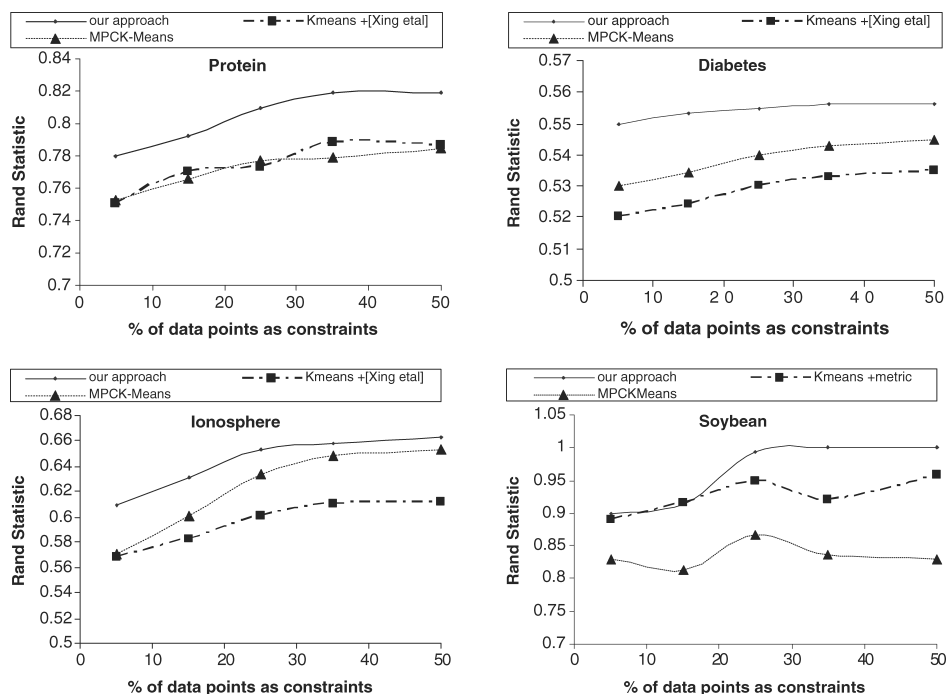


Fig. 8. Clustering accuracy versus constraints: Our approach learns quickly the subspace where a good clustering can be found with a small number of constraints. The y axis corresponds to the clustering accuracy based on the Rand statistic. The x axis refers to the percentage of data points that are used as constraints for each of the considered categories (i.e., 10% corresponds to 10% of the points as must- and 10% of the points as cannot-link constraints).

where the best clustering can be defined. The contribution of SVD depends on the datasets. However, even in cases of datasets where SVD does not seem to improve the clustering accuracy of our approach (i.e., *Ionosphere*, *Soybean*), the respective results in Figures 5 and 6 show that our approach outperforms the competing methods in both experimental scenarios (i.e., with or without SVD as a preprocessing step).

Clustering accuracy versus constraints. Figure 8 shows how the quality of clustering increases as we increase the percentage of data points used as constraints in the cases of four UCI datasets (*Protein*, *Diabetes*, *Ionosphere*, *Soybean*). These datasets are selected as “difficult-to-cluster” datasets, even in case that we use partial knowledge to guide the clustering process.

The performance of our learning approach with regard to the cardinality of the constraints (as the portion of dataset size) is presented in comparison to the approaches proposed in Xing et al. [2002] (K-Means + learned metric) and Bilenko et al. [2004] (MPCKMeans). We observe that our approach systematically leads to improvement in clustering quality (see Figure 8), even in cases where few constraints are used. For the *Protein*, *Ionosphere*, and *Soybean* datasets, clustering accuracy increases with the number of constraints. A “good” clustering can be achieved using about 30% of the data points as constraints

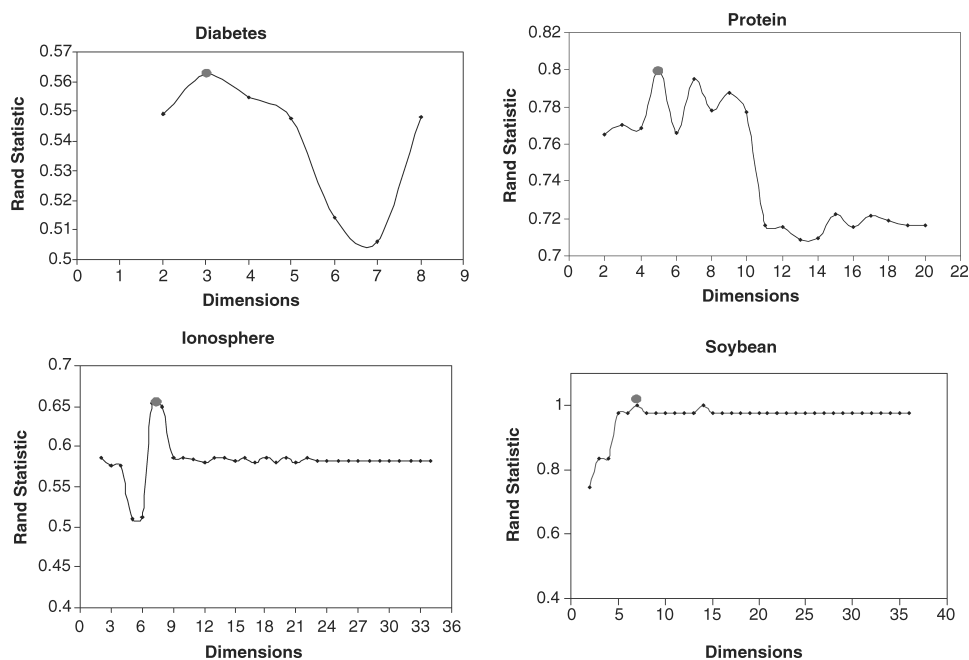


Fig. 9. Clustering accuracy versus dimensions.

for each of the considered categories (i.e., 30% must-link and 30% cannot-link), while additional ones do not seem to significantly contribute to the learning procedure. Similarly, for *Diabetes* the proposed approach achieves more accurate results than the competing two approaches, whereas additional constraints improve the clustering accuracy only insubstantially. This is due to the distribution of the underlying data, which presents a low clustering tendency and thus is not easily separable.

Based on the preceding discussion we claim that the proposed learning approach significantly improves the clustering quality. Moreover, it contributes to learning the subspace where a good clustering can be found efficiently, with only a small number of constraints.

Learning the data subspace. We evaluate the performance of our approach (in terms of clustering accuracy) in relation to the dimensions that we have to tune in order to result in a good clustering. Our experiments using four high-dimensional datasets of the UCI repository show that tuning only a small number of data dimensions, we can define the space where the best clustering can be defined. As we have already noted, the best clustering is defined in terms of quality measures, presented in Section 3.4. Figure 9 shows how the quality of clustering for the datasets of concern changes with the number of tuned dimensions. For instance, in the case of *Protein* it reaches its maximum when only 4 from the 20 dimensions are tuned, while for higher dimensions it significantly decreases. This implies that the use of more than 4 dimensions seems not to contribute to the definition of the clusters presented in *Protein*. Besides, for the *Ionosphere* dataset, we observe that clustering accuracy increases as the

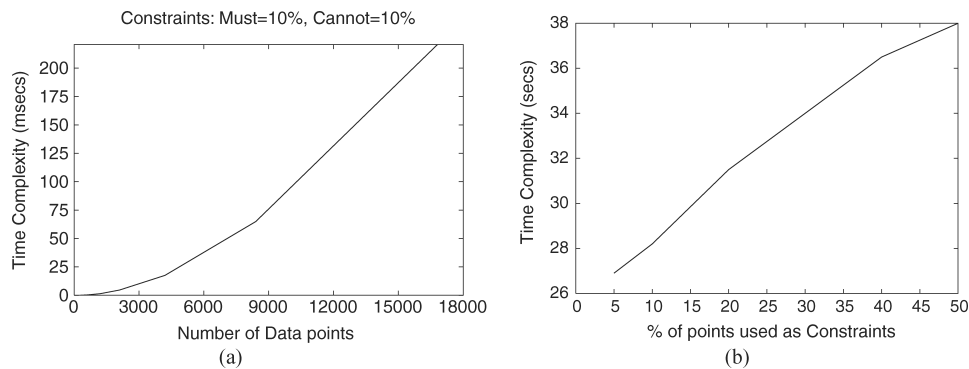


Fig. 10. Complexity of our approach versus: (a) the number of points used for learning the dimension weights to respect user constraints; and (b) the percentage of data points used as constraints for the *Iris* dataset.

number of dimensions increases from 2 to 7, while remaining vaguely the same for higher dimensions. *Diabetes* and *Soybean* show a similar behavior. Thus, using only few of the dimensions, we can efficiently learn the space where the best clustering can be defined.

Based on these observations, it is obvious that our approach does not only learn the data dimensions to satisfy the user constraints, but also assists with selecting the subspace where the best clustering can be defined.

4.3 Time Complexity Evaluation

Figure 10 shows the results of our experimental study to quantify the complexity of the proposed approach with respect to the size of dataset and the ratio of constraints used for learning. For this experiment we use K-Means for defining clusters in the underlying datasets. More specifically, Figure 10(a) shows that the complexity of our approach is nearly linear to the number of points in the dataset. In this graph we present the results of experiments using a four-dimensional dataset, while in all cases we considered that the complexity of our approach is nearly linear to the number of points in the dataset, using 20% of the original dataset for the must-link and cannot-link constraints (i.e., must= 10%, cannot= 10%). However, we have experimented with higher-dimensional datasets and the results are qualitatively similar to those presented in Figure 10(a), thus they are omitted for brevity. In addition, Figure 10(b) demonstrates that the execution time increases linearly with the percentage of constraints of concern.

5. CONCLUSIONS

In this article we propose a framework for learning the space where the best partitioning of the underlying data is achieved while user constraints are respected. It introduces a semisupervised learning approach that aims to efficiently combine both objective (i.e., related to the data structure) and subjective (i.e., in regard to user constraints) criteria in the context of clustering. The proposed approach allows the user to guide the clustering process by providing some

constraints and giving his/her feedback during the whole clustering process. Our experimental results using both real and synthetic datasets show that our approach enables significant improvement in the accuracy of the defined clustering with respect to the user constraints.

An interesting direction for our further work is the extension of our approach so that we learn the data space using locally adaptive dimension weights. There are cases in which global weights cannot satisfy the user constraints (e.g., if there are two rings of points and the user asks for separating the inside from outside ring). In this case, techniques for learning local weights are needed. Thus we aim to handle more efficiently the high-dimensional datasets containing clusters of nonstandard geometries.

ACKNOWLEDGMENTS

We would like to thank Eamonn Keogh for his valuable comments and Sugato Basu for providing us with the code for MPCKmeans.

REFERENCES

- AGGARWAL, C., PROCOPIUC, C., WOLF, J., YU, P., AND PARK, J. 1999. Fast algorithms for projected clustering. In *Proceedings of the ACM International Conference on Management of Data*.
- AGGARWAL, C. AND YU, P. 2000. Finding generalized projected clusters in high dimensional spaces. In *Proceedings of the ACM International Conference on Management of Data*.
- AGRAWAL, R., GEHRKE, J., GUNOPULOS, D., AND RAGHAVAN, P. 1998. Automatic subspace clustering of high dimensional data for data mining applications. In *Proceedings of the ACM International Conference on Management of Data*.
- ANDERSON, B., MOORE, A., AND COHN, D. 2000. A nonparametric approach to noisy and costly optimization. In *Proceedings of the International Conference on Machine Learning*.
- BAR-HILLEL, A., HERTZ, T., SHENTAL, N., AND WEINSHALL, D. 2003. Learning distance function using equivalence relations. In *Proceedings of the International Conference on Machine Learning (ICML)*.
- BASU, S., BILENKO, M., AND MOONEY, R. 2004. A probabilistic framework for semi-supervised clustering. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.
- BERRY, M. AND LINOFF, G. 1996. *Data Mining Techniques for Marketing: Sale and Customer Support*. John Wiley and Sons.
- BILENKO, M., BASU, S., AND MOONEY, R. J. 2004. Integrating constraints and metric learning in semi-supervised clustering. In *Proceedings of the International Conference on Machine Learning*.
- BLANSCH, A., GANARSKI, P., AND KORCZAK, J. 2006. Maclaw: A modular approach for clustering with local attribute weighting. *Pattern Recogn. Lett.* 27, 11 (Aug.), 1299–1306.
- BLUM, A. AND MITCHELL, T. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of the Annual Conference on Computational Learning Theory*, 92–100.
- COHN, D., CARUANA, R., AND MCCALLUM, A. 2003. Semi-Supervised clustering with user feedback. Tech. Rep. TR2003-1892, Cornell University, Ithaca, NY.
- ESTER, M., KRIEGEL, H.-P., SENDER, J., AND XU, X. 1997. Sensity-Connected sets and their application for trend detection in spatial databases. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 10–15.
- FAYYAD, U. G., PIATESKY-SHAPIRO, P. S., AND UTHURUSAMY, R. 1996. *Advances in Knowledge Discovery and Data Mining*. AAI Press.
- FISHER AND DOUGLAS. 1987. Knowledge acquisition via incremental conceptual clustering. *Mach. Learn.* 2, 139–172.
- FRIGUI, H. AND NASRAOUI, O. 2004. Unsupervised learning of prototypes and attribute weights. *Pattern Recogn.* 37, 3, 943–952.

- GAO, J., TAN, P.-N., AND CHENG, H. 2005. Semi-Supervised fuzzy clustering with pairwise-constrained competitive agglomeration. In *IEEE Conference on Fuzzy Systems*.
- HALKIDI, M., GUNOPOLOS, D., KUMAR, N., VAZIRGIANNIS, M., AND DOMENICONI, C. 2005. A framework for semi-supervised learning based on subjective and objective clustering criteria. In *Proceedings of the IEEE Conference on Data Mining (ICDM)*.
- HALKIDI, M. AND VAZIRGIANNIS, M. 2001. Clustering validity assessment: Finding the optimal partitioning of a data set. In *Proceedings of the IEEE Conference on Data Mining (ICDM)*.
- HINNEBURG, A. AND KEIM, D. 1998. An efficient approach to clustering in large multimedia databases with noise. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 58–65.
- HOGG, R. AND CRAIG, A. 1978. *Introduction to Mathematical Statistics*. Macmillan, New York.
- HUBERT, L. AND ARABIE, P. 1985. Comparing partitions. *J. Classif.*
- JAIN, A., MUTTY, M., AND FLYN, P. 1999. Data clustering: A review. *ACM Comput. Surv.* 31, 3.
- JING, L., NG, M., AND HUANG, J. X. 2005. Subspace clustering of text documents with feature weighting k-means algorithm. In *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD). Advances in Knowledge Discovery and Data Mining, Lecture Notes in Computer Science*, vol. 3518. Springer, Berlin.
- KULIS, B., BASU, S., DHILLON, I., AND MOONEY, R. 2005. Semi-Supervised graph clustering: A kernel approach. In *Proceedings of the International Conference on Machine Learning (ICML)*.
- MACQUEEN, J. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Symposium on Math, Statistics and Probability*, University of California Press, Berkeley, CA, 281–297.
- NIGAM, K., MCCALLUM, K., THRUN, S., AND MITCHELL, T. 2000. Text classification labeled and unlabeled documents using em. *Mach. Learn.* 39, 103–134.
- PRESS, W. H., TEUKOLSKY, S. A., VETTERLING, W. T., AND FLANNERY, B. P. 1997. *Numerical Recipes in C, the Art of Scientific Computing*. Cambridge University Press.
- SEGAL, E., WANG, H., AND KOLLER, D. 2003. Discovering molecular pathways from protein interaction and gene expression data. *Bioinformatics* 19, 264–272.
- STEIN, B., ZU EISSEN, S. M., AND WIBROCK, F. 2003. On cluster validity and the information need of users. In *Proceedings of the Artificial Intelligence and Applications Conference*.
- WAGSTAFF, K. AND CARDIE. 2000. Clustering with instance-level constraints. In *Proceedings of the International Conference on Machine Learning (ICML)*.
- WAGSTAFF, K., CARDIE, C., ROGERS, S., AND SCHROEDL, S. 2001. Constrained k-means clustering with background knowledge. In *Proceedings of the International Conference on Machine Learning (ICML)*. 577–584.
- XING, E. P., NG, A. Y., JORDAN, M. I., AND RUSSELL, S. 2002. Distance metric learning, with application to clustering with side-information. In *Proceedings of the Conference on Advances in Neural Information Processing Systems (NIPS)*.
- YIP, K., CHEUNG, D., AND NG, M. 2005. On discovery of extremely low-dimensional clusters using semi-supervised projected clustering. In *Proceedings of the 21st International Conference on Data Engineering*, 329–240.

Received August 2006; revised March 2007; accepted August 2007