Available online at www.sciencedirect.com
ScienceDirect
International Journal of Approximate Reasoning
47 (2008) 45–57
INTERNATIONAL JOURNAL OF APPROXIMATE REASONING
www.elsevier.com/locate/ijar

Improving fuzzy clustering of biological data by metric learning with side information

Michele Ceccarelli *, Antonio Maratea

Research Centre on Software Technology, RCOST University of Sannio, Via Traiano 11, 82100 Benevento, Italy

Received 20 April 2006; received in revised form 20 October 2006; accepted 15 March 2007
Available online 11 April 2007

Abstract

Semi Supervised methods use a small amount of auxiliary information as a guide in the learning process in presence of unlabeled data. When using a clustering algorithm, the auxiliary information has the form of side information, that is a list of co-clustered points. Recent literature shows better performance of these methods with respect to totally unsupervised ones even with a small amount of side information. This fact suggests that the use of Semi Supervised methods may be useful especially in very difficult and noisy tasks where little a priori information is available, as is the case of data deriving from biological experiments. The two more frequently used paradigms to include side information into clustering are Constrained Clustering and Metric Learning. In this paper we use a Metric Learning approach as a way to improve the classical fuzzy c-means clustering through a two steps procedure: first a series of metrics (one for each cluster) that satisfy a randomly generated set of constraints are learnt based on the data; then a generalized version of the fuzzy c-means (with the metrics computed in the previous step) is executed. We show the benefits and the limitations of this method using real world datasets and a modified version of the Partition Entropy index.

Keywords: Semi Supervised learning; Fuzzy clustering; Simulated annealing; Adaptive metric; Validity index

1. Introduction

Both classification and clustering are well formalized problems in the Machine Learning domain that find natural application in countless other disciplines. Classification is usually thought a general example of supervised learning while Clustering is thought as a general example of unsupervised learning, the choice between the two being largely determined by the available a priori information. When data labels are available the natural way to include them in the analysis process is through a supervised technique; when they are not the unsupervised methods often have no alternative. There is an intermediate case however: when we have some kind of a priori information that is not so strong to be converted into labels. To model and to use such information can be precious and loosing it with a totally unsupervised method is a waste of resources for the data analyst.

* Corresponding author.
E-mail addresses: ceccarelli@unisannio.it (M. Ceccarelli), amaratea@unisannio.it (A. Maratea).

0888-613X/$ - see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.ijar.2007.03.008

Semi Supervised are methods that use a small amount of auxiliary information to guide the other techniques, providing the data analyst with a more flexible tool to use all the available a priori knowledge. There are many reasons for considering Semi Supervised methods: often labeled data are expensive or impossible to obtain (whereas unlabeled data are abundant and easy to collect); often some a priori information is available or easily obtainable from unlabeled data; often labeling is based on human experts judgments and so is prone to errors and subjectivity, especially in presence of very difficult tasks.

In computational biology, recent techniques as Microarray Chips produce a wealth of data that need to be analyzed and interpreted. In such experiments, the level of mRNA expression of thousands of genes in a cell is simultaneously measured in various experimental conditions. The result is usually presented in form of a matrix, whose columns are the various experimental conditions (time evolution, case/control, etc.) and whose rows are the genes fragments spotted on the chips. Pattern analysis and machine learning methods are extensively used to gain insights into biological phenomena and to extract genetic information coded in the DNA chips [6,7,10,11,16]. Thanks to the application of automatic classification methods, successful results in the understanding of genes roles and interactions have been reached, although there is no literature's agreement on a general method that would work outside the tested datasets. Due to the complexity of the underlying phenomena, the results of functional genomics experiments are very hard to be validated without the aid of a well trained expert of the field. More, it is not rare the case that the dynamic of the underlying phenomenon is largely unknown and its complexity is such that the kind of information that an expert can give is necessarily intrinsically imprecise. An expert may have a general idea of what will happen, but may not be able to give a precise indication of classes. On the other side often there are genes that are biologically known to be involved in the same process under certain conditions. Using conventional unsupervised techniques this information is lost, producing poor results due to the algorithms' inability to recognize genes' correlations that are evident for an expert of the field. Such an expert may be able to indicate explicitly at least a reduced list of genes known to be "similar" and some others genes known to be "dissimilar" in a given experiment. Recent Literature shows that even a little auxiliary information can help the Classification or Clustering algorithms to reach meaningful results [3,8,17] and we believe that a fundamental step towards the availability of new and more powerful tools to analyze this kind of data is the inclusion in automatic procedures of the available a priori knowledge, not necessarily in the form of labels, supplied by the field's experts [4].

1.1. Structure of the paper

In Section 3 we formalize the metric learning problem, we show a parametrization of the side information and we introduce both the classical fuzzy c-means and the Semi Supervised Fuzzy C-means (SSFC) algorithms. Then we describe the experiments performed to evaluate SSFC's efficacy and sensitivity to the amount of auxiliary information, comparing it with both classical fuzzy c-means and other classical Semi Supervised methods; further we describe the data and the validation tools we used to make our conclusions. In Section 4 we show the obtained graphs and we state the results that are summarized in Section 5.

2. Related work

The two more frequently used paradigms to include side information into clustering are Constrained Clustering [14,18] and Metric Learning [1,3,5,8,15,17]. In the former case the objective function of a clustering algorithm is modified to include a penalty for wrongly classified points, while in the latter a suitable metric that makes similar points be closer and dissimilar points be farther away is learned prior to clustering. The first class of problems is relatively old and includes regionalization problems, a topic that will not be further discussed here. The second class is more recent and more flexible in the choice of the metric function. Among previous works in metric learning methods, in [17] a Mahalanobis distance is learned using convex optimization. This is a very effective approach, although severely limited in application to real data by the computational complexity $O(d^6)$, where d is the dimension of data. With respect to our method, it uses only one metric function for all clusters and does not account for fuzziness. In [8] the metric is learned considering pairs of samples belonging to the same class and the computational complexity is reduced to $O(d^3)$ showing similar results. Here the main problem is in the computation of the generalized eigenvalue problem, that may become

unstable. With respect to our method it does not consider pairs of cannot link ties and does not provide for fuzziness. In [3] a k-means family algorithm that joins metric learning and clustering in the same step is proposed. It is more general with respect to [8,17] because it considers a different metric for each cluster and so allows for clusters of different shape. The main problems of this approach are the computational cost and the dependency on the order of data (hence on the initialization) in the E-step. With respect to our method, it uses also a different metric for each cluster, but it does not account for fuzziness. We use a two steps approach, separating metric learning from clustering. We also use a different metric for each cluster and we generalize further to fuzzy clustering. Optimization in the learning step is done through an heuristic algorithm.

3. Materials and methods

3.1. Metric learning

The choice of a metric for a given experiment is not an easy task. Often a specific metric is suitable for some data and completely unsuitable for other, or worse it's suitable for some variables or experiments and completely inadequate for others on the same data. If the data analyst has a list of constraints that wishes the metric to fulfill, the problem is even more complicated. One solution could be to manually "adapt" the metric to the data or another to automatically "learn" from data a metric that respects these constraints. If constraints are formulated in a way that expresses the substantive knowledge of the phenomenon under study, the metric consequently learned can be thought as representing a way to include the a priori available information into the analysis performed. This can be done as a preliminary step to many classification and clustering algorithms.

In a general framework, we call $Z = \{x_1, \ldots, x_n\}$ the set of data consisting of n d-dimensional data points, $S \subset Z \times Z$ a set of pairs of similar points and $D \subset Z \times Z$ a set of pairs of dissimilar points. In an ideal scenario, S and D are provided by an expert of the field, which a priori knows at least some pairs of co-clustered and not co-clustered data points. $(x_p, x_q) \in S$ means that the two vectors x_p and x_q are known to be in the same cluster and vice versa $(x_p, x_q) \in D$ means that the two vectors x_p and x_q are known not to be in the same cluster. We look for a function f that respects triangle inequality, non-negativity and simmetry and such that

$$f(x_p, x_q)_{(x_p,x_q)\in S} \text{ is minimized}, \tag{1}$$

$$f(x_p, x_q)_{(x_p,x_q)\in D} \text{ is maximized}, \tag{2}$$

the difficulty of this problem depends on the way f and the constraints are parametrized.

3.2. Side information

The auxiliary information can have many forms and should be modeled accordingly. For clustering, auxiliary information has generally the form of side information, that is pairs of Must-Link (ML) and Cannot-Link (CL) ties: for each dataset, an expert declares the list of points that he believes should be co-clustered and the list of points that should not. We will assume, without loss in generality, that the set ML of co-clustered points and the set CL of non-co-clustered points have the following structure:

$$ML \subset \{(x_{m_p}, x_{m_q}) | x_{m_p}, x_{m_q} \in Z;\ \ell_{m_p} = \ell_{m_q}\}, \tag{3}$$

$$CL \subset \{(x_{m_p}, x_{m_q}) | x_{m_p}, x_{m_q} \in Z;\ \ell_{m_p} \neq \ell_{m_q}\}, \tag{4}$$

where $m_p \in \{1, \ldots, n\}$ and ℓ_{m_p} is the a priori known label of point x_{m_p}.

As we will learn a different metric for each cluster, we will need to subset the ML ad CL matrices in order to have the specific side information relative to each cluster j:

$$ML_j \subset \{(x_{m_p}, x_{m_q}) | x_{m_p}, x_{m_q} \in ML;\ \ell_{m_p} = j;\ \ell_{m_q} = j\}, \tag{5}$$

$$CL_j \subset \{(x_{m_p}, x_{m_q}) | x_{m_p}, x_{m_q} \in CL;\ \ell_{m_p} \neq j;\ \ell_{m_q} \neq j\}. \tag{6}$$

While if the expert gives just the *ML* list it is always possible to generate the *CL* list from the first one, it is not true the vice versa and so it is not possible to apply a Semi Supervised method with just the list of cannot link ties. On the other side, not all methods accounts for cannot link pairs (for example [8]) and it can happen that the *CL* list is not used in the learning process.

3.3. Fuzzy c-means with ordinary euclidean metric

Fuzzy Clustering is a partition–optimization technique that aims to group data based on their similarity in a non-exclusive manner, that is permitting each sample to belong to more than one group. The strength of each sample's belonging to each group is measured through a function, called 'membership' that has values between 0 and 1 and that sums to 1 on all clusters. Values closer to 1 indicate a stronger belonging of that sample to that cluster. There are various algorithms with which grouping can be performed and one of the most used is the fuzzy c-means [2]. Main known limitations of the fuzzy c-means are that:

- it can remain trapped in local optima;
- the number of clusters and the amount of fuzziness are free parameters;
- all clusters are of hyperspherical form;
- it produces in every case a grouping, even if the data have no clustering structure.

Its objective function is

$$J_b = \sum_{i}^{n} \sum_{j}^{k} f(x_i, m_j)\mu_{ij}^{b}, \tag{7}$$

where μ_{ij} are the membership, m_j are the cluster centroids', b is the overlap parameter and f is a suitable distance function.

3.4. Fuzzy c-means with learned metric

The approach pursued in this paper is based on the algorithm proposed in [17] for metric learning with side information. The main differences are threefold:

- we learn a specific metric for each cluster;
- the learned metrics are applied for the distance computation in the fuzzy *c*-means;
- The weights' optimization process is based on a stochastic search.

The method is realized in two steps: in the first step we use the *a priori* information to "tweak" the metric f_{A_j}

$$f_{A_j}(x_p, x_q) = [(x_p - x_q)^{\mathrm{T}} A_j (x_p - x_q)]^{\frac{1}{2}}. \tag{8}$$

To gain more generality and more flexibility, we considered a different matrix for each cluster $A_j\, j = 1,\ldots,k$. To define a criterion for the metric we demand that samples declared to be "similar" have small squared distance and samples declared to be "dissimilar" have high squared distance. Let us call ML_j and CL_j respectively the sets of similar points and the set of dissimilar points in the jth cluster as defined in (5) and (6), then we pose a set of k constrained problems [17]:

$$\min_{A_j} \sum_{(x_p, x_q) \in \mathcal{M}L_j} \|x_p - x_q\|_{A_j}^2 \tag{9}$$

$$\text{subject to} \sum_{(x_p, x_q) \in \mathcal{C}L_j} \|x_p - x_q\|_{A_j} \geqslant \tau, \quad \text{and } A_j \geqslant 0, \tag{10}$$

where $j = 1,\ldots,k$ and $\tau > 0$ is an arbitrary constant, the constraint term captures the notion of between cluster dissimilarity whereas the functional to minimize captures the notion of within-cluster similarity. Here, as in

[17], we use $\tau = 1$. It can be easily shown that the problems (10) are equivalent, under the assumption of diagonal matrices A_j, to the minimization of the following k convex functionals:

$$g(A_j) = \sum_{(x_p,x_q)\in ML_j} \|x_p - x_q\|^2_{A_j} - \log\left(\sum_{(x_p,x_q)\in CL_j} \|x_p - x_q\|_{A_j}\right). \tag{11}$$

The Netwon–Raphson method has been adopted in [17], it leads to an $O(d^6)$ algorithm, where d is the dimension of data. In our case, the problem is even more complex as we have a set of k such problems, moreover in microarray experiments, the data dimension can reach several thousands, therefore here we adopted a stochastic search based minimization algorithm based on the well-known Simulated Annealing (SA) [12] method. A general schema of the algorithm in the formulation that we used follows:

- set starting temperature T_0;
- compute energy function E;
- repeat until convergence or maximum number of iteration is reached:
  - go to a neighbour state through a gaussian perturbation of the current state;
  - compute the energy variation ΔE;
  - if $\Delta E < 0$ accept the new state;
  - if $\Delta E \geqslant 0$ accept the new state with probability given by the Metropolis function;

    $$\exp^{-\Delta E/T}; \tag{12}$$

  - decrease temperature according to logarithmic cooling schedule

    $$T(u) = T_0/\log(u + \alpha), \tag{13}$$

    where u is the epoch's counter and α a free parameter.

In the second step we calculate the fuzzy c-means clustering with the more general distance metrics calculated previously.

$$J^*_b = \sum_{i}^{n}\sum_{j}^{k} f_{A_j}(x_i, m_j)\mu^b_{ij}. \tag{14}$$

The convergence of the fuzzy c-means algorithm is independent from the change in the distance function if the distances are all positive and the prototypes are calculated according to the minimization of the objective function [2]. The computational complexity of the whole procedure is

$$MaxIterSA \bigcirc (\gamma^2 n^2) + MaxIterFCM \bigcirc (nd + nk), \tag{15}$$

where $MaxIterSA$ is the maximum number of iterations of the simulated annealing, $MaxIterFCM$ is the maximum number of iterations of the fuzzy c-means and $0 < \gamma < 1$ is the fraction of the data considered as side information.

3.5. Experiments

We tested the SSFC in two directions: one was versus other Semi Supervised methods to test its efficiency and in general to evaluate its sensitivity to the amount of side information provided; the other was versus conventional Unsupervised fuzzy c-means to evaluate its efficacy. The two directions have been explored using different datasets and different criteria, mainly because of the non-fuzziness of other Semi Supervised methods. As our task was to validate the efficacy and efficiency of the various methods and not to discuss cluster number issues, we considered the known true clusters' number for each dataset. For the same reason and to remain in the most general possible framework, we obtained the two sets S and D by random uniform extraction from the true labels' vectors. In the simulated annealing algorithm we used $T_0 = 1000$ and $\alpha = 5$.

Algorithm: SSFC

*I*nput:

$Z \in \Re^{n\times d}$: *data matrix of n vectors in* $\Re^d$

k: *number of clusters*

$\ell = (\ell_{m_1}, \ldots, \ell_{m_{2s}})$ where

$\ell_{m_i} \in \{1, \ldots, k\}$, $m_i \in \{1, \ldots, n\}$ $\forall i \in \{1, \ldots, 2s\}$: *a priori known true clusters' labels*

$C \in \Re^{k\times d}$: *matrix of starting clusters' centroids*

s: *number of a priori known pairs of co-clustered points*

t: *number of a priori known pairs of non-co-clustered points*

$ML \in \Re^{2s\times d} \subset \{(x_{m_p}, x_{m_q}) | x_{m_p}, x_{m_q} \in Z; \ell_{m_p} = \ell_{m_q}\}$: *Must Link matrix*

$CL \in \Re^{2t\times d} \subset \{(x_{m_p}, x_{m_q}) | x_{m_p}, x_{m_q} \in Z; \ell_{m_p} \neq \ell_{m_q}\}$: *Cannot Link matrix*

*O*utput:

$A_j \in \Re^{d\times d} j = 1, \ldots, k$: *metrics matrices for cluster j*

$C^* \in \Re^{k\times d}$: *matrix of final clusters' centroids*

$M \in [0,1]^{n\times k}$ *such that* $\sum_{j=1}^{k} m_{ij} = 1$ $\forall i$: *membership matrix*

*M*ethod:

For each cluster $j = 1, \ldots, k$ {

$ML_j \in \Re^{2s_j\times d} \subset \{(x_{m_p}, x_{m_q}) | x_{m_p}, x_{m_q} \in ML; \ell_{m_p} = j; \ell_{m_q} = j\}$:

s_j is the number of pairs of points in the ML matrix similar to current cluster

$CL_j \in \Re^{2t_j\times d} \subset \{(x_{m_p}, x_{m_q}) | x_{m_p}, x_{m_q} \in CL; \ell_{m_p} \neq j; \ell_{m_q} \neq j\}$:

t_j is the number of pairs of points in the CL matrix dissimilar from current cluster

Optimization through Simulated Annealing of the function:

$g(A_j) = \sum_{(x_p,x_q)\in ML_j} \|x_p - x_q\|^2_{A_j} - \log(\sum_{(x_p,x_q)\in CL_j} \|x_p - x_q\|_{A_j})\}$

Perform fuzzy c-means with the k metrics learnt in the previous step

3.5.1. Testing against other Semi Supervised methods

For each dataset, the data are randomly split in Train and Test (see Section 3.7). The Train set is used to perform the metrics' learning and the Test set to evaluate the performances of the algorithms with the learned metrics. On the same Train set, an increasing amount of side information is considered extracting uniformly and randomly pairs of points with the same label. This was done to evaluate the method's sensitivity to the amount of side information provided. So each clustering algorithm is executed many times on the same Train set, with the same random choice of initial centroids and an increasing amount of side information (starting from 1% to circa 35% of all Train data). For each different amount of side information, the matrix ML is generated choosing randomly an appropriate number of points from the Train set and the matrix CL is generated choosing randomly half of the points from the ML matrix. Finally, after metrics have been learnt and clustering algorithms executed on the Train set, the final centroids and learned metrics are used to cluster the Test set data on the base of a maximum proximity rule: each point is assigned to the cluster whose centroid is the closest, according to relative metric. An F-measure of concordance with the true solution is then calculated.

To make an effective and meaningful comparison among the various algorithms, as they have many different features, for each run we considered the following set of conditions:

- same splitting in Train and Test;
- same centroids;
- same ML matrix;
- same CL matrix (when used);
- same cross-validation schema;
- same quality measure of results;
- same starting metrics;
- same number of maximum iterations for the convergence check;
- same epsilon for the convergence check;
- no regularization.

The other methods used for comparison are classical k-means (KM), [17] (XJ) and [8] (BC). For the fuzziness parameter in the SSFC we found that a fixed value of 1.5 produced good results. Apart [8] metrics have been considered diagonal.

About the derivation of the CL list from the ML list, we choose randomly 30% of the ML points to fill the first half of the CL set. Then we filled the remaining half of the CL set adding points dissimilar from the corresponding first half, extracting them from the ML list if available or from the Train Set otherwise.

3.5.2. Testing against conventional fuzzy c-means

For each dataset, the full data have been clustered both with classical fuzzy c-means and SSFC. The side info matrix ML has been generated in the fixed amount of 10% of all data choosing randomly a certain number of points from the full dataset in such a way that each true cluster had at least a couple of representatives. The side info matrix CL was generated choosing randomly half of the points from the ML matrix. The initial centroids were generated randomly and independently and the overlap index b was chosen to be 1.5 for both algorithms. To Compare memberships'distributions, a modified version of the Partition Entropy index was calculated (see Section 3.7).

3.6. Data

Most of the data used for the experiments have been obtained from the UCI Machine Learning repository [13] and represent typical problems from different disciplines. All the data are fully labeled, that means the true class of each row is known as the true number of classes. Where suitable, a log-transformation has been applied. The characteristics of single datasets are resumed in Table 1.

3.6.1. Data for comparison against other Semi Supervised methods

- The ionosphere dataset I consists of datafrom a phased array of 16 high-frequency antennas trying to catch free electrons in the ionosphere. The class is a binary variable telling if they succeeded.
- The iris dataset IR is the famous dataset of Fisher that contains measurements of three species of iris plants.
- The Winsconsin Diagnostic Breast Cancer dataset WDBC contains characteristics of the cell nuclei of various Breast Cancer extracted from an image and the relative diagnosis (Benignant/Malignant). A few of the images can be found at http://www.cs.wisc.edu/street/images/.
- The Wine dataset W contains various chemical analysis of wines grown in the same region in Italy but derived from three different cultivars.

3.6.2. Data for comparison against classical fuzzy c-means

- The Yeast dataset Y has 1484 rows and 10 columns of attributes that are a series of measurements to establish the localization site of proteins. Last column is the localization site.
- The Rat dataset R is the data set of Wen, Fuhrman, Michaels, Carr, Smith, Barker and Somogyi that measures the mRNA expression levels of 112 genes during rat central nervous system development from embryonic through postnatal to adult stage (nine stages). Last column is the functional classification of genes.

Table 1
Datasets used in the experiments

Dataset	No. of instances	No. of features	No. of classes
Ionosphere (I)	351	34	2
Iris (IR)	150	4	3
Winsconsin Diagnostic Breast Cancer (WDBC)	569	30	2
Wine (W)	178	13	3
Rat (R)	112	9	4
Sporulation (S)	477	7	7
Yeast (Y)	1484	8	10

- The Sporulation dataset S is the Spellmann dataset of Yeast expression levels [16]. It has 6118 rows and 86 columns. We selected only the 477 rows that we know to be cell-cycle regulated and the seven columns of g/r ratios.

3.7. Validation

3.7.1. Validation of comparisons against other Semi Supervised methods

To have a reliable estimate of each method's average performance we computed an index of concordance with the true solution and we did K-fold cross validation. The index of concordance used is a traditional information retrieval measure: the F-measure:

$$\begin{aligned}\text{Precision} &= \frac{\text{Pairs Correctly Predicted In Same Cluster}}{\text{Total Pairs Predicted In Same Cluster}},\\ \text{Recall} &= \frac{\text{Pairs Correctly Predicted In Same Cluster}}{\text{Total Pairs In Same Cluster}},\\ F-\text{measure} &= \frac{2\times\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}}.\end{aligned} \tag{16}$$

F-measure is a pairwise measure and needs a hard partition of data to be computed. We did the final assignment of points to clusters on the base of the crude maximum proximity rule, so in the case of SSFC, we did actually a defuzzification. Even if in this way we lost the major expressiveness of membership functions, these are still considered in the training process and they help the algorithm to account for all the data.

K-fold cross validation has been implemented following this procedure: a random permutation of the dataset has been split in K parts of equal dimensions. On turn, $K-1$ part have been chosen as "Train" and the remaining part has been chosen as "Test". The Train dataset has been used to learn the metrics and to update the clusters' centroids of each specific algorithm tested, until convergence. Then the centroids and the metrics learned on the Train Dataset have been used to assign points to clusters in the Test Dataset. The assignment of points to classes in the Test Dataset has been done based on a max proximity rule, using the centroids and the metrics learned on the Train Dataset to compute distances. Just the points in the Test Dataset have been used to actually compute the F measure quality index of the solution and results have been averaged on all the K different solutions. After various trials the choice of $K=3$ seemed the most reliable.

3.7.2. Validation of comparisons against classical fuzzy c-means

In a well defined fuzzy clustering the first memberships should be much higher than the others, reflecting scarce ambiguity and good model's matching to the data structure. Considering the list of sorted memberships for each sample it is obvious that a more pronounced asymmetry towards higher values indicates a better defined clustering [9]. There are various possibilities to express quantitatively this fact. If we assimilate membership to probabilities it is possible to use the Entropy Index as a quantitative measure of asymmetry. The mean value on the whole dataset of the Entropy Index is known as Partition Entropy [2] and has the following form:

$$PE = \frac{1}{n}\sum_{i}^{n}\sum_{j}^{k}\mu_{ij}\log_a\mu_{ij}, \tag{17}$$

where n is the number of points in the dataset, μ_{ij} are the membership and k is the clusters' number. Lower values indicate more asymmetric partitions. This is a well-known index based only on memberships' values that shows an increasing trend with the number of clusters, mainly because the membership tends to spread over clusters. One way to enhance index's sensitivity to higher memberships' values and to avoid this limitation is to raise the memberships to the power p and to normalize the new values so that they sum to one. The higher the power, the higher index's sensitivity.

$$PEm_p = \frac{1}{n}\sum_{i}^{n}\sum_{j}^{k}\hat{\mu}_{ij}^{p}\log_a\hat{\mu}_{ij}^{p}, \tag{18}$$

where $\hat{\mu}_{ij}^{p}$ are memberships raised to power p and normalized.

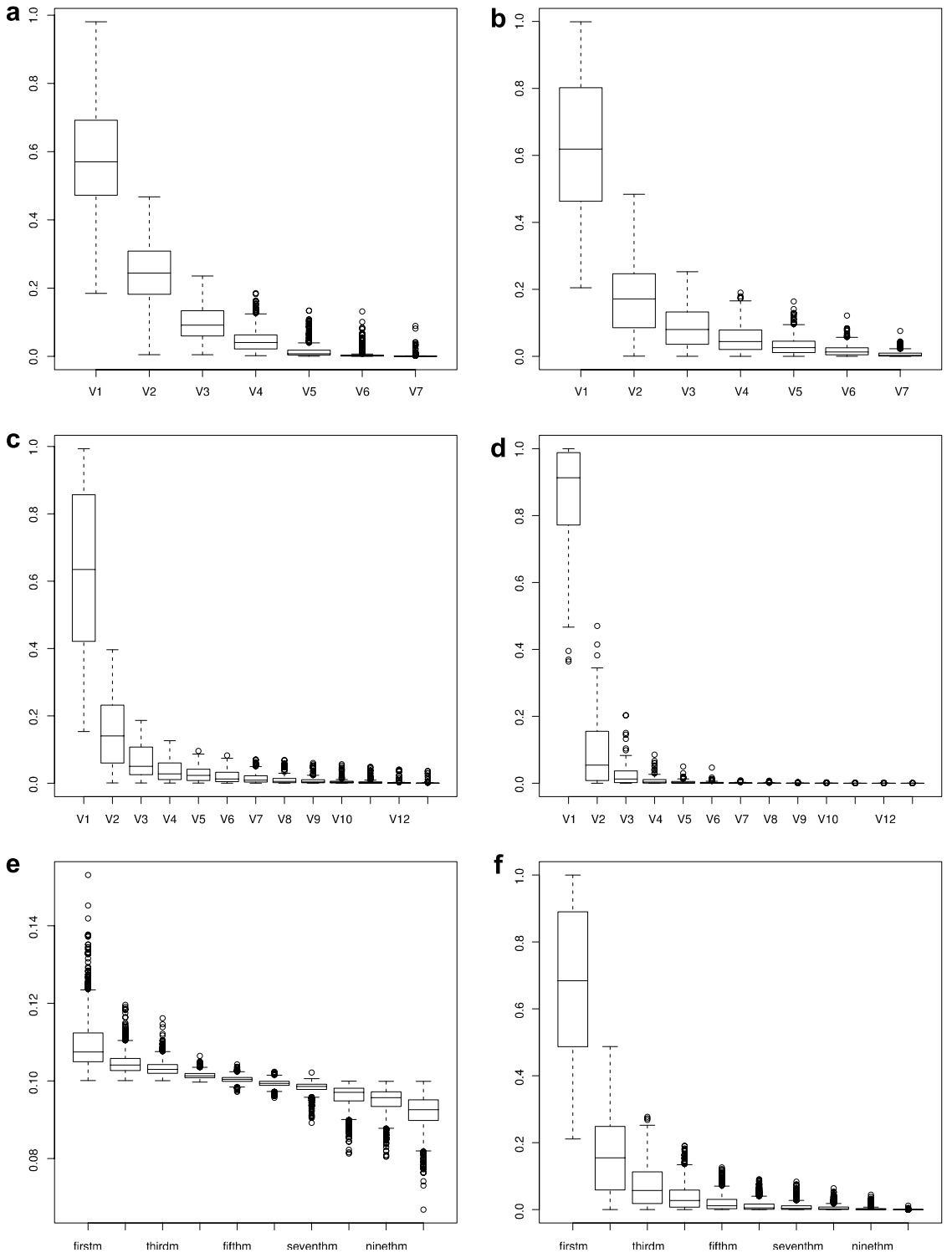

Fig. 1. *Boxplots* of ordered memberships of the three datasets analyzed. On the left data are clustered with conventional FCM and on the right data are clustered with our algorithm. (a) and (b) are referred to the Sporulation dataset; (c) and (d) to the Rat dataset; (e) and (f) to the Yeast dataset.

4. Results

Graphs of the *F*-measure on the Test set for various Semi Supervised methods (a regular *K* means is also shown as a reference) are shown in Fig. 3. On the *x* axis there is the percentage of data that has been used as

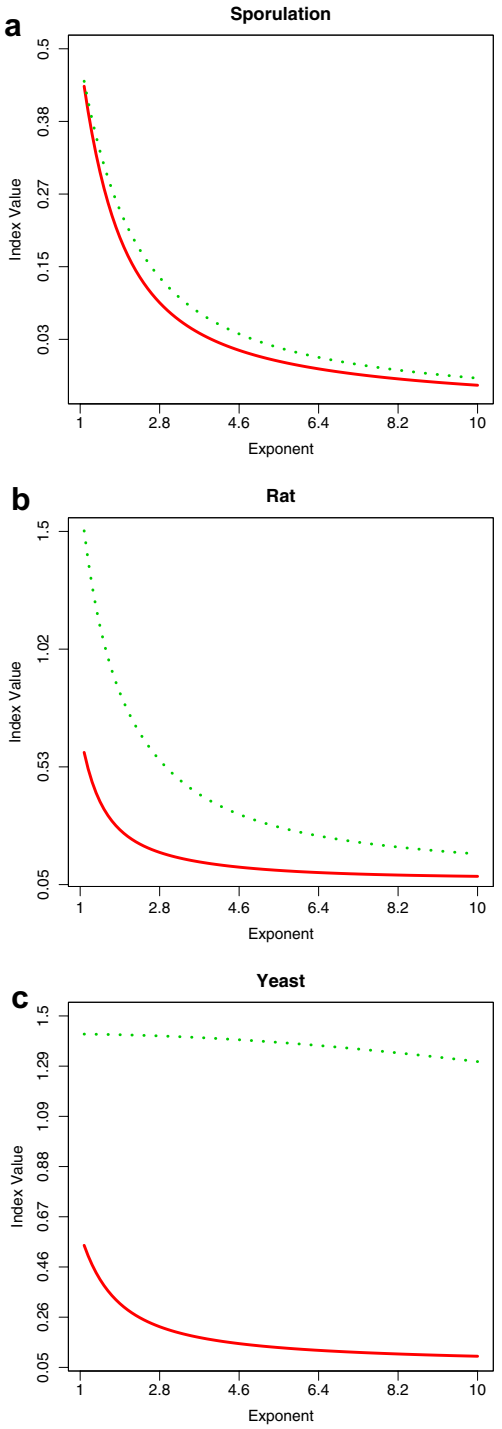

Fig. 2. Plots of *PEm* Validity Index for conventional fuzzy c-means (dotted line) and for SSFC (solid line) on the three datasets analyzed, in function of power *p*.

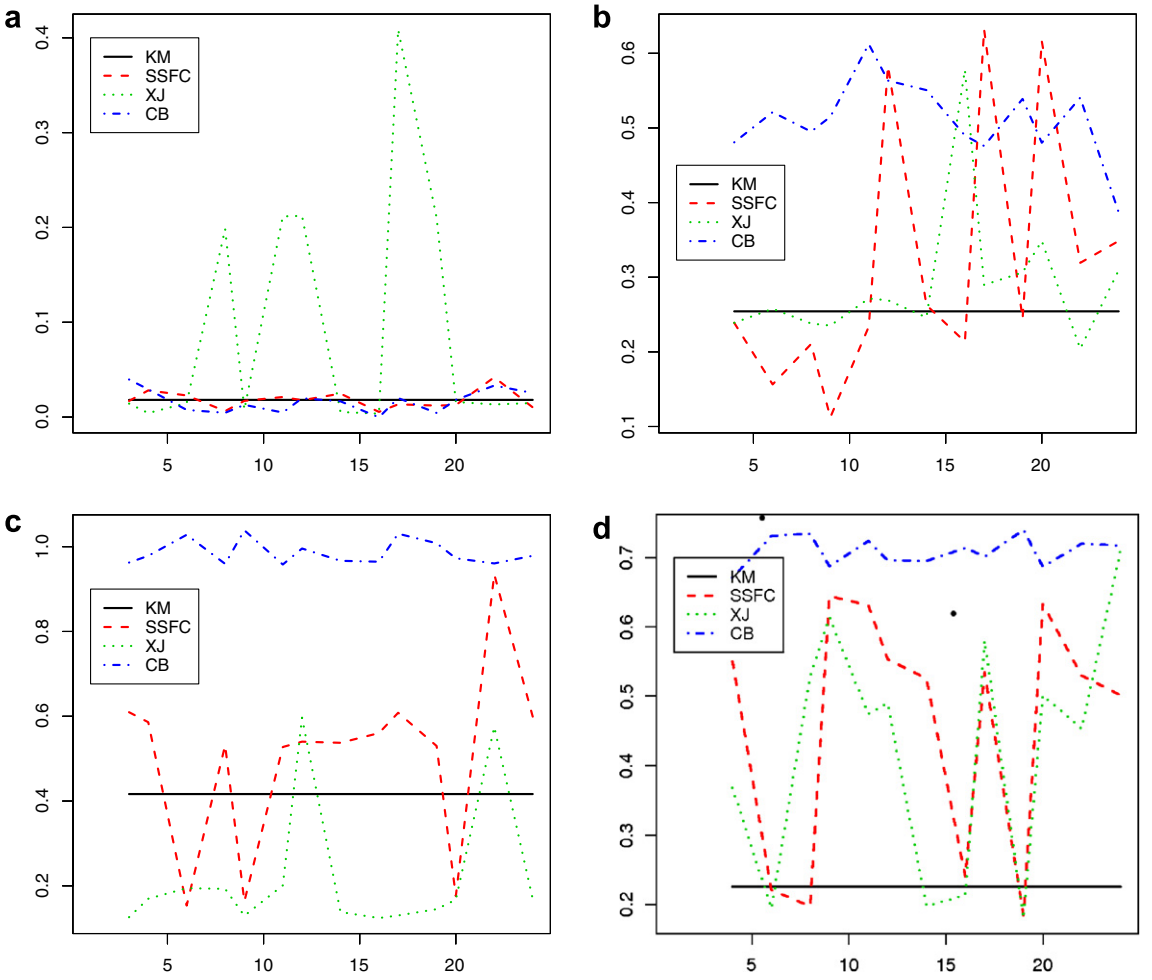

Fig. 3. Average *K*-fold *F*-measure calculated in the Test dataset (see Section 3.7) for increasing percentage of side info (a) side info (b) iris, (c) WDBC, (d) wine.

side info. We start from really a few points (provided that each cluster is represented, otherwise the computation is skipped), until a maximum of 25% of all data. The first thing to note is that there is not a regular increment of the performance with the increase of side info as one may expect. This happens mainly because of the "blind" choice of the side info points at each run. Choosing points randomly, the ones that are close together but belong to different clusters happen to be in the *CL* list, or vice versa the ones that are further away but belong to the same cluster happen to be in the *ML* list. Another frequent artifact is that a cluster is over-represented while another is underrepresented. All these circumstances alternate the learning process often leading to unstable results. However, in spite of these warnings, in average Semi Supervised methods perform better than crude *k*-means, even if with very difficult datasets (as ionoshpere) the difference is not clearly marked. Results suggest that side info points should be chosen carefully and with a solid knowledge-based reason to do so, because their inclusion in the analysis is not a priori beneficial.

Boxplots of sorted memberships for all dataset tested are shown in Fig. 1, in the case of conventional fuzzy c-means and in the case of SSFC. As we can see, in all cases our modified fuzzy c-means algorithm produces more asymmetric memberships. We can see that the ordered memberships are much more asymmetric in the case of the "learned" metric to prove a much more evident clustering structure on the data. This is particularly evident for the Yeast dataset (Fig. 1(e) and (f)) an it's a bit more attenuated for the

Rat dataset (Fig. 1(c) and (d)). It is less evident in the Sporulation dataset (Fig. 1(a) and (b)) but however there is still an improvement.

In all tested cases we have plotted the PEm indexes' values versus the power p from 1 to 10 in 0.1 steps, comparing the two algorithms, classical fuzzy c-means and SSFS (Fig. 2). As we can see the power p affects index performances and makes it much more sensible to dataset clustering structure. Specifically, in the Sporulation case, classical index *PE* fails in highlighting the distribution improvement due to the metric learning, while starting from the power of 2 the modified index *PEm* reaches the task. In all tested cases, the index *PEm* reflects distribution improvement due to metric learning and seems to be more reliable and sensible with respect to the original index *PE*.

5. Conclusions

We have shown the efficacy and the limitations of using side information in unsupervised techniques. Learning a metric that respects some user-defined constraints as a preliminary step to clustering, under the right conditions, improve clearly its performance, extending utilization possibilities to more difficult tasks without substantial changes to the technique. We used some classical real dataset to validate our method against the well known fuzzy c-means algorithm and against other Semi Supervised techniques. Results suggest the opportunity to choose with care the side info points to be included in the analysis, as a wrong choice or a blind generation may not produce a stable solution. On the other side, in average, there is and advantage in using Semi Supervised techniques both with respect to crisp k-means than to fuzzy c-means. In the fuzzy case, we quantified the advantage of using side information through a generalized version of the Partition Entropy index. The membership distribution shows a clear improvement in detecting the clustering structure for tested data. Future work is in studying more complex distance functions and in finding optimality criteria for the side info points.

References

[1] A. Bar-Hillel, T. Hertz, N. Shental, D. Weinshall, Learning a mahalanobis metric from equivalence constraints, Journal of Machine Learning Research 6 (2005) 937–965.
[2] J.C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press, NY, 1981.
[3] M. Bilenko, S. Basu, R.J. Mooney, Integrating constraints and metric learning in semi-supervised clustering, in: Proceedings of the 21st ICML, 2004, pp. 81–88.
[4] M. Ceccarelli, A. Maratea, Semi-supervised fuzzy c-means for the analysis of biological data, Lecture Notes in Artificial Intelligence 3849 (2005) 259–266.
[5] H. Chang, D.Y. Yeung, Locally linear metric adaptation for semi-supervised clustering, in: Proceedings of the 21st ICML, 2004, pp. 153–160.
[6] R.J. Cho, M.J. Campbell, E.A. Winzeler, L. Steinmetz, A. Conway, L. Wodicka, T.G. Wolfsberg, A.E. Gabrielian, D. Landsman, D.J. Lockhart, R.W. Davis, A genome-wide transcriptional analysis of the mitotic cell cycle, Molecular Cell 2 (1998) 65–73.
[7] S. Chu, J. DeRisi, M. Eisen, J. Mulholland, D. Botstein, P.O. Brown, I. Herskowitz, The transcriptional program of sporulation in budding yeast, Science 282 (1998) 699–705.
[8] T. De Bie, M. Momma, N. Cristianini, Efficiently learning the metric with side-information, Lecture Notes in Artificial Intelligence 2842 (2003) 175–189.
[9] D. Demb, P. Kastner, Fuzzy c-means method for clustering microarray data, Bioinformatics 19 (2003) 973–980.
[10] M.B. Eisen, P.T. Spellman, P.O. Brown, D. Botstein, Cluster analysis and display of genome-wide expression patterns, PNAS 95 (1998) 14863–14868.
[11] T.R. Golub, D.K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J.P. Mesirov, H. Coller, M.L. Loh, J.R. Downing, M.A. Caligiuri, C.D. Bloomfield, E.S. Lander, Molecular classification of cancer: class discovery and class prediction by gene expression monitoring, Science 286 (1999) 531–537.
[12] S. Kirkpatrick, C.D. Gelatt Jr., M.P. Vecchi, Optimization by Simulated Annealing, Science 220 (1983) 671–680.
[13] D.J. Newman, S. Hettich, C.L. Blake, C.J. Merz, UCI Repository of machine learning databases, http://www.ics.uci.edu/mlearn/MLRepository.html, Irvine, CA: University of California, Department of Information and Computer Science, 1998.
[14] N. Shental, A. Bar-Hillel, T. Hertz, D. Weinshall, Computing Gaussian mixture models with EM using equivalence constraints, in: Proceedings of Neural Information Processing Systems 2003, vol. 16, 2003.
[15] M. Schultz, T. Joachims, Learning a distance metric from relative comparisons, in: Proceedings of Neural Information Processing Systems 2003, vol. 16, 2003.

[16] P.T. Spellman, G. Sherlock, M.Q. Zhang, V.R. Iyer, K. Anders, M.B. Eisen, P.O. Brown, D. Botstein, B. Futcher, Comprehensive identification of cell cycle-regulated genes of the yeast Saccaromyces Cervesiae by microarray hybridization, Molecular Biology of the Cell 9 (1998) 3273–3297.
[17] E.P. Xing, A.Y. Ng, M.I. Jordan, S. Russell, Distance metric learning, with application to clustering with side-information, Advances in Neural Information Processing Systems 15 (2002).
[18] L. Zhengdong, T. Leen, Semi-supervised learning with penalized probabilistic clustering, in: Proceedings of Neural Information Processing Systems 2004, vol. 17, 2004.