Semi-Supervised Fuzzy Clustering with Pairwise-Constrained Competitive Agglomeration

Nizar Grira, Michel Crucianu and Nozha Boujemaa
INRIA Rocquencourt
Domaine de Voluceau, BP 105
F-78153 Le Chesnay Cedex, France
email: {Nizar.Grira, Michel.Crucianu, Nozha.Boujemaa}@inria.fr

***Abstract*— Traditional clustering algorithms usually rely on a pre-defined similarity measure between unlabelled data to attempt to identify natural classes of items. When compared to what a human expert would provide on the same data, the results obtained may be disappointing if the similarity measure employed by the system is too different from the one a human would use. To obtain clusters fitting user expectations better, we can exploit, in addition to the unlabelled data, some limited form of supervision, such as constraints specifying whether two data items belong to a same cluster or not. The resulting approach is called semi-supervised clustering. In this paper, we put forward a new semi-supervised clustering algorithm, Pairwise-Constrained Competitive Agglomeration: clustering is performed by minimizing a competitive agglomeration cost function with a fuzzy term corresponding to the violation of constraints. We present comparisons performed on a simple benchmark and on an image database.**

I. Introduction

As image collections become ever larger, effective access to their content requires a meaningful categorization of the images. Such a categorization can rely on clustering methods working on image features, but should greatly benefit from any form of supervision the user can provide, related to the visual content. Consequently, a new approach called *semi-supervised clustering*—clustering using both labelled and unlabelled data— has become a topic of significant interest. The semi-supervised clustering is between totally unsupervised clustering and fully supervised learning.

Unsupervised clustering takes an unlabelled set of data and partitions it into groups of examples, without additional knowledge, such that examples within a cluster are more ”similar” to each other than they are to data items in other clusters. Much work was dedicated to unsupervised learning and resulting algorithms can be grouped into two main categories: hierarchical or partitional.

The partitional algorithms are based on the optimization of specific objective functions and the most widely used algorithm is Fuzzy C-Means (FCM) [3], which has been constantly improved for twenty years by:

- the use of a manually engineered similarity criteria that yield good partitional of data for a given domain (e.g. the Mahalanobis distance [7]),
- the adjunction of a noise cluster [10],
- the use of competitive agglomeration [2], [6].

Due to their simplicity and computational efficiency, prototype-based clustering algorithms are very popular.

Supervised learning, on the other hand, assumes that the class structure is already known. It takes a set of examples with class labels, and returns a function that maps examples to class labels (e.g. support vector machines [11], [15]).

The main goal of the semi-supervised clustering approach is to allow a human to bias clustering with a minimum of effort by providing a small amount of knowledge concerning either class labels for some items or pairwise constraints between data items. The constraints specify whether two data items should be in the same cluster or not.

However, the few existing semi-supervised clustering algorithms, such as Pairwise Constrained K-Means (PCKmeans) [1] and Constrained K-Means [12], rely on parameters that are difficult to choose (such as the number of clusters) and require a high number of constraints to reach good results. The new semi-supervised clustering algorithm we put forward in the following, Pairwise Constrained Competitive Agglomeration (PCCA), provides solutions to these problems.

The organisation of the rest of the paper is as follows. Section II presents the background of our work. Our method is presented in section III. The results are discussed and compared with other clustering methods in section IV, while section V summarizes our concluding remarks.

II. Background

The quality of any clustering depends on how well the metric matches the user's similarity model. The Competitive Agglomeration (CA) algorithm [2] is a of the fuzzy partitional algorithm that allows the user not to specify the number of clusters [7], [10]. CA supports multiple metrics, allowing significant variations in the shapes of the clusters.

Let $X = \{\mathbf{x}_i |\ i \in \{1,..,N\}\}$ be the N feature points to cluster, $V = \{\mu_k |\ k \in \{1,..,C\}\}$ the prototypes of the C clusters and U the set of membership degrees. The objective function minimized by CA is:

$$\mathcal{J}(V,U) = \sum_{k=1}^{C}\sum_{i=1}^{N}(u_{ik})^2 d^2(\mathbf{x}_i,\mu_k) - \beta \sum_{k=1}^{C}\Big[\sum_{i=1}^{N}(u_{ik})\Big]^2 \quad (1)$$

under the constraint:

$$\sum_{k=1}^{C} u_{ik} = 1, \text{ for } i \in \{1,..,N\} \tag{2}$$

In (1), $d(\mathbf{x}_i, \mu_k)$ represents the distance between a vector $\mathbf{x}_i$ and a cluster prototype μ_k (for spherical clusters, Euclidean distance will be used) and u_{ik} is the membership of $\mathbf{x}_i$ to a cluster k. The first term is the standard FCM objective function [3]: the sum of weighted square distances. The second term progressively reduces the number of clusters.

III. Pairwise-Constrained Competitive Agglomeration

A. Semi-supervised Clustering

Existing semi-supervised clustering algorithm can be grouped into two main categories [13]: search-based and similarity-based approaches. In similarity-based approaches, the similarity metric used to retrieve clusters is first trained to satisfy the constraints in the supervised data (e.g. Mahalanobis distance trained using convex optimization [14]).

In search-based approaches, the clustering algorithm is modified so as to allow the constraints to "steer" the clustering process towards an appropriate partition. This is usually done by modifying the objective function so that it includes the available supervision provided as pairwise constraints or class labels [18], [17].

These two families of semi-supervised clustering methods rely on slightly different assumptions. Search-based methods consider that the similarities between data items provide relatively reliable information regarding the target categorization, but the algorithm needs some help in order to find the most relevant clusters. Similarity-adapting methods assume that the initial similarity measure has to be significantly modified (at a local or a more global scale) by the supervision in order to reflect correctly the target categorization.

While similarity-adapting methods appear to apply to a wider range of situations, they need either significantly more supervision (which can be an unacceptable burden for the user) or specific strong assumptions regarding the target similarity measure (which can be a strong limitation in their domain of application).

B. Principle of PCCA

PCCA corresponds to the search-based approach. The objective function to be minimized by PCCA combines the feature-based similarity between data points and cost terms for the pairwise constraints.

Let $\mathcal{M}$ be the set of *must-link* pairs such that $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{M}$ implies $\mathbf{x}_i$ and $\mathbf{x}_j$ should be assigned to the same cluster, and $\mathcal{C}$ be the set of *cannot-link* pairs such that $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{C}$ implies $\mathbf{x}_i$ and $\mathbf{x}_j$ should be assigned to the different cluster. Using the same notations as for CA, we can write the objective function PCCA must minimize:

$$\begin{aligned}\mathcal{J}(V,U) = & \sum_{k=1}^{C}\sum_{i=1}^{N}(u_{ik})^2 d^2(\mathbf{x}_i,\mu_k) \\ & + \alpha\Big(\sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{M}}\sum_{k=1}^{C}\sum_{l=1,l\neq k}^{C} u_{ik}u_{jl} \\ & + \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{C}}\sum_{k=1}^{C} u_{ik}u_{jk}\Big) - \beta\sum_{k=1}^{C}\Big[\sum_{i=1}^{N}(u_{ik})\Big]^2\end{aligned} \tag{3}$$

under the same constraint (2).

The prototypes of the clusters ($1 \leq j \leq C$) are given by

$$\mu_k = \frac{\sum_{i=1}^{N}(u_{ik})^2\mathbf{x}_i}{\sum_{i=1}^{N}(u_{ik})^2} \tag{4}$$

and cardinalities are expressed as

$$N_s = \sum_{i=1}^{N} u_{is} \tag{5}$$

The first term in (3) is the sum of squared distances to the prototypes weighted by constrained memberships (Fuzzy C-Means objective function). This term reinforces the compactness of the clusters.

The second term is composed of:

- The cost of violating the pairwise *must-link* constraints. The penalty corresponding to the presence of two such points in different clusters is weighted by the corresponding membership values.
- The cost of violating the pairwise *cannot-link* constraints. The penalty corresponding to the presence of two such points in a same cluster is weighted by their membership values.

This term is weighted by α, which is a way to specify the relative importance of the supervision.

The third component is the sum of the squares of the cardinalities of the clusters (specific to Competitive Agglomeration) and controls the number of clusters.

The final partition will minimize the sum of intra-cluster distances, while partitioning the data set into the smallest number of clusters such that only a minimum number of the constraints provided are violated.

Note that when the membership degrees are crisp and the number of clusters is pre-defined, this cost function reduces to the one used by PCKmeans [1].

It can be shown (see the Annex) that the updating of the memberships must follow the equation:

$$u_{rs} = u_{rs}^{FCM} + u_{rs}^{Constraints} + u_{rs}^{Bias} \tag{6}$$

where

$$u_{rs}^{FCM} = \frac{\frac{1}{d^2(\mathbf{x}_r,\mu_s)}}{\sum_{k=1}^{C}\frac{1}{d^2(\mathbf{x}_r,\mu_k)}} \tag{7}$$

$$u_{rs}^{Constraints} = \frac{\alpha}{2d^2(\mathbf{x}_r, \mu_s)}(\overline{C_{v_r}} - C_{v_s}) \tag{8}$$

and

$$u_{rs}^{Bias} = \frac{\beta}{d^2(\mathbf{x}_r, \mu_s)}(N_s - \overline{N_r}) \tag{9}$$

In (8), $C_{v_{rs}}$ and $\overline{C_{v_r}}$ are defined as

$$C_{v_{rs}} = \sum_{(\mathbf{x}_r,\mathbf{x}_j)\in\mathcal{M}} \sum_{l=1,l\neq s}^{C} u_{jl} + \sum_{(\mathbf{x}_r,\mathbf{x}_j)\in\mathcal{C}} u_{js} \tag{10}$$

$$\overline{C_{v_r}} = \frac{\sum_{k=1}^{C} \frac{\left(\sum_{(\mathbf{x}_r,\mathbf{x}_j)\in\mathcal{M}} \sum_{l=1,l\neq k}^{C} u_{jl} + \sum_{(\mathbf{x}_r,\mathbf{x}_j)\in\mathcal{C}} u_{jk}\right)}{d^2(\mathbf{x}_r,\mu_k)}}{\sum_{k=1}^{C} \frac{1}{d^2(\mathbf{x}_r,\mu_k)}}$$

In (9), $\overline{N_r}$ is defined as

$$\overline{N_r} = \frac{\sum_{k=1}^{C} \frac{N_k}{d^2(\mathbf{x}_r,\mu_k)}}{\sum_{k=1}^{C} \frac{1}{d^2(\mathbf{x}_r,\mu_k)}} \tag{11}$$

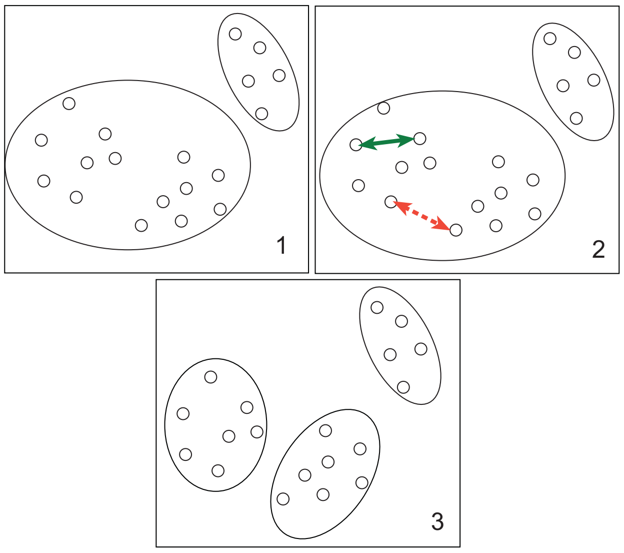

Fig. 1. Illustration of semi-supervised clustering based on pairwise constraints. Given user supervision in the form of *must-link* (plain, green) and *cannot-link* (dashed, red) constraints (2), the clustering process directly include this knowledge in the updating membership equation allowing it to find a partition that takes the constraints into account (3).

The first term in equation (6) is the membership term in the FCM algorithm and considers only distances between vectors and prototypes. The second term takes into account the available supervision: memberships are reinforced or reduced according to the pairwise constraints defined by the user (1). C_{v_s} is the cost of violation of the ”assignment” of vector r to cluster s, while $\overline{C_{v_r}}$ is the weighted average violation cost with respect to point r.

When the membership of point r to cluster s has a higher violation cost than the average, it is intuitively better to reduce its membership to the cluster s and this is exactly what the updating equation does, since the sign of $u_{rs}^{Constraints}$ is negative. Conversely, when the former cost is less than the weighted average violation cost, $u_{rs}^{Constraints}$ will have a positive sign and the membership u_{rs} will be increased by the quantity $u_{rs}^{Constraints}$.

The third term leads to a reduction of the cardinality of spurious clusters, which are discarded when their cardinality drops below a threshold. Indeed, u_{rs}^{Bias} is also a signed term that depends on the difference between the cardinality N_s of the cluster s and the weighted average of the cardinalities $\overline{N_r}$. This term is positive for clusters whose cardinality is higher than the average, so the membership of $\mathbf{x}_r$ to such clusters will be increased.

The β factor should provide a balance between the terms of (3), so β is defined at iteration t by:

$$\beta(t) = \frac{\eta_0 \exp(-|t-t_0|/\tau)}{\sum_{j=1}^{C}\left(\sum_{i=1}^{N} u_{ij}\right)^2} \times \left[\sum_{j=1}^{C}\sum_{i=1}^{N} u_{ij}^2\, d^2(\mathbf{x}_i, \mu_j)\right] \tag{12}$$

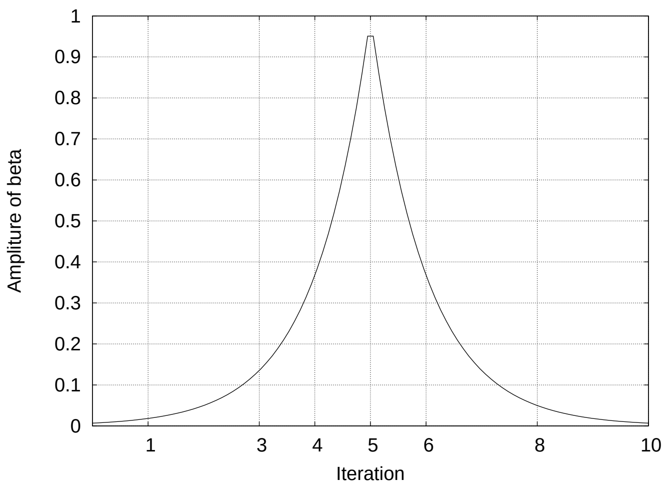

Fig. 2. How are balanced the terms of the objective function? This figure shows the time variation of the factor $\eta_0 \exp(-|t-t_0|/\tau)$ (or “amplitude”) of β, for $t_0 = 5$. The process of agglomeration should act slowly, to encourage the formation of small clusters in the beginning. Later, it should be gradually reinforced, in order to promote agglomeration until iteration t_0, when β starts to decrease. When the number of clusters becomes close to the optimum, the amplitude of β should again decrease to allow the algorithm to converge. The x-axis represents the number of iterations and the y-axis the amplitude of β.

Therefore the exponential factor makes the u_{rs}^{Bias} term small in the beginning allow cluster formation, then it is increased in order to reduce the number of clusters, and it is eventually decreased again so that the $u_{rs}^{Constraints}$ and u_{rs}^{FCM} terms can dominate, to seek the best partition of the data that respects the specified constraints.

C. Merging Process

As the algorithm proceeds, the clusters whose cardinalities drop below a threshold are discarded. The choice of this threshold is important since it reflects the size of the final clusters.

With respect to the way basic CA (see [2]) discards spurious clusters, two difficulties arise:

- The threshold has to be changed manually by the user according to the data he wants to categorize. So clustering becomes sensitive to a new parameter, when one important goal of the PCCA is to automatically find an appropriate number of clusters.
- Since good clusters may have different cardinalities, a criterion based only on the minimal cardinality of clusters is not efficient. If the minimal cardinality is too small, several prototypes can co-exist for a single large cluster. Indeed, in this cluster each point shares its membership among the prototypes and, since there are enough points, the cardinality of each cluster is larger than the threshold, see Fig. 3).
  On the other hand, if the minimal cardinality is too large, some small but distinct clusters, at equal distance from other, bigger clusters, will be lost.

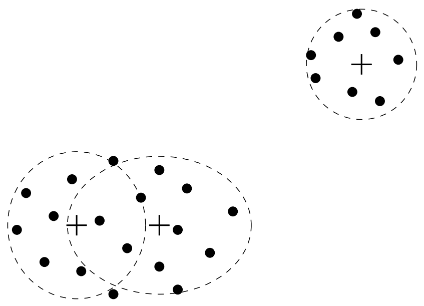
Fig. 3. The large cluster has twice the cardinality of the small one. If the minimal cardinality is small enough to retrieve the small cluster, two clusters can survive in the large one: even if each point shares its membership between two categories, the sum of these memberships for a category will be larger than the threshold under which clusters are discarded

We suggest a strategy for improving the agglomeration process in CA. First, we fix the minimum cardinality threshold according to the number of points in the data-set, such as all the small clusters can be retrieved, obtaining a weak agglomeration. Then, we build new prototypes based on pairwise merging. The proposed procedure reduces the number of prototypes by merging the best pair of prototypes among all possible pairs. This process is repeated until no more merging is possible.

At the k^{th} iteration, we first compute the $R = C(C-1)/2$ distances $d(r)$, for $r = 1,\dots,R$, between pairs of prototypes. If $\min_1$ and $\min_2$ are the two indices corresponding to the pair having the minimal distance $d_{\min}$, then we merge clusters $\min_1$ and $\min_2$ when the following criterion is satisfied:

$$d_{\min}/d_{\max} < \text{proximity threshold} \tag{13}$$

where $d_{\min} = \min\{d(r) \mid r = 1,\dots,R\}$ and $d_{\max} = \max\{d(r) \mid r = 1,\dots,R\}$. Since our aim is to make the clustering independent of such parameters, the results presented here were obtained with a fixed proximity threshold of 0.01.

D. Algorithm

The algorithm we propose is based on an iterative reallocation that partitions a data set into an optimal number of clusters by locally minimizing the sum of intra-cluster distances while respecting as many as possible of the constraints provided. PCCA alternates between membership updating step and centroid estimation step.

After the initialization step, we continue by computing β, the factor that will determine which term of the membership updating equation will dominate. Afterwards, memberships will be updated. In the second step, based on the cardinalities of different clusters and their relative proximity, spurious clusters will be discarded and close ones will be merged, thus obtaining the centroids of good clusters. The resulting PCCA algorithm is summarized below.

```
PCCA algorithm outline
  • Fix the maximum number of clusters $C$.
  • Randomly initialize prototypes for all clusters.
  • Initialize memberships: equal membership of
    every feature point to every cluster.
  • Compute initial cardinalities for all clusters.
  • Repeat
      – Compute $\beta$ using equation (12).
      – Compute memberships $u_{ij}$ using equation (6).
      – Compute cardinalities $N_j$ for $1 \le j \le C$ using
        equation (5).
      – For $1 \le j \le C$, if $N_j <$ threshold then discard
        cluster $j$.
      – Update number of clusters $C$.
        Repeat
        Merge nearest prototypes using (13).
        Until no further merging is required.
      – Update the prototypes using equation (4).
  • Until prototypes stabilize.
```

IV. Experimental results

We compared our PCCA algorithm to the basic CA algorithm and to PCKmeans. The first comparison was performed on the well-known IRIS database (also used in [1]), containing 3 classes of 50 instances each.

The second comparison was performed on a ground-truth image database containing 8 classes and a total of 187 images.

For all the experiments presented here, the constraints provided are randomly selected.

A. Clustering the Iris Data

The first database we categorized was used in [1] for evaluating the PCKmeans algorithm and has a long history in the pattern recognition literature [16]. This data set contains 3 classes of 50 instances (Iris flowers) each, a class corresponding to a variety of Iris. Every Iris flower is described by four numerical attributes, which are the length and the width of its petals and sepals. The classes are not spherical and only one class is linearly separable from the other two.

B. Clustering the Image Database

The second database we categorized is composed of images of different phenotypes of *Arabidopsis thaliana*, corresponding to slightly different genotypes. The categories of phenotypes provide indications about the role of the different genes.

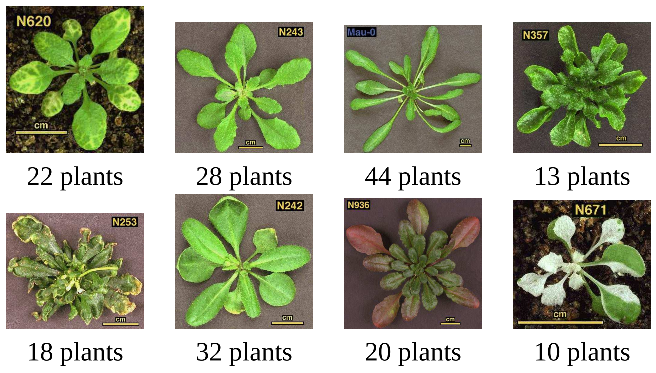

Fig. 4. A sample of the *Arabidopsis* image database, with the number of plants in each category

A sample of the images is shown in Figure 4. There are 8 categories, defined by visual criteria and described below, for a total of 187 plant images, but different categories contain very different numbers of instances. The intra-class diversity is also rather high. The categories we attempted to find in our study are:

- textured plants,
- plants with long stems and round leaves,
- plants with long stems and fine leaves,
- plants with dense, round leaves,
- plants with desiccated or yellow leaves,
- plants with large green leaves,
- plants with reddish leaves,
- plants with partially white leaves.

We first present the image descriptors we employ, the method for dimensionality reduction and our choice for the distance.

a) Image content description: Finding good image descriptors that can accurately describe the visual aspect of many different classes of images is a challenging task. For our experiments we selected the following descriptors used in our CBIR software IKONA [8]:

- **Weighted Color histograms:** statistical color signature weighted by the local color activity in the image. Information regarding the neighborhood of the pixels can be taken into account with the help of weighting functions. We employ a Laplacian weighted histogram and a probability weighted histogram (please refer to [9] for further details).
- **Shape descriptor:** to describe the shape content of an image we use a histogram based on the Hough transform, which gives the global behavior along straight lines in different directions.
- **Texture descriptor:** texture feature vectors are based on the Fourier transform, providing a distribution of the spectral power density along the frequency axes.

b) Dimensionality reduction: The resulting feature vector has 640 dimensions. This very high number of dimensions of the joint feature vector can produce difficulties during clustering and, also, can make clustering impractical for huge databases. In order to reduce the dimension of the feature vectors, we use linear principal component analysis (PCA), which is actually applied separately to each of the types of features previously described. The number of dimensions we retain is 5 times smaller than the original one.

c) Choice of the distance: Since the shape of the clusters is usually not spherical, we use the Mahalanobis distance (as in [7]) rather than the classical Euclidean one. For clusters $1 \leq k \leq C$, distances are computed using:

$$d^2(\mathbf{x}_i, \mu_k) = |C_k|^{1/p}(\mathbf{x}_i - \mu_k)^T C_k^{-1}(\mathbf{x}_i - \mu_k) \qquad (14)$$

where p is the dimension of the subspace after the PCA-based reduction and C_k is the covariance matrice of the cluster k:

$$C_k = \frac{\sum_{i=1}^{N}(u_{ik})^2(\mathbf{x}_i - \mu_k)(\mathbf{x}_i - \mu_k)^T}{\sum_{i=1}^{N}(u_{ik})^2} \qquad (15)$$

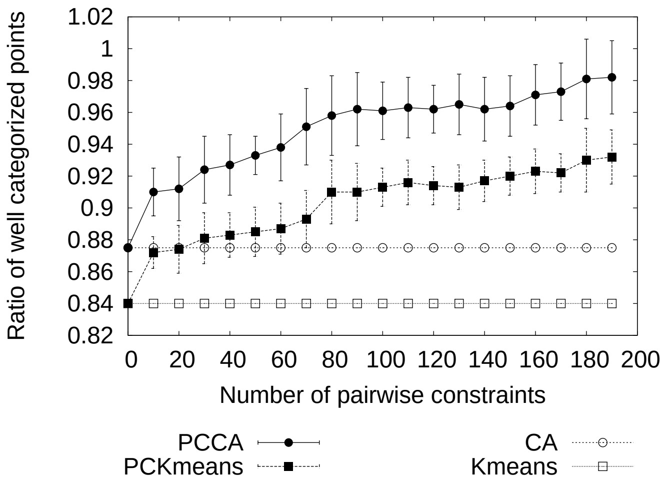

Fig. 5. Results obtained on the Iris dataset

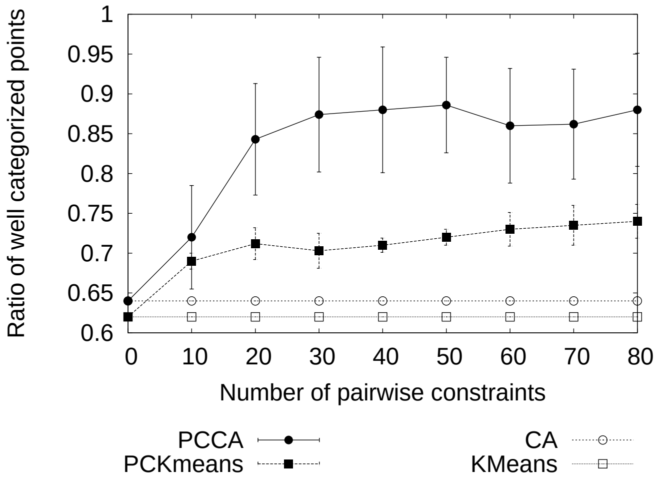

Fig. 6. Results obtained on the *Arabidopsis* database

K-means and PCKmeans were given the correct number of clusters. CA and PCCA found the number of clusters themselves, starting from a higher initial value. For the fuzzy algorithms (CA and PCCA), every data point is assigned to the cluster to which its membership value is the highest.

For every number of constraints, 100 experiments were performed with different random selections of the constraints in order to produce the error bars for PCKmeans and PCCA.

Figures 5 and 6 present the dependence between the percentage of well-categorized data points and the number of pairwise constraints considered, for each of the two datasets. The graphs for the CA and K-means algorithms (both ignoring the constraints) are only given as a reference. We can first notice that, by providing simple semantic information in the form of pairwise constraints, the user can significantly improve the quality of the categories obtained. The number of pairwise constraints required for reaching such an improvement is relatively low with respect to the number of items in the dataset.

Also, with a similar number of constraints, PCCA performs significantly better than PCKmeans by making a better use of the available constraints; the signed constraint terms in (8), part of the fuzzy memberships, directly include the non-violation of pairwise constraints in the fuzzy clustering process.

The relatively high values for the variance of the results, both for PCCA and for PCKmeans, indicate that the random selection of the pairs of data points for which the user is required to provide constraints is suboptimal. Further assumptions regarding the data may let us improve the results by using more adequate methods for selecting the constraints.

V. Conclusion

We attempted to show that by providing a limited amount of simple semantic information in the form of pairwise constraints, the user can bring the automatic categorization of the images in a database much closer to her expectations. We put forward a new semi-supervised clustering algorithm, PCCA, based on a fuzzy cost function that takes pairwise constrains into account.

Experiments on the Iris dataset and, especially, on a ground-truth image database show that PCCA performs considerably better than unconstrained CA and than PCKmeans. By making better use of the constraints, PCCA allows the number of constraints to remain sufficiently low for this approach to be interesting. Also, the computational complexity of PCCA is linear in the number of data vectors, making this algorithm suitable for real-world clustering applications.

VI. Acknowledgments

We are very grateful to NASC (European Arabidopsis Stock Centre, UK (http://arabidopsis.info/)) for allowing us to use the *Arabidopsis* images and to Ian Small from INRA (*Institut National de la Recherche Agronomique*, France) who provided us with this image database.

VII. Appendix

To minimize (3) with respect to U under the constraints (2), we use Lagrange multipliers and obtain

$$\begin{aligned}\mathcal{J}(V,U)=&\sum_{k=1}^{C}\sum_{i=1}^{N}(u_{ik})^2 d^2(\mathbf{x}_i,\mu_k)-\beta\sum_{k=1}^{C}\Big[\sum_{i=1}^{N}(u_{ik})\Big]^2\\ +\quad&\alpha\Big(\sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{M}}\sum_{k=1}^{C}\sum_{l=1,l\neq k}^{C}u_{ik}u_{jl}+\sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{C}}\sum_{k=1}^{C}u_{ik}u_{jk}\Big)\\ -\quad&\sum_{i=1}^{N}\lambda_i\left(\sum_{k=1}^{C}u_{ik}-1\right)\end{aligned}\tag{16}$$

We fix the prototypes and solve

$$\begin{aligned}\frac{\partial\mathcal{J}(V,U)}{\partial u_{rs}}=&2u_{rs}d^2(\mathbf{x}_r,\mu_s)-2\beta\sum_{i=1}^{N}u_{is}-\lambda_t\\ &+\alpha\Big(\sum_{(\mathbf{x}_r,\mathbf{x}_j)\in\mathcal{M}}\sum_{l=1,l\neq s}^{C}u_{jl}+\sum_{(\mathbf{x}_r,\mathbf{x}_j)\in\mathcal{C}}u_{js}\Big)=0\end{aligned}\tag{17}$$

where $s\in\{1,\ldots,C\},\ r\in\{1,\ldots,N\}$.

The solution can be simplified by assuming that the membership values do not change significantly from an iteration to the next, and computing $\sum_{i=1}^{N}u_{is}$ in (17) using the membership values from the previous iteration. With this assumption, (17) reduces to

$$\begin{aligned}u_{rs}=&\frac{2\beta N_s+\lambda_t}{2d^2(\mathbf{x}_r,\mu_s)}\\ &-\alpha\frac{\sum_{(\mathbf{x}_r,\mathbf{x}_j)\in\mathcal{M}}\sum_{l=1,l\neq s}^{C}u_{jl}+\sum_{(\mathbf{x}_r,\mathbf{x}_j)\in\mathcal{C}}u_{js}}{2d^2(\mathbf{x}_r,\mu_s)}\end{aligned}\tag{18}$$

where $N_s=\sum_{i=1}^{N}u_{is}$ is the cardinality of cluster s. With the notations in (10)–(11) we obtain the expressions (6)–(9).

References

[1] S. Basu, A. Banerjee and R. Mooney, Semi-supervised clustering by seeding, Proc. of 19th International Conference on Machine Learning, 2002.
[2] H. Frigui and R. Krishnapuram, Clustering by competitive agglomeration, Pattern Recognition, vol. 30, No. 7, pp. 1109–1119, 1997.
[3] J. C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press, 1981.
[4] R. N. Dave, Characterization and detection of noise in clustering, Pattern Recognition Letters, vol. 12, pp. 657–664, 1991.
[5] C. G. Looney, Interactive clustering and merging with a new fuzzy expected value, Pattern Recognition, vol. 35, No. 11, pp. 2413–2423, 2002.
[6] N. Boujemaa, On Competitive Unsupervized Clustering, Proc. of International Conference on Pattern Recognition, Barcelona, Spain, 2000.
[7] E. E. Gustafson and W. C. Kessel, Fuzzy clustering with a fuzzy covariance matrix, Proc. of IEEE CDC, San Diego, California, 1979.
[8] N. Boujemaa, J. Fauqueur, M. Ferecatu, F. Fleuret, V. Gouet, B. Le Saux and H. Sahbi, Interactive Specific and Generic Image Retrieval, Proc. of MMCBIR'2001, Rocquencourt, France.
[9] C. Vertan and N. Boujemaa, Upgrading Color Distributions for image retrieval: Can we do better ?, Proc. of Visual 2000, Lyon, France.
[10] R.N.Dave, Use of the Adaptive Fuzzy Clustering Algorithm to Detect Lines in Digital Images, Intelligent Robots and computer Vision VIII, vol.1, 192, pp. 600–611, 1989.
[11] V. Vapnik, Statistical Learning Theory. Wiley, 1998.
[12] K. Wagstaff, C. Cardie, S. Rogers and S. Schroedl, Constrained K-Means clustering with background knowledge, Proc. of 18th International Conference on Machine Learning, 2001.
[13] S. Basu, M. Bilenko, and R. J. Mooney, Comparing and unifying search-based and similarity-based approaches to semi-supervised clustering. Proc. of the ICML 2003 Workshop on the Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining, pp. 42–49, Washington DC, 2004.
[14] E.P. Xing, A. Ng, M.I. Jordan and S. Russel, Distance metric learning, with application to clustering with side-information, Advances in Neural Information Processing Systems 15, MIT Press, 2003.
[15] B. Scholkopf and A. Smola, Learning with Kernels, MIT Press, 2002.
[16] R.A. Fisher, Contributions to Mathematical Statistics, John Wiley, NY, 1950.
[17] A. Bensaid and J.C. Bezdek, Semi-supervised Point Prototype Clustering, International Joint Conference on Pattern Recognition and Artificial Intelligence, vol. 12-4, pp. 625–643, 1998.
[18] W. Pedrycz and J. Waletzky, Fuzzy clustering with partial supervision, IEEE Transactions on Systems, Man, and Cybernetics, Part B, vol. 27-5, pp. 787–795, 1997.