ACTIVE SEMI-SUPERVISED CLUSTERING FOR IMAGE DATABASE CATEGORIZATION

Nizar Grira, Michel Crucianu and Nozha Boujemaa

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

ABSTRACT

Clustering algorithms are increasingly employed for the categorization of image databases, in order to provide users with database overviews and make their access more effective. By including information provided by the user, the categorization process can produce results that come closer to user's expectations. To make such a *semi-supervised* categorization approach acceptable for the user, this information must be of a very simple nature and the amount of information the user is required to provide should be minimized. For a semi-supervised fuzzy clustering algorithm we developed, Pairwise-Constrained Competitive Agglomeration, we put forward here a criterion for the *active* selection of constraints. We show that this selection criterion allows a significant reduction in the number of pairwise constraints required, making the resulting algorithm an attractive alternative in the categorization of image databases.

1. INTRODUCTION

To let users easily apprehend the contents of image databases and to make their access to the images more effective, a relevant organisation of the contents must be achieved. While it is easy to apply standard unsupervised clustering algorithms to the descriptors of the images in a database, the results of this fully automatic categorization are rarely satisfactory. Some supervision information provided by the user is then needed for obtaining results that are closer to user's expectations. Supervised classification and *semi-supervised clustering* are both candidates for such a categorization approach.

When the user is able and willing to provide class labels for a significant sample of images in the database, supervised classification is the method of choice. In practice, this will be the case for specific classification/recognition tasks, but not so often for database categorization tasks.

When the goal is general image database categorization, not only the user does not have labels for images, but he doesn't even know *a priori* what most of the classes are and how many classes should be found. Instead, he expects the system to "discover" these classes for him. To improve results with respect to what an unsupervised algorithm would produce, the user may accept to provide some supervision if this information is of a very simple nature and in a rather limited amount. A semi-supervised clustering approach should then be employed.

We assume that users can easily evaluate whether two images should be in the same category or rather in different categories, so they can easily define *must-link* or *cannot-link* constraints between pairs of images. Following previous work by Demiriz et al. [1], Wagstaff et al. [2] or Basu et al. [3], in [4] we introduce Pairwise-Constrained Competitive Agglomeration (PCCA), a fuzzy semi-supervised clustering algorithm that exploits the simple information provided by pairwise constraints.

In the original version of PCCA [4] we do not make further assumptions regarding the data, so the pairs of items for which the user is required to define constraints are randomly selected. But in many cases, such assumptions regarding the data *are* available. We argue here that quite general assumptions let us perform a more adequate, *active* selection of the pairs of items and thus significantly reduce the number of constraints required for achieving a desired level of performance.

In section 2 we give a brief overview of existing algorithms for semi-supervised clustering. The semi-supervised fuzzy clustering algorithm we developed, Pairwise-Constrained Competitive Agglomeration, is shortly presented in section 3. We then introduce in section 4 our criterion for the active selection of constraints. The experimental results obtained with this criterion on a benchmark dataset and on a real-world problem are then given in section 5.

2. SEMI-SUPERVISED CLUSTERING

To organize a collection of data items into clusters, unsupervised clustering (or *cluster analysis*, see the surveys in [5], [6]) relies exclusively on a measure of similarity between data items. Semi-supervised clustering also takes into ac-

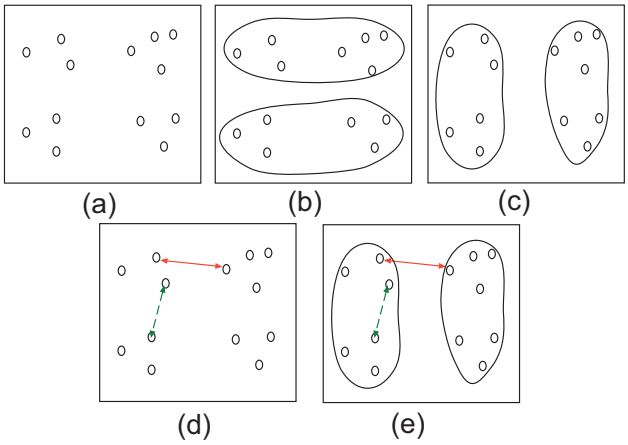

Fig. 1. Influence of pairwise constraints on clustering: (a) data items to cluster, (b) and (c) alternative potential solutions for unsupervised clustering, (d) specification of pairwise constraints (green dashed line for the *must-link* and red continuous line for the *cannot-link*), (e) solution obtained by semi-supervised clustering using these constraints.

count information regarding either the membership of some data items to specific clusters or, more often, pairwise constraints (must-link, cannot-link) between data items. For semi-supervised clustering to be successful, the supervision information should correct rather than completely contradict the similarities between data items.

Figure 1 shows a simple and not all-encompassing example of the role pairwise constraints can have when used for semi-supervised clustering in addition to the similarities between data items. Any of the partitions (b) and (c) of the data items in (a) can be solutions to an unsupervised clustering algorithm, and for some algorithms the choice will depend on random factors (such as the initialisation of the prototypes). By providing pairwise constraints like the ones pictured in Figure 1(d), the user can guide clustering to the solution he prefers.

Unlike unsupervised clustering, the semi-supervised approach to clustering has a short history and few methods were published until now. The main distinction between these methods concerns the way the two sources of information are combined [7]: either by modifying the search for appropriate clusters or by adapting the similarity measure.

- In search-based methods, the clustering algorithm itself is modified so that user-provided constraints or labels can be used to bias the search for an appropriate clustering. This can be done in several ways, such as by performing a transitive closure of the constraints and using them to initialize clusters [3], by including in the cost function a penalty for lack of compliance with the specified constraints [1], or by requiring constraints to be satisfied during cluster assignment in the clustering process [2].
- In similarity-adapting methods, an existing clustering algorithm using some similarity measure is employed, but the similarity measure is adapted so that the available constraints can be easier satisfied. Several similarity measures were employed for similarity-adapting semi-supervised clustering: the Jensen-Shannon divergence trained with gradient descent [8], the Euclidean distance modified by a shortest-path algorithm [9] or Mahalanobis distances adjusted by convex optimization [10], [11]. Among the clustering algorithms using such adapted similarity measures we can mention hierarchical single-link [11] or complete-link [9] clustering and k-means [10], [11].

It is important to note that 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).

3. PAIRWISE-CONSTRAINED COMPETITIVE AGGLOMERATION

Let $\mathbf{x}_i$, $i \in \{1, .., N\}$ be a set of N vectors representing the data items to be clustered, $\mathbf{V}$ the matrix having as columns the prototypes μ_k, $k \in \{1, .., C\}$ of C clusters ($C \ll N$) and $\mathbf{U}$ the matrix of the membership degrees, such as u_{ik} is the membership of $\mathbf{x}_i$ to the cluster k. Let $d(x_i, \mu_k)$ be the distance between the vector $\mathbf{x}_i$ and the cluster prototype μ_k. The CA algorithm minimizes the following objective function [12]:

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

under the constraint

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

The first term in (1) is the standard Fuzzy C-Means (FCM, [13]) objective function. The second term defines a competition that progressively reduces the number of clusters. The $\beta(t)$ factor sets a balance between the terms and progressively decreases with t, the iteration number.

For PCCA, the objective function to be minimized must combine the feature-based similarity between data items and knowledge of the pairwise constraints. Let $\mathcal{M}$ be the set of available must-link constraints, i.e. $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{M}$ implies $\mathbf{x}_i$ and $\mathbf{x}_j$ should be assigned to the same cluster, and $\mathcal{C}$ the set of cannot-link constraints, i.e. $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{C}$ implies $\mathbf{x}_i$ and $\mathbf{x}_j$ should be assigned to different clusters. Using the same notations as for CA, we can write the objective function PCCA must minimize [4]:

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

under the same constraint (2).

The prototypes of the clusters, for $k \in \{1,..,C\}$, are given by

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

and the fuzzy cardinalities of the clusters are

$$N_k = \sum_{i=1}^{N} u_{ik} \quad (5)$$

The first term in (3) is the sum of the squared distances to the prototypes weighted by the memberships and comes from the FCM 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 and the cost of violating the pairwise cannot-link constraints. The penalty corresponding to the presence of two such points in different clusters (for a must-link constraint) or in a same cluster (for a cannot-link constraint) is weighted by their membership values. The term taking the constraints into account is weighted by α, a constant factor that specifies the relative importance of the supervision.

The last term in (3) is the sum of the squares of the cardinalities of the clusters (comes from the CA objective function) and controls the competition between clusters.

When all these terms are combined and β is well chosen, the final partition will minimize the sum of intra-cluster distances, while partitioning the data set into the smallest number of clusters such that the specified constraints are respected as well as possible. Note that when the memberships are crisp and the number of clusters is pre-defined, this cost function reduces to the one used by the PCKmeans algorithm in [3].

It can be shown (see [4]) that the equation for updating memberships is

$$u_{rs} = u_{rs}^{FCM} + u_{rs}^{Constraints} + u_{rs}^{Bias} \quad (6)$$

where

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

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

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

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

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

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

and $\overline{N_r}$ in (9) is

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

The first term in equation (6), u_{rs}^{FCM}, is the same as the membership in the FCM algorithm and only focusses on the distances between data items and prototypes. The second term, $u_{rs}^{Constraints}$, takes into account the available supervision: memberships are reinforced or deprecated according to the pairwise constraints given by the user. The third term, u_{rs}^{Bias}, leads to a reduction of the cardinality of spurious clusters, which are discarded when their cardinality drops below a threshold.

The β factor controls the competition between clusters and is defined at iteration t by:

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

The exponential component of β makes the last term in (3) dominate during the first iterations of the algorithm, in order to reduce the number of clusters by removing spurious ones. Later, the first three terms will dominate, to seek the best partition of the data. The resulting PCCA algorithm is given below. In the original version of PCCA, the pairs of items for which the user is required to define constraints are randomly selected, prior to running the clustering process.

Outline of the PCCA algorithm

- Fix the maximum number of clusters C.
- Randomly initialize prototypes μ_j, $j \in \{1, .., C\}$.
- Initialize memberships u_{ij}: equal membership of every data item to every cluster.
- Compute initial cardinalities N_j using (5).
- **Repeat**
  - Update β using (12).
  - Update the memberships u_{ij} using (6).
  - Update the cardinalities N_j, $j \in \{1, .., C\}$, using (5).
  - For $j \in \{1, .., C\}$, if $N_j <$ threshold then discard cluster j.
  - Update the number of clusters C.
  - Update the prototypes μ_j, $j \in \{1, .., C\}$, using (4).
- **Until** prototypes stabilize.

As distance $d(\mathbf{x}_i, \mu_j)$ between a data item $\mathbf{x}_i$ and a cluster prototype μ_j one can use either the ordinary Euclidean distance when the clusters are assumed to be spherical or the Mahalanobis distance (13) when they are assumed to be elliptical:

$$d^2(\mathbf{x}_i, \mu_k) = |C_k|^{1/p}(\mathbf{x}_i - \mu_k)^T C_k^{-1}(\mathbf{x}_i - \mu_k) \qquad (13)$$

where p is the dimension of the space considered and C_k is the covariance matrice of the cluster k:

$$C_k = \frac{\sum_{i=1}^{N}(u_{ik})^2(\mathbf{x}_i - \mu_k)(\mathbf{x}_i - \mu_k)^T}{\sum_{i=1}^{N}(u_{ik})^2} \qquad (14)$$

When the Mahalanobis distance is employed, the computation of C_k in (14) are performed at the beginning of the main loop, right before the update of β.

4. ACTIVE SELECTION OF CONSTRAINTS

We consider that, most of the time, users can easily define must-link or cannot-link constraints between pairs of images, so it is possible to rely on such pairwise constraints to perform a semi-supervised categorization of image databases. To make this approach attractive for the user, it is also important to minimize the number of constraints he has to provide for reaching some given level of quality. This can be done by asking the user to define must-link or cannot-link constraints for the pairs of data items that are expected to have the strongest corrective effect on the clustering algorithm (i.e. that are *maximally informative*).

But does the *identity* of the constraints one provides have a significant impact on performance or all constraints are relatively alike and only the *number* of constraints matters? In a series of repeated experiments with PCCA using random constraints, we found a significant variance in the quality of the final clustering results. So the selection of constraints can have a strong impact and we must find appropriate selection criteria. Such criteria may depend on further assumptions regarding the data; for the criteria to be relatively general, the assumptions they rely on should not be too restrictive.

In previous work on this issue, Basu et al. [14] developed a scheme for selecting pairwise constraints *before* running their semi-supervised clustering algorithm (PCK-means). They defined a farthest-first traversal scheme of the set of data items, with the goal of finding k items that are far from each other to serve as support for the constraints.

This issue was also explored in unsupervised learning for cases where prototypes cannot be defined. In such cases, clustering can only rely on the evaluation of pairwise similarities between data items, implying a high computational cost. In [15], the authors consider subsampling as a solution for reducing cost and perform an active selection of new data items by minimizing the estimated risk of making wrong predictions regarding the true clustering from the already seen data. This active selection method can also be seen as maximizing the expected value of the information provided by the new data items. The authors find that their active unsupervised clustering algorithm spends more samples of data to disambiguate clusters that are close to each other and less samples for items belonging to well-separated clusters.

As for other search-based semi-supervised clustering methods (see section 2), when using PCCA we consider that the similarities between data items provide relatively reliable information regarding the target categorization and the constraints only help in order to find the most relevant clusters. There is then little uncertainty in identifying well-separated compact clusters. To be maximally informative, supervision effort (i.e. constraints) should rather be spent for defining

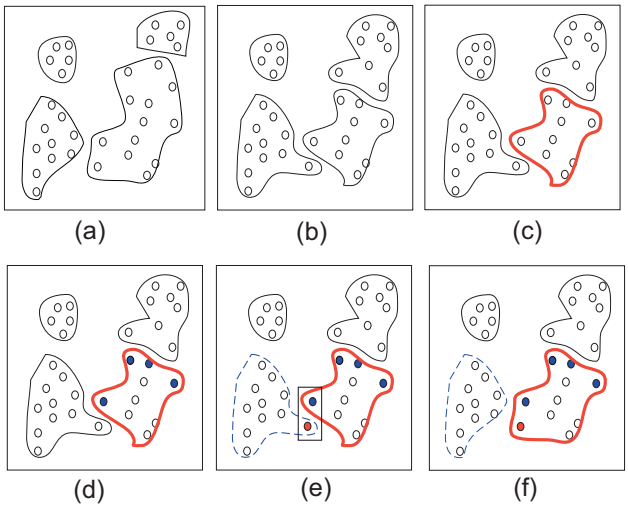

Fig. 2. Illustration of the clustering process. (a) Raw data and desired categorization. (b) Clusters formed at iteration t. (c) Least well-defined cluster (thick frontier) found as the one having the highest fuzzy hyper-volume (FHV). (d) Data items at the frontier of the least well-defined cluster (depicted as filled circles) are the items having the lowest membership degrees among the items assigned to this cluster. (e) For each of these items, we select the nearest cluster as the one corresponding to the second highest membership value of the data item under focus (frontier of the nearest cluster is dashed in the figure). To define a constraint, we then take the nearest item in the nearest cluster. (f) Generating constraints between the two selected items to bias categorization towards the one expected in (a).

those clusters that are neither compact, nor well-separated from their neighbors. One can note that this is consistent with the findings in [15] regarding unsupervised clustering.

To find the data items that provide the most informative must-link or cannot-link pairwise constraints, we shall then focus on the least well-defined clusters (see Fig. 2) and, more specifically, on the frontier with their neighbors (see Fig. 3).

To identify the least well-defined cluster at some iteration t (Fig. 2b), we use the *fuzzy hypervolume* (FHV), defined by:

$$FHV = |C_k| \tag{15}$$

$|C_k|$ being the determinant of the covariance matrix C_k of cluster k, given by (14).

The FHV was introduced by Gath and Geva [16] as an evaluation of the compactness of a fuzzy cluster; the smaller the spatial volume of the cluster and the more concentrated the data items are near its center, the lower the FHV of the cluster.

We consider the least well-defined cluster at iteration t to be the one with the largest FHV at that iteration (Fig. 2c).

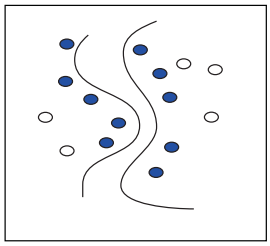

Fig. 3. Following our assumptions, for the generation of constraints we expect the most informative data items to be at the frontier of the least well-defined clusters

Note that we don't need any extra computation because we already used the FHV to find the Mahalanobis distance, see (13) and (14).

Once the least well-defined cluster at iteration t is found, we need to identify the data items near its boundary. Note that in the fuzzy setting, one can consider that a data item represented by the vector $\mathbf{x}_r$ is assigned to cluster s if u_{rs} is the highest among its membership degrees. The data items at the boundary are those having the lowest membership values to this cluster among all the items assigned to it (Fig. 2d).

Once we have a set of items that lie on the frontier of the cluster, we find for each item the closest cluster, corresponding to its second highest membership value (Fig. 2e). The user is then asked whether one of these items should be (or not) in the same cluster as the closest item from the nearest cluster ((Fig. 2f).

It is easy to see that the time complexity of this method is high. We suggest below an approximation of this method, having a much lower cost.

After finding the least well-defined cluster with the FHV criterion, we consider a virtual boundary that is only defined by a membership threshold and will usually be larger than the true one (this is why we call it "extended" boundary). The items whose membership values are closest to this threshold are considered to be on the boundary and constraints are generated directly between these items. We expect these constraints to be relatively equivalent (and not too suboptimal when compared) to the constraints that would have been obtained by the more complex method described in the previous paragraphs.

This approximate selection method has two parameters: the number of constraints at every iteration and the membership threshold for defining the boundary. The first parameter concerns the selection in general, not the approximate method specifically. In all the experiments presented next, we generate 3 constraints at every iteration of the clustering algorithm. For the comparative evaluations, we plot the ratio of well-categorized items against the number of pairwise constraints. With this selection procedure, when the maxi-

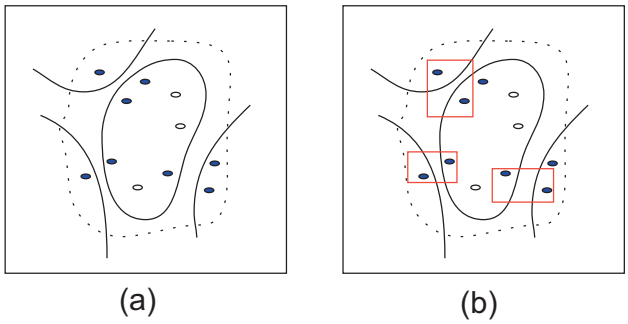

Fig. 4. Informative pairwise constraints can be selected on the extended boundary (dashed line) of the least well-defined cluster

mal number of constraints is reached, clustering continues until convergence without generating new constraints.

The second parameter is specific to the approximation of the boundary and we use a fixed value of 0.3, meaning that we consider a rather large approximation. These fixed values for the two parameters are necessarily suboptimal, but a significant increase in performance (or reduction in the number of required constraints for a given performance) is nevertheless obtained.

We use hereafter the name "most valuable pairs" (MVP) for the constraints obtained by this approximate method. Also, the PCCA algorithm including this active procedure for the selection of constraints will be called Active-PCCA.

5. EXPERIMENTAL RESULTS

We evaluated the effect our criterion for the active selection of constraints has on the PCCA algorithm and we compared it to the CA algorithm [12] (unsupervised clustering) and to PCKmeans [3] (semi-supervised clustering).

The first comparison is performed on the well-known Iris benchmark database, also used in [3], containing 3 classes of 50 instances (Iris flowers) each. 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. The simplicity and low dimension of this dataset also allows us to display the constraints that are actually selected (see Figure 8).

The second comparison is performed on a ground truth database composed of images of different phenotypes of *Arabidopsis thaliana*, corresponding to slightly different genotypes. This scientific image database is issued from studies of gene expression. There are 8 categories, defined by visual criteria: 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

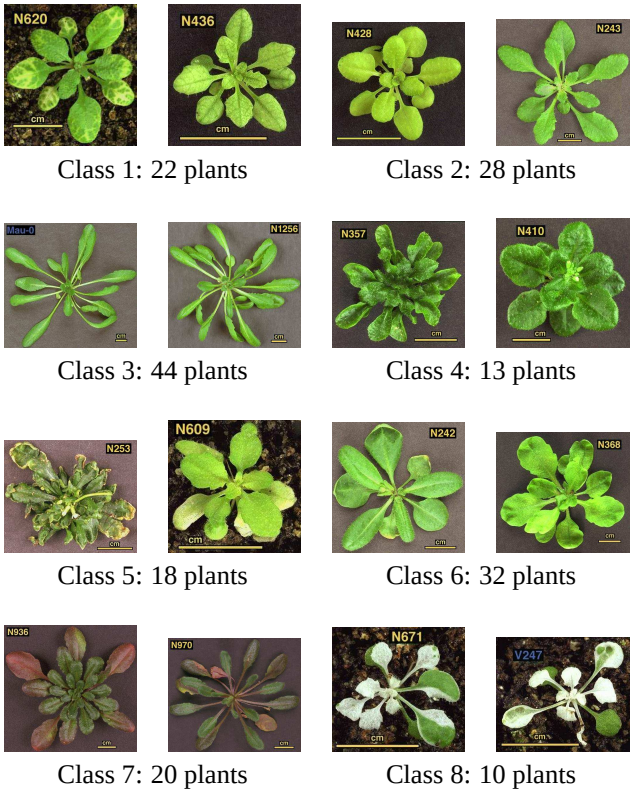

Fig. 5. A sample of the *Arabidopsis* image database, with the number of plants in each class

leaves, plants with partially white leaves. There are a total of 187 plant images, but different categories contain very different numbers of instances. The intra-class diversity is high for most classes. A sample of the images (two images from each category) is shown in Figure 5.

The global image features we used for the *Arabidopsis* database are the Laplacian weighted histogram, the probability weighted histogram, the Hough histogram, the Fourier histogram and a classical color histogram obtained in HSV color space (described in [17]).

By combining these descriptors, the resulting joint feature vector has over 600 dimensions. This very high number of dimensions of the joint feature vector can not only make clustering impractical for large databases, but also produce curse of dimensionality -related difficulties during clustering. 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 eventually retain is 5 times smaller than the original one.

In all the experiments reported here we used the Mahalanobis distance (13) rather than the classical Euclidean distance.

Figures 6 and 7 present the dependence between the

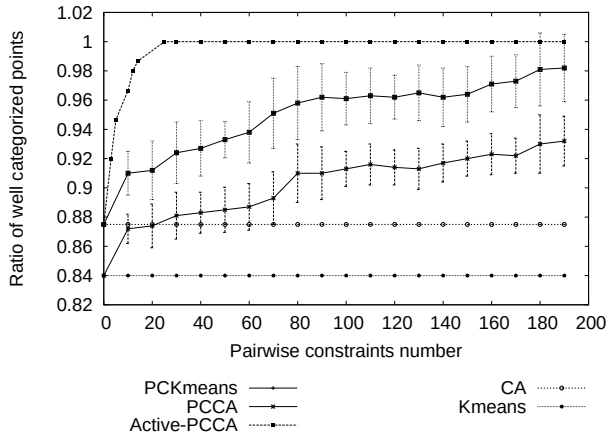

Fig. 6. Results obtained by the different clustering algorithms on the Iris database

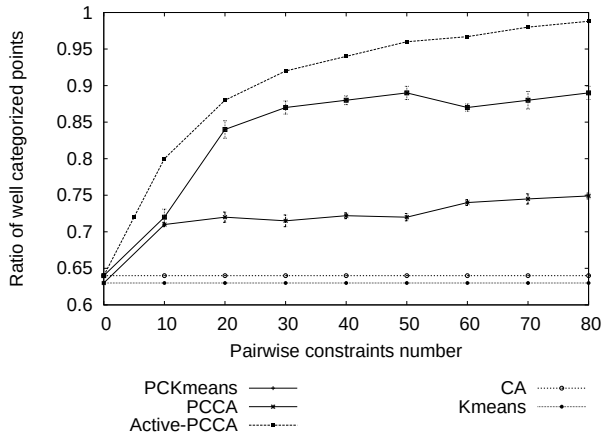

Fig. 7. Results obtained by the different clustering algorithms on the *Arabidopsis* database

percentage of well-categorized data points and the number of pairwise constraints considered, for each of the two datasets. We provide as a reference the graphs for the CA algorithm and for K-means, both ignoring the constraints (unsupervised learning).

The correct number of classes was directly provided to K-means and PCKmeans. CA, PCCA and Active-PCCA were initialized with a significantly larger number of classes and found the appropriate number by themselves.

For the fuzzy algorithms (CA, PCCA and Active-PCCA) every data item is assigned to the cluster to which its membership value is the highest. For every number of constraints, 500 experiments were performed with different random selections of the constraints in order to produce the error bars for PCKmeans and for the original PCCA.

These experimental results clearly show that the user can significantly improve the quality of the categories obtained by providing a simple form of supervision, the pairwise constraints. With a similar number of constraints, PCCA performs significantly better than PCKmeans by making a better use of the available constraints. The fuzzy clustering process directly takes into account the the pairwise constraints thanks to the signed constraint terms in (11).

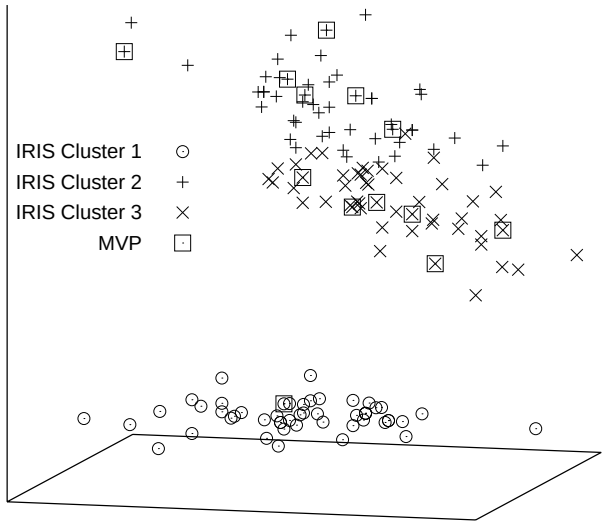

Fig. 8. Positions of the data points composing the "most valuable pairs" (MVP) produced by our selection criterion

The active selection of constraints (Active-PCCA) further reduces the number of constraints required for reaching such an improvement. The number of constraints becomes very low with respect to the number of items in the dataset.

To visualize where our selection criterion actually generates the constraints, we displayed in Figure 8 the first 13 MVPs produced on the Iris dataset, corresponding to a correct clustering of 98% of the data items. Three of the original four dimensions are shown. We can see that only one point was selected (during an early iteration) in the well-separated cluster and that the other points are spread near the frontiers of the other two clusters.

6. CONCLUSION

By including information provided by the user, general image database categorization can produce results that come much closer to user's expectations. But the user may have difficulties accepting such a semi-supervised categorisation approach unless the information he must provide is very simple in nature and in a small amount.

We put forward here a criterion for the active selection of constraints working well with our Pairwise-Constrained Competitive Agglomeration (PCCA) semi-supervised clustering algorithm, under rather general assumptions. The experiments we presented on the Iris dataset and on the

Arabidopsis image database show that the active selection of constraints combined with PCCA allows the number of constraints required to remain sufficiently low for this approach to become a really interesting alternative in the categorization of image databases. We shall evaluate this semi-supervised clustering algorithm on larger image databases that don't have a ground truth.

The approximations we made allowed us to maintain a low computational complexity for the resulting algorithm, making it suitable for real-world clustering applications. We shall continue exploring the active selection of constraints in an attempt to find a better tradeoff between performance and computational complexity.

7. ACKNOWLEDGMENTS

This work was supported by the EU Network of Excellence (http://www.muscle-noe.org/). The authors wish also to thank NASC (European Arabidopsis Stock Centre, UK, http://arabidopsis.info/) for allowing them to use the *Arabidopsis* images and Ian Small from INRA (*Institut National de la Recherche Agronomique*, France) for providing this image database.

8. REFERENCES

[1] A. Demiriz, K. Bennett, and M. Embrechts, "Semi-supervised clustering using genetic algorithms," in *Intelligent Engineering Systems Through Artificial Neural Networks 9*, C. H. Dagli et al., Ed. 1999, pp. 809–814, ASME Press.

[2] K. Wagstaff and C. Cardie, "Clustering with instance-level constraints," in *Proceedings of the 17th International Conference on Machine Learning*, 2000, pp. 1103–1110.

[3] S. Basu, A. Banerjee, and R. J. Mooney, "Semi-supervised clustering by seeding," in *Proceedings of 19th International Conference on Machine Learning (ICML-2002)*, 2002, pp. 19–26.

[4] Nizar Grira, Michel Crucianu, and Nozha Boujemaa, "Semi-supervised fuzzy clustering with pairwise-constrained competitive agglomeration," in *IEEE International Conference on Fuzzy Systems (Fuzz'IEEE 2005)*, May 2005.

[5] Anil K. Jain and Richard C. Dubes, *Algorithms for clustering data*, Prentice-Hall, Inc., 1988.

[6] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, 1999.

[7] Sugato Basu, Mikhail Bilenko, and Raymond J. Mooney, "Comparing and unifying search-based and similarity-based approaches to semi-supervised clustering," in *Proceedings of the ICML-2003 Workshop on the Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining*, Washington, DC, August 2004, pp. 42–49.

[8] D. Cohn, R. Caruana, and A. McCallum, "Semi-supervised clustering with user feedback," 2000.

[9] Dan Klein, Sepandar D. Kamvar, and Christopher D. Manning, "From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering," in *Proceedings of the 19th International Conference on Machine Learning*. 2002, pp. 307–314, Morgan Kaufmann Publishers Inc.

[10] Eric P. Xing, Andrew Y. Ng, Michael I. Jordan, and Stuart Russell, "Distance metric learning with application to clustering with side-information," in *Advances in Neural Information Processing Systems 15*, S. Thrun S. Becker and K. Obermayer, Eds., Cambridge, MA, 2003, pp. 505–512, MIT Press.

[11] M. Bilenko and R. J. Mooney, "Adaptive duplicate detection using learnable string similarity measures," in *International Conference on Knowledge Discovery and Data Mining*, Washington, DC, 2003, pp. 39–48.

[12] H. Frigui and R. Krishnapuram, "Clustering by competitive agglomeration," *Pattern Recognition*, vol. 30, no. 7, pp. 1109–1119, 1997.

[13] James C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, 1981.

[14] Sugato Basu, Arindam Banerjee, and Raymond Mooney, "Semi-supervised clustering by seeding," in *Machine Learning: Proceedings of the Nineteenth International Conference*, 2002.

[15] Thomas Hofmann and Joachim M. Buhmann, "Active data clustering," in *Advances in Neural Information Processing Systems (NIPS) 10*, 1997, pp. 528–534.

[16] Isak Gath and Amir B. Geva, "Unsupervised optimal fuzzy clustering.," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 7, pp. 773–780, 1989.

[17] Nozha Boujemaa, Julien Fauqueur, Marin Ferecatu, François Fleuret, Valérie Gouet, Bertrand Le Saux, and Hichem Sahbi, "Ikona: Interactive generic and specific image retrieval," in *Proceedings of the International workshop on Multimedia Content-Based Indexing and Retrieval (MMCBIR'2001)*, 2001.