SEMI-SUPERVISED IMAGE DATABASE CATEGORIZATION USING PAIRWISE CONSTRAINTS

N. Grira, M. Crucianu and N. Boujemaa

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

ABSTRACT

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. *Semi-supervised clustering*— learning from both labelled and unlabelled data— has consequently become a topic of significant interest. In this paper we present a new semi-supervised clustering algorithm, Pairwise-Constrained Competitive Agglomeration, which is based on a fuzzy cost function that takes *pairwise constraints* into account.

1. INTRODUCTION

Clustering methods that attempt to organize image collections for efficient content access can be grouped into two main categories: partitional or hierarchical. Partitional algorithms are based on the optimization of specific objective functions and a widely used algorithm is Fuzzy C-Means (FCM)[2], a prototype-based clustering algorithm that has been constantly improved for twenty years, by the use of the Mahalanobis distance [6], the adjunction of a noise cluster [4] or the use of competitive agglomeration [5], [3].

Unfortunately, such algorithms do not take specific user-provided information into account and the resulting image categories often do not reflect user expectations. As a consequence, *semi-supervised clustering*—letting the user provide a limited form of supervision— has recently become a topic of significant interest. More specifically, to help unsupervised clustering, a small amount of information reardding e.g. *pairwise constraints* between data items can be used; the constraints simply specify whether two data items should be in the same cluster or not. Even when a user has little or no prior knowledge of the database (ignores what the classes may be, ignores their number), she can easily provide such pairwise constraints.

The few existing semi-supervised clustering algorithms, such as Pairwise Constrained K-means (PCKmeans) [1], rely on parameters that are difficult to set (e.g. the number of clusters) and require a high number of constraints to reach good results. The new semi-supervised clustering algorithm we propose here, Pairwise Constrained Competitive Agglomeration (PCCA), provides significant improvements in these directions.

2. COMPETITIVE AGGLOMERATION: A SHORT REMINDER

Competitive Agglomeration (CA) [5] is a fuzzy partitional algorithm that does not require the user to specify the desired number of clusters. Let $X = \{\mathbf{x}_i |\ i \in \{1,..,N\}\}$ be a set of N vectors, $V = \{\mu_k |\ k \in \{1,..,C\}\}$ the set of prototypes of the C clusters and U the degrees of membership of the vectors to the clusters. CA minimizes the following objective function:

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

with the constraint

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

In (1), $d(\mathbf{x}_i,\mu_k)$ is the distance between the vector $\mathbf{x}_i$ and the cluster prototype μ_k and u_{ik} is the membership of $\mathbf{x}_i$ to a cluster k. The first term is the standard FCM objective function [2]. The second term progressively reduces the number of clusters.

3. PAIRWISE CONSTRAINED COMPETITIVE AGGLOMERATION

3.1. Principle of the Method

The objective function to be minimized should combine the feature-based similarity between data points and the pairwise constraints available. 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 different clusters. Using the same notations as for CA, the objective function PCCA must minimize is:

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

with the same constraint (2).

The prototypes of the clusters ($k \in \{1,..,C\}$) are

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

and cardinalities are computed as $N_s = \sum_{i=1}^{N} u_{is}$.

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 not respecting the pairwise *must-link* constraints and the cost of not respecting the pairwise *cannot-link* constraints. The penalty corresponding to the presence of two such points in different clusters (for must-link constraints) or in the same cluster (for cannot-link constraints) is weighted by their membership values. This second 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 (Competitive Agglomeration) and controls the number of clusters.

When the terms are combined, the final partition will minimize the sum of intra-cluster distances, while partitioning the data set into the smallest number of clusters such that a maximum number of specified constraints are respected. 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 that the equation for updating memberships is

$$u_{rs} = u_{rs}^{FCM} + u_{rs}^{Constraints} + u_{rs}^{Bias} \quad (5)$$

where

$$u_{rs}^{FCM} = \frac{\frac{1}{d^2(\mathbf{x}_r,\mu_s)}}{\sum_{k=1}^{C}\frac{1}{d^2(\mathbf{x}_r,\mu_k)}} \quad (6)$$

$$u_{rs}^{Constraints} = \frac{\alpha}{2d^2(\mathbf{x}_r,\mu_s)}(\overline{C_{v_r}} - C_{v_s}) \quad (7)$$

$$u_{rs}^{Bias} = \frac{\beta}{d^2(\mathbf{x}_r,\mu_s)}(N_s - \overline{N_r}) \quad (8)$$

In (7), C_{v_s} and $\overline{C_{v_r}}$ are defined as

$$C_{v_s} = \sum_{(\mathbf{x}_t,\mathbf{x}_j)\in\mathcal{M}}\sum_{l=1,l\neq s}^{C} u_{jl} + \sum_{(\mathbf{x}_t,\mathbf{x}_j)\in\mathcal{C}} u_{js} \quad (9)$$

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

The first term in equation (5) 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 deprecated according to the pairwise constraints defined by the user. The third term leads to a reduction of the cardinality of spurious clusters, which are discarded if their cardinality drops below a threshold.

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

$$\begin{aligned}\beta(t) &= \frac{\eta_0\exp(-t/\tau)}{\sum_{j=1}^{C}\Big[\sum_{i=1}^{N}(u_{ij})\Big]^2}\Big[\sum_{j=1}^{C}\sum_{i=1}^{N}(u_{ij})^2 d^2(\mathbf{x}_i,\mu_j) \\ &+ \alpha\Big(\sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{M}}\sum_{k=1}^{C}\sum_{l=1,l\neq k}^{C} u_{ik}u_{jl} \\ &+ \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{C}}\sum_{k=1}^{C} u_{ik}u_{jk}\Big)\Big]\end{aligned} \quad (10)$$

The exponential factor makes the last term dominant in the beginning to reduce the number of clusters, then the first 3 terms will dominate, to seek the best partition of the data.

3.2. 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 [5]) discards spurious clusters, two difficulties arise:

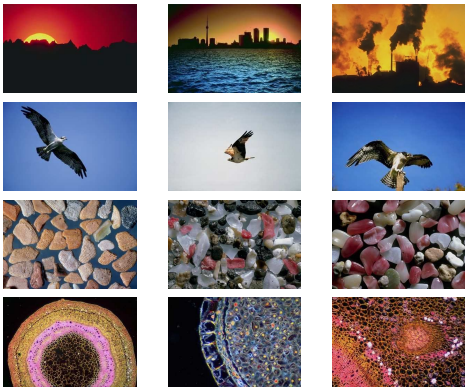

Fig. 1. Each line is a sample of one of the 4 classes of the generalist database

- The threshold has to be changed manually by the user according to the data he wants to categorize. So clustering would become sensitive to a new parameter, when one important goal of PCCA is to easily find an appropriate number of clusters.
- Good clusters may have different cardinalities, so a criterion based only on their minimal cardinality is not effective. If the minimal cardinality is too small, several prototypes can co-exist for a large cluster.

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 dataset, 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. Since we aim to reduce the sensitivity of clustering to parameters, all the results presented here were obtained with a fixed proximity threshold of 0.01.

4. EXPERIMENTAL RESULTS

To evaluate our PCCA algorithm and to compare it to the basic CA algorithm and to PCKmeans, we selected two different ground truthimage databases: a generalist one, having a few large classes and a scientific one, having several classes of very different sizes. The use of two databases with different characteristics should allow us to obtain more significant comparisons and eventually demonstrate the robustness of PCCA. The first image database contains 4 classes of 100 images each; a sample of images is shown in Figure 1. The classes are rather diverse and many images belonging to different classes are quite similar. The second database is composed of images of different phenotypes of *Arabidopsis*

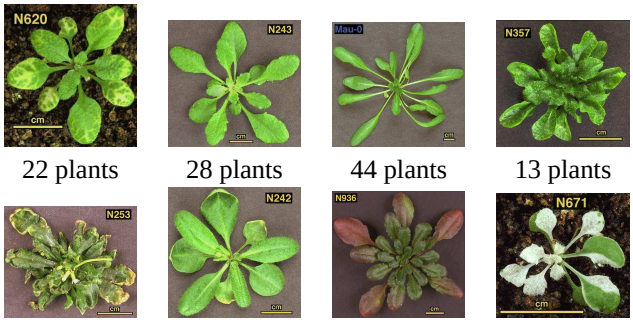

Fig. 2. A sample of the *Arabidopsis* image database, with the number of plants in each class

thaliana (corresponding to slightly different genotypes); a sample of the images is shown in Figure 2. 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 clusters we attempted to find in our study correspond to: 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. In both experiments, pairs of images are randomly selected and the (emulated) user is required to provide the corresponding constraints.

PCCA algorithm outline

- Fix the maximum number of clusters C.
- Randomly initialize prototypes for $j \in \{1, \ldots, C\}$.
- Initialize memberships: equal membership of every feature point to every cluster.
- Compute initial cardinalities for $j \in \{1, \ldots, C\}$.
- **Repeat**
  - Compute β using equation (10).
  - Compute memberships u_{ij} using equation (5).
  - Compute cardinalities N_j for $j \in \{1, \ldots, C\}$.
  - For $j \in \{1, \ldots, C\}$, if $N_j <$ threshold then discard cluster j.
  - Update number of clusters C.
  - Update the prototypes using equation (4).
- **Until** prototypes stabilize.

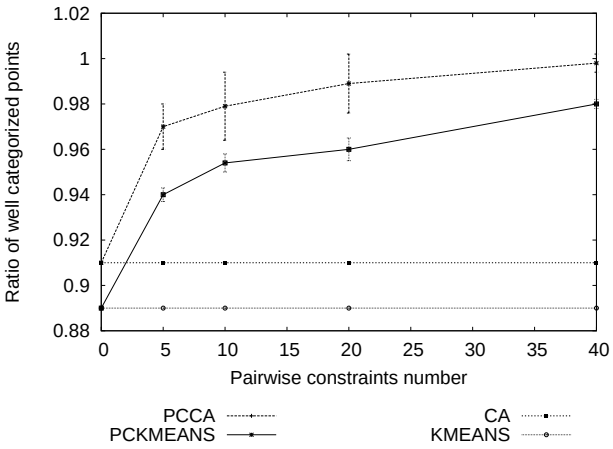

Fig. 3. Results on the groundtruth image database

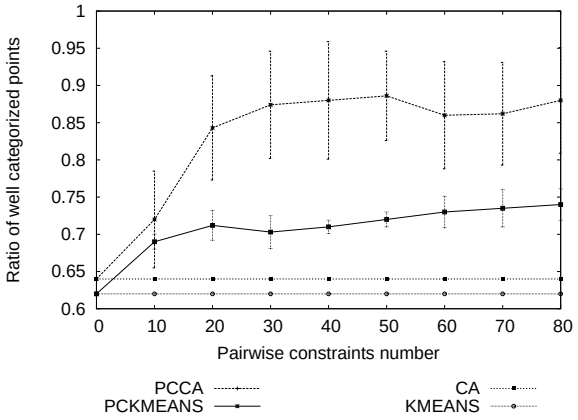

Fig. 4. Results on the *Arabidopsis* database

The image features we used 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. The joint feature vector has more than 600 dimensions, which can make clustering impractical. We used linear principal component analysis To reduce the dimension of the feature vectors; after a 5-fold reduction, we remain within a 5% overall loss of quality in the precision/recall diagrams of query by example. Since the shape of the clusters is usually not spherical, we use the Mahalanobis rather than the Euclidean distance.

Figures 3 and 4 present the dependence between the percentage of well-categorized data points and the number of pairwise constraints considered, for each of the two datatsets. The graph for the basic CA algorithm (ignoring the constraints) is 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 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 (10), part of the fuzzy memberships, directly include the pairwise constraints in the fuzzy clustering process.

5. CONCLUSION

We have shown that, by providing a limited amount of simple knowledge 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 a generalist image database and on the *Arabidopsis* database show that PCCA performs considerably better than unsupervised clustering 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 for the users. The computational complexity of PCCA is linear in the number of data vectors, making this algorithm suitable for real-world clustering applications.

6. REFERENCES

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

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

[3] N. Boujemaa. On competitive unsupervized clustering. In *Proc. of ICPR'2000*, Barcelona, Spain, 3-8 Sept. 2000.

[4] R. N. Dave. Characterization and detection of noise in clustering. *Pattern Recognition Letters*, 12:657–664, 1991.

[5] H. Frigui and R. Krishnapuram. Clustering by competitive agglomeration. *Pattern Recognition*, 30(7):1109–1119, 1997.

[6] E. E. Gustafson and W. C. Kessel. Fuzzy clustering with a fuzzy covariance matrix. In *IEEE CDC*, pages 761–766, San Diego, California, 1979.