

Adaptive Robust Clustering with Proximity-Based Merging for Video-Summary

Bertrand Le Saux, Nizar Grira and Nozha Boujemaa
 INRIA, Rocquencourt
 Domaine de Voluceau, BP-105
 F-78153 Le Chesnay Cedex, France
 email: {bertrand.le-saux, nizar.grira, nozha.boujemaa}@inria.fr

Abstract— To allow efficient browsing of large image collection, we have to provide a summary of its visual content. We present in this paper a new robust approach to categorize image databases : Adaptive Robust Competition with Proximity-Based Merging (ARC-M). This algorithm relies on a non-supervised database categorization, coupled with a selection of prototypes in each resulting category. Each image is represented by a high-dimensional vector in the feature space. A principal component analysis is performed for every feature to reduce dimensionality. Then, clustering is performed in challenging conditions by minimizing a Competitive Agglomeration objective function with an extra noise cluster to collect outliers. Agglomeration is improved by a merging process based on cluster proximity verification.

I. INTRODUCTION

Content-based Image Retrieval (CBIR) aims at indexing images by automatic description, which only depends on their objective visual content. The purpose of browsing is to help user to find rapidly his target-image, and we proceed by providing first an overview of the image database. We propose to first find the main categories of the database, and then build a summary by picking key images in each category. This categorization is performed using image signatures, which represent the visual appearance of the image. The main issues of the problem are the unknown number of categories, the high-dimensionality of the feature space, and the complexity of the natural clusters, which are often overlapping.

Prototype-based clustering algorithm are a popular way to find partitions in complex data. The fuzzy version (Fuzzy C-Means [2]) has been constantly improved for twenty years by the use of the Mahalanobis distance [6], the adjunction of a noise cluster [3] or the competitive agglomeration algorithm [1] [5]. The first versions suffer from a lack of flexibility, since the number of clusters had to be fixed before the clustering. Clustering was performed for different number of clusters, and then the validity of each partition was tested to keep only the best one according to a validity criterion [2]. Several attempts were made in the last decade to propose methods which automatically find the optimal number of clusters in the data-set, in particular the competitive agglomeration [1] which proceeds by reducing the number of clusters over iterations.

The paper is organized as follows : section II introduce some needed background material and notions, then the ARC-M approach is presented in section III. The results both on synthetic

data and on image databases are discussed in section IV and section V summarizes our concluding remarks.

II. BACKGROUND

The Competitive Agglomeration (CA) algorithm [1] is a fuzzy partitional algorithm which allows not to specify the number of clusters. Let $X = \{\mathbf{x}_i | i \in \{1, \dots, N\}\}$ be a set of N vectors representing the images. Let $B = \{\beta_j | j \in \{1, \dots, C\}\}$ represents prototypes of the C clusters. Competitive Agglomeration (CA) algorithm minimizes the following objective function:

$$J = \sum_{j=1}^C \sum_{i=1}^N (u_{ji})^2 d^2(\mathbf{x}_i, \beta_j) - \alpha \sum_{j=1}^C \left[\sum_{i=1}^N (u_{ji}) \right]^2 \quad (1)$$

Constrained by :

$$\sum_{j=1}^C u_{ji} = 1, \text{ for } i \in \{1, \dots, N\} \quad (2)$$

$d^2(\mathbf{x}_i, \beta_j)$ represents the distance from an image signature \mathbf{x}_i to a cluster prototype β_j . The choice of the distance depends on the type of clusters having to be detected. For spherical clusters, Euclidean distance will be used. u_{ji} is the membership of \mathbf{x}_i to a cluster j .

The first term is the standard FCM objective function [2] : the sum of weighted square distances. It allows us to control shape and compactness of clusters. The second term (the sum of squares of clusters' cardinalities) allows us to control the number of clusters. By minimizing both these terms together, the data set will be partitioned in the optimal number of clusters while clusters will be selected to minimize the sum of intra-cluster distances.

The cardinality of a cluster is defined as the sum of the memberships of each image to this cluster :

$$N_s = \sum_{i=1}^N (u_{si}) \quad (3)$$

Membership can be written as :

$$u_{st} = u_{st}^{FCM} + u_{st}^{Bias}, \quad (4)$$

where :

$$w_{st}^{FCM} = \frac{[1/d^2(\mathbf{x}_t, \beta_s)]}{\sum_{j=1}^C [1/d^2(\mathbf{x}_t, \beta_j)]}, \quad (5)$$

and :

$$w_{st}^{Bias} = \frac{\alpha}{d^2(\mathbf{x}_t, \beta_s)} \left(N_s - \frac{\sum_{j=1}^C [1/d^2(\mathbf{x}_t, \beta_j)] N_j}{\sum_{j=1}^C [1/d^2(\mathbf{x}_t, \beta_j)]} \right) \quad (6)$$

During the clustering process, we minimize iteratively (1). At each iteration, the clusters whose cardinality drops below a threshold are discarded, and the number of clusters reduced. As a result, only a few clusters will survive while spurious clusters will become extinct. So competitive agglomeration provides a method to find automatically a number of clusters corresponding to the data.

The parameter α should provide a balance [1] between the two terms of (1) so α at iteration k is defined by :

$$\alpha(k) = \eta_0 \exp(-k/\tau) \frac{\sum_{j=1}^C \sum_{i=1}^N (u_{ji})^2 d^2(\mathbf{x}_i, \beta_j)}{\sum_{j=1}^C \left[\sum_{i=1}^N (u_{ji}) \right]^2} \quad (7)$$

α is weighted by a factor which decreases exponentially along iterations. In the first iterations the second term of equation (1) dominates so the number of clusters drops rapidly. Then, when the optimal number of clusters is found, the first term dominates and the CA algorithm seeks the best partition of the signatures.

III. OUR METHOD : ADAPTIVE ROBUST CLUSTERING WITH PROXIMITY-BASED MERGING (ARC-M)

A. Dimensionality Reduction

Signatures computed for image retrieval are high dimensional. Weighted color histograms [9] used to describe color information have several hundreds of dimensions. To prevent the clustering to be computationally expensive, a principal component analysis is performed to reduce the dimensionality. For each feature, only the first principal components are kept.

B. Adaptive Competition

The parameter α is the weighting factor of the competition process. In equation 7, α is chosen according to the objective function and has the same value and effect for each cluster. Though, during the process, α influences the computation of memberships in equations (4) and (6). The term w_{st}^{Bias} appreciates or depreciates the membership u_{st} of data point \mathbf{x}_t to cluster s according to the cardinality of the cluster. This will cause this cluster to be conserved or discarded respectively.

Since clusters have different capacities, the problem is to attenuate the effect of w_{st}^{Bias} for loose clusters, in order to not discard them too rapidly. We introduce an average distance for each cluster s :

$$d_{moy}^2(s) = \frac{\sum_{i=1}^N (u_{si})^2 d^2(\mathbf{x}_i, \beta_s)}{\sum_{i=1}^N (u_{si})^2} \quad \text{for } 1 \leq s \leq C \quad (8)$$

And an average distance for the whole set of signatures :

$$d_{moy}^2 = \frac{\sum_{j=1}^C \sum_{i=1}^N (u_{ji})^2 d^2(\mathbf{x}_i, \beta_j)}{\sum_{j=1}^C \sum_{i=1}^N (u_{ji})^2} \quad (9)$$

Then, α in equation (6) is expressed as :

$$\alpha_s(k) = \frac{d_{moy}^2}{d_{moy}^2(s)} \alpha(k) \quad \text{for } 1 \leq s \leq C \quad (10)$$

The ratio $d_{moy}^2/d_{moy}^2(s)$ is lesser than 1 for loose clusters, so the effect of w_{st}^{Bias} is attenuated : cardinality of cluster is slowly reduced. On the contrary, $d_{moy}^2/d_{moy}^2(s)$ is greater than 1 for compact clusters, so memberships to these clusters are augmented, and their cardinality is increased : they are more resistant in the competition process. Hence we build an adaptive competition process given by $\alpha_s(k)$ for each cluster s .

C. Robust Clustering

A solution to deal with noisy data and outliers is to capture all the noise signatures in a single cluster [3]. A virtual noise prototype is defined, which is always at the same distance δ from every point in the data-set. Let this noise cluster be the first cluster, and noise prototype noted as β_1 . So we have :

$$d^2(x_i, \beta_1) = \delta^2 \quad (11)$$

Then the objective function (1) has to be minimized with the following particular conditions :

- Distances for the good clusters j are the distance used by Gustafson and Kessel [6] to discriminate ellipsoidal clusters. For clusters $2 \leq j \leq C$, distances are computed using :

$$d^2(\mathbf{x}_i, \beta_j) = |C_j|^{1/p} (\mathbf{x}_i - \beta_j)^T C_j^{-1} (\mathbf{x}_i - \beta_j) \quad (12)$$

where C_j is the fuzzy covariance matrix of the cluster j :

$$C_j = \frac{\sum_{i=1}^N (u_{ji})^2 (\mathbf{x}_i - \beta_j)(\mathbf{x}_i - \beta_j)^T}{\sum_{i=1}^N (u_{ji})^2} \quad (13)$$

The prototypes of clusters j for $2 \leq j \leq C$ are the centroids :

$$\beta_j = \frac{\sum_{i=1}^N (u_{ji})^2 \mathbf{x}_i}{\sum_{i=1}^N (u_{ji})^2} \quad (14)$$

- For the noise cluster $j = 1$, distance is given by (11).

The noise distance δ has to be specified. It would vary from one data-set to another, so it would be based on data-set statistical information. It is computed as the average distance between image signatures and good cluster prototypes :

$$\delta^2 = \delta_0^2 \frac{\sum_{j=2}^C \sum_{i=1}^N d^2(\mathbf{x}_i, \beta_j)}{N(C-1)} \quad (15)$$

The noise cluster is then supposed to catch outliers that are at an equal mean distance from all cluster prototypes. Initially, δ cannot be computed using this formula, since distances are not yet computed. It is just initialized to δ_0 , and the noise cluster becomes significant after a few iterations. δ_0 is a factor which can be used to enlarge or minimize the size of the noise cluster, though in the results that will be presented, $\delta_0 = 1$.

D. Proximity-based Merging

As the CA algorithm proceeds, the cardinality of some clusters drops below a threshold and then we discard these clusters and update the number of clusters. The choice of this threshold is important since it reflects the density of final clusters.

Two drawbacks arise in the way CA discards spurious clusters :

- The threshold has to be changed manually by the user according to the data he wants to categorize. So the clustering becomes sensitive to a new parameter, when one of the main advantages of CA was to find automatically the number of clusters.
- Since clusters may have different cardinalities, a criterion based only on the minimal cardinality of clusters is not efficient. If the minimal cardinality is too low, several prototypes can co-exist for a single large cluster (in this cluster, each point shares its membership between the prototypes, and since there are enough points, the cardinality of each cluster is larger than the threshold, see Fig. 1). On the other hand, if minimal cardinality is too large, some small but distinct clusters may have one single prototype, equidistant of the clusters.

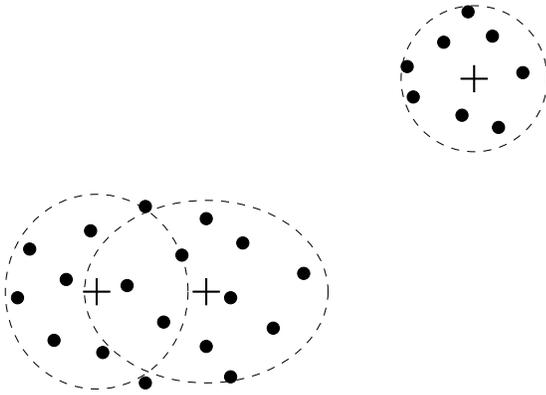


Fig. 1. The large cluster has twice the cardinality of the small one. If the minimal cardinality is small enough to retrieve the small cluster, two categories can survive in the large one, since even if each point shares its membership between two categories, the sum of these memberships for a category will be larger than the threshold under which clusters are discarded

Thus we propose in this section a strategy to improve the agglomeration process in CA. The aim of our method is to let the agglomeration process be statistically dependent from the data. 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 prototypes based on pairwise merging to fit largest clusters.

The proposed procedure reduces the number of prototypes by merging the best pair of prototypes among all possible pairs in term of a given criterion. This merging process is repeated until no more merging is required.

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. We note min_1 and min_2 the two indices corresponding to the pair which has the minimal distance d_{min} , then we merge clusters min_1 and min_2 only if the following criterion is satisfied :

$$d_{min}/d_{max} < \text{proximity measure} \quad (16)$$

where :

$$d_{min} = \min(d(r) \mid r = 1 \dots R)$$

$$d_{max} = \max(d(r) \mid r = 1 \dots R)$$

The proximity measure is a user-defined value. But since our aim is to make the clustering independent of such parameters, as in III-C, the results will be obtained with a fixed proximity threshold.

E. Outline of the New Algorithm

The new method ARC-M can now be summarized :

Fix the maximum number of clusters C .

Initialize randomly prototypes for $2 \leq j \leq C$.

Initialize memberships with equal probability for each image to belong to each cluster.

Compute initial cardinalities for $2 \leq j \leq C$.

Repeat

Compute $d^2(\mathbf{x}_i, \beta_j)$ using (12) for $2 \leq j \leq C$ and (11) for $j=1$.

Update adaptive competition factor $\alpha_s(k)$ using (10) for each cluster.

Compute memberships u_{ji} using equation (4) for each cluster and each signature.

Compute cardinalities N_j for $2 \leq j \leq C$.

For $2 \leq j \leq C$, if $N_j < \text{threshold}$ discard clusters.

Update number of clusters C .

Update prototypes using equation (14).

Update noise distance δ using equation (15).

Repeat

Merge close clusters.

while(no more merging required according to eq. 16)

until(prototypes stabilized)

IV. RESULTS

A. Results on Synthetic Data

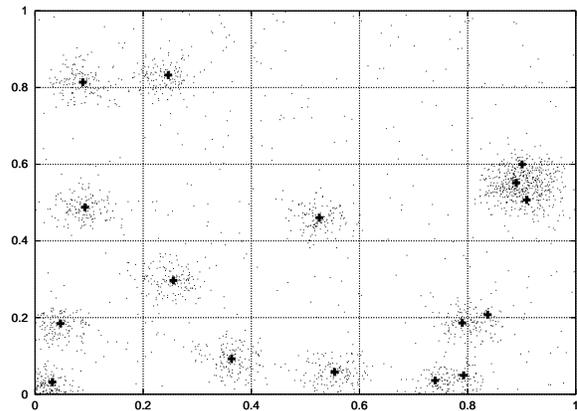


Fig. 2. Data set : 12 Gaussian clusters, with 20% of uniform noise. Basic CA fails to retrieve the largest clusters on the right.

To illustrate the efficiency of the proposed ARC-M algorithm, we use synthetic data-set examples. We compare our

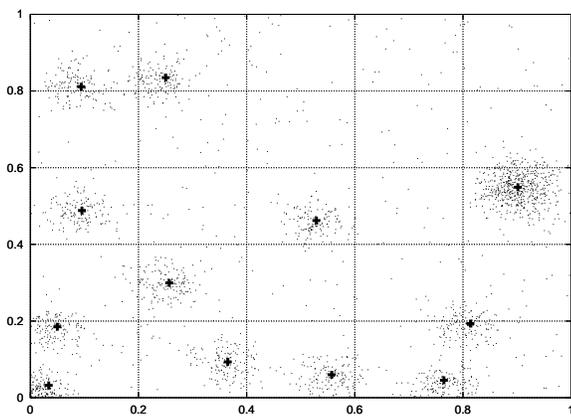


Fig. 3. Data set : 12 Gaussian clusters, with 20% of uniform noise. ARC-M finds a single prototype for the largest clusters.

method with the classic Competitive Agglomeration. With CA, the minimal cardinality threshold is tuned to approximately $N/100$ (N is the number of points in the data-set) to obtain a good clustering. With ARC-M, a cluster is discarded when its cardinality drops below $N/1000$, and the merging process fits largest clusters.

The synthetic data-set consists of 12 clusters of various densities. Fig. 2 shows what happens with CA when clusters have different densities. For the large cluster on the right, the minimal cardinality is too low, so several prototypes have been detected. On Fig. 3, our method proceeds by detecting small clusters while the merging process merges closest ones to fit large clusters.

B. Video-summary

The motivation for this study was to visually categorize images from broadcast news, and provide a compact and quickly browsable summary of this type of multimedia material. Broadcast news departments' archivists are often interested in locating studio scenes with speaker close-up, and graphics. The video-summary is processed according to the following steps :

- Key-frames are extracted from a broadcast news video and are collected as a database.
- The image database is indexed by the signature-set used in our Ikona software [8]. These image signatures were used in our former work without cluster proximity verification [7].
- Then the clustering is performed on these image signatures.

Both methods basic CA and ARC-M are compared. Fig. 4 shows the visualization of the results of clustering by CA. The image signature is projected on the two principal components obtained by principal component analysis. One image is picked in each category to build a summary of the database, which is presented in Fig. 5. Note there's redundancy in the summary : several scenes have more than one prototype, and especially the studio scenes with a speaker, since there is a large proportion of these images. The prototypes visualized on the image signatures set (Fig. 4) shows several concentrated prototypes, which correspond to these images.

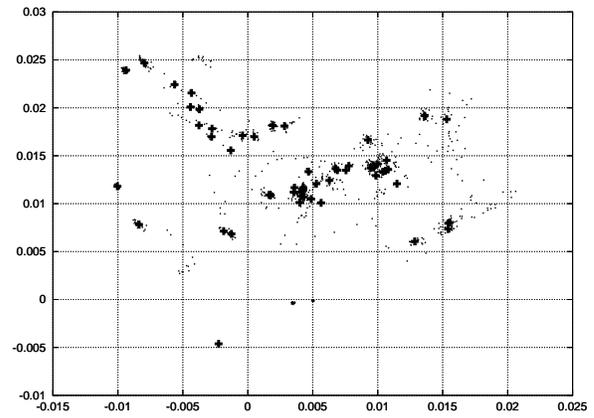


Fig. 4. Visualization in 2D of the data-set by projection of the image signatures on the two principal components obtained by PCA. Results of categorization with CA.



Fig. 5. Summary of the database obtained with CA algorithm. The large cluster of the studio scenes is split in eight categories

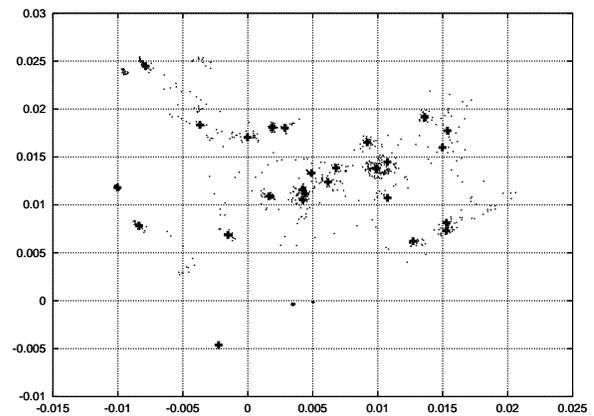


Fig. 6. Visualization in 2D of the data-set by projection of the image signatures on the two principal components obtained by PCA. Results of categorization with ARC-M.



Fig. 7. Summary of the database obtained with ARC-M algorithm. Since our method have an approach to merge clusters, only a single cluster contained all the studio scenes.

With our ARC-M method, there is no redundant images in the summary on Fig. 7 (only one studio image with the speaker is presented). On the figure 6, we can notice that there is only one prototype in the center of the data-set, instead of eight on Fig. 4, this is the cluster corresponding to the studio scenes.

V. CONCLUSION

We have presented a new unsupervised clustering algorithm : ARC-M. It allows to find the number of clusters the more adapted to the data-set. There is a noise cluster to collect ambiguous points and outliers. It uses an appropriate distance to detect clusters of various shapes. Finally, the use of a merge procedure makes the clustering adaptive to the density and the cardinality of the clusters. All these features are particularly adapted to the problems of image database categorization. The algorithm has been successfully applied to categorize images extracted from a video sequence. By picking one image by cluster, a summary of the video sequence can be built.

REFERENCES

- [1] H. Frigui and R. Krishnapuram, Clustering by competitive agglomeration, *Pattern Recognition*, vol. 30, No. 7, pp. 1109-1119, 1997.
- [2] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, 1981.
- [3] R. N. Dave, Characterization and detection of noise in clustering, *Pattern Recognition Letters*, vol. 12, pp. 657-664, 1991.
- [4] C. G. Looney, Interactive clustering and merging with a new fuzzy expected value, *Pattern Recognition*, vol. 35, No. 11, pp. 2413-2423, 2002.
- [5] N. Boujemaa, On Competitive Unsupervised Clustering, *Proc. of ICPR'2000*, 3-8 Sept., Barcelona, Spain.
- [6] E. E. Gustafson and W. C. Kessel, Fuzzy clustering with a fuzzy covariance matrix, *Proc. of IEEE CDC*, 1979, San Diego, California.
- [7] B. Le Saux and N. Boujemaa, Unsupervised Robust Clustering for Image Database Categorization, *Proc. of ICPR'2002*, Quebec, Canada.
- [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.