Unsupervised Robust Clustering for Image Database Categorization

Bertrand Le Saux and Nozha Boujemaa
INRIA, Imedia Research Group
BP 105, F-78153 Le Chesnay, France
Bertrand.Le-Saux@inria.fr,Nozha.Boujemaa@inria.fr

Abstract

Content-based image retrieval can be dramatically improved by providing a good initial database overview to the user. To address this issue, we present in this paper the Adaptive Robust Competition. This algorithm relies on a non-supervised database categorization, coupled with a selection of prototypes in each resulting category. In our approach, each image is represented by a high-dimensional signature in the feature space, and a principal component analysis is performed for every feature to reduce dimensionality. Image database overview is computed in challenging conditions since clusters are overlapping with outliers and the number of clusters is unknown.

1. 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 his target by providing first the best overview of the database. We propose to categorize the database and then to choose a key image for each category. This summary can be used as an initial overview.

The categorization is performed in the image signature space. The main issues of the problem are the high dimensionality of this feature space, the unknown number of natural categories in the data, and the variety and the complexity of these categories, which are often overlapping.

A popular way to find partitions in complex data is to use prototype-based clustering algorithms. The fuzzy version (Fuzzy C-Means [1]) 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 [5] [2]. Specific algorithms have been developed for the categorization [8] [4] and the browsing [11] of image databases.

This paper is organized as follows. §2 presents the background of our work. Our method is presented in §3. The results on image databases are discussed and compared with other clustering methods in §4, §5 summarizes our concluding remarks.

2. Background

The Competitive Agglomeration (CA) algorithm [5] is a fuzzy partitional algorithm which does not require the number of clusters to be specified, which is here unknown. Let $X = \{x_i |\ i \ \epsilon\ \{1,..,N\}\}$ be a set of N vectors representing the images. Let $B = \{\beta_j |\ j\ \epsilon\ \{1,..,C\}\}$ represents prototypes of the C clusters. CA minimizes the following objective function :

$$J = \sum_{j=1}^{C}\sum_{i=1}^{N}(u_{ji})^2 d^2(x_i,\beta_j) - \alpha \sum_{j=1}^{C}\Big[\sum_{i=1}^{N}(u_{ji})\Big]^2 \quad (1)$$

With the constraint :

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

In (1), $d^2(x_i,\beta_j)$ stands for the distance from an image signature x_i to a cluster prototype β_j (for spherical clusters, Euclidean distance will be used) and u_{ji} is the membership of x_i to a cluster j. The first term is the standard FCM objective function [1] : the sum of weighted square distances. The second term leads to reduce the number of clusters. By minimizing both terms simultaneously, the data set will be partitioned in the optimal number of clusters while clusters will be arranged in order to minimize the sum of intra-cluster distances.

Membership can be written as :

$$u_{st} = u_{st}^{FCM} + u_{st}^{Bias}, \quad (3)$$

with

$$u_{st}^{FCM} = \frac{[1/d^2(x_t,\beta_s)]}{\sum_{j=1}^{C}[1/d^2(x_t,\beta_j)]}, \quad (4)$$

and

$$u_{st}^{Bias} = \frac{\alpha}{d^2(x_t, \beta_s)} \big(N_s - \frac{\sum_{j=1}^{C} [1/d^2(x_t, \beta_j)] N_j}{\sum_{j=1}^{C} 1/d^2(x_t, \beta_j)}\big) \quad (5)$$

where the cardinality of a cluster is defined by $N_s = \sum_{i=1}^{N} (u_{si})$. The first term in equation (3) is the membership term in FCM algorithm and takes into account only relative distances to the clusters. The second term leads to a reduction of cardinality of spurious clusters, which are discarded if their cardinality drops below a threshold. So only good clusters are conserved.

α should provide a balance [5] between the two terms of (1), so α at iteration k is defined by :

$$\alpha(k) = \eta_0 \exp(-k/\tau) \frac{\sum_{j=1}^{C} \sum_{i=1}^{N} (u_{ji})^2 d^2(x_i, \beta_j)}{\sum_{j=1}^{C} \left[\sum_{i=1}^{N} (u_{ji}) \right]^2} \quad (6)$$

The exponential factor makes the second term preponderant in a first time to reduce the number of cluster, and then the first term dominates to seek the best partition of the data.

3. Adaptive Robust Competition (ARC)

3.1. Dimensionality Reduction

We have computed a signature space for the Columbia Object Image Library [9] (a 1440 gray scale image database representing 20 objects shot every 5 degrees). This feature space is high-dimensional and contains three signatures :

1. Intensity distribution (16-D) : the gray level histogram.
2. Texture (8-D) : the Fourier power spectrum is used to describe the spatial frequency of the image [10].
3. Shape and Structure (128-D) : the correlogram of edge-orientations histogram (in the same way as color correlogram presented at [7]).

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.

3.2. Adaptive Competition

α is the weighting factor of the competition process. In equation 6, 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 (3) and (5). The term u_{st}^{Bias} appreciates or depreciates the membership u_{st} of data point 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 compacities, the problem is to attenuate the effect of u_{st}^{Bias} for loose clusters, in order to not discard them too rapidly. We introduce an average distance for each cluster s :

$$d_{moy}^2(s) = \frac{\sum_{i=1}^{N} (u_{si})^2 d^2(x_i, \beta_s)}{\sum_{i=1}^{N} (u_{si})^2} \quad for\, 1 \leq s \leq C \quad (7)$$

And an average distance for the whole set of signatures :

$$d_{moy}^2 = \frac{\sum_{j=1}^{C} \sum_{i=1}^{N} (u_{ji})^2 d^2(x_i, \beta_j)}{\sum_{j=1}^{C} \sum_{i=1}^{N} (u_{ji})^2} \quad (8)$$

Then, α in equation (5) is expressed as :

$$\alpha_s(k) = \frac{d_{moy}^2}{d_{moy}^2(s)} \alpha(k) \quad for\, 1 \leq s \leq C \quad (9)$$

The ratio $d_{moy}^2 / d_{moy}^2(s)$ is lesser than 1 for loose clusters, so the effect of u_{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.

3.3. Robust clustering

A solution to deal with outliers and data points with ambiguous memberships is to capture such signatures in a single cluster [3]. Let this noise cluster be the first cluster, and let define a virtual noise prototype β_1 such as :

$$\forall\, i \;\; d^2(x_i, \beta_1) = \delta^2 \quad (10)$$

Then the objective function (1) has to be minimized under the following constraint :

- Distances for the good clusters j are defined by :

  $$d^2(x_i, \beta_j) = (x_i - \beta_j)^T A_j (x_i - \beta_j) \quad for\; 2 \leq \; j \; \leq C. \quad (11)$$

  where A_j are positive definite matrices. If A_j are identity matrices, then the distance is Euclidean distance, and the prototypes of clusters $2 \leq j \leq C$ are :

  $$\beta_j = \frac{\sum_{i=1}^{N} (u_{ji})^2 x_i}{\sum_{i=1}^{N} (u_{ji})^2} \quad (12)$$

- For the noise cluster $j = 1$, distance is given by (10).

The noise distance δ is computed as the average of distances between image signatures and good cluster prototypes :

$$\delta^2 = \delta_0^2 \frac{\sum_{j=2}^{C} \sum_{i=1}^{N} d^2(x_i, \beta_j)}{N(C-1)} \tag{13}$$

The noise cluster is then supposed to catch outliers that are at an equal mean distance from all cluster prototypes. The factor δ_0 is an initialization factor, and can be used to enlarge or minimize the size of the noise cluster, though in the results that will be presented, $\delta_0 = 1$.

3.4. Choice of distance for good clusters

What would be the most appropriate choice for (11) ? The image signatures are composed of different features which describe different attributes. The distance between signatures is defined as the weighted sum of partial distances for each feature $1 \leq f \leq F$:

$$d(x_i, \beta_j) = \sum_{f=1}^{F} w_{j,f} d_f(x_i, \beta_j) \tag{14}$$

Since the natural categories in image databases have various shapes (the more often hyper-ellipsoidal) and are overlapping, Euclidean distance is not appropriate. So the Mahalanobis distance [6] is used to discriminate image signatures. For clusters $2 \leq j \leq C$, partial distances for feature f are computed using :

$$d_f(x_i, \beta_j) = |C_{j,f}|^{1/p_f} (x_{i,f} - \beta_{j,f})^T C_{j,f}^{-1} (x_{i,f} - \beta_{j,f}) \tag{15}$$

where $x_{i,f}$ and $\beta_{j,f}$ are the restrictions of image signatures x_i and cluster prototype β_j to the feature f. p_f is the dimension of the subspace corresponding to feature f. $C_{j,f}$ is the covariance matrix of cluster j for the feature f :

$$C_{j,f} = \frac{\sum_{i=1}^{N} (u_{ji})^2 (x_{i,f} - \beta_{j,f})(x_{i,f} - \beta_{j,f})^T}{\sum_{i=1}^{N} (u_{ji})^2} \tag{16}$$

3.5. Normalization of features

The features have different orders of magnitude and different dimensions, so the distance cannot be a simple sum of partial distances. The idea is to learn the weights $w_{j,f}$ in equation (14) during the clustering process. Ordered Weight Averaging [12] is used, as proposed in [4].

First, partial distances are sorted in ascending order. For each feature f, the average rank of corresponding partial distance over images is obtained :

$$w_{j,f}^{(k)} = \frac{1}{n} \sum_{i=1}^{n} \frac{2(F - rank(d_f(x_i, \beta_j)))}{F(F+1)} \tag{17}$$

And the weight at iteration $k > 0$ is updated using :

$$w_{j,f}^{(k)} = w_{j,f}^{(k-1)} + \frac{2(F - r_f)}{F(F+1)} \tag{18}$$

In this process : 1) Features are normalized. 2) Similar images according to a single feature (i.e. which have a small partial distance) are clustered together since the weight of this feature will be increased.

3.6. Algorithm outline

```
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.
Initialize feature weights uniformly for each cluster $2 \leq j \leq C$.
Compute initial cardinalities for $2 \leq j \leq C$.
Repeat
    Compute covariance matrix for $2 \leq j \leq C$ and feature subsets $1 \leq f \leq F$ using (16).
    Compute $d^2(x_i, \beta_j)$ using (10) for $j = 1$ and (15) for $2 \leq j \leq C$.
    Update weights for clusters $2 \leq j \leq C$ using (18) for each feature.
    Compute $\alpha_j$ for $1 \leq j \leq C$ using equations (9) and (6).
    Compute memberships $u_{ji}$ using equation (3) for each cluster and each signature.
    Compute cardinalities $N_j$ for $2 \leq j \leq C$.
    For $2 \leq j \leq C$, if $N_j < threshold$ discard cluster $j$.
    Update number of clusters $C$.
    Update the prototypes using equation (12).
    Update noise distance $\delta$ using equation (13).
Until (prototypes stabilize).
```

4. Results and discussion

ARC is compared to two other clustering algorithms : the basic CA algorithm presented in §2 and the Self-Organization of Oscillator Network (SOON) algorithm [4].

The categorization is performed on the three features. For each category, a prototype is chosen according to the following steps : First, the average value of each feature is

Table 1. comparison of the results of the clustering methods with the ground-truth

	ARC	SOON	CA
mass of mis-categorized images	24%	38%	39%
noise cluster mass	4%	27%	0%

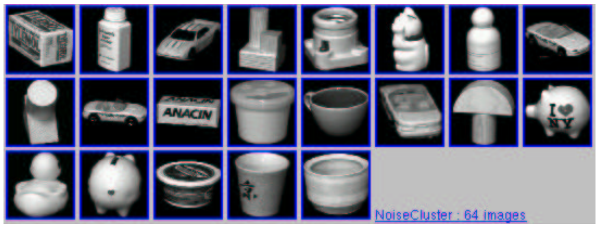

Figure 1. Summary with ARC.

computed over image ; Then the average of all images defines a virtual prototype ; The real prototype is the nearest image to the virtual one.

The three summaries are presented on figures (1), (2) and (3). Almost all the natural categories are retrieved with the three methods. But with SOON or CA algorithms, some categories are split in several clusters, so prototypes are redundant. Our method provides a better summary with less redundancy.

Then, since the CA algorithm has no cluster to collect ambiguous image signatures, the clusters are noisy. With both ARC and SOON algorithms, noise signatures are put in a separate cluster, so clusters considered as good have quite no noise. With the SOON algorithm, more than one quarter of the database is considered as noise (table 1). It leads to have really good (i.e. without noise) clusters, but the drawback is that cluster are small, and contain not more than the third of the natural category : so they do not provide a good representation of the database. Our method puts only the ambiguous images in the noise cluster, and finds almost all the images of the natural category.

5. Conclusion

We have presented a new unsupervised and adaptive clustering algorithm to categorize image databases. When prototypes of each category are picked and collected together, it provides a summary for the image database. It allows to face the problems raised by image database browsing and more specifically handle the “page zero” one. It allows to compute the optimal number of clusters in the dataset. It collects outliers and ambiguous image signatures in a noise cluster, to prevent them from biasing the categorization process. Finally, it uses an appropriate distance to retrieve clusters of various shapes and densities.

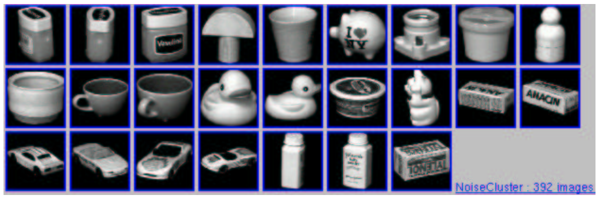

Figure 2. Prototypes of clusters with SOON.

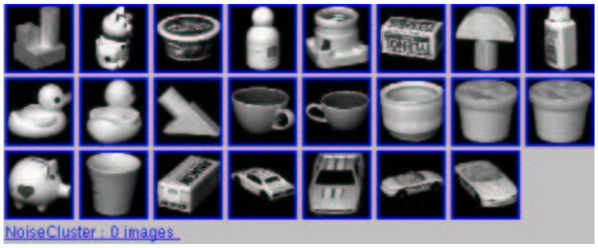

Figure 3. Prototypes of clusters with CA.

References

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

[2] N. Boujemaa. On competitive unsupervized clustering. In *Proc. of ICPR'2000*, Barcelona, Spain.

[3] R. N. Dave. Characterization and detection of noise in clustering. *Pattern Recognition Letters*, 12, 1991.

[4] H. Frigui, N. Boujemaa, and S.-A. Lim. Unsupervised clustering and feature discrimination with application to image database categorization. In *NAFIPS*, Vancouver, Canada, 2001.

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

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

[7] J. Huang, S. R. Kumar, M. M, and Z. W.-J. Spatial color indexing and applications. In *ICCV*, Bombay, India, 1998.

[8] S. Medasani and R. Krishnapuram. Categorization of image databases for efficient retrieval using robust mixture decomposition. In *IEEE CBAIVL'1998*, Santa Barbara, California.

[9] S. A. Nene, S. K. Nayar, and H. Murase. Columbia object image library (coil-20). Technical report, Columbia University, http://www.cs.columbia.edu/CAVE/, 1996.

[10] H. Niemann. *Pattern Analysis and Understanding*. Springer, Heidelberg, 1990.

[11] J. Z. Wang, G. Wiederhold, O. Firschein, and S. X. Wei. Content-based image indexing and searching using daubechies' wavelets. *Int. J. on Digital Libraries*, 1(4), 1997.

[12] R. R. Yager. On ordered weighted averaging aggregation operators in multicriteria decision making. *Systems, Man and Cybernetics*, 18(1), 1988.