Levels of details for Gaussian mixture models

Vincent Garcia[1], Frank Nielsen[1,2], and Richard Nock[3]

[1] Ecole Polytechnique
Laboratoire d'informatique LIX
91128 Palaiseau Cedex, France
[2] Sony Computer Science Laboratories, Inc.
3-14-13 Higashi Gotanda
141-0022 Shinagawa-Ku, Tokyo, Japan
[3] Université des Antilles-Guyane, CEREGMIA
Campus de Schoelcher, BP 7209
97275 Schoelcher, Martinique, France

Abstract. Mixtures of Gaussians are a crucial statistical modeling tool at the heart of many challenging applications in computer vision and machine learning. In this paper, we first describe a novel and efficient algorithm for simplifying Gaussian mixture models using a generalization of the celebrated k-means quantization algorithm tailored to relative entropy. Our method is shown to compare experimentally favourably well with the state-of-the-art both in terms of time and quality performances. Second, we propose a practical enhanced approach providing a hierarchical representation of the simplified GMM while automatically computing the optimal number of Gaussians in the simplified mixture. Application to clustering-based image segmentation is reported.

1 Introduction and prior work

A mixture model is a powerful framework to estimate the probability density function of a random variable. For instance, the Gaussian mixture models (GMMs for short) – also known as mixture of Gaussians (MoGs) – have been widely used in many different area domains such as image processing. For a given mixture model f, the probability density function evaluated at $x \in \mathbb{R}^d$ is given by

$$f(x) = \sum_{i=1}^{n} \alpha_i f_i(x) \tag{1}$$

where $0 \leq \alpha_i \leq 1$ denotes the weight of each mixture component f_i such as $\sum_{i=1}^{n} \alpha_i = 1$. Given a GMM f, each function f_i is a multivariate Gaussian function

$$f_i(x) = \frac{1}{(2\pi)^{d/2}|\Sigma_i|^{1/2}} \exp\left(-\frac{(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i)}{2}\right) \tag{2}$$

parametrized by its mean $\mu_i \in \mathbb{R}^d$ and its covariance symmetric positive-definite matrix $\Sigma_i \succ 0$. It is common to estimate model parameters from independent and

identically-distributed observations using the expectation-maximization (EM) algorithm [1].
A typical operation on mixture models is the estimation of statistical measures such as Shannon entropy or the Kullback-Leibler divergence. With large number of components in the mixture model (*e.g.* arising from a kernel-based Parzen density estimation [2]), the estimation of these measures is prohibitive in terms of computation time. The computational time can be strongly decreased by reducing the number of components in the mixture model. The simplest method to obtain a compact representation of f is to re-learn the mixture model directly from the source dataset. However, this may not be applicable for two reasons. First, the estimation of a mixture model is computationally expensive if we consider large datasets. Second, the source dataset can be unavailable. Thus, the most appropriated solution is to simplify the initial mixture model f.
Given a mixture model f composed of n components (see equation (1)), the problem of mixture model simplification consists in computing a simpler mixture model g

$$g(x) = \sum_{j=1}^{m} \alpha'_j g_j(x) \tag{3}$$

with m components ($1 \leq m < n$) such as g is the "best" approximation of f with respect to a similarity measure.

Some GMM simplification methods have been proposed in the last decade. Zhang and Kwok [3] have proposed to simplify a GMM by first grouping similar components together and then performing local fitting through function approximation. By using the squared loss to measure the distance between mixture models, their algorithm naturally combines the two different tasks of component clustering and model simplification. Goldberger *et al.* [4] have proposed a fast GMM simplification algorithm named UTAC (Unscented Transform Approximation Clustering) based on the Unscented Transform (UT) method [5] [6]. The UTAC algorithm proceeds by maximizing the UTA (Unscented Transform Approximation of the negative cross-entropy) criterion computed between the two GMMs f and g. The authors have shown that the UTA criterion can be maximized with a standard EM-like algorithm. Davis and Dhillon [7] have proposed a hard clustering algorithm based on the decomposition of the relative entropy as the sum of a Burg matrix divergence with a Mahalanobis distance parametrized by the covariance matrices. Goldberger and Roweis [8] have proposed a GMM simplification algorithm based on the k-means hard clustering.
These methods have two disadvantages. First, they only consider the problem of GMM simplification. However, other kind of mixture models have been successfully used in different applications such as multinomial mixture models in text classification [9]. Proposing a simplification algorithm working not only on GMMs but on a generic wider class of mixture models, called exponential families, is necessary. Second, they require the user to specify the number of Gaussians (denoted m) used in the simplified model g, the optimal value of m depending both on the initial GMM and on the application.

In this paper, we first describe a novel and efficient algorithm for simplifying GMMs using a generalization of the celebrated k-means quantization algorithm tailored to relative entropy (see section 2). Our algorithm extends easily to *arbitrary* mixture of exponential families. The proposed method is shown to compare favourably well with the state-of-the-art UTAC algorithm both in terms of time and quality performances. Second, we describe an algorithm based on the G-means algorithm [10] who (1) allows to automatically learn the *optimal* number of Gaussians m in the simplified model and (2) provides a progressive representation of the GMM (see section 3).

2 Entropic quantization of GMMs

2.1 Relative entropy and Bregman divergence

The fundamental measure between statistical distributions is the relative entropy, also called the Kullback-Leibler divergence (denoted by KLD). Given two distributions f_i and f_j, the KLD is an oriented distance (asymmetric) and is defined as

$$\mathrm{KLD}(f_i||f_j) = \int f_i(x) \log \frac{f_i(x)}{f_j(x)} \mathrm{d}x. \tag{4}$$

This fastidious integral computation yields for multivariate normal distributions

$$\begin{aligned}\mathrm{KLD}(f_i||f_j) =& \frac{1}{2} \log \left(\frac{\det \Sigma_j}{\det \Sigma_i} \right) + \frac{1}{2} \mathrm{tr}\left(\Sigma_j^{-1} \Sigma_i \right) \\ &+ \frac{1}{2} (\mu_j - \mu_i)^T \Sigma_j^{-1} (\mu_j - \mu_i) - \frac{d}{2}\end{aligned} \tag{5}$$

where $\mathrm{tr}(\Sigma)$ is the matrix trace operator. We can avoid the integral computation using the canonical form of exponential families [11]

$$f_F(x|\tilde{\Theta}) = \exp\left\{ \langle \tilde{\Theta}, t(x) \rangle - F(\tilde{\Theta}) + C(x) \right\} \tag{6}$$

where $\tilde{\Theta}$ are the *natural parameters* associated with the *sufficient statistics* $t(x)$. The *log normalizer* $F(\tilde{\Theta})$ is a strictly convex and differentiable function that specifies uniquely the exponential family, and the function $C(x)$ is the carrier measure. The relative entropy between two distribution members of the same exponential family is equal to the Bregman divergence defined for the log normalizer F on the natural parameter space:

$$\mathrm{KLD}(f_i||f_j) = D_F(\tilde{\Theta}_j||\tilde{\Theta}_i) \tag{7}$$

where

$$D_F(\tilde{\Theta}_j||\tilde{\Theta}_i) = F(\tilde{\Theta}_j) - F(\tilde{\Theta}_i) - \langle \tilde{\Theta}_j - \tilde{\Theta}_i, \nabla F(\tilde{\Theta}_i) \rangle. \tag{8}$$

The $\langle \cdot, \cdot \rangle$ denotes the inner product and ∇F is the gradient operator. For multivariate Gaussian distributions, we consider mixed-type vector/matrix parameters (μ, Σ). The sufficient statistics is *stacked* into a two-part d-dimensional

vector/matrix entity $t(x) = (x, -\frac{1}{2}xx^T)$ associated with the natural parameters $\tilde{\Theta} = (\theta, \Theta) = (\Sigma^{-1}\mu, \frac{1}{2}\Sigma^{-1})$. The log normalizer specifying the exponential family is [12]

$$F(\tilde{\Theta}) = \frac{1}{4}\mathrm{tr}(\Theta^{-1}\theta\theta^T) - \frac{1}{2}\log\det\Theta + \frac{d}{2}\log\pi. \tag{9}$$

The inner product $\langle\tilde{\Theta}_p, \tilde{\Theta}_q\rangle$ is then a composite inner product obtained as the sum of two inner products of vectors and matrices: $\langle\tilde{\Theta}_p, \tilde{\Theta}_q\rangle = \langle\Theta_p, \Theta_q\rangle + \langle\theta_p, \theta_q\rangle$. For matrices, the inner product is defined by the trace of the matrix product $\Theta_p\Theta_q^T$: $\langle\Theta_p, \Theta_q\rangle = \mathrm{tr}(\Theta_p\Theta_q^T)$. The gradient ∇F is given by

$$\nabla F(\tilde{\Theta}) = \left(\frac{1}{2}\Theta^{-1}\theta\ ,\ -\frac{1}{2}\Theta^{-1} - \frac{1}{4}(\Theta^{-1}\theta)(\Theta^{-1}\theta)^T\right). \tag{10}$$

2.2 Bregman *k*-means

Banerjee *et al.* [11] extended Lloyd's k-means algorithm to the class of Bregman divergences, generalizing also the former Linde-Buzo-Gray clustering algorithm. They proved that the simple Lloyd's iterative algorithm minimizes *monotonically* the Bregman (right-sided) loss function:

$$\mathrm{LossFunction}_F(\{x_1, ..., x_n\}; k) = \min_{c_1,...,c_k} \sum_k \sum_i D_F(x_i||c_k).$$

where x_i are the source point sets and c_k the respective cluster centroids. A right-sided Bregman k-means is a left-sided differential entropic (*i.e.* KLD) clustering, and vice-versa. Thus, we propose a GMM simplification algorithm based on Bregman k-means. The k-means algorithm is the repetition until convergence of two steps: First, determine membership in clusters (repartition step); second, recompute the centroids. The algorithms 1 and 2 respectively present our right-sided and left-sided Bregman k-means clustering algorithms (denoted BKMC). For these algorithms, $\tilde{\Theta}$ and $\tilde{\Theta}'$ denote natural parameters respectively for GMMs f and g.

2.3 Symmetric Bregman *k*-means

The BKMC algorithm can be modified in order to use the symmetric Bregman divergence instead of a sided one. Indeed, the use of a symmetric similarity measure is required for common applications such as content-based image retrieval. Given two Gaussians $\tilde{\Theta}_p$ and $\tilde{\Theta}_q$ (natural parameters), the symmetric Bregman divergence SD_F (used in the repartition step) is defined as the mean of the right-sided and left-sided Bregman divergences:

$$SD_F(\tilde{\Theta}_p, \tilde{\Theta}_q) = \frac{D_F(\tilde{\Theta}_q||\tilde{\Theta}_p) + D_F(\tilde{\Theta}_p||\tilde{\Theta}_q)}{2} \tag{15}$$

Algorithm 1 BKMC right-sided(f,m)

1: Initialize the GMM g.
2: **repeat**
3: Compute the cluster C: the Gaussian f_i belongs to cluster C_j if and only if

$$D_F(\tilde{\Theta}_i \| \tilde{\Theta}'_j) < D_F(\tilde{\Theta}_i \| \tilde{\Theta}'_l),\ \forall l \in [1,m] \setminus \{j\} \tag{11}$$

4: Compute the centroids: the weight and the natural parameters of the j-th centroid (*i.e.* Gaussian g_j) are given by:

$$\alpha'_j = \sum_i \alpha_i, \quad \theta'_j = \frac{\sum_i \alpha_i \theta_i}{\sum_i \alpha_i}, \quad \Theta'_j = \frac{\sum_i \alpha_i \Theta_i}{\sum_i \alpha_i} \tag{12}$$

The sum $\sum_i$ is performed on $i \in [1,m]$ such as $f_i \in C_j$.
5: **until** the cluster does not change between two iterations.

Algorithm 2 BKMC left-sided(f,m)

1: Initialize the GMM g.
2: **repeat**
3: Compute the cluster C: the Gaussian f_i belongs to cluster C_j if and only if

$$D_F(\tilde{\Theta}'_j \| \tilde{\Theta}_i) < D_F(\tilde{\Theta}'_l \| \tilde{\Theta}_i),\ \forall l \in [1,m] \setminus \{j\}$$

4: Compute the centroids: the weight and the natural parameters of the j-th centroid (*i.e.* Gaussian g_j) are given by:

$$\alpha'_j = \sum_i \alpha_i, \quad \tilde{\Theta}'_j = \nabla F^{-1}\left(\sum_i \frac{\alpha_i}{\alpha'_j} \nabla F\left(\tilde{\Theta}_i\right)\right) \tag{13}$$

where

$$\nabla F^{-1}(\tilde{\Theta}) = \left(-\left(\Theta + \theta\theta^T\right)^{-1}\theta\ ,\ -\frac{1}{2}\left(\Theta + \theta\theta^T\right)^{-1}\right) \tag{14}$$

The sum $\sum_i$ is performed on $i \in [1,m]$ such as $f_i \in C_j$.
5: **until** the cluster does not change between two iterations.

Similarly, the symmetric centroid c_s is computed from the right-sided and left-sided centroids (respectively denoted c_r and c_l). The symmetric centroid c_s belongs to the geodesic link between c_r and c_l. A point on this link is given by

$$c_\lambda = \nabla F^{-1} \left(\lambda \nabla F(c_r) + (1 - \lambda) \nabla F(c_l) \right) \tag{16}$$

where $\lambda \in [0, 1]$. The symmetric centroid $c_s = c_\lambda$ verifies

$$SD_F(c_\lambda, c_r) = SD_F(c_\lambda, c_l). \tag{17}$$

A standard dichotomy search on λ allows to quickly find the symmetric centroid c_s for a given precision.

3 Hierarchical GMM representation

Hamerly and Elkan [10] proposed to adapt the k-means clustering algorithm to learn automatically the number of clusters (parameter k) during the process. Their algorithm, called G-means for Gaussian-means, starts with a small number of centroids (usually 1) and splits iteratively the centroids. G-means repeatedly makes decisions based on the statistical Anderson-Darling test [13]: If the data currently assigned to a centroid follow a normal distribution, then the data are represented by their centroid; otherwise, the data are split into two subsets. The G-means algorithm directly provides a hierarchical clustering of the input data. In this section, we propose a GMM simplification algorithm based on G-means and BKMC algorithms. This algorithm, named Bregman G-means clustering algorithm (BGMC for short) and described in algorithm 3, first allows to automatically learn the *optimal* number of Gaussians m in the simplified model, and second provides a progressive representation of the GMM. The problem here is to determine if a set of Gaussians (GMM) follows a Gaussian distribution. If so, the set is represented by one Gaussian: its centroid (right-sided, left-sided, or symmetric). Otherwise, the Gaussian set is divided in two subsets. We reasonably assume that a GMM (Gaussian set) is a Gaussian distribution if a large set of l points drawn from this GMM verify the Anderson-Darling test. In our experiments, l was set to $l = 10000$ and the confidence level (here denoted β) used in the Anderson-Darling test was set to $\beta = 95\%$. The algorithm 3 starts with BGMC(N, f, c, α) where N is the root of an empty binary tree, f is a GMM, c is the centroid (right-sided, left-sided, or symmetric) of f, and $\alpha = \sum_{i=1}^{n} \alpha_i = 1$. N_{left} and N_{right} respectively denote the left-child and the right-child of the node N.

The hierarchical structure of the simplified GMM g allows us to introduce the notion of *resolution*, the successive resolutions given a progressive representation of g. Each node of the tree contains a weighted Gaussian. The resolution r corresponds to all the weighted Gaussians contained in nodes of depth r. The resolution 0 corresponds to a GMM containing only one Gaussian: the tree root. The maximal resolution (*i.e.* the tree height) contains all the leafs of the tree. The *optimal* value of m is given by the GMM size at the maximal resolution.

Algorithm 3 Calculate BGMC(N, f, c, α)

Store the centroid c and the weight α in the node N.
Draw a set of l points $X = \{x_1, \cdots, x_l\}$ from f.
Split the centroid c into two centroids c_1 and c_2.
Perform a Bregman k-means on c_1 and c_2. Let f_1 (resp. f_2) be the set containing the weighted Gaussians of f closer to c_1 (resp. c_2) than c_2 (resp. c_1). Let α_1 (resp. α_2) be the sum of all the weights of the Gaussians contained in f_1 (resp. f_2).
Compute the projection vector $v = \mu_1 - \mu_2$ where μ_1 and μ_2 are respectively the mean of c_1 and c_2.
Given X and v, use the Anderson-Darling statistical test [13] to detect if f is a normal distribution (at confidence level $\beta = 0.95$).
if f is a normal distribution **then**
    Stop the process; the current node N is a leaf (N_{left} and N_{right} are null).
else
    Compute BGMC(N_{left}, f_1, c_1, α_1).
    Compute BGMC(N_{right}, f_2, c_2, α_2).
end if

4 Experiments

4.1 Bregman *k*-means clustering

In this section, we compare the influence of the Bregman divergence type (right-sided, left-sided, or symmetric) on the quality of the simplified GMM g. This quality is evaluated through the standard right-sided Kullback-Leibler divergence (KLD) between f and g estimated with a classical Monte-Carlo algorithm [14] since it does not admit any closed-form solution. For this experiment, the initial GMM f is composed of 32 Gaussians and is computed from the image Baboon: First we perform a standard k-means algorithm to gather RGB pixels in 32 classes, and second we compute each f_i with a standard EM algorithm. The dimension of the Gaussians is 3 (components RGB: red, green, blue).
The figure 1 shows the evolution of the KLD as a function of m (number of the Gaussians in the simplified GMM) for the different Bregman divergence types. First, the KLD decreases with m as expected whatever the Bregman divergence type used. Indeed, the quality of the approximation of the initial GMM f increases with the number of Gaussians in the simplified model g. Second, the left-sided Bregman divergence gives the best results and the right-sided the worst. Indeed, the measure used to evaluate the quality of the simplification is the right-sided KLD. The left-sided Bregman clustering on natural parameters amounts to compute a right-sided KLD clustering on corresponding probability measures. The symmetric BKMC provides better results than right-sided BKMC but worse than left-sided BKMC. In the paper remainder, we will use the left-sided BKMC.

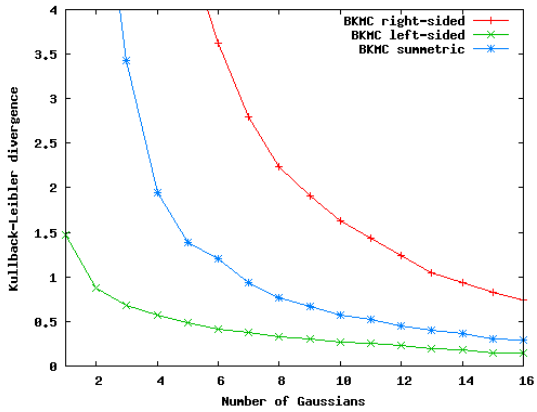

Fig. 1. Evolution of the KLD as a function of m for algorithms right-sided, left-sided, and symmetric BKMC. The left-sided BKMC provides the best approximation of the initial GMM.

4.2 Method comparison

4.3 BKMC versus UTAC

The figure 2 shows the evolution of the KLD as a function of m (number of components in the simplified GMM) for algorithms UTAC and BKMC (left-sided). Both algorithms are written in Java. The initial GMM f is computed as in section 4.1. First, whatever the algorithm used (UTAC and BKMC), the KLD decreases with m. Second, BKMC provides the best results and is faster than UTAC: for $m = 16$, the clustering process is performed in 20 milliseconds for BKMC and 107 milliseconds for UTAC on a Dell Precision M6400 laptop (Intel Core 2 duo @ 2.53GHz, 4Go DDR2 memory, Windows Vista 64 bits, Java 1.6). Indeed, BKMC is based on a k-means algorithm which generally quickly converges. UTAC uses a EM method known to slowly converge (*i.e.* within a threshold after a large number of iterations). We automatically stop the UTAC process after 30 iterations if the process has not converged.

4.4 Clustering-based image segmentation

In this section, we apply the GMM simplification methods in the context of clustering-based image segmentation problem. Given an image, a pixel x can be considered as a point in $\mathbb{R}^3$. Given a GMM g of m Gaussians, the segmentation is performed by classifying each pixel x to the most probable class C_i:

$$g_i(x) > g_j(x)\ \forall j \in [1, m] \setminus \{i\}$$

This segmentation is illustrated by assigning to the pixel x the value of the class representative μ'_i (see figure 3). The images used for the experiments are

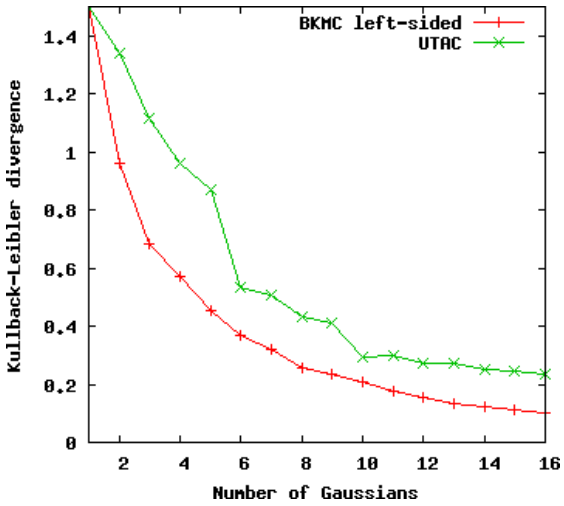

Fig. 2. Evolution of the KLD as a function of m for algorithms BKMC and UTAC.

Baboon, Lena, Colormap, and Shantytown. The first and second rows show respectively the input image and the segmentation computed from the initial GMM f composed 32 Gaussians. The third and fourth rows show the segmentations computed after the simplification of f respectively with the algorithms UTAC and BKMC. With all images tested, the algorithm BKMC provides the best results (visually and according to the KLD value).

4.5 Hierarchical GMM representation

In this section, we apply the BGMC algorithm (hierarchical GMM) in the context of clustering-based image segmentation. The figure 4 shows the segmentation obtained from different resolution of the hierarchical GMM. The segmentation quality increases with the resolution. A resolution equal to 0 provides a GMM composed only of one Gaussian: all the pixel of the input image belongs to the same class. The *optimal* value of m is given by the GMM at the maximal resolution. For each image, we give below this optimal value m, the maximum resolution, and the KLD between the initial GMM f and the *optimal* simplified GMM:
- Baboon: $m = 14$, max. res.=8, KLD=0.18
- Lena: $m = 14$, max. res.=7, KLD=0.13
- Colormap: $m = 14$, max. res.=9, KLD=0.59
- Shantytown: $m = 13$, max. res.=5, KLD=0.28

On average, the construction of the hierarchical GMM is performed in 466 ms.

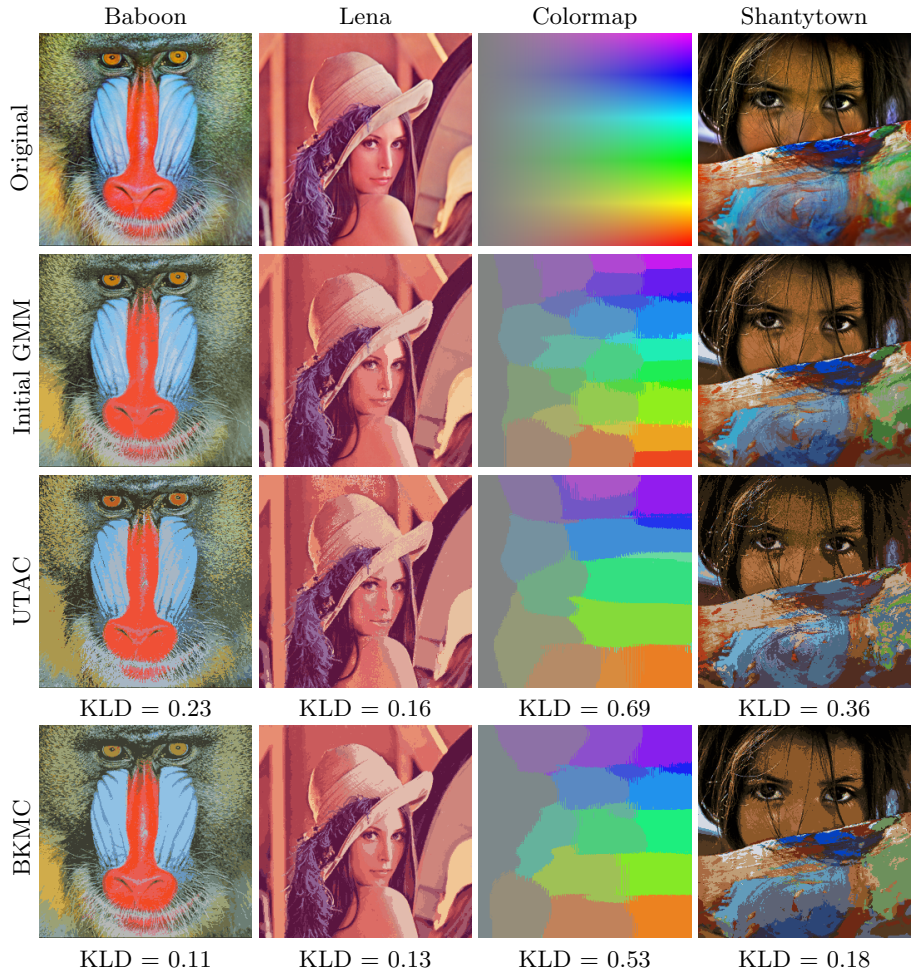

Fig. 3. Application of GMM simplifying algorithms (UTAC and BKMC) to clustering-based image segmentation. The BKMC algorithm provides the best results (visually and according to the KLD value).

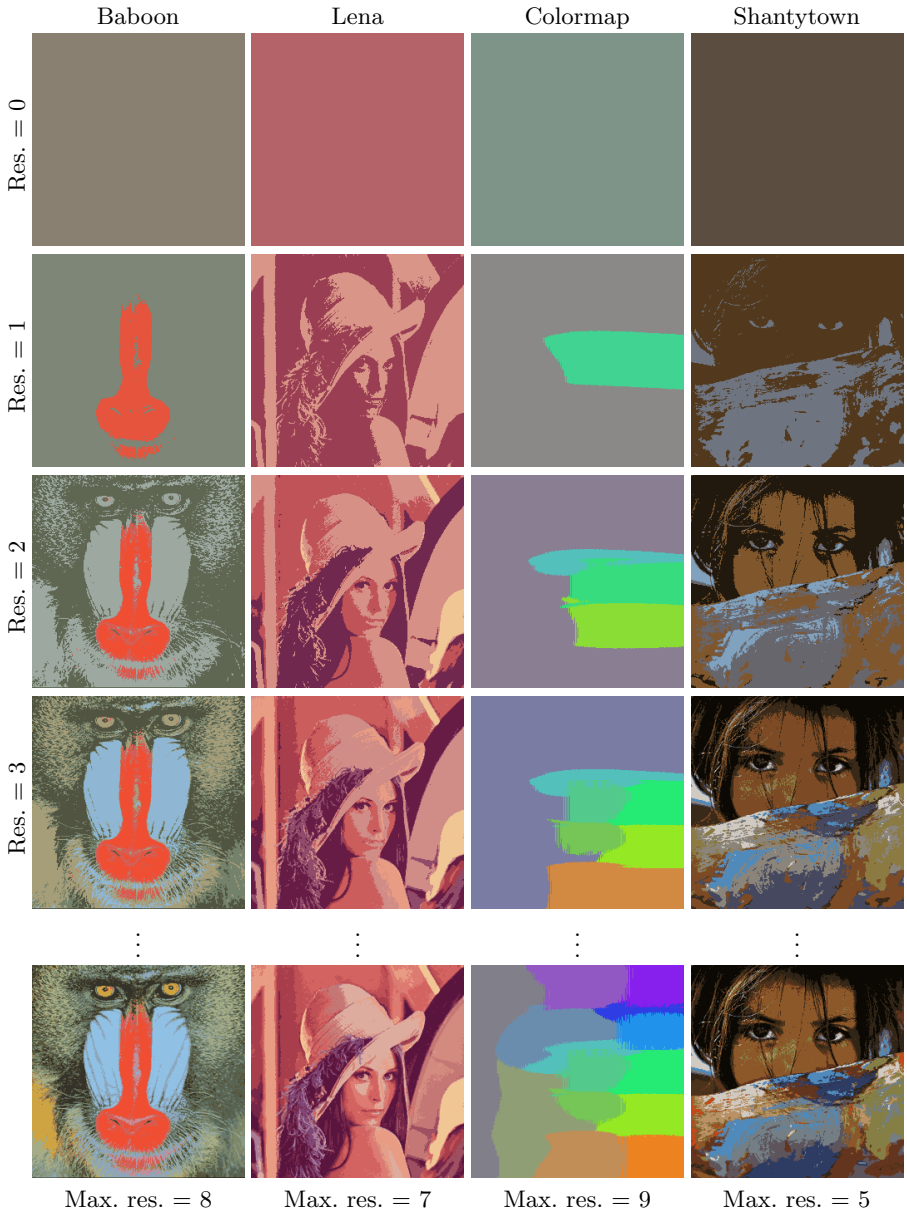

Fig. 4. Application of BGMC algorithm to clustering-based image segmentation. The figure shows (from top to bottom) the simplified GMM from resolution 0 to the maximal resolution. The GMM simplification quality increases with the resolution.

5 Concluding remarks

In this paper, we have proposed two algorithms for the simplification of Gaussian mixtures models. The first one, named BKMC, is based on the k-means algorithm. Experiments corroborate that BKMC yields better results in shorter computational time in comparison to the state-of-the-art. The second proposed algorithm, named BGMC, is based on the G-means algorithm. BGMC allows to automatically learn the *optimal* number of Gaussians in the simplified model and provides a progressive representation of the initial GMM. Note that although we have presented our algorithms to simplify GMM, our framework is generic and applies to any mixture model of an exponential family. The Java library implementing these algorithms is available at www.lix.polytechnique.fr/∼nielsen/MEF.

References

1. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum-likelihood from incomplete data via the EM algorithm. Journal of Royal Statistical Society B **39** (1977) 1–38
2. Parzen, E.: On the estimation of a probability density function and mode. Annals of Mathematical Statistics **33** (1962) 1065–1076
3. Zhang, K., Kwok, J.T.: Simplifying mixture models through function approximation. In: Neural Information Processing Systems. (2006)
4. Goldberger, J., Greenspan, H., Dreyfuss, J.: Simplifying mixture models using the unscented transform. IEEE Transactions Pattern Analysis Machine Intelligence **30** (2008) 1496–1502
5. Goldberger, J., Gordon, S., Greenspan, H.: An efficient image similarity measure based on approximations of kl-divergence between two gaussian mixtures. In: IEEE International Conference on Computer Vision. (2003)
6. Julier, S.J., Uhlmann, J.K.: Unscented filtering and nonlinear estimation. Proceedings of the IEEE **92** (2004) 401–422
7. Davis, J.V., Dhillon, I.: Differential entropic clustering of multivariate gaussians. In: Neural Information Processing Systems. (2006)
8. Goldberger, J., Roweis, S.: Hierarchical clustering of a mixture model. In: Neural Information Processing Systems. (2004)
9. Novoviov, J., Malk, A.: Application of multinomial mixture model to text classification. In: Pattern Recognition and Image Analysis. (2003)
10. Hamerly, G., Elkan, C.: Learning the k in k-means. In: Neural Information Processing Systems. (2003)
11. Banerjee, A., Merugu, S., Dhillon, I., Ghosh, J.: Clustering with bregman divergences. Journal of Machine Learning Research **6** (2005) 234–245
12. Nielsen, F., Boissonnat, J.D., Nock, R.: On Bregman Voronoi diagrams. In: SIAM Symposium on Discrete Algorithms. (2007)
13. Anderson, T.W., Darling, D.A.: Asymptotic theory of certain "goodness of fit" criteria based on stochastic processes. In: Annals of Mathematical Statistics. (1952)
14. Hershey, J.R., Olsen, P.A.: Approximating the Kullback Leibler divergence between gaussian mixture models. In: IEEE International Conference on Acoustics, Speech, and Signal Processing. (2007)