HIERARCHICAL GAUSSIAN MIXTURE MODEL

Vincent Garcia, Frank Nielsen

École Polytechnique / Sony CSL Inc.
Laboratoire d'informatique LIX
91128 Palaiseau Cedex, France

Richard Nock

Université des Antilles-Guyane, CEREGMIA
Campus de Schoelcher, BP 7209
97275 Schoelcher, Martinique, France

ABSTRACT

Gaussian mixture models (GMMs) are a convenient and essential tool for the estimation of probability density functions. Although GMMs are used in many research domains from image processing to machine learning, this statistical mixture modeling is usually complex and further needs to be simplified. In this paper, we present a GMM simplification method based on a hierarchical clustering algorithm. Our method allows one to first to quickly compute a compact version of the initial GMM, and second to automatically learn the optimal number of components of the simplified GMM. Using the framework of Bregman divergences, this simplification algorithm, although presented here for GMMs, is suitable for any mixture of exponential families.

***Index Terms*—** Gaussian mixture model, mixture model simplification, hierarchical clustering, Bregman divergence, exponential family

1. INTRODUCTION

A mixture model is a powerful framework commonly used to estimate the probability density function of a random variable. For instance, the Gaussian mixture model (GMM for short) has been widely used in many different application domains including statistics, image and signal processing, physics, biology, finance, *etc.* Let us consider a mixture model f of size n. The probability density function f evaluated at $x \in \mathbb{R}^d$ is given by

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

where $\alpha_i \in [0,1]$ denotes the weight of the i^{th} mixture component f_i such that $\sum_{i=1}^{n} \alpha_i = 1$. If f is a GMM, 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 variance-covariance symmetric positive-definite matrix $\Sigma_i \succ 0$. The model parameters are usually estimated from independent and identically-distributed observations using the Expectation-Maximization (EM) algorithm.
Mixture models are usually used to estimate statistical measures such as Shannon entropy. However, the computation of such measures can be prohibitive in terms of computation time for large mixtures, for instance for mixtures arising from a kernel-based Parzen density estimator. The computational time can strongly be 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 [1]. Thus, the most appropriated solution is to simplify the initial mixture model f.
Given a mixture model f composed of n components (see Eq. (1)), the problem of mixture model simplification consists in computing a simpler mixture model g

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

composed of m components ($1 \leq m < n$) such as g is the *best* approximation of f with respect to a similarity measure.
The parameter m impacts on the computation time and on the approximation quality of f as well. In most cases, a small value of m implies a fast but rough approximation of f, and conversely. Most of state-of-the-art mixture model simplification methods [1, 2, 3] consider the value m as a user-defined parameter. However, a given m cannot be adapted for every mixture models. This parameter should be learnt from f in order to achieve a given quality. It appears indeed more appropriate to constrain the mixture quality instead of the mixture size.
In this paper, we propose a GMM simplification algorithm based on the hierarchical clustering algorithm. This algorithm provides a hierarchical representation of the initial GMM f. From this representation, the algorithm automatically learns the *optimal* value m of components for an expected and user-defined mixture quality. Although described for GMMs, our method is suitable for the wide class of mixture of exponential families. Experiments on image processing are reported.

2. HIERARCHICAL GMMS

2.1. Hierarchical clustering

In the context of cluster analysis, *hierarchical clustering* is a group of methods consisting in building a hierarchical clustering of a set of objects (points, symbols, *etc.*). Hierarchical clustering is generally subdivided into two categories: *agglomerative* and *divisive* methods. The agglomerative methods usually start with elementary subsets and successively merge subsets until having a single set containing all objects. The divisive methods start with the initial set and recursively split the set and subsets until having elementary subsets. In this paper, we will only consider agglomerative methods.
Let P be a set of objects and let $P_1, P_2, \cdots, P_n$ be a partition of P ($\cup_i P_i = P$ and $P_i \cap P_j = \emptyset$ for all $i \neq j$). Let us consider a distance $D(.,.)$ (potentially asymmetric) between two subsets. Note

that we will present afterward the most common distances. The first step of the hierarchical clustering algorithm is to determine the two closest subsets P_i and P_j relatively to $D(.,.)$ among the $n(n-1)$ possible combinations. The second step is to merge the subsets P_i and P_j into a single subset. The algorithm usually starts with subsets containing one object (if $|P| = n$ then the initial partition contains n item subsets) and successively merges pairs of subsets until having one set equal to P. The initial subsets and the way these subsets were merged are stored into a hierarchical structure called a *dendrogram*.
The distance $D(.,.)$ between subsets, also know as linkage criterion, determines the order of the subset merging. Let A and B be two sets of objects (*e.g.* points) and let $d(.,.)$ be a distance between two objects (*e.g.* the Euclidean distance). The three most used linkage criteria are the minimum distance (Eq. (4)), the maximum distance (Eq. (5)), and the average distance (Eq. (6)):

$$D_{\min}(A,B) = \min\{d(a,b) \;\big|\; a \in A,\ b \in B\} \tag{4}$$

$$D_{\max}(A,B) = \max\{d(a,b) \;\big|\; a \in A,\ b \in B\} \tag{5}$$

$$D_{\text{av}}(A,B) = \frac{1}{|A||B|} \sum_{a \in A} \sum_{b \in B} d(a,b) \tag{6}$$

2.2. 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

$$D_{\mathrm{KL}}(f_i||f_j) = \int f_i(x) \log \frac{f_i(x)}{f_j(x)} \mathrm{d}x. \tag{7}$$

If, for instance, f_i and f_j are two multivariate Gaussian distributions parametrized by their mean μ_i and μ_j and by their variance-covariance matrix Σ_i and Σ_j, the fastidious integral computation of Eq. (7) leads to a closed form expression of the KLD:

$$\begin{aligned} D_{\mathrm{KL}}(f_i||f_j) =& \frac{1}{2} \log \frac{|\Sigma_j|}{|\Sigma_i|} + \frac{1}{2} \mathrm{tr}\left(\Sigma_j^{-1}\Sigma_i\right) \\ &+ \frac{1}{2}(\mu_j - \mu_i)^\top \Sigma_j^{-1}(\mu_j - \mu_i) - \frac{d}{2} \end{aligned} \tag{8}$$

where $|\Sigma|$ and $\mathrm{tr}(\Sigma)$ are respectively the determinant and the trace operator. We can avoid the integral computation using the canonical form of exponential families [4]

$$f_F(x;\boldsymbol{\Theta}) = \exp\left\{\langle \boldsymbol{\Theta}, \mathrm{t}(x)\rangle - F(\boldsymbol{\Theta}) + \mathrm{k}(x)\right\} \tag{9}$$

where $\boldsymbol{\Theta}$ are the *natural parameters* associated with the *sufficient statistic* $\mathrm{t}(x)$. The *log normalizer* $F(\boldsymbol{\Theta})$ is a strictly convex and differentiable function that characterizes uniquely the exponential family, and the function $\mathrm{k}(x)$ is the *carrier measure*. The classical distributions Gaussian, Laplacian, Poisson, binomial, Bernoulli, multinomial, Rayleigh, Gamma, Beta, and Dirichlet are all exponential families. For instance, let us consider the case of a multivariate Gaussian distribution:

$$f(x;\mu,\Sigma) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(-\frac{(x-\mu)^T\Sigma^{-1}(x-\mu)}{2}\right) \tag{10}$$

This distribution is an exponential family which can be written in a canonical form by defining:

$$\boldsymbol{\Theta} = (\theta, \Theta) = \left(\Sigma^{-1}\mu, \frac{1}{2}\Sigma^{-1}\right) \tag{11}$$

$$F(\boldsymbol{\Theta}) = \frac{1}{4}\mathrm{tr}(\Theta^{-1}\theta\theta^\top) - \frac{1}{2}\log|\Theta| + \frac{d}{2}\log\pi \tag{12}$$

$$\mathrm{t}(x) = (x, -xx^\top) \tag{13}$$

$$\mathrm{k}(x) = 0 \tag{14}$$

where $\boldsymbol{\Theta}$ is a mixed-type vector/matrix parameters. We have gathered in [5] the different formula allowing one to write the most classical distributions as exponential families (canonical form).

The relative entropy between two members of the same exponential family is equal to the Bregman divergence on the swapped natural parameters and defined for the log normalizer F:

$$D_{\mathrm{KL}}(f_i||f_j) = \mathrm{D}_F(\boldsymbol{\Theta}_j||\boldsymbol{\Theta}_i) \tag{15}$$

$$= F(\boldsymbol{\Theta}_j) - F(\boldsymbol{\Theta}_i) - \langle \boldsymbol{\Theta}_j - \boldsymbol{\Theta}_i, \nabla F(\boldsymbol{\Theta}_i)\rangle \tag{16}$$

In Eq. (16), $\langle\cdot,\cdot\rangle$ denotes the inner product and ∇F is the gradient operator. In the case of mixed-type vector/matrix parameters (*e.g.* Gaussian distributions), the inner product $\langle \boldsymbol{\Theta}_p, \boldsymbol{\Theta}_q\rangle$ is a composite inner product obtained as the sum of two inner products of vectors and matrices $\langle \boldsymbol{\Theta}_p, \boldsymbol{\Theta}_q\rangle = \langle \Theta_p, \Theta_q\rangle + \langle \theta_p, \theta_q\rangle$, where the inner product of two matrices is defined by $\langle \Theta_p, \Theta_q\rangle = \mathrm{tr}(\Theta_p\Theta_q^\top)$.
Using this formalism, we can define a mixture model simplification algorithm suitable for any mixture of exponential families. However, for the sake of clarity, we present below our method in the case of GMMs.

2.3. Hierarchical mixture models

Let f be a GMM of n components. By considering f as a set of n weighted Gaussians $(\alpha_i, \boldsymbol{\Theta}_i)$, the adaptation of hierarchical clustering algorithm towards mixtures of exponential families defines an elegant method to simplify f. First, we need to define the distance, denoted $d(.,.)$, between two weighted Gaussians. Since all the distributions belong to the same mixture, a symmetric distance seems appropriate:

$$d((\alpha_i, \boldsymbol{\Theta}_i), (\alpha_j, \boldsymbol{\Theta}_j)) = \alpha_i\alpha_j SD_F(\boldsymbol{\Theta}_i, \boldsymbol{\Theta}_j) \tag{17}$$

where $SD_F(.,.)$ is the symmetric Bregman divergence [6]

$$SD_F(\boldsymbol{\Theta}_i, \boldsymbol{\Theta}_j) = \frac{D_F(\boldsymbol{\Theta}_i||\boldsymbol{\Theta}_j) + D_F(\boldsymbol{\Theta}_j||\boldsymbol{\Theta}_i)}{2}. \tag{18}$$

However, a sided distance can be used as well:

$$d((\alpha_i, \boldsymbol{\Theta}_i), (\alpha_j, \boldsymbol{\Theta}_j)) = \alpha_i\alpha_j D_F(\boldsymbol{\Theta}_i||\boldsymbol{\Theta}_j) \tag{19}$$

Since all the possible pairs are considered, the chosen sided Bregman divergence (right-sided or left-sided) does not impact on the hierarchical construction.
Used with a linkage criterion (see section 2.1, Eq. (4)-(6)), the distance $d(.,.)$ allows us to identify the two *closest* subsets in order to merge them into a single subset. Due to the subset merging, the number of subsets (starting at n) decreases by 1 after each iteration of the algorithm until having 1 set containing all the weighted Gaussians.

The proposed algorithm creates a hierarchical structure, called *hierarchical mixture model* (similar to level of details in computer graphics), containing the weighted distributions and the information relative to the subset merging. This structure allows to quickly compute a simpler mixture g with an arbitrary number of components. Thus, we define the mixture of resolution r as the mixture g of r components

$$g = \sum_{j=1}^{r} \beta_j g_j \tag{20}$$

builds from the r subsets $\mathcal{P}_1, \cdots, \mathcal{P}_r$ remaining after $n-r$ iterations of the Bregman hierarchical clustering algorithm, subsets extracted from the hierarchical mixture model. The parameters of the j^{th} component are computed from the weighted distributions belonging to the subset $\mathcal{P}_j$. The distribution g_j is the Bregman centroid (right-sided, left-sided, or symmetric centroid, *c.f.* [3, 6]) of the subset $\mathcal{P}_j$. The weight β_j is computed as

$$\beta_j = \sum_i \alpha_i \quad \text{s.t.} \quad (\alpha_i, \boldsymbol{\Theta}_i) \in \mathcal{P}_j. \tag{21}$$

The type of Bregman centroid used must correspond to the sided / symmetric Bregman divergence used to build the hierarchical mixture model. Indeed, the extracted mixture g depends on the chosen sided Bregman divergence since the right-sided, the left-sided, and the symmetric Bregman centroids are different.

2.4. Automatic learning of the *optimal* GMM size

Let f be a GMM of n components that we want to simplify into a GMM g under the two following constraints: first g has to be as compact as possible; second g must reach a minimum simplification quality $D_{\mathrm{KL}}(f, g) \leq t$ defined by the user. The right-sided Kullback-Leibler divergence $D_{\mathrm{KL}}(f, g)$ between two GMMs is estimated by a classical Monte-Carlo method since it does not admit a closed-form expression. The GMM simplification method introduced in section 2.3 allows to quickly compute, from the hierarchical mixture model, a simplified version of f with an arbitrary number of components. We reasonably assume that the simplification quality increases (KLD decreases) with the resolution (see Fig. 4). Therefore, a standard dichotomy search on the resolution allows to quickly find the *optimal* number of components in the simplified mixture.

3. EXPERIMENTS

3.1. Mixture simplification

In this section, we compare the influence of the Bregman divergence and of the linkage criterion on the GMM simplification quality. The simplification quality is given by the right-sided KLD estimated by a Monte-Carlo method (sample of 1000 points). The initial GMM f with 32 Gaussian components was computed from the image Baboon (see Fig. 3): first the RGB pixels (dimension 3) were gathered into 32 classes C_i using a standard k-means algorithm; second the parameters of f_i were computed from the points of C_i using a standard Expectation-Maximization algorithm. The weight α_i was given by the proportion of points in the class C_i.

The Fig. 1 and 2 respectively show the evolution of the simplification quality as a function of the resolution for the different Bregman divergences and for the different linkage criteria. The simplification quality increased (KLD decreased) with the resolution. The Fig. 1 shows that the left-sided Bregman hierarchical clustering provided the best simplification quality. Indeed, estimating the left-sided Bregman divergence on natural parameters amounts to estimate the right-sided Kullback-Leibler divergence on distributions. Since the simplification quality measure is the right-sided Kullback-Leibler divergence, obtaining the best simplification with the left-sided Bregman hierarchical clustering was then the expected behavior. The Fig. 2 shows that the maximum and the average distances provided two simplified GMMs similar in terms of quality while the minimum distance provided a lower-quality mixture. In the paper remainder, we will use the left-sided Bregman divergence and the maximum-distance as they provided the best results over experiments. Once the hierarchical mixture model has been built, the mixture simplification was performed in less than one millisecond on a standard laptop.

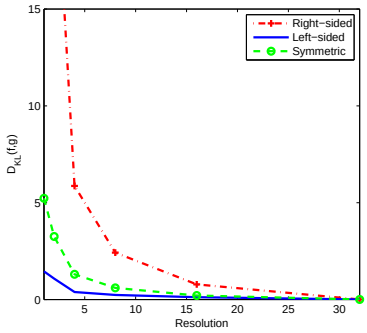

Fig. 1. Evolution of the simplification quality as a function of the resolution and for the right-sided, left-sided, and symmetric Bregman hierarchical clustering algorithms. The linkage criterion used was the maximum-distance.

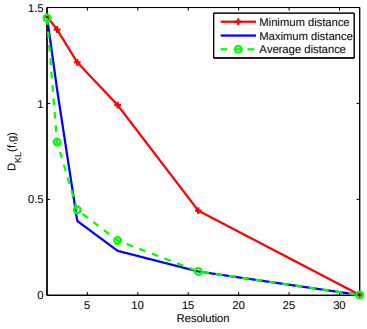

Fig. 2. Evolution of the simplification quality as a function of the resolution and for the minimum, the maximum, and the average distances (linkage criteria). The Bregman hierarchical clustering used was the left-sided one.

3.2. Image segmentation and optimal mixture model

The Fig. 3 illustrates the influence of the resolution on the simplified GMM g in the context of clustering-based image segmentation. Given a color image I, a pixel x can be considered as a point in $\mathbb{R}^3$ (RGB channels). The image segmentation is performed by assigning each image pixel x to the class C_i having the highest density value: $g_i(x) > g_j(x)\ \forall j \in [1, m] \setminus \{i\}$. Then, the image segmentation is illustrated by replacing the color value of the pixel x by the mean μ_i of the Gaussian g_i. The images used for the experiments were Baboon, Lena, Colormap, and Shantytown (256×256 pixels). Fig. 4 shows the evolution of the simplification quality as a function of the

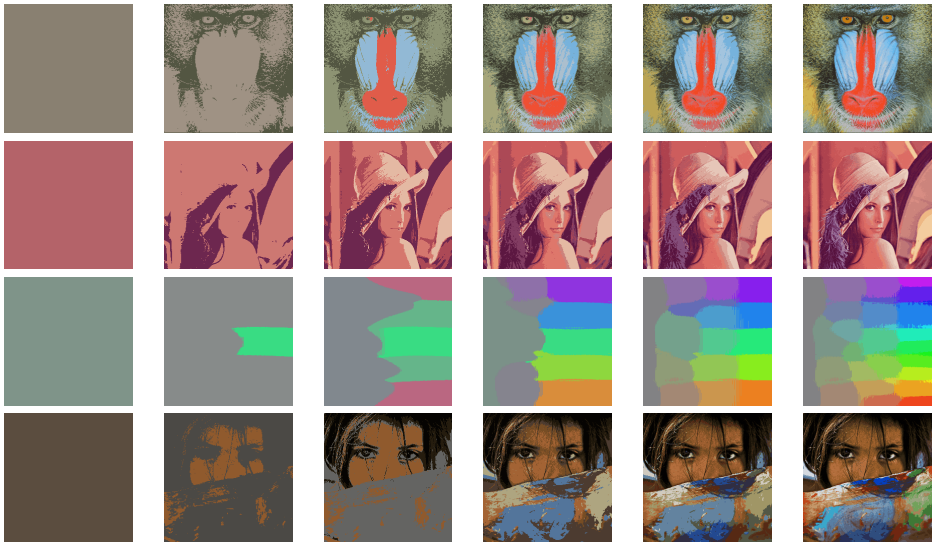

Fig. 3. Impact of the resolution r on the mixture simplification for a clustering-based image segmentation application.

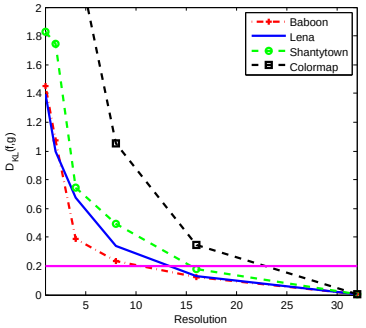

Fig. 4. Evolution of the simplification quality as a function of the resolution for images Baboon, Lena, Colormap, and Shantytown.

resolution for the same set of images. The simplification quality (visual quality for Fig. 3 and objective quality for Fig. 4) increased with the resolution. The number of class is by definition equal to the resolution of the simplified mixture. A GMM of resolution 4 is precise enough to recognize the initial image from the image segmentation (Baboon, Lena, and Shantytown). In the case of Colormap, the image does not have any *structure* usually present in natural images. Due to this lack of structure, the GMM simplification is less efficient than with natural images.

In Section 2.4, we introduced a method to learn the optimal number of components in the simplified mixture given a minimum mixture quality threshold t. We set the minimum quality $t = 0.2$ (horizontal red dashed line in Fig. 4). The automatic learning process found out that the optimal GMM size satisfying this quality constraint contains 10 components for the image Baboon, 14 components for Lena, 16 for Shantytown, and 23 for Colormap.

4. CONCLUSION

This paper described a fast Gaussian mixture model (GMM) simplification method based on hierarchical clustering. This method allows one to quickly compute a compact version of a source GMM of an arbitrary size. Moreover, our method is able to automatically learn the optimal number of components for the simplified GMM. Using the framework of Bregman divergences, this method extends to any mixture of exponential families. An open-source Java library, named jMEF (Java Mixtures of Exponential Families), implementing this algorithm has been designed and is available at `www.lix.polytechnique.fr/~nielsen/MEF`.

5. ACKNOWLEDGMENTS

We gratefully acknowledge financial support from DIGITEO GAS 2008-16D and ANR GAIA 07-BLAN-0328-01.

6. REFERENCES

[1] J. Goldberger, H. Greenspan, and J. Dreyfuss, "Simplifying mixture models using the unscented transform," *IEEE Transactions Pattern Analysis Machine Intelligence*, vol. 30, pp. 1496–1502, 2008.

[2] K. Zhang and J. T. Kwok, "Simplifying mixture models through function approximation," in *Neural Information Processing Systems*, 2006.

[3] F. Nielsen, V. Garcia, and R. Nock, "Simplifying Gaussian mixture models via entropic quantization," in *17th European Conference on Signal Processing (EUSIPCO)*, 2009.

[4] A. Banerjee, S. Merugu, I. Dhillon, and J. Ghosh, "Clustering with Bregman divergences," *Journal of Machine Learning Research*, vol. 6, pp. 234–245, 2005.

[5] F. Nielsen and V. Garcia, "Statistical exponential families: A digest with flash cards," arXiv 0911.4863, December 2009, http://arxiv.org/abs/0911.4863.

[6] F. Nielsen and R. Nock, "Sided and symmetrized Bregman centroids," *IEEE Transactions on Information Theory*, vol. 55, pp. 2048–2059, 2009.