Signal Processing ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Simplification and hierarchical representations of mixtures of exponential families

V. Garcia [a,*], F. Nielsen [a,b]

[a] École polytechnique, Laboratoire d'informatique, 91128 Palaiseau Cedex, France
[b] Sony Computer Science Laboratories Inc., 3-14-13 Higashi Gotanda, Shinagawa-Ku, 141-0022 Tokyo, Japan

ARTICLE INFO

Article history:
Received 1 December 2009
Received in revised form
26 March 2010
Accepted 18 May 2010

Keywords:
Mixtures models
Kullback–Leibler divergence
Bregman divergence
Exponential family
Mixtures of exponential families
Clustering algorithms

ABSTRACT

A mixture model in statistics is a powerful framework commonly used to estimate the probability measure function of a random variable. Most algorithms handling mixture models were originally specifically designed for processing mixtures of Gaussians. However, other distributions such as Poisson, multinomial, Gamma/Beta have gained interest in signal processing in the past decades. These common distributions are unified in the framework of exponential families in statistics. In this paper, we present three generic clustering algorithms working on arbitrary mixtures of exponential families: the Bregman soft clustering, the Bregman hard clustering, and the Bregman hierarchical clustering. These algorithms allow one to estimate a mixture model from observations, to simplify such a mixture model, and to automatically learn the "optimal" number of components in a simplified mixture model according to a resolution parameter. In addition, we present jMEF, an open source Java™ library allowing users to create, process and manage mixtures of exponential families. In particular, jMEF includes the three aforementioned Bregman clustering algorithms.

1. Introduction

A mixture model is a powerful framework commonly used to estimate the probability measure function of a random variable. For instance, the mixtures of Gaussians have 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 n components. The probability density function f evaluated at $x \in \mathbb{R}^d$ is given by

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

where f_i is a statistical distribution and $\alpha_i \in [0,1]$ are the normalized weights associated to the f_i's such that $\sum_{i=1}^{n} \alpha_i = 1$.

The spread of mixture models implies to use algorithms devoted to such mixtures (e.g. mixture simplification algorithms). Most of these seminal algorithms were specifically designed for mixtures of Gaussians since they represent in practice the most commonly met mixture models. However, the use of other distribution models such as Poisson, multinomial, Gamma and Beta have gained interest in signal processing in the past decades (cf. [1,2] for non-Gaussianity of data sets). Usually, adapting the Gaussian toolbox of standard algorithms to these newly met distribution models is a laborious and complex task. The exponential families are a wide class of distributions including the most common distribution models: Gaussian, Poisson, Laplacian, binomial, multinomial, Bernoulli, Rayleigh, Beta, Gamma, etc. (cf. [3]). Based on these families, we can define generic algorithms adapted not only to mixtures of Gaussians, but adapted to the most common mixture models.

Since the amount of algorithms processing mixture models is quite large, we focus in this paper on the clustering algorithms: soft/hard membership and

* Corresponding author.
E-mail addresses: garciav@lix.polytechnique.fr (V. Garcia), nielsen@lix.polytechnique.fr (F. Nielsen).

0165-1684/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.sigpro.2010.05.024

hierarchical clusterings. Our first contribution is to generalize those classical clustering algorithms to arbitrary mixtures of exponential families using the framework of Bregman divergences. Therefore, these algorithms are, respectively, called Bregman soft clustering, Bregman hard clustering, and Bregman hierarchical clustering. The Bregman soft clustering algorithm is the adaptation of the expectation–maximization (EM) algorithm allowing one to estimate the parameters of a mixture of exponential families from a set of observation points (random samples). The Bregman hard clustering algorithm is the adaptation of the celebrated k-means algorithm toward mixtures of exponential families and Bregman divergence. This hard clustering algorithm appears to be a very efficient mixture simplification algorithm. The Bregman hierarchical clustering algorithm creates a hierarchical mixture model from an initial mixture of exponential families. This hierarchical mixture model allows first to quickly simplify the initial mixture, and second to automatically learn the optimal number of components in the simplified mixture, according to some resolution parameter.

Our second contribution consists in briefly introducing the open source `jMEF` library: a Java™ library for Mixtures of Exponential Families. This cross-platform library allows users to create, process and manage mixtures of exponential families. In particular, `jMEF` includes the implementations of the three described Bregman clustering algorithms. The open source library is freely available at http://www.lix.polytechnique.fr/~nielsen/MEF.

The paper is organized as follows: Section 2 introduces the theoretical background (Bregman divergence and Bregman centroids) required in Section 3. Section 3 presents the Bregman soft, hard and hierarchical clusterings with a brief introduction to the `jMEF` library. Section 4 proposes some experiments with applications to image processing, and compare our approaches to the state of the art. Finally, Section 5 concludes the paper.

2. Exponential family and Bregman divergence

In this section, we introduce the theoretical background required for the clustering part of Section 3. First, we present in Section 2.1 the link existing between the Kullback–Leibler (also known as the relative entropy) and Bregman divergences. Second, we briefly introduce (Section 2.2) the dual canonical parameterizations of distributions belonging to an exponential family. Finally, we present in Section 2.3 the concept of Bregman centroids.

2.1. Relative entropy, exponential family, and Bregman divergence

The fundamental measure between statistical distributions is the relative entropy, also called the Kullback–Leibler divergence (denoted by KLD, cf. [4,5]). Given two distributions f_i and f_j, the KLD is an oriented distance (asymmetric) and is defined as

$$D_{\mathrm{KL}}(f_i\|f_j)=\int_x f_i(x)\log\frac{f_i(x)}{f_j(x)}\,\mathrm{d}x \tag{2}$$

The relative entropy is not a metric but satisfies positive-definiteness $D_{\mathrm{KL}}(f\|g)\geq 0$ (called Gibb's inequality in the literature). If, for instance, f_i and f_j are two multivariate Gaussian distributions parametrized by their respective mean μ_i and μ_j and by their corresponding variance–covariance matrix Σ_i and Σ_j, the fastidious integral computation of Eq. (2) leads to a closed form expression of the KLD:

$$D_{\mathrm{KL}}(f_i\|f_j)=\frac{1}{2}\log\frac{|\Sigma_j|}{|\Sigma_i|}+\frac{1}{2}\mathrm{tr}(\Sigma_j^{-1}\Sigma_i)+\frac{1}{2}(\mu_j-\mu_i)^{\top}\Sigma_j^{-1}(\mu_j-\mu_i)-\frac{d}{2} \tag{3}$$

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 (cf. [6])

$$f_F(x;\boldsymbol{\Theta})=\exp\{\langle\boldsymbol{\Theta},\mathrm{t}(x)\rangle-F(\boldsymbol{\Theta})+\mathrm{k}(x)\} \tag{4}$$

where $\boldsymbol{\Theta}$ are the *natural parameters* (see Section 2.2), and $\mathrm{t}(x)$ the *sufficient statistics*. 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:

$$f(x;\mu,\Sigma)=\frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}}\exp\left(-\frac{(x-\mu)^{\top}\Sigma^{-1}(x-\mu)}{2}\right) \tag{5}$$

This distribution is an exponential family which can be written in a canonical form (see Eq. (4)) by defining

$$\boldsymbol{\Theta}=(\theta,\Theta)=(\Sigma^{-1}\mu,\tfrac{1}{2}\Sigma^{-1}) \tag{6}$$

$$F(\boldsymbol{\Theta})=\frac{1}{4}\mathrm{tr}(\Theta^{-1}\theta\theta^{\top})-\frac{1}{2}\log|\Theta|+\frac{d}{2}\log\pi \tag{7}$$

$$\mathrm{t}(x)=(x,-xx^{\top}) \tag{8}$$

$$\mathrm{k}(x)=0 \tag{9}$$

where $\boldsymbol{\Theta}$ is a mixed-type vector/matrix parameter. The computation of these expressions can be laborious and prone to errors. Therefore, we have gathered in Nielsen and Garcia [3] the decomposition as exponential families (canonical form) of most classical distributions.

Interestingly, 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:

$$\mathrm{D}_{\mathrm{KL}}(f_i\|f_j)=\mathrm{D}_F(\boldsymbol{\Theta}_j\|\boldsymbol{\Theta}_i) \tag{10}$$

where

$$\mathrm{D}_F(\boldsymbol{\Theta}_j\|\boldsymbol{\Theta}_i)=F(\boldsymbol{\Theta}_j)-F(\boldsymbol{\Theta}_i)-\langle\boldsymbol{\Theta}_j-\boldsymbol{\Theta}_i,\nabla F(\boldsymbol{\Theta}_i)\rangle \tag{11}$$

In Eq. (11), $\langle\cdot,\cdot\rangle$ denotes the inner product and ∇ is the gradient operator. In the case of mixed-type vector/matrix parameters (e.g. Gaussian distributions), the inner

Please cite this article as: V. Garcia, F. Nielsen, Simplification and hierarchical representations of mixtures of exponential families, Signal Process. (2010), doi:10.1016/j.sigpro.2010.05.024

product $\langle \mathbf{\Theta}_p, \mathbf{\Theta}_q \rangle$ is a composite inner product obtained as the sum of two inner products of vectors and matrices

$$\langle \mathbf{\Theta}_p, \mathbf{\Theta}_q \rangle = \langle \Theta_p, \Theta_q \rangle + \langle \theta_p, \theta_q \rangle \quad (12)$$

where the inner product of vectors is the dot product $\langle \theta_p, \theta_q \rangle = \theta_p^\top \theta_q$, and the inner product of two matrices is defined by

$$\langle \Theta_p, \Theta_q \rangle = \mathrm{tr}(\Theta_p \Theta_q^\top) = \mathrm{tr}(\Theta_q \Theta_p^\top) \quad (13)$$

Thanks to this formalism, we can define a family of algorithms suitable to any mixture of exponential families.

2.2. Distribution parameters: dual canonical coordinate systems

We now present an essential notion of convex analysis: the Legendre–Fenchel transformation (informally called the slope transformation). Any convex function F admits a dual conjugate convex function F^*:

$$F^*(x') = \sup_x \{\langle x', x \rangle - F(x)\}$$

x' is called the dual variable. The supremum is reached at the *unique* point where the gradient of $G(x') = \langle x', x \rangle - F(x)$ vanishes or, equivalently, when $x' = \nabla F(x)$. x and x' are called dual parameterizations.

Thus a distribution f_F belonging to an exponential family can be equivalently parametrized by its source parameters $\mathbf{\Lambda}$, its natural parameters $\mathbf{\Theta}$, or by its dual expectation parameters $\mathbf{H}$ (see [3]). The conversion procedures from natural to expectation parameters (and conversely) are the following:

$$\mathbf{H} = \nabla F(\mathbf{\Theta}) \quad (14)$$

$$\mathbf{\Theta} = \nabla F^*(\mathbf{H}) \quad (15)$$

where F^* is the dual Legendre of the log normalizer F, also denoted by $G = F^*$ in the remainder.

For the sake of brevity, the conversion procedures from source to natural parameters (and conversely) and from source to expectation parameters (and conversely) for the most classical distributions (exponential families) are given in Nielsen and Garcia [3]. The algorithms proposed/presented in this paper consider either natural parameters or expectation parameters. The context should be clear enough to avoid any confusion.

2.3. Bregman centroids

Given a set $\mathcal{S}$ of n weighted distributions parametrized by their natural parameters

$$\mathcal{S} = \{\{\alpha_1, \mathbf{\Theta}_1\}, \{\alpha_2, \mathbf{\Theta}_2\}, \ldots, \{\alpha_n, \mathbf{\Theta}_n\}\} \quad (16)$$

the Bregman centroid (cf. [7]) is the point (parameter in the natural coordinate system) minimizing the average Bregman divergence with the distribution parameters of $\mathcal{S}$. Similarly to the Kullback–Leibler divergence, the Bregman divergence is an asymmetric[1] measure. As a consequence, we must consider three types of centroids: right-sided centroid $\overline{\mathbf{\Theta}}_R$, left-sided centroid $\overline{\mathbf{\Theta}}_L$, and symmetric centroids $\overline{\mathbf{\Theta}}_S$, respectively, satisfying the following equations:

$$\overline{\mathbf{\Theta}}_R = \operatorname*{argmin}_{\mathbf{\Theta}} \frac{1}{\sum_i \alpha_i} \sum_i \alpha_i \mathrm{D}_F(\mathbf{\Theta}_i \| \mathbf{\Theta}) \quad (17)$$

$$\overline{\mathbf{\Theta}}_L = \operatorname*{argmin}_{\mathbf{\Theta}} \frac{1}{\sum_i \alpha_i} \sum_i \alpha_i \mathrm{D}_F(\mathbf{\Theta} \| \mathbf{\Theta}_i) \quad (18)$$

$$\overline{\mathbf{\Theta}}_S = \operatorname*{argmin}_{\mathbf{\Theta}} \frac{1}{\sum_i \alpha_i} \sum_i \alpha_i \mathrm{SD}_F(\mathbf{\Theta}, \mathbf{\Theta}_i) \quad (19)$$

where SD_F is the symmetric Bregman divergence given by

$$\mathrm{SD}_F(\mathbf{\Theta}, \mathbf{\Theta}_i) = \frac{\mathrm{D}_F(\mathbf{\Theta}_i \| \mathbf{\Theta}) + \mathrm{D}_F(\mathbf{\Theta} \| \mathbf{\Theta}_i)}{2} \quad (20)$$

The minimization of Eqs. (17) and (18) provides closed-form expressions for both right-sided and left-sided centroids (see [7]):

$$\overline{\mathbf{\Theta}}_R = \frac{\sum_i \alpha_i \mathbf{\Theta}_i}{\sum_i \alpha_i} \quad (21)$$

$$\overline{\mathbf{\Theta}}_L = \nabla F^* \left(\frac{\sum_i \alpha_i \nabla F(\mathbf{\Theta}_i)}{\sum_i \alpha_i} \right) \quad (22)$$

where ∇F^* is the gradient of the dual Legendre of the log normalizer F. The symmetric centroid does not have a closed-form expression and has to be estimated from both the right- and the left-sided centroids. Indeed, the symmetric centroid $\overline{\mathbf{\Theta}}_S$ belongs to the geodesic link between $\overline{\mathbf{\Theta}}_R$ and $\overline{\mathbf{\Theta}}_L$ (cf. [7]). The following equation allows one to walk on this geodesic link

$$\mathbf{\Theta}_\lambda = \nabla F^*(\lambda \nabla F(\overline{\mathbf{\Theta}}_R) + (1-\lambda) \nabla F(\overline{\mathbf{\Theta}}_L)) \quad (23)$$

where $\lambda \in [0,1]$. For $\lambda = 0$ we have $\mathbf{\Theta}_\lambda = \overline{\mathbf{\Theta}}_L$, and for $\lambda = 1$ we have $\mathbf{\Theta}_\lambda = \overline{\mathbf{\Theta}}_R$. The symmetric centroid must verify the following constraint:

$$\mathrm{SD}_F(\overline{\mathbf{\Theta}}_S, \overline{\mathbf{\Theta}}_R) = \mathrm{SD}_F(\overline{\mathbf{\Theta}}_S, \overline{\mathbf{\Theta}}_L) \quad (24)$$

A standard bisection search on the parameter λ (cf. Eq. (23)) allows to estimate in a few steps the parameters of the symmetric centroid $\overline{\mathbf{\Theta}}_S$ for a given precision.

The concept of Bregman centroid is somehow *abstract* and need to be clarified. We give here a visual example in order to help the reader grasp a good understanding of those different types of Bregman centroids. We created a set $\mathcal{S}$ of four univariate Gaussians with similar variance $\sigma^2 = 6$. The means of these Gaussians were, respectively, set to $\mu_1 = 10$, $\mu_2 = 20$, $\mu_3 = 30$, and $\mu_4 = 40$. Then, we computed from $\mathcal{S}$ the centroids $\overline{\mathbf{\Theta}}_R$, $\overline{\mathbf{\Theta}}_L$, and $\overline{\mathbf{\Theta}}_S$. The parameters of these centroids are reported bellow:

$$\overline{\mathbf{\Theta}}_R = (\mu = 25, \sigma^2 = 131)$$

$$\overline{\mathbf{\Theta}}_L = (\mu = 25, \sigma^2 = 6)$$

$$\overline{\mathbf{\Theta}}_S = (\mu = 25, \sigma^2 = 28)$$

Fig. 1 illustrates the four Gaussians of $\mathcal{S}$ as well as the three computed centroids. First, the means of the three centroids were all equal to 25. This is due to the fact that the variance σ^2 was shared by the four Gaussians. Second, the variance of the left-sided centroid is equal to the

[1] Right-sided, left-sided, and symmetric Bregman divergences.

Please cite this article as: V. Garcia, F. Nielsen, Simplification and hierarchical representations of mixtures of exponential families, Signal Process. (2010), doi:10.1016/j.sigpro.2010.05.024

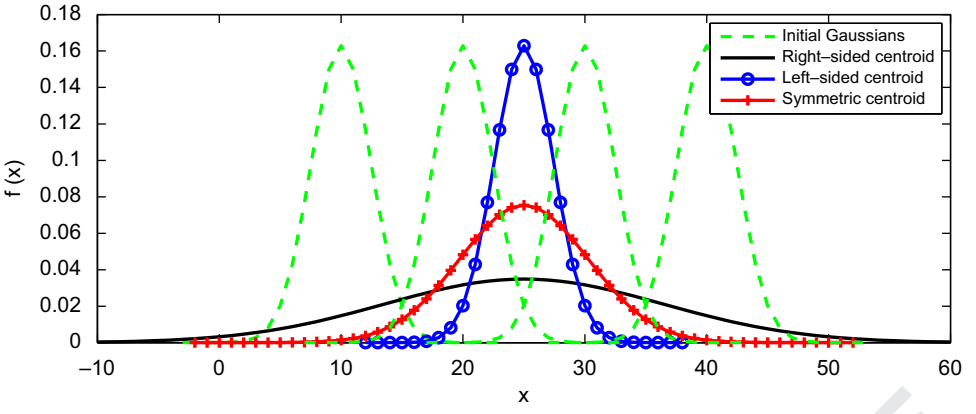

Fig. 1. Bregman centroids.

variance of the Gaussians of $\mathcal{S}$ (average of the variances). The left-sided centroid is indeed called "generalized centroid". On the contrary, the variance of the right-sided centroid is very large, this centroid trying to encompass the Gaussians of $\mathcal{S}$ (that is, to adhere more to the support of all Gaussians). The variance of the symmetric centroid is "between" the variances of the right-sided centroid and the left-sided centroid.

As a consequence, the choice of the centroid (i.e. Bregman divergence) used should depend on the application: the left-sided centroid provides an average distribution, the right-sided centroid provides an encompassing distribution, and symmetric centroid provides a trade-off between the right-sided and the left-sided centroids.

3. Bregman clustering

The literature of clustering algorithms for mixture of exponential families barely exists. As a consequence, this section presents the state of the art as well as the contributions of the paper. Some existing methods (soft and hard clustering algorithms, Sections 3.1 and 3.2) are presented and modified to fit our needs. The hierarchical clustering (Section 3.3) is an original contribution, and Section 3.4 presents the open source `jMEF` library implementing these three clustering algorithms.

3.1. Bregman soft clustering

Proposed by Dempster et al. in 1977 (cf. [8]), the expectation–maximization (EM) algorithm is an efficient and widely used method to estimate the model parameters from a set of observations (e.g. points) by maximizing the likelihood. EM is an iterative method which alternates the expectation step (E-step) and the maximization step (M-step). EM algorithm is also known as *soft clustering*: each observation has a probability to belong to each model.

The Bregman soft clustering, originally proposed by Banerjee et al. [6], is the adaptation of the EM algorithm allowing one to estimate the parameters of a mixture of exponential families from a point set $\mathcal{S}$ (observations). As in the classical EM algorithm, the Bregman soft clustering starts with an initialization step and then alternates the E-step and the M-step. The process stops if the increase in likelihood is smaller than a given threshold or if the number of iterations is greater than a user prescribed value.

Initialization step: The initialization step computes an initial mixture of exponential families from the set of observations. This initial mixture is indeed needed in the first call of the E-step. First, the points (observations) are gathered into n subsets $\mathcal{S}_1,\ldots,\mathcal{S}_n$ using a classical k-means algorithm (cf. [9] and Section 3.2). The weight α_j and the expectation parameters $\mathbf{H}_j$ of the jth mixture component are given by

$$\alpha_j = \frac{|\mathcal{S}_j|}{|\mathcal{S}|} \tag{25}$$

$$\mathbf{H}_j = \frac{1}{|\mathcal{S}_j|}\sum_{\{i|x_i\in\mathcal{S}_j\}} \mathrm{t}(x_i) \tag{26}$$

where $|\mathcal{S}_j|$ denotes the cardinality of the subset $\mathcal{S}_j$ and $\mathrm{t}(x)$ is the sufficient statistic. (This is because the maximum likelihood estimator for an exponential family of a set of i.i.d. observations is provably the center of mass in the expectation coordinate system. This point is called the observation point in information geometry. Note that to estimate the $\mathbf{H}$ expectation parameter, we only rely on the sufficient statistics aggregating compactly all information contained in the observation sample, hence its name.)

E-step: The E-step consists in computing the probability $p(i,j)$ that the observation x_i has been generated by the jth distribution

$$p(i,j) = \frac{\alpha_j \exp(-\mathrm{D}_G(\mathrm{t}(x_i),\mathbf{H}_j))\exp(\mathrm{k}(x_i))}{\sum_{l=1}^{n}\alpha_l \exp(-\mathrm{D}_G(\mathrm{t}(x_i),\mathbf{H}_l))\exp(\mathrm{k}(x_i))} \tag{27}$$

where $\mathrm{k}(x)$ is the carrier measure and $\mathrm{D}_G(\cdot,\cdot)$ is the divergence associated with $G{=}F^*$ (dual Legendre of the log normalizer F) given by

$$\mathrm{D}_G(p\|q) = G(p) - G(q) - \langle p-q, \nabla G(q)\rangle \tag{28}$$

However, the computation of $p(i,j)$ as proposed in Eq. (27) implies to compute $G(\mathrm{t}(x_i))$ which is equal to $-\frac{1}{2}\log 0$ in

Please cite this article as: V. Garcia, F. Nielsen, Simplification and hierarchical representations of mixtures of exponential families, Signal Process. (2010), doi:10.1016/j.sigpro.2010.05.024

the case Gaussian distributions. To solve[2] this problem, we simply expand Eq. (28) to Eq. (27). We then factorize and simplify $G(\mathrm{t}(x_i))$ in both numerator and denominator. Thus, $p(i,j)$ is given by

$$p(i,j)=\frac{\alpha_j \exp(G(\mathbf{H}_j)+\langle \mathrm{t}(x_i)-\mathbf{H}_j, \nabla G(\mathbf{H}_j)\rangle)}{\sum_{l=1}^{n}\alpha_l \exp(G(\mathbf{H}_l)+\langle \mathrm{t}(x_i)-\mathbf{H}_l, \nabla G(\mathbf{H}_l)\rangle)} \tag{29}$$

M-step: Then, based on $p(i,j)$, the M-step computes the weight and distribution parameters of the jth distribution as follows:

$$\alpha_j=\frac{1}{N}\sum_{i=1}^{N} p(i,j) \tag{30}$$

$$\mathbf{H}_j=\frac{\sum_{i=1}^{N} p(i,j)\mathrm{t}(x_i)}{\sum_{i=1}^{N} p(i,j)} \tag{31}$$

The difference here with the original work (cf. [6]) is first the modification of Eq. (27) into Eq. (29), and second the adaptation of the algorithm for mixtures of exponential families. Although small, the contribution of this section is fundamental (if not crucial) to estimate the parameters of *any* mixture of exponential families (including the familiar Gaussian distributions) from a given point set.

3.2. Bregman hard clustering

Let f be a mixture model of n components:

$$f(x)=\sum_{i=1}^{n}\alpha_i f_i(x) \tag{32}$$

A mixture simplification algorithm consists in computing, from the initial mixture f, a simpler mixture g of m components with $0\le m<n$ such that g is the *best* approximation of f. The simplification allows one to avoid the fastidious estimation (in terms of computation time) of g from the initial observations, to speed-up the computation of statistical measures, or to compare mixtures of different size [10]. In this section, we present a simplification algorithm adapted for mixtures of exponential families [11].

The *k-means* algorithm [9], also known as Lloyd's algorithm or *hard clustering* algorithm, is a method of cluster analysis which aims to partition a set of observations $\mathcal{S}$ (e.g. points) into k clusters (subsets) $\mathcal{S}_1,\ldots,\mathcal{S}_k$, each cluster being determined by its centroid. The k-means algorithm alternates an assignment step, which assign points to the cluster with the closest centroid, and an update step, which update centroids from the point repartition. The original Bregman hard clustering algorithm [6] is the adaptation of the k-means algorithm towards exponential families. Here, we improve this method and define a fast simplification method for mixtures of exponential families.

[2] We thank Banerjee (cf. [6]) for acknowledging the problem and email correspondences concerning this matter. Indeed, in Banerjee et al. [6], the authors consider a compact "equivalent" expression of exponential families: $\exp(\langle x,\theta\rangle - F(\theta))$.

A mixture of exponential families f can be considered as a set $\mathcal{S}$ of n weighted distributions parametrized by their natural parameters:

$$\mathcal{S}=\{\{\alpha_1,\boldsymbol{\Theta}_1\},\ldots,\{\alpha_n,\boldsymbol{\Theta}_n\}\}$$

The Bregman hard clustering consists in computing from $\mathcal{S}$ a set of m weighted distributions $C_j=\{\overline{\alpha}_j,\overline{\boldsymbol{\Theta}}_j\}$, for j in $[1,m]$, called centroid. The value m is usually user defined and depends on the application requirements. However, this value can be learnt from the initial mixture as presented in Section 3.3. Recall that the Bregman divergence is an asymmetric measure. As a consequence, three versions of the Bregman hard clustering are considered, respectively, based on the right-sided, left-sided, and symmetric Bregman divergences. We present below the Bregman hard clustering based on the right-sided Bregman divergence. Indeed, the adaptation to the left-sided and to the symmetric Bregman divergences is trivial. As in the classical k-means algorithm, the Bregman hard clustering starts with an initialization step and then alternates the assignment and the update steps (centroid relocation).

Initialization step: The first step of the algorithm initializes the m clusters $\mathcal{S}_j$ by initializing the m centroids $C_j=\{\overline{\alpha}_j,\overline{\boldsymbol{\Theta}}_j\}$ for j in $[1,m]$. The simplest method consists in randomly picking m weighted distributions among $\mathcal{S}$, the m centroids having to be different from each other (different distribution parameters). This makes sure that, after the assignment step, no cluster will be empty.

Recently, Arthur and Vassilvitskii (cf. [12]) proposed the k-means++ algorithm which optimizes the centroid initialization based on a random technique. The experimentations proposed by the authors show that their approach improves both the speed and the accuracy of k-means. The k-means++ algorithm can be easily adapted to mixture of exponential families and Bregman divergence. Indeed, the classical distance (usually square Euclidean distance) is simply replaced by the Bregman divergence.

Assignment step: Given a set of m centroids $C_j=\{\overline{\alpha}_j,\overline{\boldsymbol{\Theta}}_j\}$ for j in $[1,m]$, the assignment step assigns each weighted distribution $\{\alpha_i,\boldsymbol{\Theta}_i\}$ cluster with the closest centroid. For instance, the distribution $\{\alpha_i,\boldsymbol{\Theta}_i\}$ belongs to the cluster $\mathcal{S}_j$ if and only if

$$D_F(\boldsymbol{\Theta}_i\|\overline{\boldsymbol{\Theta}}_j)\le D_F(\boldsymbol{\Theta}_i\|\overline{\boldsymbol{\Theta}}_l),\quad \forall l\in[1,m]. \tag{33}$$

Update step: The parameters of the centroid C_j are updated from the distributions belonging to $\mathcal{S}_j$. Actually, C_j is the right-sided Bregman centroid of the cluster $\mathcal{S}_j$ as presented in Section 2.3. The weight $\overline{\alpha}_j$ is the sum of the weights belonging to $\mathcal{S}_j$:

$$\overline{\alpha}_j=\sum_i \alpha_i \quad \text{s.t. } \{\alpha_i,\boldsymbol{\Theta}_i\}\in\mathcal{S}_j \tag{34}$$

The algorithm alternates assignment and update steps until the assignment step no longer changes.

The extension of the Bregman hard clustering to left-sided Bregman divergence (resp. symmetric Bregman divergence) is obtained by replacing the right-sided Bregman divergence (initialization and assignment steps) and the right-sided centroid (update step), respectively, by the left-sided Bregman divergence (resp. symmetric

Please cite this article as: V. Garcia, F. Nielsen, Simplification and hierarchical representations of mixtures of exponential families, Signal Process. (2010), doi:10.1016/j.sigpro.2010.05.024

Bregman divergence) and by the left-sided centroid (resp. symmetric centroid).

As mentioned before, the output of the algorithm is a set of centroids (weighted distributions) C_j. This set can be considered as a mixture of exponential families g. It turns out that g is a simplified version of the initial mixture f. Thus, applied in the context of mixture simplification, the Bregman hard clustering appears to be a very efficient method both in terms of quality and computation time. Indeed, experiments (cf. Section 4.5) report that the Bregman hard clustering compare favorably to the state of the art UTAC algorithm (cf. [13]). One major drawback of the UTAC method is that it requires to compute the eigenvectors of the covariance matrices, a highly time-consuming operation.

3.3. Bregman hierarchical clustering

In this section, we propose a new method which (1) provide a hierarchical representation of an initial mixture of exponential families f, (2) allow to quickly simply f into a mixture of an arbitrary size m, and (3) learn the *optimal* value of m.

In the context of cluster analysis, *hierarchical clustering* is a set of methods consisting in building a hierarchical clustering of a set of objects (e.g. points). Hierarchical clustering is generally subdivided into two categories: agglomerative methods and divisive methods. In this paper, we will only consider the agglomerative methods. Let $\mathcal{S}$ be a set of objects and let $\{\mathcal{S}_1,\mathcal{S}_2,\ldots,\mathcal{S}_n\}$ be a partition of $\mathcal{S}$ such that $\sum_{i=1}^{n}|\mathcal{S}_i|=|\mathcal{S}|$ where $|\mathcal{S}_i|$ denotes the cardinality of $\mathcal{S}_i$. The first step of the algorithm determines the two *closest* subsets $\mathcal{S}_i$ and $\mathcal{S}_j$, relatively to a distance D(.,.), among the $n(n-1)$ possible combinations. The second step merges the subsets $\mathcal{S}_i$ and $\mathcal{S}_j$ into a single subset. The algorithm usually starts with subsets containing one object (if $|\mathcal{S}|=n$, then the initial partition contains n subsets) and alternates the first and the second steps until having a single set equal to $\mathcal{S}$. The initial subsets and the way these subsets were merged are stored into a hierarchical structure called *dendrogram*. The distance D(.,.) between subsets, also known 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). Table 1 presents the three most classical linkage criteria: the minimum, the maximum, and the average distances.

The Bregman hierarchical clustering algorithm is the instantiation of the hierarchical clustering to the case of mixtures of exponential families and Bregman divergence.

Table 1
Classical linkage criteria: minimum, maximum, and average distances.

Criterion	Formula
Minimum distance	$\mathrm{D}_{\min}(A,B)=\min\{d(a,b)\|a\in A,b\in B\}$
Maximum distance	$\mathrm{D}_{\max}(A,B)=\max\{d(a,b)\|a\in A,b\in B\}$
Average distance	$\mathrm{D}_{\mathrm{av}}(A,B)=\frac{1}{\|A\|\|B\|}\sum_{a\in A}\sum_{b\in B}d(a,b)$

As for the Bregman hard clustering algorithm, we consider a mixture of exponential families f as a set of n weighted distributions:

$$\mathcal{S}=\{\{\alpha_1,\boldsymbol{\Theta}_1\},\ldots,\{\alpha_n,\boldsymbol{\Theta}_n\}\}$$

Similarly to the classical algorithm, the initial clusters contain each one a weighted distribution $\{\alpha_i,\boldsymbol{\Theta}_i\}$. In the step one, the distance $d(.,.)$ is then replaced by the Bregman divergence to compute the linkage criterion. Then, step two is a classical merging procedure. Keeping in mind that the Bregman divergence is an asymmetric measure, we can define three different Bregman hierarchical clusterings. The use of a symmetric measure (symmetric Bregman divergence) seems to be more appropriate since all considered weighted distributions belong to the same mixture of exponential families. However, a non-symmetric measure can be used as well. This will be discussed later on.

The proposed algorithm creates a hierarchical structure (dendrogram), called *hierarchical mixture model*, containing the weighted distributions and the information relative to the subset merging. Interestingly, this structure allows one to quickly compute a simpler mixture g with an arbitrary number of components. We define the mixture of resolution r as the mixture g of r components

$$g=\sum_{j=1}^{r}\beta_j g_j \tag{35}$$

builds from the r subsets $\mathcal{S}_1,\ldots,\mathcal{S}_r$ remaining after the iteration $n-r$ of the Bregman hierarchical clustering algorithm, subsets extracted from the hierarchical mixture model. The parameters of the jth component are computed from the weighted distributions belonging to the subset $\mathcal{S}_j$. The distribution g_j is the centroid (right-sided, left-sided, or symmetric centroid) of the subset $\mathcal{S}_j$ (cf. Section 2.3). The weight β_j is computed as

$$\beta_j=\sum_i \alpha_i \quad \text{s.t. } \{\alpha_i,\boldsymbol{\Theta}_i\}\in\mathcal{S}_j \tag{36}$$

Section 4.6 will show that this mixture simplification process is almost instantaneous. Since all the combinations are considered during the first step of the algorithm, the two subsets selected to be merged are the same using either the right-sided or the left-sided Bregman divergence. As a consequence, the hierarchical mixture model is strictly the same with both left- and right-sided Bregman divergences. However, the mixture simplification based on the hierarchical mixture model depends on the divergence chosen since the right-sided and left-sided centroids are different. The hierarchical mixture model built using the symmetric Bregman divergence is different from the one built using a sided Bregman divergence. Moreover, the use of a symmetric measure allows to speed-up the step one of the algorithm (selection step) by considering $n(n-1)/2$ couples of subsets instead of $n(n-1)$.

The hierarchical mixture model gathers into a same structure all the possible resolutions (from 1 to n). This hierarchical structure allows us to introduce a new method to automatically learn the *optimal* number of

Please cite this article as: V. Garcia, F. Nielsen, Simplification and hierarchical representations of mixtures of exponential families, Signal Process. (2010), doi:10.1016/j.sigpro.2010.05.024

components in the simplified mixture g. The notion of optimality is relative to the two following constraints that the proposed method must satisfy:

1. g has to be as compact as possible,
2. g must reach a minimum prescribed quality $D_{KL}(f,g) \leq \tau$,

where $D_{KL}(f\|g)$ is here the Kullback–Leibler divergence estimated by a Monte-Carlo method since it does not admit a closed-form expression for mixture of exponential families. The minimum quality τ is a user defined parameter. Indeed, any learning method is based at least on one parameter and constraining the mixture quality instead of constraining the number of components is the most appropriated approach.

The learning process is based on the following assumption (verified in practice): the simplification quality increases ($D_{KL}(f,g)$ decreases) with the resolution (see Fig. 10). Therefore, a standard binary search algorithm on the resolution allows one to quickly learn the optimal number of components m in the simplified mixture.

3.4. jMEF: Java library for mixtures of exponential families

In this section, we briefly introduce `jMEF`.[3] `jMEF` is a cross-platform Java™ library designed to create, process and manage mixtures of exponential families. To be more specific, `jMEF` allows one to:

- create and manage mixtures of exponential families,
- estimate the parameters of a mixture of exponential families using Bregman soft clustering (cf. Section 3.1),
- simplify mixtures using Bregman hard clustering (cf. Section 3.2),
- define a hierarchical mixture model using the Bregman hierarchical clustering (cf. Section 3.3),
- and automatically retrieve the optimal number of components in the mixture using the hierarchical mixture model (cf. Section 3.3).

The source code is available on-line at http://www.lix.polytechnique.fr/~nielsen/MEF and is licensed under an open source MIT License (a Matlab® wrapper is also available). The experiments presented in Section 4 were based on a code entirely written using the jMEF library. As a consequence, these experiments are reproducible, and are provided as self-contained tutorials of this library.

4. Experiments

4.1. Introduction and setup

All the procedures used in the following experiments were written using the `jMEF` library presented in Section 3.4. The computer used was a Dell Precision M6400 laptop (Intel Core 2 duo at 2.53 GHz, 4Go DDR2 memory, Windows XP 32 bits, Java 1.6).

4.2. Bregman soft clustering

The first experiment is similar to an experiment proposed by Banerjee et al. [6]. We created three mixtures of exponential families, each mixture having three components:

- Mixture of Gaussian distributions with $\mu_1 = 10$, $\mu_2 = 20$, $\mu_3 = 40$, and $\sigma_1^2 = \sigma_2^2 = \sigma_3^2 = 25$.
- Mixture of Poisson distributions with $\lambda_1 = 10$, $\lambda_2 = 20$, $\lambda_3 = 40$.
- Mixture of binomial distributions with $p_1 = 0.1$, $p_2 = 0.2$, $p_3 = 0.4$ and the number of trials $n = 100$.

The mixtures were created to be similar in shape with distributions, respectively, centered on 10, 20 and 40. The distributions' weight was set to $\alpha = \frac{1}{3}$ for all the mixture distributions. We randomly drew 1000 points from each mixture that we stored in three different tables (one for each mixture). For each point set, we estimated a mixture of Gaussian distributions, a mixture of Poisson distributions, and a mixture of binomial distributions using the proposed Bregman soft clustering algorithm. The maximum number of iterations of the Bregman soft clustering was set to 30. The quality of the clustering was measured in terms of normalized mutual information (denoted NMI, cf. [14]) between the predicted clusters and original clusters. The NMI is equal to 1 is the two clusters are similar, and is close to 0 if they are mostly different. The results were averaged over 100 trials. A point assigned to one class (highest distribution value) for the generative mixture can indeed be assigned to another class for the estimated mixture. Table 2 presents the NMI for the different possible combinations. We can see that the clustering quality is better when the mixture model estimated using the Bregman soft clustering algorithm matches the generative mixture model.

This experiment first confirms the experiment of Banerjee et al. [6] and second validates the `jMEF` code.

4.3. Bregman soft clustering and statistical images

Mixtures of Gaussians are, by far, the most often used mixture models. In this experiment, we show that we can roughly describe an image (visual representation) using a

Table 2
Normalized mutual information (NMI) between the original cluster and the estimated clusters.

Generative model	Gaussian	Poisson	Binomial
Gaussian	$\mathbf{0.9148 \pm 0.0615}$	0.8752 ± 0.0687	0.8980 ± 0.0706
Poisson	0.7374 ± 0.0577	$\mathbf{0.8364 \pm 0.0818}$	0.8114 ± 0.0723
Binomial	0.8555 ± 0.0456	0.9503 ± 0.0446	$\mathbf{0.9526 \pm 0.0474}$

The rows correspond to different generative models and the columns to the estimated models.

[3] Java library for Mixtures of Exponential Families.

Please cite this article as: V. Garcia, F. Nielsen, Simplification and hierarchical representations of mixtures of exponential families, Signal Process. (2010), doi:10.1016/j.sigpro.2010.05.024

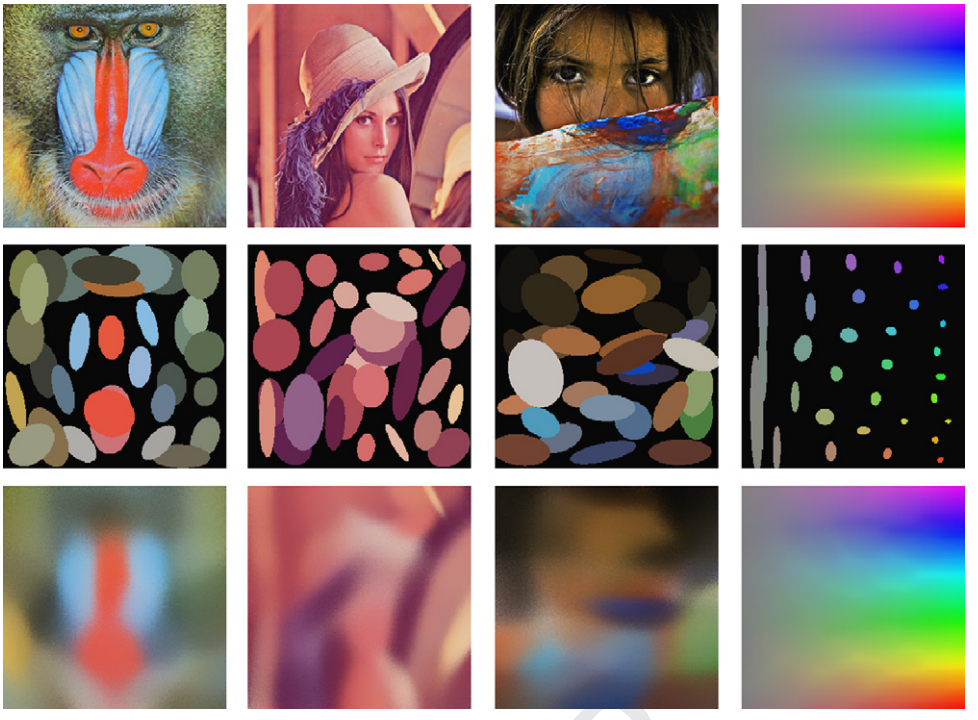

Fig. 2. Application of Bregman soft clustering to statistical images. Input images (first row) are (from left to right) Baboon, Lena, Shantytown, and Colormap. The second row illustrates the mixtures of Gaussians estimated from the input images. The third row shows the statistical images generated from mixtures of Gaussians.

compact mixture of Gaussians. A typical application of such a representation is color image retrieval.

Given an input image (see Fig. 2, first row), we considered a pixel p as a point in a five-dimensional space

$$p = (p_R, p_G, p_B, p_X, p_Y)^\top$$

where $\{p_R, p_G, p_B\}$ represent the RGB[4] color information of the pixel, and $\{p_X, p_Y\}$ are spatial information (the position of the pixel, respectively, the column and the row index). Each input image contained 65 536 pixels (image size 256 $\times$ 256 pixels). Using the Bregman soft clustering on the set pixels, we estimated a mixture of Gaussians f of 32 components. As presented below, the value 32, chosen arbitrary, allowed to capture the image structure and provided a relatively compact mixture model. However, another value could have been chosen instead.

Each Gaussian distribution of the estimated mixture was parametrized by its mean

$$\mu = (\mu_R, \mu_G, \mu_B, \mu_X, \mu_y)^\top$$

and its covariance matrix Σ. The second row of Fig. 2 illustrates the estimated mixture for each input image. Each distribution is represented by an ellipse centered on $\{\mu_X, \mu_Y\}$. The color of the ellipse is $\{\mu_R, \mu_G, \mu_B\}$ in the RGB colorspace and its shape is determined by the covariance matrix Σ. The input images were Baboon, Lena, Shantytown, and Colormap. At this stage, the fact that an image can be represented by a mixture of Gaussians is not clear yet. Then, we filled an empty image of size 256 $\times$ 256 pixels with pixels p randomly drawn (color+position) from the estimated mixture, each point being rounded to the nearest neighbor such that $p \in [0,255]$[5]. Points out of range were simply ignored. In order to have a smooth result, we drew points from f until having at least 20 sample points by pixel position $\{p_X, p_Y\}$. The color value of the statistical image pixel at the position $\{p_X, p_Y\}$ was the average color value of the drawn points at the same position. The resulting image is called *statistical image* since it was generated from a mixture of statistical distributions. The third row of Fig. 2 illustrates the statistical images for the input images. One can roughly recognize image Baboon, Lena, and Shantytown. The main structure of the images was captured[5] by the mixtures such as, for instance, the nose and the cheeks of Baboon, and the hat of Lena. The statistical image generated from Colormap was a very precise representation of the initial image. Indeed, Colormap is, by definition, a statistical image. It does not have well defined structures as in natural images but the probability of a color appearance at a given position is well captured by the mixture components.

The proposed experiment shows that the image structure can be captured into a mixture of Gaussians. The image is then represented by a small set of parameters (in comparison to the number of pixels) which is well adapted to applications such as color image

[4] RGB: color components red, green, and blue.

[5] As expected, high-frequencies of edges are not revealed.

Please cite this article as: V. Garcia, F. Nielsen, Simplification and hierarchical representations of mixtures of exponential families, Signal Process. (2010), doi:10.1016/j.sigpro.2010.05.024

retrieval. Considering an input image represented by its mixture of Gaussians, it is then trivial to retrieve, in an image database, a set of images that have a similar color organization.

One can ask why the RGB colorspace has been chosen here since other colorspaces such as YUV or Lab are known to be more suited to image processing? The use of such a colorspace implies to write conversion procedures. Using the RGB colorspace, we have been able to clearly present statistical images without increasing the code complexity.

4.4. Mixture simplification applied to image segmentation

In this experiment, we use the Bregman hard clustering (cf. Section 3.2) to simplify mixtures of Gaussians in the context of clustering-based image segmentation. Let f be a mixture of Gaussians of n components in a three-dimensional space. Given a color image I, a pixel p is here considered as a point in $\mathbb{R}^3$

$$p = (p_R, p_G, p_B)^\top$$

where p_R, p_G, p_B are the RGB color information. The image segmentation is performed by assigning each image pixel p to the class C_i having the highest density value:

$$f_i(p) > f_j(p), \quad \forall j \in [1,n] \backslash \{i\}$$

Then, the image segmentation is illustrated by replacing the color value of the pixel p by the mean μ_i of the Gaussian f_i.

Given an input image, the first step of this experiment was to estimate, using the Bregman soft clustering algorithm, an initial mixture of Gaussians f of 32 components from the image pixels. Second, we simplified f using the Bregman hard clustering into a mixture of

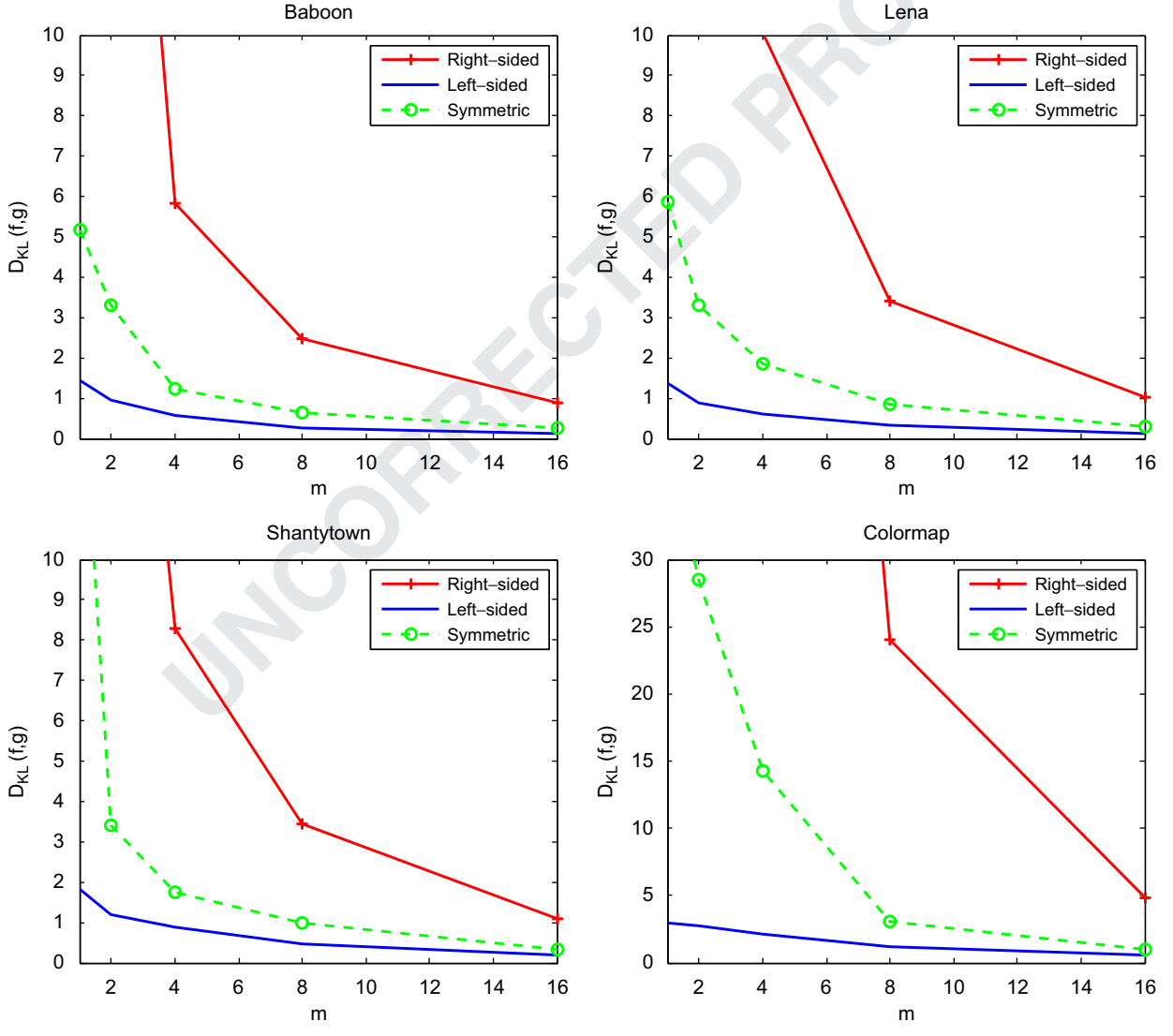

Fig. 3. Evolution of the mixture simplification quality $D_{KL}(f,g)$ as a function of number m of components in the simplified mixture and for the right-sided, left-sided, and symmetric Bregman hard clustering algorithms.

Please cite this article as: V. Garcia, F. Nielsen, Simplification and hierarchical representations of mixtures of exponential families, Signal Process. (2010), doi:10.1016/j.sigpro.2010.05.024

Gaussians g of m components with m={1,2,4,8,16}. More specifically, we simplified f with the right-sided, left-sided, and symmetric Bregman hard clustering algorithms. Fig. 3 shows the evolution of the simplification quality as a function of m and for the three mentioned Bregman hard clustering algorithms. The input images were Baboon, Lena, Shantytown, and Colormap. The simplification quality is given by the Kullback–Leibler divergence $D_{KL}(f,g)$ estimated by a Monte-Carlo method. Five thousand points were randomly drawn to estimate $D_{KL}(f,g)$. The simplification quality increased ($D_{KL}(f,g)$ decreased) with m whatever the Bregman divergence used. Indeed, the quality of the approximation of the initial mixture f increases with the number of Gaussians in the simplified model g. Second, the left-sided Bregman divergence gave the best results and the right-sided the worst. 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 hard clustering was then the expected behavior. The symmetric Bregman hard clustering provided better results than the right-sided one but worse than the left-sided one since it is computed from right-sided and left-sided Bregman divergences and centroids.

Fig. 4 illustrates the mixture simplification, based on the left-sided Bregman hard clustering, in the context of image segmentation. The number of classes is, by construction, equal to the number of components

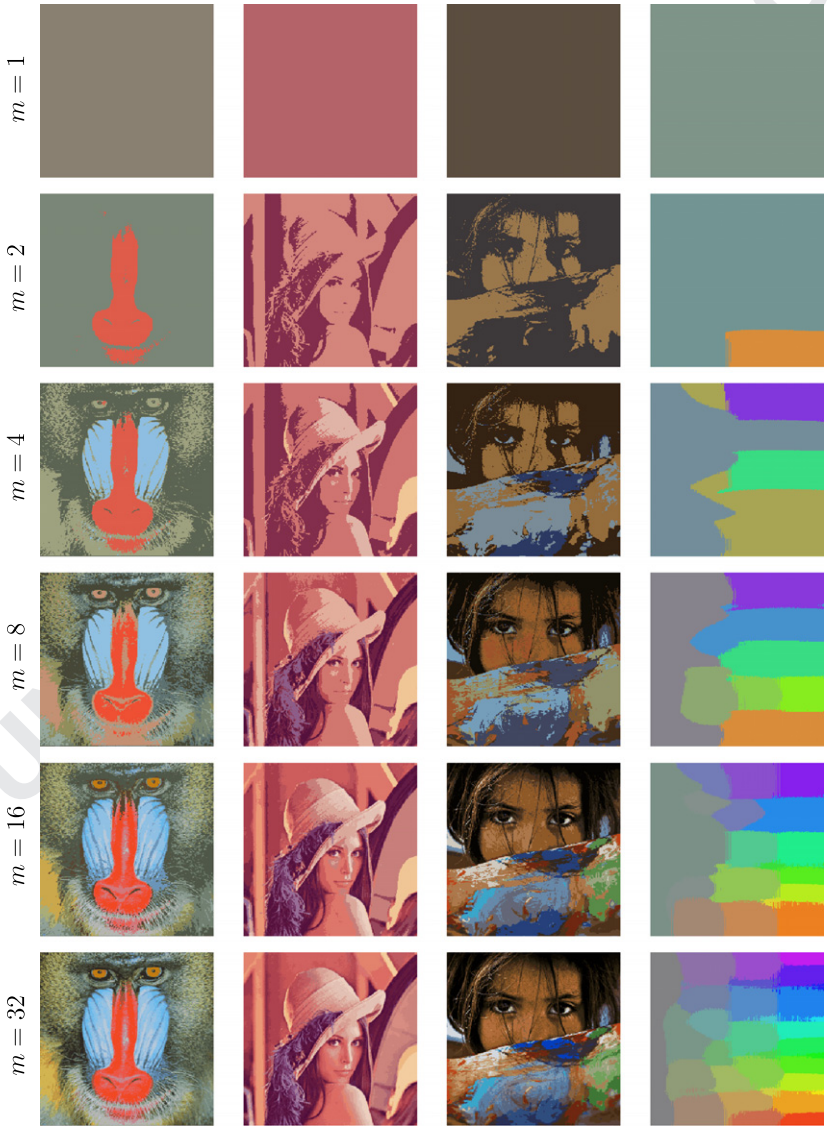

Fig. 4. Illustration of the mixture simplification using the left-sided Bregman hard clustering algorithm in the context of clustering-based image segmentation. The value m denotes the number of components in the simplified mixture. The initial mixture contained 32 components (last row).

Please cite this article as: V. Garcia, F. Nielsen, Simplification and hierarchical representations of mixtures of exponential families, Signal Process. (2010), doi:10.1016/j.sigpro.2010.05.024

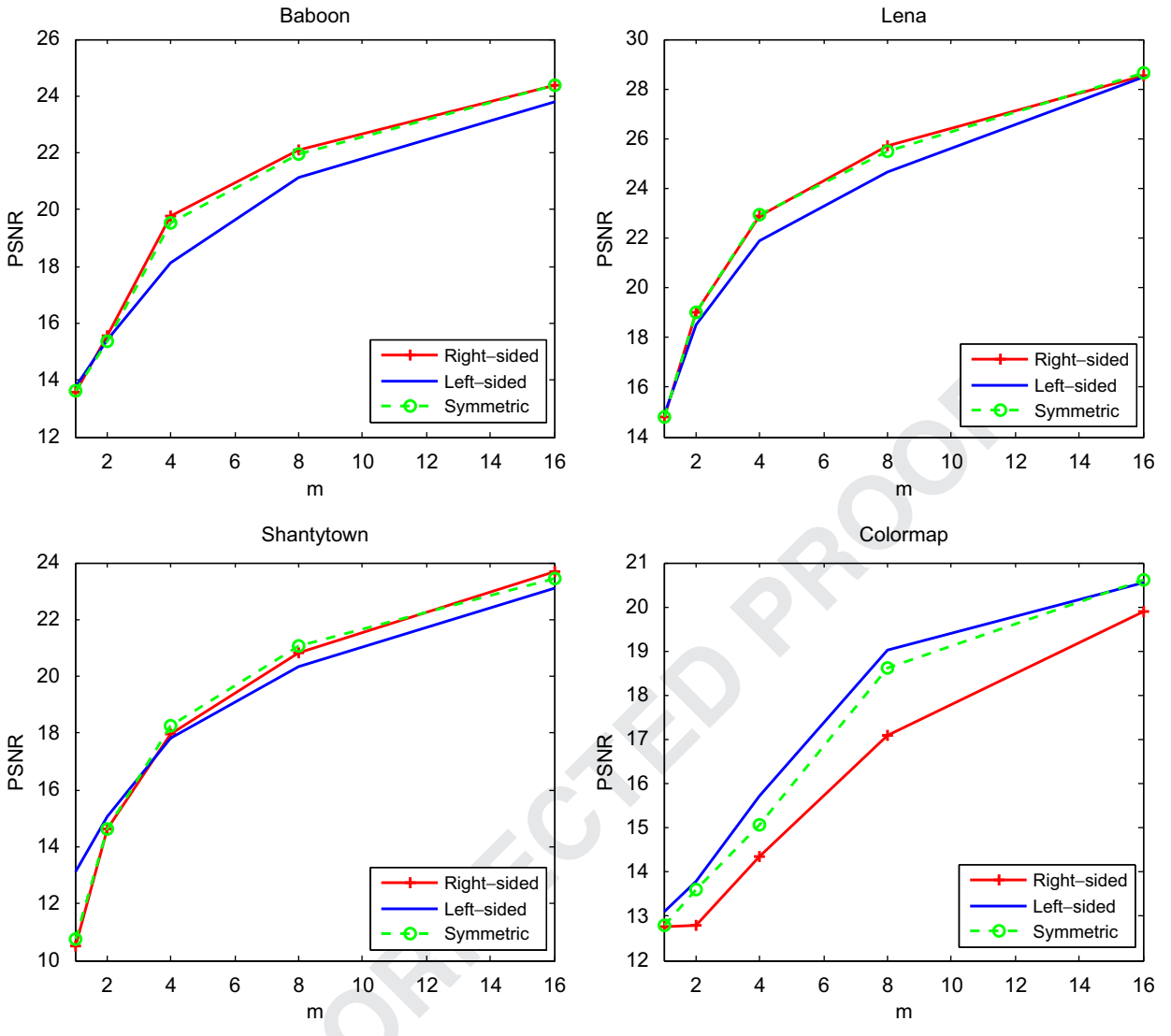

Fig. 5. Evolution of the segmentation quality (PSNR) as a function of number m of components in the simplified mixture and for the right-sided, left-sided, and symmetric Bregman hard clustering algorithms.

m in g. The visual quality of the image segmentation (visual similarity with the input image) increases m.

Fig. 3 clearly indicates that, according to $D_{KL}(f,g)$, the left-sided Bregman hard clustering is a more efficient mixture simplification method than both the right-sided and symmetric Bregman hard clusterings. However, one can ask what is the influence of the method on the image segmentation? Since there is no ground-truth on image segmentation, we propose to use the classical PSNR[6] measure between the initial image and the segmented image to evaluate the segmentation quality. Fig. 5 shows the evolution of this segmentation quality as a function of m for the right-sided, left-sided, and symmetric Bregman hard clustering algorithms. For the three natural images (Baboon, Lena, and Shantytown), it appears that, according to the segmentation quality, the right-sided and the symmetric Bregman hard clusterings provide a better mixture simplification than the left-sided Bregman hard clustering. Indeed, in the context of image segmentation, it is natural to represent a set of Gaussians by an encompassing Gaussian. As presented in Section 2.3, the right-sided and the symmetric centroids try to encompass the set of distributions while the left-sided centroid is simply an average distribution. Thus, the right-sided and the symmetric Bregman hard clusterings are more adapted to image segmentation than the left-sided Bregman hard clustering. The image Colormap is a special case since it is, by definition, a statistical image. In this case, the left-sided Bregman hard clustering provides the best segmentation (and the right-sided one the worst).

[6] Peak Signal-to-Noise Ratio.

Please cite this article as: V. Garcia, F. Nielsen, Simplification and hierarchical representations of mixtures of exponential families, Signal Process. (2010), doi:10.1016/j.sigpro.2010.05.024

We conclude this experiment by studying the computation time of the proposed mixture simplification method. The main goal of the mixture simplification algorithms is to reduce the computation time by avoiding to recompute a simpler mixture from the initial point set. Let $\mathcal{S}$ be a set of 5000 points in a five-dimensional space, and let f be a mixture of 32 Gaussians estimated from $\mathcal{S}$ using the Bregman soft clustering. The estimation of a simpler mixture g of 10 components from $\mathcal{S}$ (soft clustering) is performed in 138 s while the simplification of f using the Bregman hard clustering is performed in 694 ms (200 times faster). The speed-up depends on size of the point set, on the mixture size, and on the dimension.

4.5. Mixture simplification: comparison with state of the art

Goldberger et al. proposed a fast mixture of Gaussians simplification algorithm (cf. [13]) named UTAC (unscented transform approximation clustering) based on the unscented transform (UT) method (cf. [15,16]). The UTAC algorithm proceeds by maximizing the UTA (unscented transform approximation of the negative cross-entropy) criterion computed between the initial mixture f and the estimated mixture g. The authors show that the UTA criterion can be maximized with a standard EM-like algorithm. UTAC is, as far as we know, the most efficient mixture simplification algorithm (yet) both in terms of quality and computation time.

In this section, we compare the UTAC algorithm to the proposed Bregman hard clustering for simplifying mixtures of Gaussians. Given an input image, we learnt a mixture of Gaussians f of 32 components as presented in Section 4.4. The value 32 was arbitrary chosen: this value is large enough to show the algorithm behavior and is small enough to have fast computation. We then simplified f into a simpler mixture of Gaussians using the algorithms UTAC and left-sided Bregman hard clustering. Fig. 6 presents the evolution of the simplification quality as a function of m (number of Gaussians in the simplified mixture) for both algorithms. The input images used for this experiment were Baboon and Lena. The Bregman hard clustering and the UTAC algorithms provided a simplified mixture of similar quality. However, the Bregman hard clustering was faster than the UTAC algorithm. For instance, let us consider a mixture f of 100 Gaussians in a five-dimensional space, the simplification into a mixture of 10 components is performed in 6275 ms using UTAC and in 694 ms using the Bregman hard clustering (9 times faster than UTAC). The speed-up depends on the mixture size and on the dimension. In both algorithms, the maximum number of iterations was set to 30.

To summarize, the proposed Bregman hard clustering provides a mixture simplification similar to the one computed by the UTAC algorithm. However, the proposed mixture simplification method is faster than UTAC, and is generic as it applies to any mixture of exponential families.

4.6. Hierarchical mixture model applied to image segmentation

The Bregman hierarchical clustering (cf. Section 3.3) allows first to simplify a mixture of exponential families, and second to automatically learn the optimal number of components in the mixture model. In this section, we study the effect of the linkage criterion (Section 4.6.1) and of the Bregman divergence type (Section 4.6.2) on the simplification quality. Then, we compare the computation time of the Bregman hard clustering and of the Bregman hierarchical clustering to simplify a given mixture model. Finally, we used the method described in Section 3.3 to learn the optimal number of components in the mixture model.

The initial mixture of Gaussians used in this section are composed of 32 and are estimated form the images Baboon, Lena, Shantytown and Colormap (cf. Section 4.4).

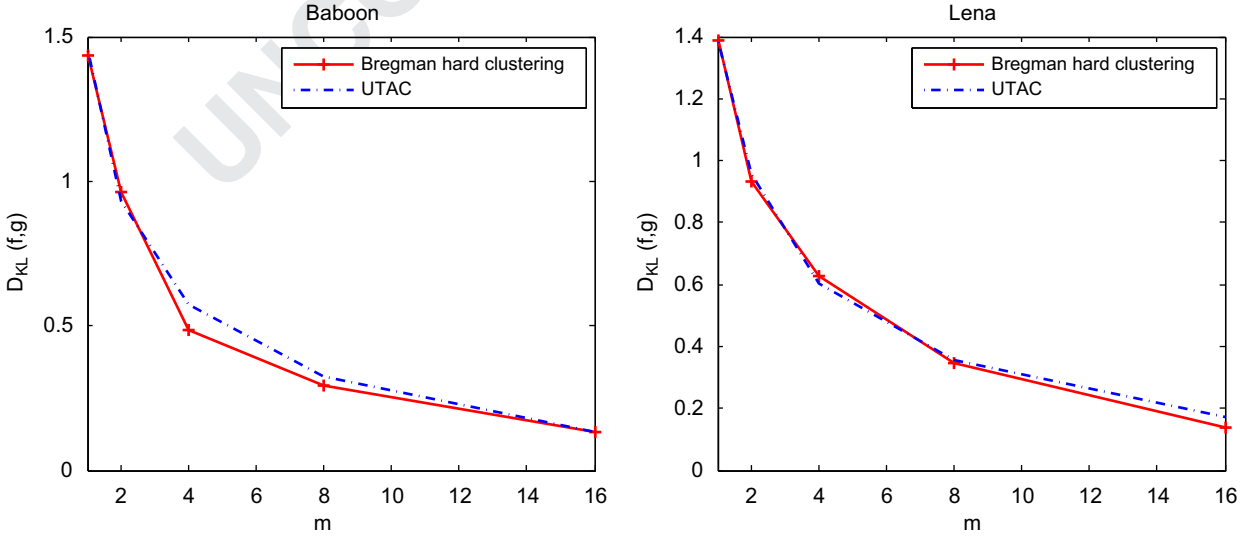

Fig. 6. Evolution of the mixture simplification quality $D_{KL}(f,g)$ as a function of m for algorithms UTAC and left-sided Bregman hard clustering.

Please cite this article as: V. Garcia, F. Nielsen, Simplification and hierarchical representations of mixtures of exponential families, Signal Process. (2010), doi:10.1016/j.sigpro.2010.05.024

4.6.1. Linkage criterion

Given an initial mixture f, we built three hierarchical mixtures of Gaussians using the left-sided Bregman hierarchical clustering, the three hierarchical mixtures being, respectively, based on the minimum, the maximum, and the average distances as linkage criterion. We deduced from these hierarchical mixtures three simplified mixture models for different resolutions. Fig. 7 shows the evolution of the simplification quality as a function of the resolution and for the three hierarchical mixtures of Gaussians. The maximum and average distances provided a better mixture simplification method (lower $D_{KL}(f,g)$) than the minimum distance. Indeed, the maximum and average distances are known to be more reliable and more robust to outliers than the minimum distance. In the paper remainder, we will use the maximum distance to build any hierarchical mixture model.

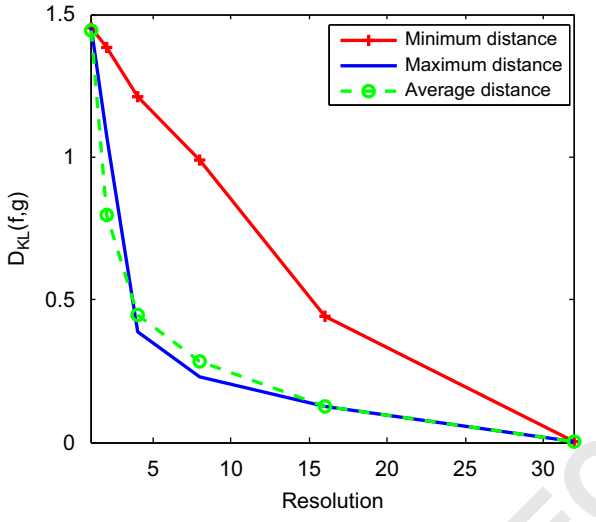

Fig. 7. Evolution of the simplification quality $D_{KL}(f,g)$ as a function of the resolution. The linkage criteria used to built the hierarchical mixture models are the minimum, the maximum, and the average distances.

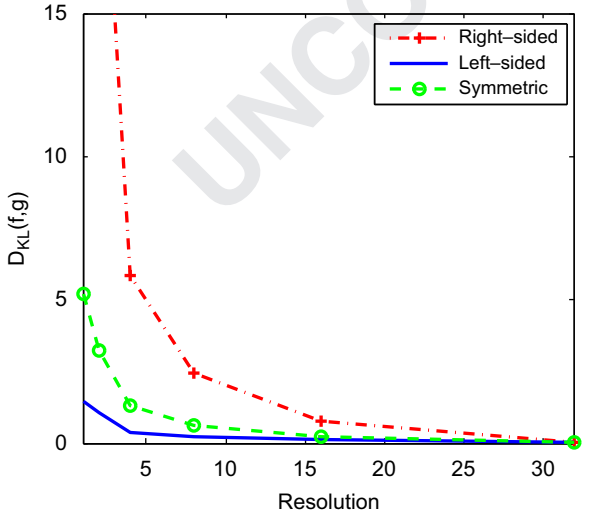

Fig. 8. Evolution of the simplification quality $D_{KL}(f,g)$ as a function of the resolution and for the right-sided, left-sided, and symmetric Bregman hard clustering algorithms.

4.6.2. Bregman divergence type

Given an initial mixture f, we built three hierarchical mixture of Gaussians, respectively, using the right-sided, the left-sided, and the symmetric Bregman hierarchical clustering. We deduced from these hierarchical mixtures three simplified mixture models for different resolutions. Fig. 8 shows the evolution of the simplification quality as a function of the resolution and for the three mentioned Bregman hierarchical clustering algorithms. The simplification quality increased ($D_{KL}(f,g)$ decreased) with resolution whatever the Bregman divergence used. Moreover, the left-sided Bregman hierarchical clustering is the most efficient (in terms of quality) method and the right-sided one is the worst one (see Section 4.4 for explanation). Fig. 9 illustrates the mixture simplification, using the left-sided Bregman hierarchical clustering, applied to image segmentation. The visual quality of the image segmentation increases with the resolution.

4.6.3. Computational time

By comparing Figs. 3 and 8, one can notice that the simplification quality is similar using the Bregman hard clustering and Bregman hierarchical clustering. However, even if the mixture simplification process is fast using the Bregman hard clustering, it is almost instantaneous using Bregman hierarchical clustering. For instance, let us consider a mixture f of 100 Gaussians in a five-dimensional space, the simplification into a mixture of 10 components is performed in 694 ms using the Bregman hard clustering and in 2 ms using the Bregman hierarchical clustering. The construction of the hierarchical mixture model (87 s in this experiment) has not been taken into account since this process is usually done off-line.

4.6.4. Learning of the optimal mixture model

In Section 3.3, we have presented a method to automatically learn the optimal number of components in a given mixture of exponential family. Given an initial mixture of Gaussians f learnt from an input image, we built a hierarchical mixture model h using the left-sided Bregman hierarchical clustering. We set the constraint on the minimum quality $\tau = 0.2$. Fig. 10 shows the evolution of the simplification quality as a function of the number of components m (i.e. resolution) and for the images Baboon, Lena, Shantytown, and Colormap. We plotted on the same figure the threshold $\tau = 0.2$ (magenta horizontal solid line). The learning process provided a mixture of 10 components for Baboon, 14 components for Lena, 16 for Shantytown, and 23 for Colormap (see Fig. 11).

The learning process is not a fast method since it is based on a Monte-Carlo method. For instance, given a mixture f of 100 Gaussians in a five-dimensional space, the optimal mixture model is computed in 35 s: more

Please cite this article as: V. Garcia, F. Nielsen, Simplification and hierarchical representations of mixtures of exponential families, Signal Process. (2010), doi:10.1016/j.sigpro.2010.05.024

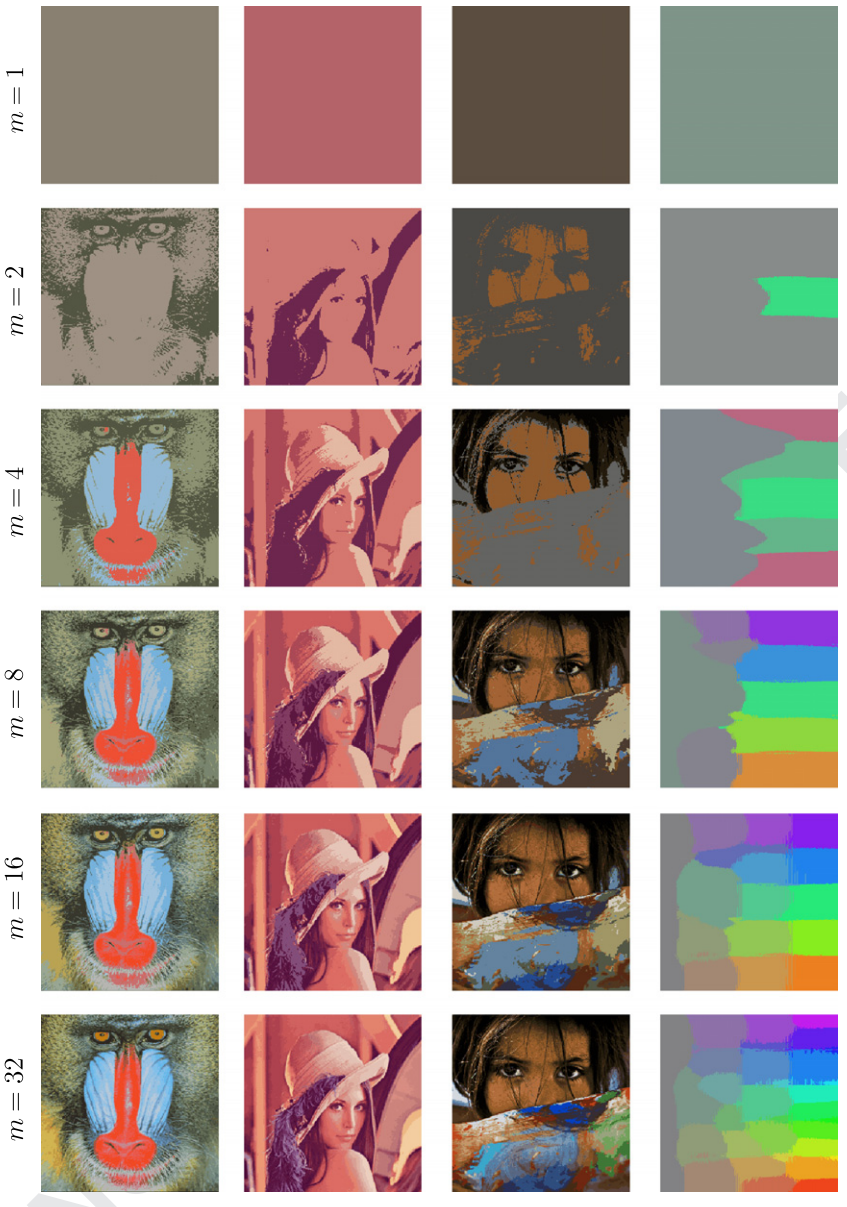

Fig. 9. Bregman hierarchical clustering (left-sided) and application to image segmentation. The value m denotes the resolution which is equal to the number of components in the simplified mixture. The maximum resolution contained 32 components (last row).

than 99% of the computation time was used to estimate the Kullback–Leibler divergence from 1000 sample points.

To summarize this experiment, the Bregman hierarchical clustering allowed to efficiently (quality and computation time) simplify a given mixture of Gaussians or a mixture of exponential families. Using the maximum or the average distance as linkage criterion, the Bregman hierarchical clustering provided a simplification quality similar to the Bregman hard clustering. Finally, the learning process seemed to be an efficient method (in terms of quality) to retrieve the optimal number of components in the simplified mixture model.

5. Conclusion

The exponential family is a wide class of distributions including, among others, Gaussian, Poisson, Laplacian, binomial, multinomial, Bernoulli, and Rayleigh distributions. In this paper, we have presented three clustering algorithms adapted to mixtures of exponential families: the Bregman soft clustering, the Bregman hard clustering, and the Bregman hierarchical clustering.

The Bregman soft clustering algorithm is the adaptation of the expectation–maximization (EM) algorithm allowing one to estimate the parameters of a mixture of exponential families from a set of observation points. Our

Please cite this article as: V. Garcia, F. Nielsen, Simplification and hierarchical representations of mixtures of exponential families, Signal Process. (2010), doi:10.1016/j.sigpro.2010.05.024

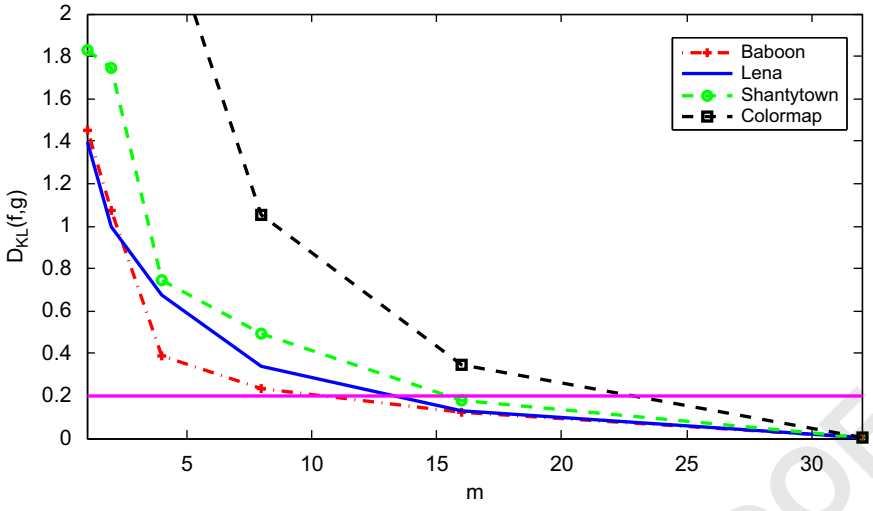

Fig. 10. Evolution of the simplification quality $D_{KL}(f,g)$ as a function of the resolution. The horizontal line is the threshold on the minimum simplification quality set to $\tau = 0.2$.

Fig. 11. Automatic learning of the optimal number of components in the simplified mixture. The threshold on the minimum simplification quality was $\tau = 0.2$.

algorithm uses the most generic canonical form of exponential families, and thus bypasses a technical difficulty (i.e., degeneracy occur for some mixtures including Gaussian mixtures in the work of Banerjee et al. [6]). In particular, it has been shown that, using this algorithm, an image can be represented (color organization) by a compact mixture of Gaussians. Such a statistical image representation could be very useful in various applications such as color image retrieval.

The Bregman hard clustering algorithm is the adaptation of the celebrated *k*-means algorithm to the case of mixtures of exponential families, relying on the fact that the relative entropy of members of the same exponential family is equivalent to a corresponding Bregman divergence. The proposed method is slightly but importantly different from the original version (cf. [6]) since it is suitable to weighted distributions. Thus, by considering a mixture of exponential families as a set of weighted distributions, the Bregman hard clustering appears to be a very efficient mixture simplification algorithm. Indeed, this algorithm provides a similar mixture simplification quality in a shorter computation time than the state of the art UTAC algorithm (cf. [13]).

The Bregman hierarchical clustering algorithm is the adaptation of the hierarchical clustering to the case of mixtures of exponential families in the context of Bregman divergences. This algorithm creates a hierarchical mixture model from an initial mixture of exponential families. This hierarchical mixture model allows first to quickly compute a compact version of the initial mixture, and second to automatically learn the optimal number of components for the simplified mixture.

Both Bregman hard clustering and Bregman hierarchical clustering algorithms have been applied to mixture simplification and clustering based image segmentation.

In this paper, we also have presented `jMEF`.[7] `jMEF` is a cross-platform open source Java™ library designed to create, process and manage mixtures of exponential families. This library implements the Bregman soft clustering, the Bregman hard clustering, and the Bregman hierarchical clustering. The `jMEF` library is available on-line at http://www.lix.polytechnique.fr/~nielsen/MEF and is licensed under a MIT License.

Acknowledgments

We warmly thank Arindam Banerjee for e-mail correspondences. We gratefully acknowledge financial support from DIGITEO GAS (contract 2008-16D) and ANR GAIA (contract 07-BLAN-0328-01).

[7] Java library for Mixtures of Exponential Families.

Please cite this article as: V. Garcia, F. Nielsen, Simplification and hierarchical representations of mixtures of exponential families, Signal Process. (2010), doi:10.1016/j.sigpro.2010.05.024

References

[1] C. Elkan, Clustering documents with an exponential-family approximation of the Dirichlet compound multinomial distribution, in: International Conference on Machine Learning, New York, NY, USA, 2006, pp. 289–296.

[2] T. Masada, S. Kiyasu, S. Miyahara, Clustering images with multinomial mixture models, in: International Symposium on Advanced Intelligent Systems, 2007.

[3] F. Nielsen, V. Garcia, Statistical exponential families: a digest with flash cards. December 2009, arXiv 0911.4863 〈http://arxiv.org/abs/0911.4863〉.

[4] S. Kullback, R.A. Leibler, On information and sufficiency, The Annals of Mathematical Statistics 22 (1) (1951) 79–86.

[5] S. Kullback, Information Theory and Statistics, John Wiley and Sons, 1959.

[6] A. Banerjee, S. Merugu, I. Dhillon, J. Ghosh, Clustering with Bregman divergences, Journal of Machine Learning Research 6 (2005) 234–245.

[7] F. Nielsen, R. Nock, Sided and symmetrized Bregman centroids, IEEE Transactions on Information Theory 55 (2009) 2048–2059.

[8] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, Journal of the Royal Statistical Society. Series B (Methodological) 39 (1977) 1–38.

[9] S.P. Lloyd, Least squares quantization in PCM, IEEE Transactions on Information Theory 28 (2) (March 1982) 129–137.

[10] A. Peter, A. Rangarajan, A new closed-form information metric for shape analysis, in: Medical Image Computing and Computer Assisted Intervention, 2006, pp. 249–256.

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

[12] D. Arthur, S. Vassilvitskii, *k*-means++: the advantages of careful seeding, in: Annual ACM-SIAM Symposium on Discrete Algorithms, Philadelphia, PA, USA, 2007, pp. 1027–1035.

[13] J. Goldberger, H. Greenspan, J. Dreyfuss, Simplifying mixture models using the unscented transform, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (2008) 1496–1502.

[14] A. Strehl, J. Ghosh, Cluster ensembles—a knowledge reuse framework for combining multiple partitions, Journal of Machine Learning Research 3 (2002) 583–617.

[15] J. Goldberger, S. Gordon, H. Greenspan, An efficient image similarity measure based on approximations of KL-divergence between two Gaussian mixtures, in: IEEE International Conference on Computer Vision, 2003.

[16] S.J. Julier, J.K. Uhlmann, Unscented filtering and nonlinear estimation, Proceedings of the IEEE 92 (2004) 401–422.

Please cite this article as: V. Garcia, F. Nielsen, Simplification and hierarchical representations of mixtures of exponential families, Signal Process. (2010), doi:10.1016/j.sigpro.2010.05.024