

# jMEF: A Java Library for Mixture of Exponential Families

**Vincent Garcia**

*École Polytechnique*

91128 Palaiseau Cedex, France

GARCIAV@LIX.POLYTECHNIQUE.FR

**Frank Nielsen**

*École Polytechnique / Sony Computer Science Laboratories Inc.*

91128 Palaiseau Cedex, France

FRANK.NIELSEN@ACM.ORG

**Editor:** XXX

## Abstract

We describe the jMEF library: a Java library allowing one to create and manage mixture of exponential families in statistics. jMEF also allows on to learn a mixture from a given set of observations, to simplify a mixture, to provide a hierarchical representation of a given mixture, or to automatically learn the optimal number of components of the simplified mixture. Because our framework is generic, it is suitable for any mixture of exponential families including, among others, mixtures of Gaussian, Poisson, and multinomial distributions.

**Keywords:** Mixture models, Exponential family, Bregman divergence, Bregman clustering

## 1. Introduction

A mixture model is a powerful framework commonly used to estimate the probability density function of a random variable. For instance, the mixtures of Gaussians (MoG) have been widely used in many different application domains including statistics, image and signal processing, physics, biology, *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  $f(x) = \sum_{i=1}^n \alpha_i f_i(x)$ , where  $f_i$  is a statistical distribution and  $\alpha_i \in [0, 1]$  is the weight associated to  $f_i$  such that  $\sum_{i=1}^n \alpha_i = 1$ . The use of mixture models implies to use algorithms adapted to such mixtures. Most of them were specifically designed for MoG since they represent the huge majority of the mixture model used. However, in the last decades, other distribution models such as Poisson or multinomial have been successfully used.

In this paper, we briefly introduce the jMEF library: a Java library for Mixtures of Exponential Families. Our framework is generic as it is defined for the wide class of exponential family including the following distributions: Gaussian, Poisson, binomial, Bernoulli, multinomial, Beta, Gamma, and Rayleigh distributions<sup>1</sup>. An exponential family is a family of statistical distributions that admits the following canonical decomposition  $p_F(x; \Theta) = \exp(\langle t(x), \Theta \rangle - F(\Theta) + k(x))$ , where  $t(x)$  is the sufficient statistics,  $\Theta$  the natural parameters,  $F$  the convex log normalizer function and  $k(x)$  the carrier measure. The representation of most common distributions as exponential families are given in Nielsen and Garcia (2009). In short, this cross-platform library allows one to create and manage mixture of exponential families. More interesting, jMEF is able (1) to learn a mixture from a set of observations using the Bregman soft clustering algorithm Banerjee et al. (2005), (2) to simplify a mixture using the Bregman hard clustering algorithm, and (3) to provide a hierarchical representation of a given mixture. The open source library is freely available at <http://www.lix.polytechnique.fr/~nielsen/MEF>.

---

1. More distributions will be added in the future.

## 2. Description of the library

### 2.1 Bregman soft clustering

The Bregman soft clustering make us of the duality Bregman divergence/exponential families to generalize the expectation-maximization (EM) algorithm allowing one to estimate the parameters of a mixture of exponential families from a point set  $\mathcal{S}$ . The initialization of the mixture consists first in performing a classical  $k$ -means algorithm to gather the points of  $\mathcal{S}$  in  $n$  subsets  $\mathcal{S}_1, \dots, \mathcal{S}_n$ . Then, the weight  $\alpha_j$  and the expectation parameters  $\mathbf{H}_j = \nabla F(\Theta_j)$  of the  $j^{\text{th}}$  mixture components are given by

$$\alpha_j = \frac{|\mathcal{S}_j|}{|\mathcal{S}|}, \quad \mathbf{H}_j = \frac{1}{|\mathcal{S}_j|} \sum_{i/x_i \in \mathcal{S}_j} t(x_i) \quad (\text{the observed point Amari and Nagaoka (2000)}) \quad (1)$$

where  $|\mathcal{S}_j|$  denotes the cardinal of the set  $\mathcal{S}_j$  and  $t(x)$  is the sufficient statistic. The Bregman soft clustering algorithm then alternates iteratively the following expectation (E) step and maximization (M) step:

- **E step**

$$p(i, j) = \frac{\alpha_j \exp(G(\mathbf{H}_j) + \langle t(x_i) - \mathbf{H}_j, \nabla G(\mathbf{H}_j) \rangle)}{\sum_{l=1}^n \alpha_l \exp(G(\mathbf{H}_l) + \langle t(x_i) - \mathbf{H}_l, \nabla G(\mathbf{H}_l) \rangle)} \quad (2)$$

- **M step**

$$\alpha_j = \frac{1}{|\mathcal{S}|} \sum_{i=1}^{|\mathcal{S}|} p(i, j), \quad \mathbf{H}_j = \frac{\sum_{i=1}^{|\mathcal{S}|} p(i, j) t(x_i)}{\sum_{i=1}^{|\mathcal{S}|} p(i, j)} \quad (3)$$

### 2.2 Bregman hard clustering

The  $k$ -means algorithm, also known as *hard clustering*, partitions a point set into  $k$  classes. The  $k$ -means algorithm alternates an *assignment step*, which assigns points to the cluster with the closest centroid, and an *update step*, which update centroids from the point repartition. The process is iterated until convergence. By considering a mixture of exponential families as a set of weighted distributions, we can easily adapt the  $k$ -means clustering toward the mixtures of exponential families. Indeed, the Bregman divergence provides a convenient measure to first assign the distributions to the different classes and second to update centroids (Bregman centroid, Nielsen and Nock (2009)). The proposed Bregman hard clustering implementation returns the  $k$  class centroids. This algorithm provides an efficient method (both in terms of quality and computation time) to simplify a mixture of exponential families by quantizing the mixture parameters.

### 2.3 Bregman hierarchical clustering

Hierarchical clustering is a group of methods consisting in building a hierarchy of clusters. Hierarchical clustering is generally subdivided in two categories: agglomerative and divisive methods. Here, we only consider the agglomerative methods which start with elementary subsets and successively merge subsets until having a single set containing all the objects. The Bregman hierarchical clustering is the adaptation of the hierarchical clustering toward the mixtures of exponential families. We consider a mixture of exponential families as a set of weighted distributions. The classical linkage criteria (minimum, maximum, and average distances) are then based on the Bregman divergence between distributions instead of the Euclidean distance between points. By generating a hierarchical mixture model, the proposed Bregman hierarchical clustering implementation allows one (1) to obtain in one step (fast computation) a simplified version of the initial mixture model, and (2) to automatically learns the *optimal* value number of classes in the simplified mixture satisfying a mixture criterion quality.

Generative model	Gaussian	Poisson	Binomial
Gaussian	<b>0.9148±0.0615</b>	0.8752±0.0687	0.8980±0.0706
Poisson	0.7374±0.0577	<b>0.8364±0.0818</b>	0.8114±0.0723
Binomial	0.8555±0.0456	0.9503±0.0446	<b>0.9526±0.0474</b>

Table 1: NMI between the original (row) and the estimated(column) clusters.

### 3. Getting started with jMEF with tutorials

#### 3.1 Bregman soft clustering

The first tutorial is similar to an experiment proposed by Banerjee et al. (2005). We create three 1-dimensional datasets of 1000 sample each, based on mixture models of Gaussian, Poisson and Binomial distributions respectively. All the mixture models have three components with equal priors centered at 10, 20 and 40 respectively. The standard deviation  $\sigma$  of the Gaussian densities was set to 5 and the number of trials  $N$  of the Binomial distribution was set to 100 so as to make the three models somewhat similar to each other, in the sense that the variance is approximately the same for all the models. For each dataset, we estimate the parameters of three mixture models of Gaussian, Poisson and Binomial distributions using the proposed Bregman soft clustering implementation. The quality of the clustering was measured in terms of the normalized mutual information (denoted NMI, see Strehl and Ghosh (2002)) between the estimated clusters and original clusters, and the results were averaged over 100 trials. The table 1 presents the NMI for the different possible combinations.

#### 3.2 Parameter estimation of a mixture of Gaussians

We propose in this tutorial to compare the Bregman soft clustering to the classical EM algorithm for estimating the parameters of a mixture of univariate Gaussians. Given an initial mixture of univariate Gaussians  $f$  of  $n$  components, we draw a set  $\mathcal{S}$  of  $m$  points from  $f$ . Then, we estimate from  $\mathcal{S}$  the parameters of two mixtures of univariate Gaussians  $f_1$  and  $f_2$  of  $n$  components respectively using the classical EM algorithm and the Bregman soft clustering algorithm. By this tutorial, we first show that the Bregman soft clustering is exactly similar to the classical EM algorithm as  $f_1 = f_2$ . Second, we show that the parameters of the initial mixture  $f$  are correctly recovered in  $f_1$  and  $f_2$ , provided that  $m$  is large enough.

#### 3.3 Mixture model simplification and image segmentation

We propose here to simplify mixture of exponential families using the Bregman hard clustering algorithm. To be more precise, we illustrate the impact of the mixture model simplification in the context of clustering-based image segmentation (see Garcia et al. (2009)). The tutorial also describes basic operations on mixtures such as saving (resp. loading) a mixture in (resp. from) an output (resp. input) file. The considered distributions are multivariate Gaussians. Given an image  $I$ , let  $n$  and  $m$  be the number of components respectively of the initial and of the simplified mixtures:

1. Load the mixture  $f$  of  $n$  components from an input file. If the file doesn't exist yet, the mixture is estimated from the pixels of the image (points RGB in  $\mathbb{R}^3$ ) using Bregman soft clustering, and the mixture is saved in an output file.
2. Compute the image segmentation from  $I$  and  $f$  and save the segmentation in an output file.
3. Simplify the mixture model  $f$  into a mixture  $g$  of  $m$  components.
4. Compute the image segmentation from  $I$  and  $g$  and save the segmentation in an output file.

The Fig. 1 illustrates several image segmentation for different values of  $m$ .

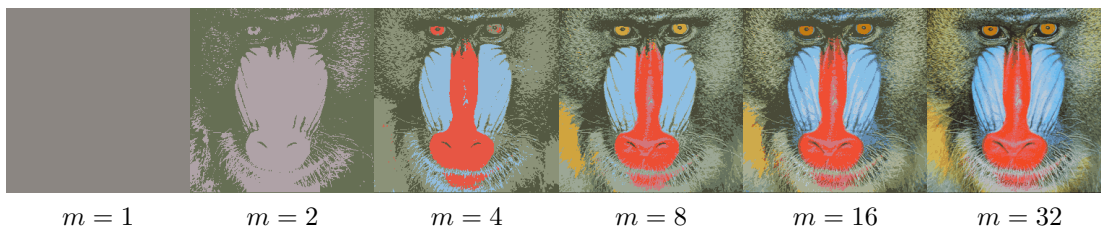


Figure 1: Application of mixture model simplification using the Bregman hard clustering to clustering-based image segmentation. The input image is Baboon.

### 3.4 Hierarchical mixture models and image segmentation

Finally, we present how to use the Bregman hierarchical clustering to build a hierarchical mixture model from an initial mixture  $f$ . As in Section 3.3, we apply the mixture simplification in the context of image segmentation and the considered distributions are multivariate Gaussians.

1. Load the mixture  $f$  of  $n$  components from an input file.
2. Compute the image segmentation from  $I$  and  $f$  and save the segmentation in an output file.
3. Build the hierarchical mixture model  $h$  from  $f$ .
4. Extract a simpler mixture model  $g$  of  $m$  components (resolution  $m$ ) from  $h$ .
5. Compute the image segmentation from  $I$  and  $g$  and save the segmentation in an output file.

The hierarchical mixture model allows one to automatically learn the number of components in the mixture model. In the tutorial, just replace `getResolution( $m$ )` by `getOptimalMixtureModel()`.

## 4. Availability and Requirements

The jMEF library is an open source library released under a MIT license: the library is free of charge, it can be used, copy, modify, merge, publish, distribute, sublicense, and it can be incorporated into both open-source and commercial software. The library is written in Java and is consequently cross-platform (MacOS X, MS Windows, and Linux). Its requires an installed Java virtual machine but without additional libraries.

## References

- S.i Amari and H. Nagaoka. *Methods of Information Geometry*. Oxford University Press, 2000.
- A. Banerjee, S. Merugu, I. Dhillon, and J. Ghosh. Clustering with Bregman divergences. *Journal of Machine Learning Research*, 6:234–245, 2005.
- V. Garcia, F. Nielsen, and R. Nock. Levels of details for gaussian mixture models. In *Asian Conference on Computer Vision*, 2009.
- F. Nielsen and V. Garcia. Statistical exponential families: A digest with flash cards. arXiv, December 2009. <http://arxiv.org/abs/0911.4863>.
- F. Nielsen and R. Nock. Sided and symmetrized Bregman centroids. *IEEE Transactions on Information Theory*, 55:2048–2059, 2009.
- A. Strehl and J. Ghosh. Cluster ensembles – a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3:583–617, 2002.