Fast learning of Gamma mixture models with *k*-MLE

Olivier Schwander[1] and Frank Nielsen[2]

[1] École Polytechnique, Palaiseau, France
[2] Sony Computer Science Laboratories, Tokyo, Japan

Abstract. We introduce a novel algorithm to learn mixtures of Gamma distributions. This is an extension of the k-Maximum Likelihood estimator algorithm for mixtures of exponential families. Although Gamma distributions are exponential families, we cannot rely directly on the exponential families tools due to the lack of closed-form formula and the cost of numerical approximation: our method uses Gamma distributions with a fixed rate parameter and a special step to choose this parameter is added in the algorithm. Since it converges locally and is computationally faster than an Expectation-Maximization method for Gamma mixture models, our method can be used beneficially as a drop-in replacement in any application using this kind of statistical models.

1 Introduction and prior work

Statistical mixtures are among the most used tools in many applications which require to model experimental data with probability distributions. Such a mixture is a weighted sum of components which are themselves probability distributions (usually the same distribution is shared by all the components):

$$m(x) = \sum_{i=1}^{k} \omega_i p(x; \theta_i) \tag{1}$$

The big challenge here is to learn the parameter vectors ω and θ and the number of components k (we limit us to the case of finite mixtures but some algorithms may output mixtures with an infinite number of components [1]). One of the most famous algorithms to learn the parameters ω and θ is the Expectation-Maximization (EM) algorithm [2].

We address here the problem of learning mixtures of Gamma distributions (see Fig. 1)). Although not as common as Gaussian mixture models, Gamma mixtures are of interest in many applications as various as bioinformatics [3], communication networks modeling [4] or health services analysis [5] and a lot of work has been devoted to these mixtures.

Our new algorithm is an extension of the k-Maximum Likelihood estimator (k-MLE) algorithm by Nielsen [6]. It relies on the same principle which was already used for mixtures of generalized Gaussians [7]. Our contribution is to

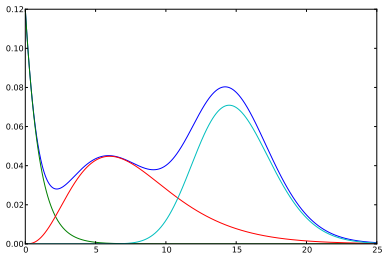


Fig. 1. A mixture of Gamma distributions with 3 components: $\omega_1 = 0.12, \alpha_1 = 1, \beta_1 = 1$; $\omega_2 = 0.4, \alpha_2 = 4, \beta_2 = 2$; $\omega_3 = 0.48, \alpha_3 = 30, \beta_3 = 0.5$.

provide a new algorithm for Gamma mixtures which is faster than methods based on Expectation-Maximization

Since the studied method relies on the exponential families framework, the necessary background about exponential families is recalled and we show that Gamma distributions are members of the exponential families. After a description of two algorithms designed to learn mixtures of exponential families, Bregman Soft Clustering, which relies on EM and k-MLE, we explain why they are not well suited for the particular case of Gamma mixtures. In the following section we present our extension of k-MLE which allows to efficiently learn mixtures of Gamma distributions. In the last section we evaluate the effectiveness of our proposed algorithm both in terms of computational cost and in terms of quality of the produced models.

2 Exponential families and their parametrizations

2.1 Definition

Exponential families are a widespread class of distributions and many commonly used distributions belong to this class (with the notable exception of the uniform distribution): for example Gaussian, Beta, Gamma, Rayleigh, Von Mises are all members of this class ([8] provides a vast list of exponential families with their decomposition). An exponential family is a set of probability mass or probability density functions which admits the following canonical decomposition:

$$p(x;\theta) = \exp(\langle t(x), \theta \rangle - F(\theta) + k(x)) \tag{2}$$

with

- $t(x)$ the sufficient statistic,
- θ the natural parameters,
- $\langle \cdot, \cdot \rangle$ the inner product,

– F the log-normalizer,
– $k(x)$ the carrier measure.

The log-normalizer characterizes the exponential family and is equal to:

$$F(\theta) = \log \int_x \exp(\langle t(x), \theta \rangle + k(x))\, dx \tag{3}$$

Since this log-normalizer F is a strictly convex and differentiable function, it admits a dual representation by the Legendre-Fenchel transform:

$$F^\star(\eta) = \sup_\theta \{\langle \theta, \eta \rangle - F(\theta)\} \tag{4}$$

We get the maximum for $\theta = (\nabla F)^{-1}(\eta)$ and $F^\star$ can be computed with:

$$F^\star(\eta) = \langle \eta, \theta \rangle - F(\theta) \tag{5}$$

Thus we deduce that the gradient of F and of its dual $F^\star$ are inversely reciprocal:

$$\nabla F = (\nabla F^\star)^{-1} \tag{6}$$

The duality between F and its Legendre transform $F^\star$ leads to a new parametrization for the exponential families, which is the dual of the natural parameters: the so-called expectation parameters $\eta = \nabla F(\theta)$. The parameters η are called expectation parameters since $\eta = E[t(x)]$ [8].

In the general case, the dual $F^\star$ may be not known in closed-form and thus may require numerical approximation (which is time consuming and submitted to various practical problems like the choice of the initialization for an iterative procedure).

2.2 Bregman divergences

Bregman divergences are a family of divergences parametrized by the set of strictly convex and differentiable functions and is written as:

$$B_F(p\|q) = F(p) \; - \; F(q) \; - \; \langle p \; - \; q, \; \nabla F(q) \rangle \tag{7}$$

The function F is called the *generator* of the Bregman divergence.

The family of Bregman divergences generalizes a lot of usual divergences, for example:

– the squared Euclidean distance, for $F(x) = x^2$,
– the Kullback-Leibler (KL) divergence, with the Shannon negative entropy $F(x) = \sum_{i=1}^{d} x_i \log x_i$ (also called Shannon information).

2.3 Bijection between exponential families and Bregman divergences

Banerjee *et al.* [9] showed that Bregman divergences are in bijection with the exponential families through the generator F. For each exponential family with a log-normalizer F there is one and only one Bregman divergence whose generator is $F^\star$, the Legendre dual of F. We can rewrite the exponential family in terms of the corresponding Bregman divergence:

$$p(x;\theta) = \exp(\langle t(x), \theta\rangle - F(\theta) + k(x)) \tag{8}$$
$$= \exp(-B_{F^\star}(t(x)\|\eta) + F^\star(t(x)) + k(x)) \tag{9}$$

where η is the expectation parameter of the family ($\eta = \nabla F(\theta)$).

This bijection allows in particular to compute the Kullback-Leibler divergence between two members of the same exponential family:

$$\mathrm{KL}\left(p(x,\theta_1); p(x,\theta_2)\right) = \int_x p(x;\theta_1)\log\frac{p(x;\theta_1)}{p(x;\theta_2)}\,dx \tag{10}$$
$$= B_F(\theta_2\|\theta_1) \tag{11}$$

where F is the log-normalizer of the exponential family and the generator of the associated Bregman divergence.

Thus, computing the Kullback-Leibler divergence between two members of the same exponential family is equivalent to computing a Bregman divergence between their natural parameters (with swapped order).

2.4 Gamma is an exponential family

The general case of the Gamma distribution is

$$p(x;\alpha,\beta) = \frac{\beta^\alpha x^{\alpha-1}\exp(-\beta x)}{\Gamma(\alpha)} \tag{12}$$

with $\alpha, \beta > 0$ and x is a positive real number.

The parameter α is called the **shape** parameter and β is called the **rate** parameter (or **inverse scale** parameter). It is common to find another parametrization which replace the rate parameter by the **scale** parameter $\theta = \frac{1}{\beta}$.

This distribution is an exponential family with the following parametrization:

Natural parameters $(\theta_1, \theta_2) = (-\beta, \alpha - 1)$

Sufficient statistics $t(x) = (x, \log x)$

Log normalizer $F(\theta_1, \theta_2) = (-(\theta_2 + 1)\log(-\theta_1) + \log \Gamma(\theta_2 + 1))$

Gradient log normalizer $\nabla F(\theta_1, \theta_2) = \left(\frac{\theta_2+1}{-\theta_1}, -\log(-\theta_1) + \psi(\theta_2 + 1)\right)$

Dual log normalizer $F^\star(\eta_1, \eta_2) = \left\langle (\nabla F)^{-1}(\eta_1, \eta_2), (\eta_1, \eta_2)\right\rangle - F\left((\nabla F)^{-1}(\eta_1, \eta_2)\right)$

Although the log-normalizer F and its gradient ∇F are known in closed-form, it is not the case for its dual $F^\star$ and for the gradient of the dual $\nabla F^\star = (\nabla F)^{-1}$. It thus requires numerical approximation, which is computationally costly.

3 Learning mixtures of exponential families

3.1 Bregman Soft Clustering

The Bregman Soft Clustering for mixtures of exponential families has been introduced in [9]. It is actually a meta-algorithm which takes the considered family as an input of the algorithm and which does not require specific adaptation for each family, contrary to most of the previously proposed methods. As a variant of EM, it still relies on the usual two steps:

Expectation step: The usual Expectation-Maximization algorithm gives us the following formulation for the posterior probabilities:

$$p(i|x_t, \eta) = \frac{\omega_i p(x_t; \eta_i)}{\sum_{j=1}^k \omega_j p(x_t; \eta_j)} \tag{13}$$

Using the bijection between exponential families, we can replace the probability density function of the exponential family by its expression using the associated Bregman divergence.

$$p(i|x_t, \eta) = \frac{\omega_i \exp\left(-B_{F^\star}(t(x_i)\|\eta_i)\right) \exp k(x_t)}{\sum_{j=1}^k \omega_j \exp\left(-B_{F^\star}(t(x_t)\|\eta_j)\right) \exp k(x_t)} \tag{14}$$

$$= \frac{\omega_i \exp\left(-B_{F^\star}(t(x_t)\|\eta_i)\right)}{\sum_{j=1}^k \omega_j \exp\left(-B_{F^\star}(t(x_t)\|\eta_j)\right)} \tag{15}$$

$$\tag{16}$$

Since $B_{F^\star}(p\|q) = F^\star(p) - F^\star(q) - \langle p - q, \nabla F^\star(q)\rangle$ we can expand the expression of the Bregman divergence in the previous expression:

$$p(i|x_t, \eta) = \frac{\omega_i \exp\left(-F^\star(t(x_t)) - F^\star(\eta_i) - \langle t(x_t) - \eta_i, \nabla F^\star(\eta_i)\rangle\right)}{\sum_{j=1}^{k} \omega_j \exp\left(-F^\star(t(x_t)) - F^\star(\eta_j) - \langle t(x_t) - \eta_j, \nabla F^\star(\eta_j)\rangle\right)} \quad (17)$$

$$= \frac{\omega_i \exp\left(F^\star(\eta_i) + \langle t(x_t) - \eta_i, \nabla F^\star(\eta_i)\rangle\right)}{\sum_{j=1}^{k} \omega_j \exp\left(F^\star(\eta_j) + \langle t(x_t) - \eta_j, \nabla F^\star(\eta_j)\rangle\right)} \quad (18)$$

Maximization step: The maximization step is done with the maximum likelihood estimator for exponential families [9]. It can be computed as the average of the sufficient statistics on the observations:

$$\hat{\eta} = E\left[t(x)\right] = \frac{1}{n} \sum t(x_i) \quad (19)$$

Notice that we get an estimate which lives in the space of the expectation parameters. If one wants the associated natural parameter $\hat{\theta} = \nabla F^\star(\hat{\eta})$, the $\nabla F^\star$ function will be needed, either in closed-form or with a numerical approximation (which will be computationally costly).

3.2 *k*-Maximum Likelihood Estimator

Assume we have a set $\mathcal{X} = \{x_1, \ldots, x_n\}$ of n observations which have been sampled from a finite mixture model with k components. The joint probability distribution of theses samples with the missing components z_i (indicating from which component each observation x_i comes from) is:

$$p(x_1, z_1, \ldots, x_n, z_n) = \prod_i p(z_i|\omega) p(x_i|z_i, \theta) \quad (20)$$

Since the variables z_i are not observed in practice, we marginalize these variable and we get:

$$p(x_1, \ldots, x_n|\omega, \theta) = \prod_i \sum_j p(z_i = j|\omega) p(x_i|z_i = j, \theta) \quad (21)$$

The straightforward way to optimize this distribution would be to test the k^n labels but this is not tractable in practice. Instead, Expectation-Maximization optimizes the following quantity, the expected log-likelihood:

$$\bar{l}(x_1, \ldots, x_n) = \frac{1}{n} \log p(x_1, \ldots, x_n) \quad (22)$$

$$= \frac{1}{n} \sum_i \log \sum_j p(z_i = j|\omega) p(x_i|z_i = j, \theta) \quad (23)$$

Contrary to this approach, the k-Maximum Likelihood Estimator maximizes the average complete log-likelihood:

$$\bar{l}'(x_1, z_1, \ldots, x_n, z_n) = \frac{1}{n} \log p(x_1, z_1, \ldots, x_n, z_n) \tag{24}$$

$$= \frac{1}{n} \sum_i \log \prod_j \left((\omega_j p_F(x_i, \theta_j))^{\delta(z_i)} \right) \tag{25}$$

$$= \frac{1}{n} \sum_i \sum_j \delta(z_i) \left(\log p_F(x_i, \theta_j) + \log \omega_j \right) \tag{26}$$

Since p_F is an exponential family, we have:

$$\log p_F(x_i, \theta_j) = -B_{F^*}(t(x), \eta_j) + \underbrace{F^\star(t(x)) + k(x)}_{\text{does not depend on } \theta} \tag{27}$$

The terms which do not depend on θ are of no interest for the maximization problem and can be removed: we can then rewrite Eq. (26) to get the equivalent problem:

$$\arg\min \sum_i \sum_j \delta(z_i) \left(B_{F^*}(t(x), \eta_j) - \log \omega_j \right) \tag{28}$$

As stated in [6] this problem can be solved for a fixed set of weights ω_i using the Bregman k-means algorithm with the Bregman divergence B_{F^*} (actually, any heuristic for k-means is convenient).

The weights can now be optimized by taking $\omega_i = \frac{|\mathcal{C}_i|}{n}$ (where $|C_i|$ is the number of observations put in the cluster $\mathcal{C}_i$ by the solution of the previous clustering problem). This step amounts to maximize the cross-entropy of the mixture [6].

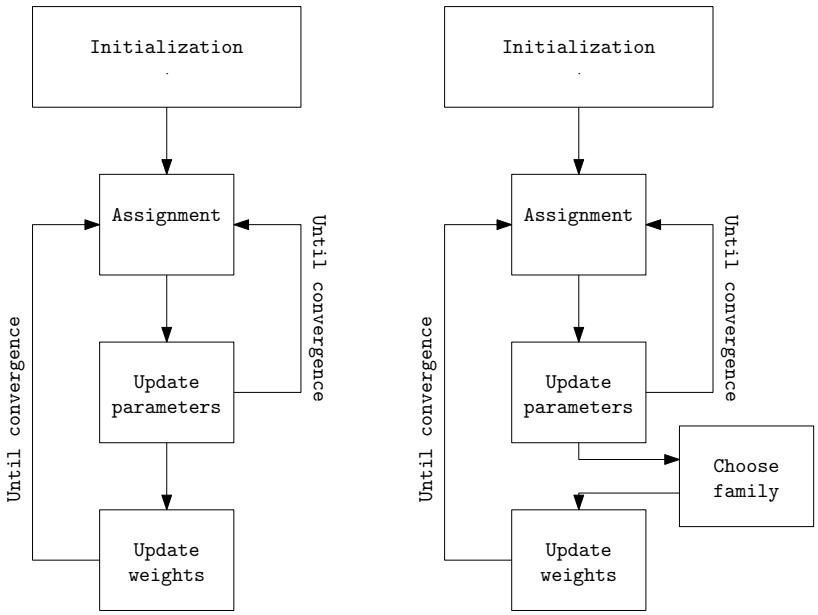
The full algorithm can be summarized as follows (see Fig. 2(a) for a block diagram):

1. **Initialization** (random or using k-MLE ++[6]);
2. **Assignment** $z_i = \arg\max_j \log(\omega_j p_F(x_i|\theta_j))$;
3. **Update** of the η parameters $\eta_i = \frac{1}{n_j} \sum_{x \in \mathcal{C}_j} t(x_i)$;
   Goto step 2 until local convergence;
4. **Update** of the parameters ω_j;
   Goto step 2 until local convergence of the complete likelihood.

4 *k*-MLE for Gamma

4.1 Gamma with fixed rate parameter

The algorithms described in the two previous sections needs frequent conversions between natural parameters θ and expectation parameters η. The bijection between the two parameter spaces uses the functions ∇F and $\nabla F^\star$ which are not



(a) Original k-MLE (b) Extended k-MLE

Fig. 2. Block diagram for the original algorithm and its extension

known in closed-form for the Gamma distribution. Moreover, the evaluation of the Bregman divergence $B_{F^\star}$ is also needed, but the function $F^\star$ is also missing in closed-form. k-MLE may still be applicable to Gamma mixtures but the numerical approximations needed would dramatically reduce the speed of the algorithm, which is one of its main interests [10].

To avoid the computational difficulties for the functions which are not known in closed form, we introduce the Gamma distribution with a fixed rate parameter. The parameter β is not any more a member of the source parametrization and is instead a parameter of the distribution.

$$p_\beta(x;\alpha) = \frac{\beta^\alpha x^{\alpha-1}\exp(-\beta x)}{\Gamma(\alpha)} \tag{29}$$

This is still an exponential family with the following parametrization (a comprehensive list of formulas is given in Table 1):

Natural parameters $\theta = \alpha - 1$

Log normalizer $F(\theta) = -(\theta+1)\log(\beta) + \log\Gamma(\theta+1)$

Gradient log normalizer $\nabla F(\theta) = -\log(\beta) + \psi(\theta + 1)$

Dual log normalizer $F^\star(\eta) = \langle \nabla F^\star(\eta), \eta \rangle - F\left(\nabla F^\star(\eta)\right)$

Gradient of the dual log normalizer $\nabla F^\star(\eta) = (\nabla F)^{-1}(\eta)$

The ∇F function can be inverted in closed-form with respect to the inverse digamma function ψ^{-1}, giving:

$$(\nabla F)^{-1}(\eta) = \psi^{-1}(\eta + \log\beta) - 1 = \nabla F^\star(\eta) \tag{30}$$

We can now compute the $F^\star$ function by directly applying the Legendre transform to the log-normalizer F:

$$F^\star(\eta) = \langle \nabla F^\star(\eta), \eta \rangle - F\left(\nabla F^\star(\eta)\right) \tag{31}$$

$$\begin{aligned} &= \eta\left(\psi^{-1}(\eta + \log\beta) - 1\right) + \psi^{-1}(\eta + \log\beta)\log\beta \\ &\quad - \log\Gamma\left(\psi^{-1}(\eta + \log\beta)\right) \end{aligned} \tag{32}$$

Strictly speaking, this is still not a closed-form but, contrary to the functions we get for the full Gamma distribution, the two missing functions Γ and ψ^{-1} can be computed efficiently: algorithms for the Γ function are well known [11] and ψ^{-1} is numerically well behaved and can be computed efficiently computed with a dichotomic search [3].

4.2 Maximum likelihood estimator

Results from exponential families [9] give an estimator for the expectation parameters of the fixed rate family:

$$\hat{\eta} = \frac{1}{n}\sum t(x_i) = \frac{1}{n}\sum \log(x_i) = -\log\hat{\alpha} + \psi(\beta) \tag{33}$$

By derivation of the likelihood function, we get an estimator for the rate parameter β [4]:

$$\hat{\beta} = \frac{n\hat{\alpha}}{\sum x_i} \tag{34}$$

[3] See `http://hips.seas.harvard.edu/files/invpsi.m` for a working Matlab implementation which can be easily translated in any language

PDF	$p_\beta(x;\alpha) = \frac{\beta^\alpha x^{\alpha-1}\exp(-\beta x)}{\Gamma(\alpha)}$
$\Lambda \to \Theta$	$\theta = \alpha - 1$
$\Theta \to \Lambda$	$\alpha = \theta + 1$
$\Lambda \to H$	$\eta = -\log\beta + \psi(\alpha)$
$H \to \Lambda$	$\alpha = \psi^{-1}(\eta + \log\beta)$
$\Theta \to H$	$\eta = \nabla F(\theta)$
$H \to \Theta$	$\theta = \nabla F^\star(\eta)$
Log normalizer	$F(\theta) = -(\theta+1)\log\beta + \log\Gamma(\theta+1)$
Gradient log normalizer	$\nabla F(\theta) = -\log\beta + \psi(\theta+1)$
Dual log normalizer	$F^\star(\eta) = \eta(\psi^{-1}(\eta+\log\beta)-1) + \psi^{-1}(\eta+\log\beta)\log\beta + \log\Gamma(\psi^{-1}(\eta+\log\beta))$
Gradient dual log normalizer	$\nabla F^\star(\eta) = \psi^{-1}(\eta+\log\beta) - 1$
Sufficient statistic	$t(x) = \log x$
Carrier measure	$k(x) = -\beta x$

Table 1. Gamma distribution with fixed rate as an exponential family

4.3 Learning mixtures

The original k-MLE algorithm builds mixture models where all the components belong to the same exponential family. Although generic Gamma distributions are exponential families, Gamma distributions with fixed rate are not in the **same** exponential family if the rate parameter is not the same across components. In order to build a mixture with a different β parameter for each component, we will follow the approach introduced in [7] (for generalized Gaussian) which adds a supplementary step to the k-MLE procedure (see Fig. 2(b)): before updating the weights, the family of each component is chosen using a maximum likelihood estimator. In the Gamma case, it amounts to choosing the rate parameter of each component, using the MLE given in Eq. (34).

The new k-MLE algorithm for Gamma mixtures (k-MLE-Gamma) can be summarized as follows:

1. **Initialization** (random or using k-MLE ++[6]);
2. **Assignment** $z_i = \arg\max_j \log(\omega_j p_{F_j}(x_i|\theta_j))$;
3. **Update** of the η parameters $\eta_i = \frac{1}{n_j}\sum_{x\in\mathcal{C}_j}\log(x_i)$;
   Goto step 2 until stability (local convergence of the k-means);
4. **Update** of the parameters ω_j and β_j (for all j);
   Goto step 2 until local convergence of the complete likelihood.

4.4 Convergence to a local maximum

As the one proposed for generalized Gaussian, this algorithm converges to a local maximum of the complete log-likelihood. We want to minimize the same cost function as the original k-MLE algorithm, the complete log-likelihood of the mixture, with the slight difference that the log-normalizer is not shared among components but now depends on the values β_j and is now written F_j instead of F:

$$\bar{l}(x_1, z_1, ..., x_n, z_n|w, \theta) = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{k} \delta_j(z_i)(\log p_{F_j}(x_i|\theta_j) + \log \omega_j) \tag{35}$$

$$= \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{k} \delta_j(z_i) \Big(- B_{F_j{}^*}(t(x_i), \eta_j) + {F_j}^*(t(x_i)) + k_j(x_i) + \log \omega_j \Big) \tag{36}$$

Let $\mathcal{C}_j$ be the set of the indices of the observations sampled from the j-th component. Maximizing the log-likelihood $\bar{l}$ is equivalent to minimizing the cost function $-\bar{l}$:

$$\bar{l'} = -\bar{l} = \frac{1}{n} \sum_{j=1}^{k} \sum_{i \in \mathcal{C}_j} U_j(x_i, \eta_j) \tag{37}$$

where

$$U_j(x_i, \eta_j) = -\left(\log p_{F_j}(x_i|\theta_j) + \log \omega_j\right) \tag{38}$$

$$= B_{F_j{}^*}(t(x_i) : \eta_j) - {F_j}^*(t(x_i)) - k_j(x_i) - \log \omega_j \tag{39}$$

is the cost for the observation i to have been sampled from the component j. Notice this cost depends on j since each component has a different generator F_j and a different carrier measure k_j.

This minimization problem can be solved with the Lloyd k-means algorithm [12] using the cost function U (which is not a distance nor a divergence and can even be negative). A proof of the convergence of the Lloyd algorithm for this cost function is given in [7].

After the execution of the Lloyd algorithm, the log-likelihood has been optimized for fixed ω_j and β_j. The final step is to update these two parameters using the proportion of samples in each cluster for the weights and the estimator for β (from Eq. (34)).

5 Expectation-Maximization for Gamma mixtures

Almhana *et al.* [4] proposed a specific variant of Expectation-Maximization for Gamma mixtures. The E step is unchanged compared to the classical EM algorithm, the only changes are in the M step: a specific update step is used for the α and β parameters. We will use this algorithm as a reference in the experiments presented in Section 6.

Maximization step Given the current estimate for the parameters ω, α and β, the new values can be computed with:

$$\omega_i^{(k+1)} = \frac{1}{n} \sum_{t=1}^{n} p(i|x_t, \theta^{(k)}) \tag{40}$$

$$\beta_i^{(k+1)} = \frac{\alpha_i^{(k)} \sum_{t=1}^{n} p(i|x_t, \theta^{(k)})}{\sum_{t=1}^{n} x_t p(i|x_t, \theta^{(k)})} \tag{41}$$

$$\alpha_i^{(k+1)} = \alpha_i^{(k)} + \frac{1}{k}\, G \tag{42}$$

with

$$G = \frac{1}{n} \sum_{t=1}^{n} \left(\log x_t + \log \beta_i^{(k)} - \psi(\alpha_i^{(k)}) \right) p(i|x_t, \theta^{(k)}) \tag{43}$$

6 Experiments

6.1 On synthetic data

The first experiment evaluates the convergence of k-MLE and the convergence of EM on a synthetic example: 15000 observations are sampled from a known three components Gamma mixture and the two evaluated methods are used to estimate Gamma mixture models with three components. We draw in Fig. 3 the log-likelihood of each mixture at each iteration of the two algorithms. Although the goal of k-MLE is to maximize the complete log-likelihood (Eq. (24)) and not the log-likelihood (Eq. (22) we see that both algorithms converge to a (local) maximum of the log-likelihood. Moreover k-MLE provides better results and converges way faster than EM.

6.2 On a real dataset

The second experiment describes experimental results on a real dataset which collects distances between atoms inside RNA molecules in order to predict the 3D structure of these molecules. Gaussian mixture models were successfully used to model the density of these distance [13] [14] but since the observations are

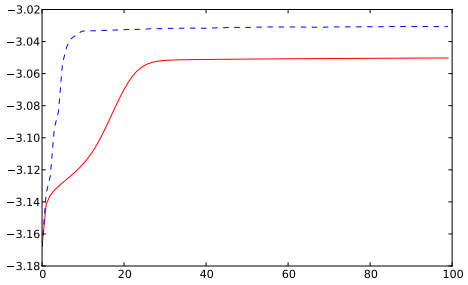
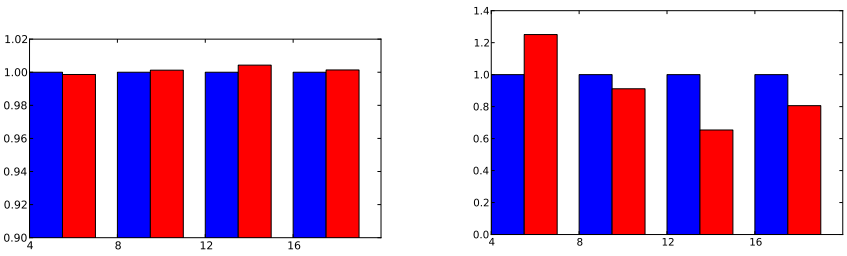


Fig. 3. Log-likelihood with respect to the number of components for k-MLE (dashed curve) and EM (plain curve). Higher curve (k-MLE model) means better model.



(a) Log-likelihood ratio (b) Time ratio

Fig. 4. Log-likelihood and computation time ratios for k-MLE (right red bars) and EM (left blue bars) with respect to the number of components in the mixture. EM is our reference for comparison and thus has the score 1.

intrinsically positive a mixture model with a positive support (remember that Gaussian distribution is defined on $\mathbb{R}$ whereas the Gamma distribution is defined on $\mathbb{R}_+$) would be more statistically meaningful.

Fig. 4 presents results on this dataset, in terms of log-likelihood and computation time with respect to the number of components in the mixture (4, 8, 12 and 16 components). Since absolute value for likelihood and time are difficult to compare meaningfully, we plot the mean ratio between the values got with k-MLE and the one got with EM (which is our reference for comparison and represented by 1 on the graphics). We see that k-MLE for Gamma mixtures performs similarly (or even better) to EM for Gamma mixtures for the quality of the built models and outperforms EM for the computation time (between 10% and 40%). The only case where k-MLE is worse than EM is for 4 components: k-MLE seems to be less robust when the number of components is not enough to model accurately the observations.

7 Conclusion

We presented a new algorithm for mixtures of Gamma distributions which is both fast and accurate. Accuracy is important since it means that the quality of the produced models (and thus the performances in the considered applications) will not decrease: our new algorithm could thus be considered as a drop-in replacement for other Gamma mixtures algorithms. The faster speed not only means that the computation time will decrease in applications where Gamma mixtures are already used but also that these mixtures will become of new interest in areas where the use of the Gamma distribution was theoretically interesting but not feasible in practice due to the high computation time. Moreover, this new extension of the k-Maximum Likelihood estimator shows the power and the genericity of the method which allows interesting perspectives for new and unexplored kinds of mixtures.

Acknowledgments Thanks to the anonymous referees for their insightful comments and their careful proofreading.

References

1. Neal, R.M.: Markov chain sampling methods for dirichlet process mixture models. Journal of Computational and Graphical Statistics **9**(2) (2000) 249–265
2. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society. Series B (Methodological) (1977) 1–38
3. Mayrose, I., Friedman, N., Pupko, T.: A Gamma mixture model better accounts for among site rate heterogeneity. Bioinformatics **21**(Suppl 2) (2005)
4. Almhana, J., Liu, Z., Choulakian, V., McGorman, R.: A recursive algorithm for gamma mixture models. In: IEEE International Conference on Communications, 2006. ICC '06. Volume 1. (June 2006) 197–202

5. Venturini, S., Dominici, F., Parmigiani, G.: Gamma shape mixtures for heavy-tailed distributions. The Annals of Applied Statistics **2**(2) (June 2008) 756–776 Zentralblatt MATH identifier: 05591297; Mathematical Reviews number (MathSciNet): MR2524355.
6. Nielsen, F.: k-MLE: A fast algorithm for learning statistical mixture models. CoRR (2012)
7. Schwander, O., Schutz, A.J., Nielsen, F., Berthoumieu, Y.: k-MLE for mixtures of generalized Gaussians. In: 2012 21st International Conference on Pattern Recognition (ICPR). (November 2012) 2825–2828
8. Nielsen, F., Garcia, V.: Statistical exponential families: A digest with flash cards. CoRR **09114863** (2009)
9. Banerjee, A., Merugu, S., Dhillon, I.S., Ghosh, J.: Clustering with Bregman divergences. The Journal of Machine Learning Research **6** (2005) 1705–1749
10. Nielsen, F.: K-MLE: A fast algorithm for learning statistical mixture models. In: 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). (March 2012) 869–872
11. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: Numerical recipes 3rd edition: The art of scientific computing. Cambridge University Press (2007)
12. Lloyd, S.P.: Least squares quantization in PCM. IEEE Transactions on Information Theory **28**(2) (1982) 129–137
13. Sim, A.Y., Schwander, O., Levitt, M., Bernauer, J.: Evaluating mixture models for building rna knowledge-based potentials. Journal of Bioinformatics and Computational Biology **10**(02) (2012)
14. Bernauer, J., Huang, X., Sim, A.Y., Levitt, M.: Fully differentiable coarse-grained and all-atom knowledge-based potentials for rna structure evaluation. RNA **17**(6) (2011) 1066–1075