k-MLE for mixtures of generalized Gaussians

Olivier Schwander
École Polytechnique
Palaiseau, France
schwander@lix.polytechnique.fr

Frank Nielsen
Sony Computer Science Laboratories, Inc
Tokyo, Japan
Frank.Nielsen@acm.org

Aurélien J. Schutz
Univ. Bordeaux / IPB
Groupe Signal, IMS
aurelien.schutz@ims-bordeaux.fr

Yannick Berthoumieu
Univ. Bordeaux / IPB
Groupe Signal, IMS
yannick.berthoumieu@ims-bordeaux.fr

Abstract

We introduce an extension of the k-MLE algorithm, a fast algorithm for learning statistical mixture models relying on maximum likelihood estimators, which allows to build mixture of generalized Gaussian distributions without a fixed shape parameter. This allows us to model finely probability density functions which are made of highly non Gaussian components. We theoretically prove the local convergence of our method and show experimentally that it performs comparably to Expectation-Maximization methods while being more computationally efficient.

1. Introduction

Mixture models are fruitful and universal tools for estimating unknown densities. The most common mixtures are the mixtures of Gaussians but a lot of work have been devoted to other kinds of distributions. Nevertheless the choice of the underlying component distribution is often difficult. Once the family of distributions for the components has been chosen, the task is to learn the parameters (ω_i, θ_i) of the mixture $\sum_{i=1}^{k} \omega_i p(x;\theta_i)$. The most famous method is the Expectation-Maximization algorithm which is designed to maximize the expected complete likelihood of the mixture. This algorithm is well-known for the celebrated Gaussian mixture models, and some variants have been proposed to learn mixtures of various types of distributions, including generalized Gaussian [1] and Laplace laws [5]. One of the most promising extension is the Bregman Soft Clustering algorithm [2] which allows to use any exponential family for the underlying distribution. In recent work, the k-Maximum Likelihood estimator [8] has been proposed: this algorithm relies on the maximum likelihood estimator of the underlying distribution and on k-means like update steps to maximize the complete log-likelihood function, but the method was first described for the case where all the components belong to the same exponential family, which is not the case of the generalized Gaussians with varying shape parameter.

The paper is organized as follows: Section 2 recalls basics on generalized Gaussian distributions; Section 3 presents the original k-MLE method [8]. Section 4 describes the extension to generalized Gaussians and proves the local convergence. Section 5 presents some experimental results on synthetic data.

2. Generalized Gaussian

2.1. Definition

The generalized Gaussian (GG) replaces the square in the usual Gaussian distribution by a parameter β. This family thus contains the normal law ($\beta = 2$), the Laplace law ($\beta = 1$) and even the uniform law in the limit case ($\beta \to +\infty$).

$$f(x;\mu,\alpha,\beta) = \frac{\beta}{2\alpha\Gamma(1/\beta)} \exp\left(-\frac{|x-\mu|^\beta}{\alpha}\right) \quad (1)$$

with $\alpha > 0$ (the scale parameter) and $\beta > 0$ (the shape parameter).

Generalized Gaussian distributions have been successfully used in problems of texture classification [4, 3] and mixtures of these distributions have been used

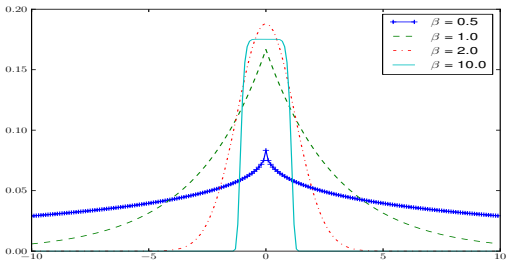

Figure 1. Probability densities of generalized Gaussians for various shape parameters β.

for image/video segmentation [1]. We focus here on one-dimensional distributions but a multivariate generalized Gaussian can be written as a product of such one-dimensional laws.

A member of an exponential family [9] admits the following canonical decomposition:

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

with $t(x)$ the sufficient statistic, θ the natural parameters, F the log-normalizer, and $k(x)$ the carrier measure. Generalized Gaussian are exponential families for both fixed μ and β with the parameters: $t(x) = -|x-\mu|^\beta$, $\theta = \alpha^{-\beta}$, $F(\theta) = -\beta\log(\theta) + \log\left(\frac{\beta}{2\Gamma(1/\beta)}\right)$, and $k(x) = 0$.

2.2. Maximum likelihood estimator (MLE)

There is no maximum likelihood estimator known in closed-form but [4] proposed a numerical scheme to estimate the parameters α and β (with $\mu = 0$). Using results from the framework of exponential families, we can estimate the expectation parameters $\eta = \nabla F^*(\theta) = -\frac{1}{\theta} = \sum_{i=1}^N t(x_i)$ (where F^* is the Legendre dual of F, see [9] for more details).

For a given θ, β can be estimated as the solution of the equation:

$$1 + \frac{\psi(1/\beta)}{\beta^2} - \frac{\theta}{N}\sum_{i=1}^N |x_i|^\beta \log\left(\theta|x_i|^\beta\right) = 0 \qquad (3)$$

This can be solved using the Newton-Raphson method initialized by a dichotomic search between $\beta = 0$ and $\beta = 20$ (for a high enough β the generalized Gaussian law is very close to a uniform law). In some applications, it may be worth limiting the estimation to this dichotomic search in order to reduce the computation time.

Notice that in the case $\mu \neq 0$, it is sufficient to translate the observation and to learn the parameters for $y_i = x_i - \hat{\mu}$ with $\hat{\mu} = \frac{1}{N}\sum_{i=1}^N x_i$.

3. k-Maximum Likelihood estimators

Mixture models are traditionally learned using the expectation-maximization (EM) soft clustering technique that monotonically increases the *expected incomplete likelihood*. Given prescribed mixture weights, the hard clustering k-MLE algorithm iteratively assigns data to the most likely weighted component and updates the component models using Maximum Likelihood Estimators. After this step, which can be reduced to a k-means problem, the weights are updated. The algorithm loops until it reaches a maximum of the *complete log-likelihood* [8]:

$$\bar{l}(\{x_i, z_i\}_{i=1}^n \ldots | w, \theta) = \frac{1}{n}\sum_{i=1}^n \log \prod_{j=1}^k (\omega_j p_F(x_i|\theta_j))^{\delta_j(z_i)}$$

$$= \frac{1}{n}\sum_{i=1}^n \sum_{j=1}^k \delta_j(z_i)(\log p_F(x_i|\theta_j) + \log\omega_j)$$

where $\omega_i \geq 0$ is the weight of the i-th component ($\sum \omega_i = 1$, the z_i are the missing component labels and $\delta_j(z_i) = 1$ if and only if x_i has been sampled from the j-th component, and 0 otherwise.

4. k-MLE with component-wise families

4.1. Algorithm

The original k-MLE algorithm [8] was described to learn mixture models where all the components belong to the same exponential family. Although the generalized Gaussian distribution is an exponential family this is only true for a fixed shape parameter β. This is interesting by itself since it would allow to build a mixture of Gaussian distributions (for $\beta = 2$) or a mixture of Laplace distribution (for $\beta = 1$) but it does not exploit the full power of generalized Gaussian mixtures. We present here an extension of the k-MLE algorithm which allows to learn a different shape parameter β for each component. We minimize the same cost function, the complete log-likelihood of the mixture:

$$\bar{l}(x_1, z_1, ..., x_n, z_n | w, \theta) = \frac{1}{n}\sum_{i=1}^n \sum_{j=1}^k \delta_j(z_i)(\log p_{F_j}(x_i|\theta_j) + \log\omega_j) \qquad (4)$$

Using the bijection between exponential families and dual Bregman divergences [2] which states that

$\log p_{F_j}(x|\theta_j) = -B_{F_j{}^*}(t(x) : \eta_j) + {F_j}^*(t(x)) + k(x)$, where $\eta_j = \nabla F_j(\theta_j)$ is the moment parameterization of the j-th component exponential family distribution, we mathematically rewrite the log-likelihood as:

$$\begin{aligned}\bar{l} = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{k}\delta_j(z_i)\left(-B_{F_j{}^*}(t(x_i) : \eta_j)\right. \\ \left. + {F_j}^*(t(x_i)) + k_j(x_i) + \log\omega_j\right)\end{aligned} \tag{5}$$

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

$$\bar{l}' = -\bar{l} = \frac{1}{n}\sum_{j=1}^{k}\sum_{i\in\mathcal{C}_j} U_j\left(x_i, \eta_j\right) \tag{6}$$

where

$$U_j(x_i, \eta_j) = -\left(\log p_{F_j}(x_i|\theta_j) + \log\omega_j\right) \tag{7}$$

$$\begin{aligned}= B_{F_j{}^*}(t(x_i) : \eta_j) - {F_j}^*(t(x_i)) \\ - k_j(x_i) - \log\omega_j\end{aligned} \tag{8}$$

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.

The problem of the minimization of the cost function $-\bar{l}$ can be seen as the resolution of a generalized k-means problem for the costs D_j (this is not a distance nor a divergence, and it can even be negative). The resolution of this problem with the Lloyd algorithm is discussed in Subsection 4.2.

The k-means problem has been solved for fixed ω_j and for fixed β_j (thus for fixed F_j and k_j): we can improve the log-likelihood by maximizing over these two parameters. After the iterations of the Lloyd algorithm the log-likelihood is equal to:

$$\begin{aligned}\bar{l} = \sum_{j=1}^{k}\gamma_j\left(\sum_{i=1}^{n} -B_{F_j{}^*}(t(x_i) : \eta_j)\right) \\ + \sum_{j=1}^{k}\gamma_j\left(\sum_{i=1}^{n} {F_j}^*(t(x_i)) + k_j(x_i) + \log\omega_j\right)\end{aligned} \tag{9}$$

where $\gamma_j = \frac{\delta_j(z_i)}{n}$ is the proportion of observations which have been assigned to cluster j.

The first term of this sum has been optimized by the Lloyd algorithm so we can focus on the second term. The ω_j parameters should be chosen in order to maximize the cross-entropy quantity [8] $\sum_{j=1}^{k}\gamma_j\log\omega_j$ which reaches its maximum for $\omega_j = \gamma_j$ for all j.

The remaining term $\sum_{j=1}^{k}\gamma_j(\sum_{i=1}^{n}{F_j}^*(t(x_i)) + k_j(x_i))$ can be maximized by taking well-chosen F_j and k_j on each cluster. In the case of the generalized Gaussian distributions this amounts to finding the best μ_j and β_j parameters (since the α_j are contained within the expectation parameters η_j, they have been fixed during the k-means step). This can be done by using the maximum likelihood estimator for the shape and location parameters of the generalized Gaussian:

$$\mu_j = \sum_{i\in\mathcal{C}_j} x_i, \qquad \beta_j = \text{ the estimator in Eq. 3} \tag{10}$$

The full algorithm can be summarized as follows:

1. **Initialization** (random or using k-MLE++[8]);
2. **Assignment** $z_i = \arg\min_j \log(\omega_j p_{F_j}(x_i|\theta_j))$;
3. **Update** of the η parameters $\eta_i = \frac{1}{n_j}\sum_{x\in\mathcal{C}_j} t(x_i)$; **Goto** step 2 until local convergence;
4. **Update** of the parameters ω_j, μ_j, β_j; **Goto** step 2 until local convergence of the complete likelihood.

4.2. Local convergence of the k-means

We now prove that the widespread Lloyd method allows to find a local minimum by decreasing monotonically the cost function. The Lloyd method iterates over two main steps: **assignment** and **centroid updates**. The sketch of the proof is very similar to the usual proof, but we emphasize the fact the costs U_j are not distances nor divergences and may even be negative.

Let us denote by $\mathcal{C}_i^{(t)}$ the content of the cluster i at the t-th iteration and by $\eta_i^{(t)}$ the center of the cluster i at the t-th iteration. The cost function at the iteration t is:

$$\bar{l}'^{(t)} = \frac{1}{n}\sum_{j=1}^{k}\sum_{i\in\mathcal{C}_j^{(t)}} U_j\left(x_i, \eta_j^{(t)}\right) \tag{11}$$

The assignment step allocates each point x_i to the center η_j which minimizes the cost $U_j(x_i, \eta_j)$, so we have:

$$\bar{l}'^{(t)} \leq \frac{1}{n}\sum_{j=1}^{k}\sum_{i\in\mathcal{C}_j^{(t+1)}} U_j\left(x_i, \eta_j^{(t)}\right) \tag{12}$$

For each cluster, the centroid update step must solve the optimization problem $\eta_j^{(t+1)} = \arg\min_{\eta_j^*}\sum_{i\in\mathcal{C}_j^{(t+1)}} U_j\left(x_i, \eta_j^*\right)$ where the cost function is:

$$\sum_{i\in\mathcal{C}_j^{(t+1)}} B_{F_j{}^*}(t(x_i) : \eta_j) - {F_j}^*(t(x_i)) - k_j(x_i) - \log\omega_j$$

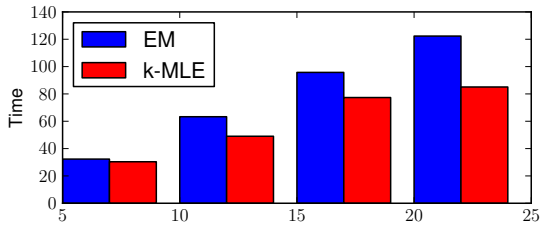

Figure 2. Computation time wrt. k

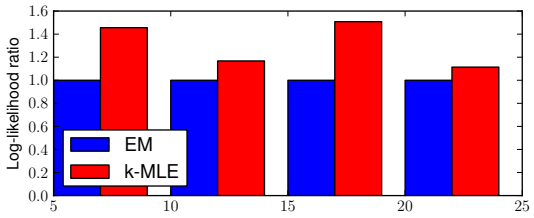

Figure 3. Ratio between the log-likelihood of EM and of k-MLE wrt. k

By removing the constant terms $-{F_j}^*(t(x_i)) - k_j(x_i) - \log \omega_j$ which does not depend on the centroids $\eta_i^{(t)}$ the original problem becomes equivalent to the following problem: $\eta_j^{(t+1)} = \arg\min_{\eta^*} \sum_{i \in \mathcal{C}_j^{(t+1)}} B_{{F_j}^*}(t(x_i) : \eta_j)$ which is the problem of the computation of a right-sided Bregman centroid [7]: this centroid is known in closed-form [2] $\eta_j^{(t+1)} = \sum_{i \in \mathcal{C}_j^{(t+1)}} t(x_i)$. Since the centroid update minimizes the average cost to each centroid, we now have:

$$\frac{1}{n} \sum_{j=1}^{k} \sum_{i \in \mathcal{C}_j^{(t+1)}} U_j\left(x_i, \eta_j^{(t+1)}\right) \leq \frac{1}{n} \sum_{j=1}^{k} \sum_{i \in \mathcal{C}_j^{(t+1)}} U_j\left(x_i, \eta_j^{(t)}\right) \tag{13}$$

Combining the equations (12) and (13) we get a global inequality which characterizes the monotonic decrease of the cost function.

5. Experiments on synthetic data

We randomly generate mixtures of 5 generalized Gaussians. The parameters are chosen according to a uniform law. To draw observations, random samples from generalized Gaussian are generated using the method described in [6]. We learn mixtures using the classical Gaussian EM and our generalized Gaussian k-MLE. Figure (3) shows the expected improvement in terms of log-likelihood by not using Gaussian to learn a non Gaussian mixture. Figure (2) shows the computation time: the improvement of the log-likelihood has been done at no cost, showing the computational efficiency of our algorithm.

6. Conclusion

We described a provably locally converging method to learn mixtures of generalized Gaussian distribution without a fixed shape parameter. This method extends the algorithm k-MLE [8] to the case where the components of the mixture models do not belong to the same exponential family. Experiments show that this extension competes favorably with the classical EM since it allows to build more precise mixtures without an increase of the computational cost. Moreover, the method is not limited to generalized Gaussians and a direct extension would be to replace the β updates step by a choice among a predefined set of exponential families.

References

[1] M. Allili, D. Ziou, N. Bouguila, and S. Boutemedjet. Image and video segmentation by combining unsupervised generalized gaussian mixture modeling and feature selection. *IEEE Trans. Circuits and Systems for Video Technology*, 20(10):1373–1377, 2010.

[2] A. Banerjee, S. Merugu, I. Dhillon, and J. Ghosh. Clustering with Bregman divergences. *Journal of Machine Learning Research*, 6:1705–1749, 2005.

[3] S.-K. Choy and C.-S. Tong. Supervised texture classification using characteristic generalized gaussian density. *Journal of Mathematical Imaging and Vision*, 29:35–47, 2007.

[4] M. Do and M. Vetterli. Wavelet-based texture retrieval using generalized gaussian density and Kullback-Leibler distance. *IEEE Trans. Image Processing*, 11(2):146–158, 2002.

[5] N. Mitianoudis and T. Stathaki. Overcomplete source separation using Laplacian mixture models. *IEEE Signal Processing Letters*, 12(4):277–280, 2005.

[6] M. Nardon and P. Pianca. Simulation techniques for generalized gaussian densities. *Journal of Statistical Computation and Simulation*, 79(11):1317–1329, 2009.

[7] F. Nielsen and R. Nock: Sided and symmetrized Bregman centroids. IEEE Transactions on Information Theory, 55(6):2882-2904, 2009.

[8] F. Nielsen. k-MLE: A fast algorithm for learning statistical mixture models. In *International Conference on Acoustics, Speech and Signal Processing*, *CoRR*, 1203.5181, 2012.

[9] F. Nielsen and V. Garcia. Statistical exponential families: A digest with flash cards. *CoRR*, 09114863, 2009.