Apprentissage rapide de modèles de mélanges avec k-MLE et ses extensions

Olivier SCHWANDER[1], Frank NIELSEN[2]

[1]École Polytechnique
Palaiseau, France

[2]Sony Computer Science Laboratories, Inc
Tokyo, Japan
schwander@lix.polytechnique.fr, nielsen@lix.polytechnique.fr

Résumé – Les modèles de mélange sont parmi les outils les plus répandus pour estimer des densités de probabilité inconnues. Certains algorithmes se concentrent sur un type particulier de distributions, d'autres se veulent très génériques, comme k-MLE qui permet de produire des mélanges de familles exponentielles. Nous proposons ici une extension de k-MLE qui permet d'apprendre de façon efficace des mélanges de Gaussiennes généralisées, qui ne sont pas des familles exponentielles dans leur cas le plus général, et des mélanges de lois Gamma, qui sont bien des familles exponentielles mais pour lesquelles l'absence de formules closes dans certains cas rendrait l'algorithme original peu efficace.

Abstract – Mixture models are among the most used tools for modeling unknown probability densities. Some algorithms focus on a particular type of distributions, others are generic like k-MLE which allows to build mixtures of exponential families. We introduce here an extension of k-MLE which allows to efficiently produce mixtures of generalized Gaussian, which are not exponential families in the general case, and mixtures of Gamma laws, which are exponential families but for which the lack of closed-form formula in some cases makes the original algorithm inefficient.

1 Introduction et motivation

Les modèles de mélange sont parmi les outils les plus répandus pour l'estimation de densités inconnues. Parmi tout ceux-ci, les mélanges de Gaussiennes sont certainement les plus employés, mais beaucoup de travaux ont été consacrés à d'autres sortes de mélanges (mélanges de lois Rayleigh, de lois Laplace, de lois Gamma [2], de Gaussiennes généralisées [1] ou de familles exponentielles en général [3]). On s'intéresse ici à des mélanges de Gaussiennes généralisées qui ont déjà été utilisées avec succès dans des travaux sur la classification de textures [1] et à des mélanges de loi Gamma (voir Figure 4) qui sont utilisées dans des applications variées, par exemple en bio-informatique [7] [12] ou pour la modélisation de réseaux [12].

Nous rappelons dans un premier temps le principe de l'algorithme k-Maximum de vraissemblance [8] (k-MLE) qui permet d'apprendre des mélanges de familles exponentielles. Contrairement à EM qui est souvent qualifié de *soft clustering*, il peut être vu comme une méthode de *hard clustering* et présente l'avantage d'être beaucoup plus rapide qu'EM. Construire un modèle de mélange avec une méthode de clustering n'est pas une idée nouvelle en soi [5] [4], des idées similaires ont également été utilisées pour des modèles de Markov cachés, dont les modèles de mélange sont un cas particulier [6]. La principale contribution de k-MLE est de fournir un cadre générique adapté à toutes les familles exponentielles.

Nous présentons ensuite deux extensions de cet algorithme, qui ont pour but de l'appliquer à des mélanges de distributions difficiles à manipuler directement avec k-MLE : la première permet d'apprendre des mélanges de Gaussiennes généralisées [10] (qui ne sont pas des familles exponentielles dans le cas général) ; la deuxième permet d'apprendre des mélanges de lois Gamma (qui sont bien des familles exponentielles mais pour lesquelles l'absence de formes closes pour certaines fonctions de la décomposition en famille exponentielle rend impossible en pratique l'application directe de k-MLE). Ces deux extensions sont ensuite comparées à EM (dans une version pour les Gaussiennes généralisées, et dans une version pour les lois Gamma) en termes de log-vraisemblance et de temps de calcul.

2 Algorithme k-MLE

2.1 Familles exponentielles

Les familles exponentielles forment une classe de lois de probabilités admettant la décomposition canonique suivante :

$$p_F(x;\theta) = \exp(\langle t(x), \theta\rangle - F(\theta) + k(x)) \qquad (1)$$

avec $t(x)$ la statistique suffisante, θ les *paramètres naturels*, $\langle\cdot,\cdot\rangle$ un produit scalaire, F le *générateur*, $k(x)$ le coefficient de normalisation. Le générateur F caractérise la famille exponentielle ; c'est une fonction strictement convexe et dérivable.

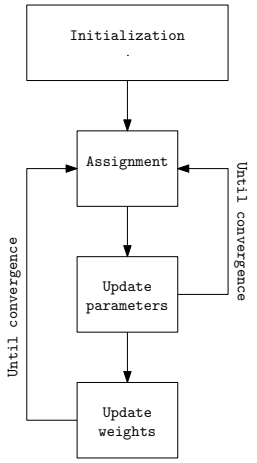


FIGURE 1 – Étapes de l'algorithme k-MLE

Une famille exponentielle admet deux types de paramétrisation : les paramètres naturels de la décomposition donnée ci-dessus et une paramétrisation duale, appelée *paramètres d'espérance* η. Ces deux espaces de paramètres sont en bijection à travers les fonctions ∇F et $\nabla F^\star$ (où $F^\star$ est le dual de F par la transformée de Legendre) : on a $\eta = \nabla F(\theta)$ et $\theta = \nabla F(\eta)$.

2.2 Log-vraisemblance complète

Pour un ensemble de n observations x_i provenant d'un mélange à k composantes, la loi de probabilité jointe de ces observations avec les variables cachées z_i qui indiquent de quelle composante ils proviennent s'écrit :

$$p(x_1, z_1, \ldots, x_n, z_n) = \prod_i p(z_i|\omega)p(x_i|z_i, \theta) \tag{2}$$

EM maximise la log-vraisemblance moyenne qui, après marginalisation des variables inconnues z_i, s'écrit :

$$\bar{l} = \frac{1}{n}\sum_i \log \sum_j p(z_i = j|\omega)p(x_i|z_i = j, \theta) \tag{3}$$

k-MLE, au contraire, maximise la log-vraisemblance complète moyenne :

$$\bar{l}' = \frac{1}{n}\log p(x_1, z_1, \ldots, x_n, z_n) \tag{4}$$

$$= \frac{1}{n}\sum_i \sum_j \delta(z_i)\left(\log p_F(x_i, \theta_j) + \log \omega_j\right) \tag{5}$$

2.3 Algorithme

La log-vraisemblance complète moyenne est optimisée par k-MLE en deux étapes [8] : avec des poids ω_j fixés, on cherche les meilleurs θ_j (cette opération est équivalente à résoudre un problème de clustering, par exemple avec l'algorithme des k-moyennes) ; les poids ω_j sont ensuite mis à jour étant donnés les θ_j calculés à l'étape précédente. Ce procédé donne l'algorithme suivant (résumé Figure 1) :

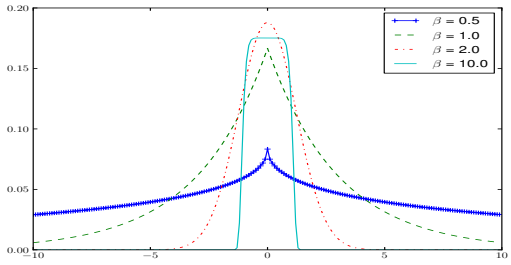


FIGURE 2 – Gaussiennes généralisées pour plusieurs valeurs de β

1. **Initialisation** (aléatoire ou avec k-MLE ++[8]) ;
2. **Assignation** $z_i = \arg\max_j \log(\omega_j p_{F_j}(x_i|\theta_j))$;
3. **Mise à jour des paramètres** $\eta_i = \frac{1}{n_j}\sum_{x\in\mathcal{C}_j}\log(x_i)$; **Retourner à l'étape 2** jusqu'à obtenir la convergence locale du k-means ;
4. **Mettre à jour** les poids ω_j ; **Retourner à l'étape 2** jusqu'à obtenir la convergence de la log-vraisemblance complète.

3 Extensions pour les lois Gamma et les Gaussiennes généralisées

3.1 Gaussiennes généralisées

Dans une Gaussienne généralisée, le carré de la Gaussienne classique est remplacé par un paramètre β, dit paramètre de forme. Cette famille contient bien la loi normale ($\beta = 2$) mais aussi par exemple la loi Laplace ($\beta = 1$). La loi uniforme peut même être vue comme un cas limite ($\beta \to +\infty$). Différents exemples sont présentés Figure 2. Elle s'écrit :

$$f(x; \mu, \alpha, \beta) = \frac{\beta}{2\alpha\Gamma(1/\beta)}\ \exp\left(-\frac{|x-\mu|^\beta}{\alpha^\beta}\right) \tag{6}$$

avec $\alpha > 0$ (*échelle*) and $\beta > 0$ (*forme*).

Lorsque les paramètres μ et β sont fixés, il s'agit d'une famille exponentielle à un seul paramètre α. Les algorithmes existants pour les familles exponentielles permettraient bien d'apprendre un mélange de Gaussiennes généralisées, mais seulement avec des paramètres μ et β partagés entre toutes les composantes, ce qui n'est pas suffisant dans la plupart des cas.

3.2 k-MLE étendu

On peut modifier l'algorithme k-MLE pour rajouter une étape de choix de la famille exponentielle après la mise à jour des poids, ce qui revient à déterminer le paramètre β associé à chaque composante. Ce choix se fait à l'aide d'un estimateur

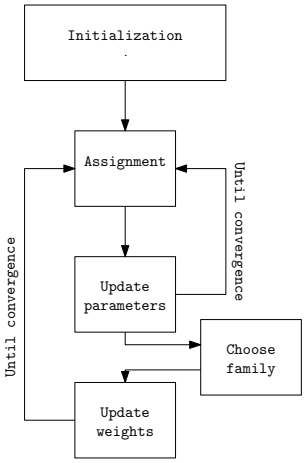


FIGURE 3 – Étape supplémentaire de choix de la famille pour la version étendue de l'algorithme k-MLE

de maximum de vraisemblance sur les points associés au cluster correspondant à chaque composante. On obtient alors l'algorithme suivant (voir Figure 3) :

1. **Initialisation** (aléatoire ou avec k-MLE ++[8]) ;
2. **Assignation** $z_i = \arg\max_j \log(\omega_j p_{F_j}(x_i|\theta_j))$;
3. **Mise à jour des paramètres** $\eta_i = \frac{1}{n_j} \sum_{x \in \mathcal{C}_j} \log(x_i)$;
   Retourner à l'étape 2 jusqu'à obtenir la convergence locale du k-means ;
4. **Mettre à jour** les poids ω_j ;
   Estimer les paramètres β_j ;
   Retourner à l'étape 2 jusqu'à obtenir la convergence de la log-vraisemblance complète.

Cette version étendue converge elle aussi vers un maximum local de la log-vraisemblance complète : la preuve est analogue à celle de k-MLE [8] et est donnée en détails dans [10].

3.3 Lois Gamma

La densité de probabilité d'une loi Gamma peut s'exprimer de la façon suivante :

$$p(x; \alpha, \beta) = \frac{\beta^\alpha x^{\alpha-1} \exp(-\beta x)}{\Gamma(\alpha)} \tag{7}$$

avec α (*échelle*), β (*intensité*) et x des réels positifs.

Il s'agit bien d'une famille exponentielle mais malheureusement, plusieurs formules essentielles pour appliquer k-MLE ou même un Bregman Soft Clustering [3] ne sont pas connues en formule close (notamment $F^\star$ et $\nabla F^\star$). Il serait bien sûr possible d'utiliser quand même ces deux méthodes en utilisant un schéma d'approximation numérique pour ces deux fonctions mais cela réduirait à néant le principal avantage de k-MLE : sa rapidité. Pour se ramener à un cas similaire à celui utilisé pour

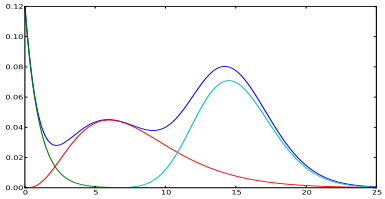


FIGURE 4 – Mélange de Gamma avec 3 composantes : $\omega_1 = 0.12, \alpha_1 = 1, \beta_1 = 1$; $\omega_2 = 0.4, \alpha_2 = 4, \beta_2 = 2$; $\omega_3 = 0.48, \alpha_3 = 30, \beta_3 = 0.5$.

les Gaussiennes généralisées, nous allons d'abord introduire la loi Gamma avec une intensité fixée :

$$p_\beta(x; \alpha) = \frac{\beta^\alpha x^{\alpha-1} \exp(-\beta x)}{\Gamma(\alpha)} \tag{8}$$

On remarque qu'il s'agit toujours d'une famille exponentielle, dont toutes les fonctions sont connues en forme close (l'intensité ne fait plus partie des paramètres source, elle est désormais simplement un paramètre de la distribution) :

- $\theta = \alpha - 1$
- $F(\theta) = -(\theta + 1)\log(\beta) + \log\Gamma(\theta + 1)$
- $\nabla F(\theta) = -\log(\beta) + \psi(\theta + 1)$

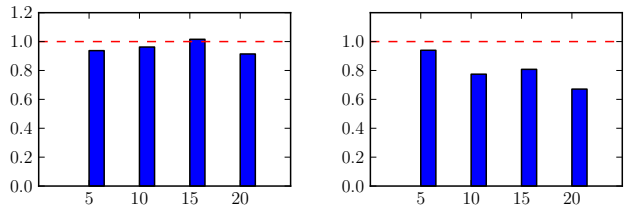
ainsi que

$$F^\star(\eta) = \eta\left(\psi^{-1}(\eta + \log\beta) - 1\right) + \psi^{-1}(\eta + \log\beta)\log\beta - \log\Gamma\left(\psi^{-1}(\eta + \log\beta)\right) \tag{9}$$

Ce cas est donc identique à celui décrit pour les Gaussiennes généralisées : il est possible d'apprendre efficacement un mélange de lois Gamma à intensité fixée où l'intensité est fixée indépendemment pour chaque composante, autrement dit, un mélange de lois Gamma. On a toujours la convergence vers un maximum local de la log-vraisemblance, dont la preuve est décrite dans [9].

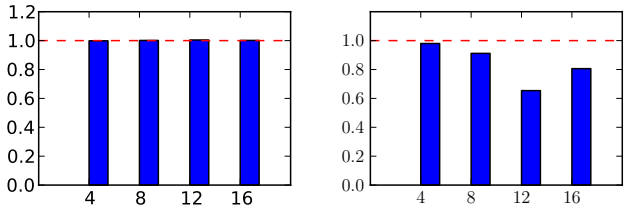
4 Résultats expérimentaux

Les deux algorithmes proposés (une implémentation est disponible à l'adresse `http://www.lix.polytechnique.fr/~schwander/libmef`) sont comparés à EM en termes de temps de calcul et en termes de qualité des modèles produits (en utilisant la log-vraisemblance). La version de k-MLE pour les Gaussiennes généralisées est comparée à l'EM proposé par [1], sur un jeu de données synthétique. Le k-MLE pour les lois Gamma est comparé à la version de EM introduite dans [2], sur un jeu de données réelles de distances entre atomes dans des molécules d'ARN [11]. Pour les quatre algorithmes ainsi considérés, nous évaluons le temps de calcul et la log-vraisemblance



(a) Log-vraisemblance (b) Temps de calcul

FIGURE 5 – Écarts relatifs entre k-MLE et EM pour les mélanges de Gaussiennes généralisées. EM est ici la référence et a donc un score de 1. Les qualités obtenues sont comparables mais le temps de calcul pour k-MLE est toujours meilleur.



(a) Log-vraisemblance (b) Temps de calcul

FIGURE 6 – Écarts relatifs entre k-MLE et EM pour les mélanges de lois Gamma. EM est ici la référence et a donc un score de 1. On retrouve le même comportement que Figure 5 pour k-MLE : même qualité et temps de calcul inférieur.

obtenue en fonction du nombre de composantes du modèle et nous présentons l'écart relatif avec EM qui est à chaque fois notre référence.

On constate tout d'abord sur les figures 5(a) et 6(a) que les deux k-MLE produisent des modèles de qualité comparable : il est intéressant de noter que même si k-MLE n'est pas conçu pour optimiser la même grandeur que EM, il obtient dans les deux cas une valeur de la log-vraisemblance comparable à celle obtenue grâce à EM.

L'apport le plus intéressant de k-MLE est sa rapidité : on voit sur les figures 5(b) et 6(b) que dans les deux cas les temps de calcul obtenus avec k-MLE sont très inférieurs à ceux obtenus avec EM.

5 Conclusion

Nous avons présenté deux méthodes permettant d'appliquer l'idée de l'algorithme k-MLE à des mélanges pour lesquels son utilisation directe était impossible ou peu efficace : sur des Gaussiennes généralisées, qui ne sont pas des familles exponentielles pour tous leurs paramètres ; et sur des lois Gamma dont la décomposition en famille exponentielle n'est pas totalement connue en forme close. Dans les deux cas nos algorithmes ont produit des modèles d'aussi bonne qualité que les méthodes traditionnelles tout en fournissant un gain important sur le temps de calcul.

Références

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

[2] J. Almhana, Z. Liu, V. Choulakian, and R. McGorman. A recursive algorithm for Gamma mixture models. In *IEEE International Conference on Communications, 2006. ICC '06*, volume 1, pages 197–202, June 2006.

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

[4] Arindam Banerjee, Inderjit S Dhillon, Joydeep Ghosh, and Suvrit Sra. Clustering on the unit hypersphere using von Mises-Fisher distributions. *Journal of Machine Learning Research*, 6(2) :1345, 2006.

[5] Michael Kearns, Yishay Mansour, and Andrew Y Ng. An information-theoretic analysis of hard and soft assignment methods for clustering. In *Learning in graphical models*, pages 495–520. Springer, 1998.

[6] Alexey Koloydenko, Meelis Käärik, and Jüri Lember. On adjusted Viterbi training. *Acta Applicandae Mathematicae*, 96(1-3) :309–326, 2007.

[7] I. Mayrose, N. Friedman, and T. Pupko. A Gamma mixture model better accounts for among site rate heterogeneity. *Bioinformatics*, 21(Suppl 2), 2005.

[8] Frank Nielsen. k-MLE : A fast algorithm for learning statistical mixture models. *CoRR*, 2012.

[9] Olivier Schwander and Frank Nielsen. Fast learning of Gamma mixture models with k-MLE. In *SIMBAD*, 2013.

[10] Olivier Schwander, Aurelien J. Schutz, Frank Nielsen, and Yannick Berthoumieu. k-MLE for mixtures of generalized Gaussians. In *2012 21st International Conference on Pattern Recognition (ICPR)*, pages 2825–2828, November 2012.

[11] Adelene YL Sim, Olivier Schwander, Michael Levitt, and Julie Bernauer. Evaluating mixture models for building RNA knowledge-based potentials. *Journal of Bioinformatics and Computational Biology*, 10(02), 2012.

[12] Sergio Venturini, Francesca Dominici, and Giovanni Parmigiani. Gamma shape mixtures for heavy-tailed distributions. *The Annals of Applied Statistics*, 2(2) :756–776, June 2008. Zentralblatt MATH identifier : 05591297 ; Mathematical Reviews number (MathSciNet) : MR2524355.