Évaluation des algorithmes LEM et *e*LEM pour données continues

F.-X. Jollois *, M. Nadif **

*CRIP5, Université de Paris 5,
45 rue des Saint-Pères,
75270 Paris Cedex 06, France
francois-xavier.jollois@univ-paris5.fr
**LITA - UFR MIM, Université de Metz,
Ile du Saulcy,
57045 METZ Cedex 1, France
nadif@iut.univ-metz.fr

Résumé. Très populaire et très efficace pour l'estimation de paramètres d'un modèle de mélange, l'algorithme EM présente l'inconvénient majeur de converger parfois lentement. Son application sur des tableaux de grande taille devient ainsi irréalisable. Afin de remédier à ce problème, plusieurs méthodes ont été proposées. Nous présentons ici le comportement d'une méthode connue, LEM, et d'une variante que nous avons proposée récemment *e*LEM. Celles-ci permettent d'accélérer la convergence de l'algorithme, tout en obtenant des résultats similaires à celui-ci. Dans ce travail, nous nous concentrons sur l'aspect classification, et nous illustrons le bon comportement de notre variante sur des données continues simulées et réelles.

1 Introduction

Plusieurs méthodes de classification utilisées sont basées sur une distance ou une mesure dissimilarité. Or, l'utilisation des modèles de mélange dans la classification est devenue une approche classique et très puissante (voir par exemple Banfield et Raftery (1993), et Celeux et Govaert (1995)). En traitant la classification sous cette approche, l'algorithme EM (Dempster et al., 1977), composé de deux étapes : *Estimation* et *Maximisation*, est devenu quasiment incontournable. Celui-ci est très populaire pour l'estimation de paramètres. Ainsi, de nombreux logiciels sont basés sur cette approche, comme Mclust-EMclust (Fraley et Raftery, 1999), EMmix (McLachlan et Peel, 1998), Mixmod (Biernacki et al., 2001) ou AutoClass (Cheeseman et Stutz, 1996).

Malheureusement, le principal inconvénient de EM réside dans sa lenteur due au nombre élevé d'itérations parfois nécessaire pour la convergence, ce qui rend son utilisation inappropriée pour les données de grande taille. Ayant testé plusieurs méthodes (Nadif et Jollois, 2004), nous avons retenu l'algorithme LEM (Thiesson et al, 2001) qui utilise une étape partielle d'*Estimation* au lieu d'une étape complète. A partir de cet algorithme, nous avons cherché à améliorer sa performance et avons proposé une variante plus efficace, *e*LEM. Sur des données qualitatives simulées et réelles, les performances de cette nouvelle version ont été très encourageantes. Le principal objectif de

ce travail est d'étendre l'étude expérimentale au cas des données continues en utilisant les modèles de mélange Gaussiens.

2 Modèle de mélange et algorithme EM

Dans l'approche modèle de mélange, les individus $\mathbf{x}_i, \ldots, \mathbf{x}_n$ à classifier sont supposés provenir d'un mélange de s densités dans des proportions inconnus $p_1, \ldots, p_s$. Pour des données continues, nous utilisons classiquement des distributions Gaussiennes. Ainsi, chaque objet $\mathbf{x}_i$ est une réalisation d'une densité de probabilité (p.d.f.), décrite par

$$f(\mathbf{x}_i|s, \boldsymbol{\theta}) = \sum_{k=1}^{s} p_k \varphi(\mathbf{x}_i|\mathbf{a}_k),$$

où les p_k représentent les proportions du mélange ($0 < p_k < 1$ pour $k = 1, \ldots, s$ et $\sum_k p_k = 1$) et $\varphi(.|\mathbf{a}_k)$ représente la densité (Gaussienne) de dimension d de la classe k, avec le vecteur moyenne μ_k et la matrice de covariance Σ_k, avec $\mathbf{a}_k = (\mu_k, \Sigma_k)$,

$$\varphi(\mathbf{x}_i; (\mu_k, \Sigma_k)) = (2\pi)^{-d/2} |\Sigma_k|^{-1/2} \exp\left(-\frac{1}{2}(\mathbf{x}_i - \mu_k)^T \Sigma_k^{-1} (\mathbf{x}_i - \mu_k)\right).$$

Et $\boldsymbol{\theta} = (p_1, \ldots, p_s; \mathbf{a}_1, \ldots, \mathbf{a}_s)$ représente le vecteur des paramètres du mélange à estimer. Les classes sont de forme ellipsoidale, centrées sur μ_k. La matrice de covariance Σ_k détermine leurs caractéristiques géométriques. Dans ce travail, nous avons choisi de prendre le modèle sphérique, avec $\Sigma_k = \lambda_k I$, où λ_k représente le volume de la classe, propre à chacune ici.

La log-vraisemblance de $\boldsymbol{\theta}$ pour $\mathbf{x} = (\mathbf{x}_i, \ldots, \mathbf{x}_n)$, est donnée par

$$L(\mathbf{x}, \theta) = \sum_{i=1}^{n} \log\left(\sum_{k=1}^{s} p_k \varphi_k(\mathbf{x}_i; \alpha_k)\right). \tag{1}$$

Dans la suite, nous allons aborder le problème de la classification sous l'approche estimation : les paramètres sont d'abord estimés, puis la partition en est déduite par la méthode du maximum a posteriori (MAP). L'estimation des paramètres du modèle passe par la maximisation de $L(\mathbf{x}, \theta)$. Une solution itérative pour la résolution de ce problème est l'algorithme EM (Dempster et al., 1977). Le principe de cet algorithme est de maximiser de manière itérative l'espérance de la log-vraisemblance complétée conditionnellement à l'estimation courante de $\theta^{(q)}$ et aux données $\mathbf{x}$:

$$Q(\theta|\theta^{(q)}) = \sum_{i=1}^{n} \sum_{k=1}^{s} t_{ik}^{(q)} \left(\log(p_k) + \log \varphi_k(\mathbf{x}_i; \alpha_k)\right)$$

où $t_{ik}^{(q)} \propto p_k^{(q)} \varphi_k(\mathbf{x}_i; \alpha_k^{(q)})$ est la probabilité conditionnelle a posteriori. Chaque itération de EM a deux étapes :

Estimation : Calculer $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(c)}) = \mathbf{E}[L(\mathbf{x}, \boldsymbol{\theta})|\mathbf{x}; \boldsymbol{\theta}^{(c)}]$. dans le contexte mélange, ceci revient à calculer les probabilités a posteriori $t_{ik} \propto p_k \varphi(\mathbf{x}_i; \mathbf{a}_k)$.

Maximisation : Calculer $\boldsymbol{\theta}^{(c+1)} = (\mathbf{p}^{(c+1)}, \mathbf{a}^{(c+1)})$ qui maximise l'estimation conditionnelle $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(c)})$.

3 Accélération

3.1 Lazy EM (LEM)

L'algorithme Lazy EM (Thiesson et al, 2001), ou LEM, cherche à réduire le temps de l'étape Estimation. Pour ceci, il cherche à identifier régulièrement les individus importants, et à les utiliser ensuite pendant plusieurs itérations. Un individu i est considéré comme important si le changement de sa probabilité t_{ik} entre deux itérations successives est grande. Notons $y_{\overline{lazy}}$ cet ensemble d'individus importants, et y_{lazy} l'ensemble des restants. Chaque itération requiert soit une étape Estimation standard, soit une étape Estimation *lazy*, suivie ensuite par une étape Maximisation standard. L'étape complète calcule pour tous les individus les probabilités a posteriori. De plus, elle établit la liste des individus importants. Une étape *lazy* ne met à jour qu'une partie des probabilités a posteriori.

Estimation :

Étape standard : Calculer les probabilités a posteriori. Identifier y_{lazy} comme l'ensemble d'individus à ignorer durant les étapes *lazy*.

Étape *lazy* : Dans cette étape, on calcule les probabilités a posteriori $t_{ik}^{(q)}$ pour toutes les observations appartenant au bloc $y_{\overline{lazy}}$, quand aux autres observations (appartenant à y_{lazy}), nous avons $t_{ik}^{(q)} = t_{ik}^{(q-1)}$. Seule l'espérance conditionnelle associée au bloc $y_{\overline{lazy}}$ notée $Q_{\overline{lazy}}$ est mise à jour. Autrement dit la quantité globale qu'on cherchera à maximiser dans l'étape maximisation est

$$Q(\theta|\theta^{(q)}) = Q(\theta|\theta^{(q-1)}) - Q_{\overline{lazy}}(\theta|\theta^{(q-1)}) + Q_{\overline{lazy}}(\theta|\theta^{(q)}).$$

Maximisation : On cherche comme dans l'algorithme EM classique, le paramètre $\theta^{(q+1)}$ qui maximise $Q(\theta|\theta^{(q)})$.

Le déroulement de LEM débute avec une itération standard suivie de it itérations. Ce schéma est répété jusqu'à la convergence de l'algorithme.

La viabilité de l'algorithme LEM réside partiellement dans l'idée que toutes les données ne sont pas d'importance égale. Elle dépend aussi du coût de calcul pour déterminer l'importance de chaque individu et du coût de stockage pour garder cette information. Pour les modèles de mélange, ces deux coûts peuvent être grandement réduits, voire simplement supprimés pour le coût de stockage, grâce à un critère d'importance. L'idée derrière ce critère est la suivante : si un individu a une forte probabilité d'appartenir à une classe, il n'est pas approprié de l'assigner à une autre. Et s'il le fallait, cela ne serait pas soudainement mais plutôt progressivement. Ainsi, nous supposons que les observations qui ne sont pas fortement liées à une classe contribuent le plus à l'évolution des paramètres. Un individu est considéré donc comme important s'il a toutes les probabilités d'appartenance t_{ik} inférieures à un certain seuil.

Due à la démonstration de Neal et Hinton (1998), la convergence de l'algorithme est théoriquement justifiée et est applicable pour chaque découpage arbitraire des individus, du moment qu'on visite régulièrement tous les cas.

3.2 La version eLEM

L'idée de départ de Thiesson et al (2001) est d'écarter un certain nombre d'individus que l'on peut considérer comme peu important dans les calculs. Cette notion d'importance peut être rapportée à l'évolution des probabilités a posteriori. En effet, si un individu ne montre pas d'évolution importante entre deux étapes, c'est qu'il est a priori stable et donc, il a de fortes chances de le rester un long moment. Dans ce cas, on prend la décision de l'écarter des calculs et de ne plus le prendre en compte pendant un certain nombre d'itérations. Au contraire, si son évolution est significative, il est intéressant de le garder dans les calculs. Pour ceci, nous avons choisi de mesurer les différences entre les probabilités a posteriori avant et après l'étape Estimation standard, où on remet à jour tous les t_{ik} de tous les individus. Plus précisément, nous comparons la moyenne des valeurs absolues de ces différences pour chaque classe avec un seuil th :

$$\frac{\sum_{k=1}^{s} |t_{ik}^{(q)} - t_{ik}^{(q-1)}|}{s} < th$$

Cette version de EM notée eLEM s'est avérée très efficace et meilleure que LEM sur des données qualitatives (Nadif et Jollois, 2004). Nous étudions ci-après son comportement sur des données continues en utilisant un modèle de mélange Gaussien.

4 Expériences numériques

Pour illustrer les performances de EM, LEM et eLEM, nous les avons appliquées sur des données simulées suivant le modèle de mélange Gaussien présenté dans la section 2 (avec $n = 5000$, $d = 2$, $s = 5$, trois situations : classes bien séparées (+), moyennement séparées (++) et peu séparées (+++), voir la figure 1). Nous avons restreint les paramètres après plusieurs tests. Ainsi, nous utilisons les seuils $th \in \{0.5, 0.6, 0.7, 0.8, 0.9, 0.95, 0.99\}$ pour LEM et $th \in \{0.001, 0.005, 0.010, 0.015, 0.020\}$ pour eLEM. Pour le nombre d'itérations, nous utilisons $it \in \{1, 2, 3, 4\}$ pour LEM et eLEM.

Dans le tableau 1, nous présentons le coefficient d'accélération moyen calculé avec

$$\frac{temps_{EM}}{temps_{LEM/eLEM}}$$

(avec l'écart-type) pour toutes les paramétrisations qui donnent la même log-vraisemblance que EM. A partir des résultats présentés dans Tab. 1, nous observons clairement que eLEM est plus rapide que LEM, pour les trois situations.

Nous avons également appliqué et comparé ces deux méthodes sur des données réelles, tirées du site de l'UCI Machine Learning Repository[1]. Le premier tableau,

[1] http ://www.ics.uci.edu/˜mlearn/MLRepository.html

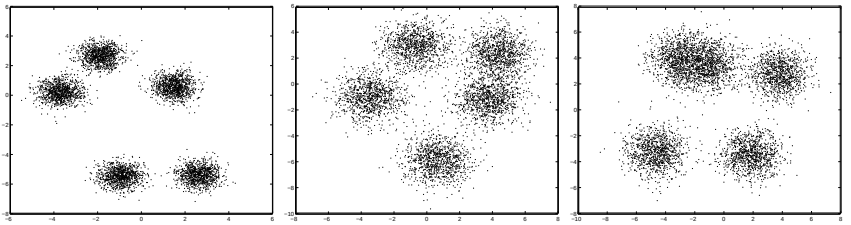

FIG. 1 – Distribution des données simulées avec cinq classes : classes bien séparées (+), moyennement séparées (++) et peu séparées (+++).

Yeast, concerne la localisation de site de protéines, selon certaines mesures ou scores calculés à partir de mesures. Il contient 1484 instances et 8 attributs. Ces données sont réparties en 10 classes. Le second tableau de données, German Credit, concerne des crédits banquiers, en Allemagne. Il contient 1000 instances, décrites par 24 variables numériques (dont certaines sont des scores). Il y a deux classes présentes (bon payeur ou mauvais payeur). Les partitions obtenus sont les mêmes par EM, LEM et eLEM et les performances en terme de rapidité enregistrées par LEM et eLEM sont reportées dans la table 1, elles montrent la supériorité de eLEM sur LEM.

Données		LEM	eLEM
Simulées	+	1.05 (± 1.98)	3.03 (± 1.29)
	++	0.67 (± 0.24)	1.39 (± 0.33)
	+++	0.86 (± 0.19)	1.71 (± 0.31)
Réelles	Yeast	1.16 (± 0.47)	2.30 (± 0.81)
	German Credit	0.98 (± 0.16)	1.48 (± 0.30)

TAB. 1 – Coefficient d'accélération moyen (et écart-type) pour LEM et eLEM.

5 Conclusion et Perspectives

Dans ce travail, nous nous sommes intéressés au problème de l'accélération de l'algorithme EM. Nous avons présenté deux variantes de cet algorithme : la première due à Thiesson et al (2001) appelée LEM et la seconde eLEM qui tient compte de l'évolution des probabilités a posteriori. Notre méthode eLEM s'avère plus performante sur des données continues, confirmant les résultats déjà obtenus sur des données qualitatives. Actuellement nous sommes menons des expériences intensives à partir d'autres modèles de mélange Gaussiens afin de valider les performances de eLEM. Aussi, nous cherchons à proposer une stratégie efficace permettant de surmonter la difficulté du choix des paramètres tout en réduisant le temps d'exécution.

Références

Banfield, J. D. and Raftery, A. E. (1993), Model-based Gaussian and non-Gaussian Clustering, Biometrics, 49, 803–821, 1993.

C. Biernacki and G. Celeux and G. Govaert and F. Langrognet and Y. Vernaz (2001), MIXMOD : High Performance Model-Based Cluster and Discriminant Analysis, http ://www-math.univ-fcomte.fr/MIXMOD/index.php, 2001.

Celeux, G. and Govaert, G. (1995), Gaussian Parcimonious Clustering Methods, Patt. Rec., 28, 781–793, 1995.

Cheeseman, P. and Stutz, J. (1996), Bayesian Classification (AutoClass) : Theory and Results, in Advances in Knowledge Discovery and Data Mining, Fayyad, U. and Piatetsky-Shapiro, G. and Uthurusamy, R., AAAI Press, 61–83, 1996.

Dempster, A. and Laird, N. and Rubin, D. (1977), Mixture Densities, Maximum Likelihood from Incomplete Data via the EM Algorithm, Journal of the Royal Statitical Society, 39, 1, 1–38, 1977.

Fraley, C. and Raftery, A. E. (1999), MCLUST : Software for Model-Based Cluster and Discriminant Analysis, University of Washington, 342, 1999.

McLachlan, G. J. and Peel, D. (1998), User's guide to EMMIX-Version 1.0, University of Queensland, 1998.

Nadif M., Jollois F.-X. (2004), Accélération de EM pour données qualitatives : étude comparative de différentes versions, Extraction et Gestion des Connaissances, RNTI-E-2, 253–264, 2004.

Neal, R. and Hinton, G. (1998), A View of the EM Algorithm that Justifies Incremental, Sparse, and Other Variants, Jordan, M., in Learning in Graphical Models, 355–371, 1998.

Thiesson, B. and Meek, C. and Heckerman, D. (2001), Accelerating EM for Large Databases, Machine Learning 45, 279–299, 2001.

Summary

Very popular and efficient for mixture parameters estimation, the EM algorithm has the major inconvenient to converge sometimes slowly. Its application on large data sets becomes unsuitable. Then, some accelerating methods were proposed. We present here the behavior of a known variant, LEM, and of a new one that we have proposed, *e*LEM. These variants speed-up the convergence of EM, and yield similar results. In this work, we focus on clustering context, and we illustrate the good behavior of our method on synthetic and real continuous data.