

Algorithme EM régularisé

Pierre HOUDOUIN¹, Matthieu JONCKHEERE², Frédéric PASCAL¹, Esa OLLILA³

¹Université Paris-Saclay, CNRS, CentraleSupélec, Laboratoire des signaux et systèmes, 91190, Gif-sur-Yvette, France

²LAAS-CNRS, Université de Toulouse, 31077 Toulouse, France

³Aalto University, Helsinki, Finland

pierre.houdouin@centralesupelec.fr, frederic.pascal@centralesupelec.fr, mjonckheer@laas.fr,
esa.ollila@aalto.fi

Résumé – L’algorithme Expectation-Maximization (EM) est un algorithme itératif, notamment utilisé pour estimer les maximum de vraisemblance de données issues d’un modèle de mélange gaussien (GMM). Lorsque la taille de l’échantillon des données est proche de leur dimension, les estimations successives de la matrice de covariance peuvent être singulières ou mal conditionnées, entraînant une baisse des performances. Nous présentons dans ce papier une nouvelle version régularisée de l’algorithme EM adapté au cas où le rapport entre taille de l’échantillon et la dimension est faible. Cette méthode maximise une version pénalisée de la vraisemblance de l’algorithme EM-GMM qui assure que les covariances soient toujours définies positives. Des tests sur données réelles illustrent enfin l’intérêt de cette approche pour un problème de clustering.

Abstract – Expectation-Maximization (EM) algorithm is a widely used iterative algorithm for computing maximum likelihood estimate when dealing with Gaussian Mixture Model (GMM). When the sample size is smaller than the data dimension, this could lead to a singular or poorly conditioned covariance matrix and, thus, to performance reduction. This paper presents a regularized version of the EM algorithm that efficiently uses prior knowledge to cope with a small sample size. This method aims to maximize a penalized GMM likelihood where regularized estimation may ensure positive definiteness of covariance matrix updates by shrinking the estimators towards some structured target covariance matrices. Finally, experiments on real data highlight the good performance of the proposed algorithm for clustering purposes.

1 Introduction

L’algorithme EM [1] est un algorithme fréquemment utilisé en apprentissage non supervisé et en modélisation statistique permettant de trouver un maximum local de la vraisemblance de données non labellisées et d’estimer les labels associés. En procédant itérativement, cet algorithme estime les paramètres inconnus du modèle qui augmentent l’espérance de la vraisemblance des données complétées grâce aux paramètres de l’itération précédente.

Historiquement élaboré pour les modèles de mélange de gaussien (GMM) [2], l’algorithme a été étendu aux distributions de Student par [3] pour mieux faire face aux données aberrantes et aux données à queues lourdes. Plus récemment, une généralisation aux distributions elliptiques symétriques a été développée ([4] pour le clustering et [5] pour la classification).

En traitement du signal, la dimension m des données est souvent élevée par rapport à leur nombre n : $n \sim m$. Dans de telles conditions, des problèmes de convergence surviennent lors de l’estimation des matrices de covariance qui ne sont plus forcément bien conditionnées ou même inversibles à chaque itération. L’estimation de matrices de covariance régularisées est une technique couramment

utilisée pour surmonter cette difficulté dans les modèles de clustering [7, 8, 9].

En 2022, [10] présente une nouvelle version régularisée de l’algorithme EM, RG-EM, qui utilise une nouvelle pénalisation de la vraisemblance permettant de tirer parti de la structure sous-jacente supposée des matrices de covariance. [10] montre que des estimateurs mieux conditionnés sont obtenus, avec de meilleures performances de clustering dans les régimes où la dimension est élevée par rapport au nombre de données. Nous proposons ici d’évaluer les performances de RG-EM sur des données réelles. La structure du papier est la suivante : la section 2 rappelle les éléments théoriques de l’algorithme, la section 3 contient les expériences sur données réelles et les conclusions, remarques et perspectives sont établies dans la section 5.

2 Algorithme EM régularisé

On suppose que chaque observation $\mathbf{x}_i \in \mathbb{R}^m$ est issue d’un GMM où chaque cluster \mathcal{C}_k a son propre vecteur moyenne $\boldsymbol{\mu}_k \in \mathbb{R}^m$, sa propre matrice de covariance symétrique définie positive $\boldsymbol{\Sigma}_k \in \mathbb{R}^{m \times m}$ et sa probabilité

d'appartenance $\pi_k \in [0, 1]$ avec $\sum_k \pi_k = 1$. La densité de probabilité de \mathbf{x}_i s'écrit alors :

$$f(\mathbf{x}_i|\boldsymbol{\theta}) = (2\pi)^{-\frac{m}{2}} \sum_{k=1}^K \pi_k |\boldsymbol{\Sigma}_k|^{-\frac{1}{2}} e^{-\frac{1}{2}(\mathbf{x}_i - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1}(\mathbf{x}_i - \boldsymbol{\mu}_k)}$$

avec $\boldsymbol{\theta} = (\pi_1, \dots, \pi_K, \boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_K, \boldsymbol{\Sigma}_1, \dots, \boldsymbol{\Sigma}_K)$, le vecteur de tous les paramètres inconnus.

Supposons également qu'une information a priori sur la structure des matrices de covariance de chaque cluster est disponible : e.g., elles sont proches de matrices cibles \mathbf{T}_k , $k = 1, \dots, K$. On exploite cette structure en pénalisant la vraisemblance avec la divergence de Kullback-Leibler (définie dans [7]) entre chaque $\boldsymbol{\Sigma}_k$ et \mathbf{T}_k :

$$\Pi_{\text{KL}}(\boldsymbol{\Sigma}_k, \mathbf{T}_k) = \frac{1}{2} (\text{tr}(\boldsymbol{\Sigma}_k^{-1} \mathbf{T}_k) - \log |\boldsymbol{\Sigma}_k^{-1} \mathbf{T}_k| - m).$$

Soit $\mathbf{X} = (\mathbf{x}_1 \cdots \mathbf{x}_n)$ la matrice des données issues de notre modèle GMM, notre vraisemblance pénalisée est alors :

$$\ell_\eta(\boldsymbol{\theta}|\mathbf{X}) = \ell(\mathbf{X}|\boldsymbol{\theta}) - \sum_{k=1}^K \eta_k \Pi_{\text{KL}}(\boldsymbol{\Sigma}_k, \mathbf{T}_k)$$

où $\eta_1, \dots, \eta_K \geq 0$ sont des paramètres automatiquement ajustés.

Proposition 2.1. *L'étape E de l'algorithme EM régularisé n'est pas modifiée, on a $\forall i \in \llbracket 1, n \rrbracket, \forall k \in \llbracket 1, K \rrbracket$:*

$$p_{ik}^{(t)} = \frac{\hat{\pi}_k^{(t)} |\hat{\boldsymbol{\Sigma}}_k^{(t)}|^{-\frac{1}{2}} e^{-\frac{1}{2}(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k^{(t)})^\top \hat{\boldsymbol{\Sigma}}_k^{(t)-1}(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k^{(t)})}}{\sum_{j=1}^K \hat{\pi}_j^{(t)} |\hat{\boldsymbol{\Sigma}}_j^{(t)}|^{-\frac{1}{2}} e^{-\frac{1}{2}(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_j^{(t)})^\top \hat{\boldsymbol{\Sigma}}_j^{(t)-1}(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_j^{(t)})}} \quad (1)$$

Preuve. Voir [10]

Proposition 2.2. *Les mises à jours de l'étape M sont les suivantes :*

$$\pi_k^{(t+1)} = \frac{1}{n} \sum_{i=1}^n p_{ik}^{(t)}, \quad \hat{\boldsymbol{\mu}}_k^{(t+1)} = \sum_{i=1}^n w_{ik}^{(t)} \mathbf{x}_i$$

$$\hat{\boldsymbol{\Sigma}}_k^{(t+1)} = \beta_k^{(t+1)} \sum_{i=1}^n w_{ik}^{(t)} (\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k^{(t)}) (\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k^{(t)})^\top + (1 - \beta_k^{(t+1)}) \mathbf{T}_k,$$

$$\text{où } \beta_k^{(t+1)} = \frac{n \pi_k^{(t+1)}}{\eta_k + n \pi_k^{(t+1)}} \text{ et } w_{ik}^{(t)} = \frac{p_{ik}^{(t)}}{\sum_{i=1}^n p_{ik}^{(t)}}$$

Preuve. Voir [10]

La matrice cible \mathbf{T}_k permet ainsi d'injecter des connaissances a priori sur $\boldsymbol{\Sigma}_k$ dans l'estimation. Si aucune information a priori n'est disponible, on peut choisir $\mathbf{T}_k = \hat{\theta}_k^0 \mathbf{I}_m$, ce qui permet simplement d'assurer le bon conditionnement des estimateurs. On utilise alors l'estimateur classique du paramètre d'échelle $\theta_k = \text{tr}(\boldsymbol{\Sigma}_k)/m$. Dans nos expériences, on utilise $\hat{\theta}_k^0 = \text{tr}(\hat{\boldsymbol{\Sigma}}_k^0)/m$ où $\hat{\boldsymbol{\Sigma}}_k^0$ est la valeur initiale de l'estimation de la matrice de covariance,

Algorithm 1 L -fold validation croisée de η_k pour le cluster k

Entrées : L'échelle $\hat{\theta}_k^0$ et les indices \mathcal{D}^0 des éléments du cluster k . Un ensemble de candidats $\{\eta_j\}_{j=1}^J$ pour la valeur du paramètre de pénalisation.

- 1: Séparer \mathcal{D}^0 en L sous-ensembles distincts $\mathcal{D}_1, \dots, \mathcal{D}_L$ t.q $\mathcal{D}^0 = \cup_{l=1}^L \mathcal{D}_l$ et initialiser $\mathbf{T}_k^0 = \hat{\theta}_k^0 \cdot \mathbf{I}_m$ comme matrice cible.
 - 2: Initialiser $\text{Err}_j \equiv \text{Err}(\eta_j) = 0$ pour $j = 1, \dots, J$.
 - 3: **for** $l \in \llbracket 1, L \rrbracket$ **do**
 - 4: Initialiser $\mathcal{D}_{\text{val}} = \mathcal{D}_l$ and $\mathcal{D}_{\text{tr}} = \mathcal{D} / \mathcal{D}_{\text{val}}$
 - 5: $\mathbf{S}_{\text{val}} = \frac{1}{|\mathcal{D}_{\text{val}}|} \sum_{i \in \mathcal{D}_{\text{val}}} (\mathbf{x}_i - \bar{\mathbf{x}}_{\text{val}}) (\mathbf{x}_i - \bar{\mathbf{x}}_{\text{val}})^\top$
 - 6: $\hat{\boldsymbol{\Sigma}} = \frac{1}{|\mathcal{D}_{\text{tr}}|} \sum_{i \in \mathcal{D}_{\text{tr}}} (\mathbf{x}_i - \bar{\mathbf{x}}_{\text{tr}}) (\mathbf{x}_i - \bar{\mathbf{x}}_{\text{tr}})^\top$
 - 7: **for** $\eta \in \{\eta_1, \dots, \eta_J\}$ **do**
 - 8: $\hat{\boldsymbol{\Sigma}}_\eta = \frac{|\mathcal{D}_{\text{tr}}|}{\eta + |\mathcal{D}_{\text{tr}}|} \hat{\boldsymbol{\Sigma}} + \frac{\eta}{\eta + |\mathcal{D}_{\text{tr}}|} \mathbf{T}_k^0$
 - 9: $\text{Err}_l = \text{Err}_l + \text{tr}(\hat{\boldsymbol{\Sigma}}_\eta^{-1} \mathbf{S}_{\text{val}}) + \log |\boldsymbol{\Sigma}_\eta|$
 - 10: Choisir le η_j qui minimise $\{\text{Err}(\eta_j)\}_{j=1}^J$
-

obtenue grâce à un premier clustering avec l'algorithme K-means. Dans l'algorithme EM, la valeur du paramètre d'échelle est périodiquement mise à jour avec la nouvelle valeur de $\hat{\boldsymbol{\Sigma}}_k$.

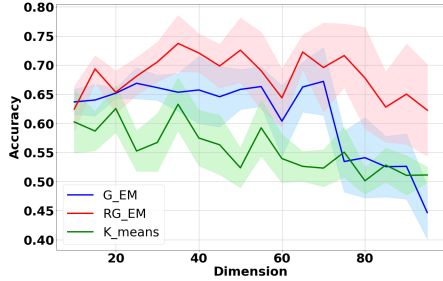
Le choix du paramètre de régularisation est également essentiel. On utilise une sélection par validation croisée qui maximise la log-vraisemblance gaussienne [9]. Chaque η_k est estimé indépendamment parmi un ensemble de candidats $\{\eta_1, \dots, \eta_J\}$ par la procédure décrite dans l'algorithme 1.

3 Expériences sur données simulées

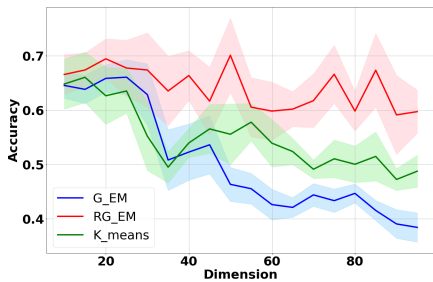
L'algorithme EM régularisé, RG-EM, est comparé à l'EM classique, noté G-EM, ainsi qu'à l'algorithme K-means. Les deux versions de l'EM sont implémentées et la version de Scikit-learn pour le K-means est utilisée. Afin que l'EM classique converge même lorsque la dimension est élevée et qu'il y a peu de données, on ajoute une régularisation classique avec la matrice $\epsilon \mathbf{I}_m$ à chaque itération. On utilise pour **K-means** $n_{\text{init}} = 10$ et $\text{max}_{\text{iter}} = 200$, pour **G-EM** $\epsilon = 10^{-4}$ et $\text{max}_{\text{iter}} = 40$ et pour **RG-EM** $L = 5$ (Algorithme 1) et $\text{max}_{\text{iter}} = 40$. Comme indiqué dans la section 2, on utilise pour matrices cibles $\mathbf{T}_k = \text{tr}(\hat{\boldsymbol{\Sigma}}_k^0)/m \mathbf{I}_m$. Pour **RG-EM**, on recalcule les η_k optimaux toute les 10 itérations. Les données générées à partir de distributions gaussiennes sont réparties en $K = 3$ clusters avec les priors $\pi_k = \frac{1}{3}$. Le vecteur moyenne est tiré aléatoirement sur la sphère centrée de rayon 2 tandis qu'on utilise une structure autorégressive pour les covariances. On choisit $(\boldsymbol{\Sigma}_k)_{i,j} = \rho_k^{|i-j|}$ avec les coefficients 0.8, 0.5 et 0.2. Cela traduit une structure autorégressive dans les données.

On teste deux configurations avec, respectivement, $n = 1000$ et $n = 500$. On évalue la performance des modèles

en calculant leur précision. Pour calculer la précision en clustering, on commence par calculer la matrice de confusion, puis on permute les colonnes de sorte à maximiser la somme des éléments diagonaux. Les résultats sont présentés en figure 1.



(a) $n = 1000$



(b) $n = 500$

FIGURE 1 – Evolution de la précision en fonction de la dimension

Dans les deux configurations, il y a une dimension à partir de laquelle les performances de l'EM classique chutent et cela correspond au ratio $\frac{n}{m} \approx 14$. A l'inverse, l'EM régularisé parvient à conserver des performances similaires entre la dimension 10 et la dimension 100. En effet, les matrices de covariance des clusters ont une structure proche d'une identité, surtout lorsque ρ est proche de 0. La matrice cible choisie s'avère donc ici particulièrement pertinente

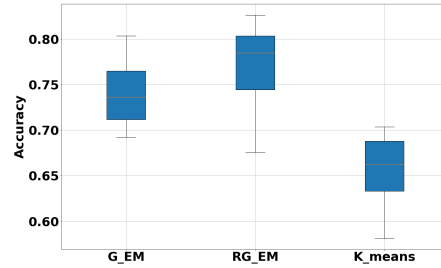
4 Expériences sur données réelles

On teste chaque méthode sur des jeux de données réelles issus de l'UCI machine learning repository [11]. Deux datasets sont utilisés :

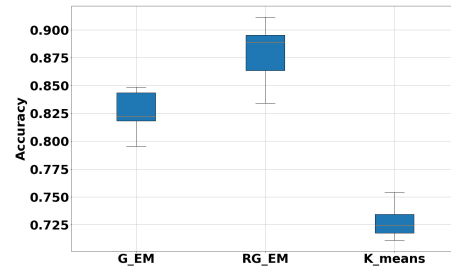
- **Ionosphere** : $n = 351$, $p = 34$ et $K = 2$
- **Breast cancer** : $n = 699$, $p = 9$ et $K = 2$

On utilise 70% des données pour l'entraînement et 30% pour l'évaluation des performances. Les résultats sont moyennés sur 100 simulations et les datasets sont recomposés toutes les 10 simulations. Utiliser une matrice cible circulaire n'est pas adapté si certaines valeurs propres des

matrices de covariance sont proche de 0. On effectue donc une analyse en composantes principales pour réduire la dimension, on choisit la nouvelle dimension comme la plus petite permettant de conserver 95% de l'information (variance). Cela correspond à $m = 8$ pour Breast cancer et $m = 26$ pour Ionosphere. Les nouvelles matrices étant proches de matrices diagonales, le choix de $\mathbf{T}_k = \hat{\theta}_k^0 \cdot \mathbf{I}_m$ semble pertinent. On obtient les résultats présentés sur la figure 2.



(a) ionosphere

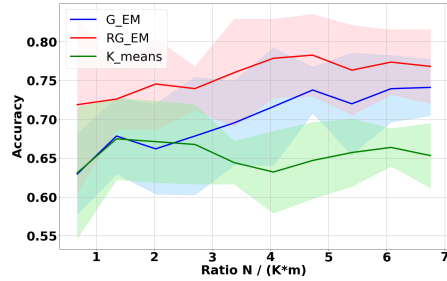


(b) Breast cancer

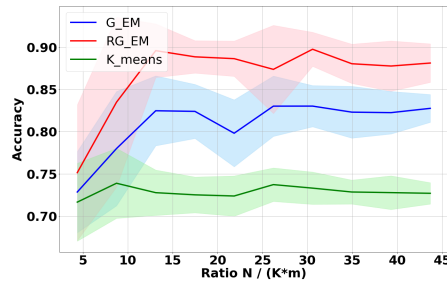
FIGURE 2 – Précision médiane

Sur les deux jeux de données, K-means obtient des performances sensiblement inférieures aux méthodes EM, avec un écart d'environ 10% de précision. La version régularisée de l'EM conduit, sur les deux jeux de données, à de meilleurs résultats que l'algorithme GMM classique, la réduction de dimension ayant rendu pertinent l'utilisation d'une matrice cible proportionnelle à l'identité. On peut maintenant s'intéresser à l'évolution des performances de chaque méthode lorsque le rapport $\frac{n}{m}$ devient de plus en plus faible. Pour observer cela, on supprime progressivement des données du dataset d'entraînement pour réduire sa taille de 100% à 10%, ce qui fait diminuer le rapport $\frac{n}{m}$. Les résultats sont présentés sur la figure 3.

Sur les deux jeux de données, K-means n'est pas très impacté par la diminution du nombre de données. En effet, la suppression des données ne change pas la structure géométrique des clusters, et K-means construit une frontière similaire avec peu de données. A l'inverse, les estimateurs des algorithmes EM sont impactés par la baisse du nombre



(a) ionosphere



(b) Breast cancer

FIGURE 3 – Evolution de la précision en fonction du rapport $\frac{n}{m}$

de données, ce qui provoque une diminution des performances. Sur le dataset breast cancer wisconsin, les deux méthodes conservent des performances similaires jusqu'à ce que le nombre de données soit réduit de 80%. La performance chute alors rapidement pour rejoindre celle des autres méthodes. Sur le dataset ionosphere, les deux algorithmes EM baissent progressivement, mais encore une fois, la version régularisée chute moins vite et conserve de meilleures performances.

5 Conclusion

Nous avons présenté dans cet article une version régularisée de l'algorithme EM-GMM qui surpasse les méthodes classiques de clustering dans les régimes où le nombre de données est faible par rapport à la dimension. Dans cette nouvelle approche, l'estimation de la matrice de covariance est régularisée avec un terme de pénalisation qui oriente l'estimation vers une matrice cible. Les coefficients de régularisation η_k optimaux sont sélectionnés grâce à un algorithme de validation croisée et régulièrement mis à jour au cours des itérations. Les performances obtenues avec ce nouvel algorithme sont meilleures que celles obtenues avec des algorithmes classiques. De plus, la méthode proposée, qui peut être vue comme une amélioration de l'EM classique, est relativement stable en fonction du rapport m/n . Les perspectives de ces travaux vont se focaliser sur

l'apprentissage des matrices cibles, ainsi que sur la version totalement non-supervisée de RG-EM.

Références

- [1] A. P. Dempster and N. M. Laird and D. B. Rubin *Maximum Likelihood from Incomplete Data via the EM Algorithm*. Journal of the Royal Statistical Society, 1977
- [2] Guorong Xuan and Wei Zhang and Peiqi Chai *EM algorithms of gaussian mixture model and hidden Markov model*. Proceedings 2001 International Conference on Image Processing
- [3] Ingrassia, Salvatore and Minotti, Simona C and Incarbone, Giuseppe *An EM algorithm for the student-t cluster-weighted modeling*. Challenges at the Interface of Data Analysis, Computer Science, and Optimization, 2012
- [4] Roizman, Violeta and Jonckheere, Matthieu and Pascal, Frédéric *A flexible EM-like clustering algorithm for noisy data*. arXiv preprint arXiv:1907.01660, 2019
- [5] Houdouin, Pierre and Wang, Andrew and Jonckheere, Matthieu and Pascal *Robust classification with flexible discriminant analysis in heterogeneous data*. ICASSP 2022
- [6] Mahdi Teimouri *EM algorithm for mixture of skew-normal distributions fitted to grouped data*. Journal of Applied Statistics, 2021
- [7] Ying Sun and Prabhu Babu and Daniel P. Palomar *Regularized Tyler's Scatter Estimator : Existence, Uniqueness, and Algorithms*. IEEE Transactions on Signal Processing, 2014
- [8] Pascal, Frédéric and Chitour, Yacine and Quek, Yihui *Generalized robust shrinkage estimator and its application to STAP detection problem*. IEEE Transactions on Signal Processing, 2014
- [9] Yi, Mengxi and Tyler, David E *Shrinking the Covariance Matrix using Convex Penalties on the Matrix-Log Transformation*. Journal of Computational and Graphical Statistics, 2020
- [10] Pierre Houdouin and Esa Ollila and Frederic Pascal *Regularized EM algorithm*. <https://arxiv.org/abs/2303.14989>, 2022
- [11] Dua Dheeru and Graff Casey. *UCI Machine Learning Repository*. University of California, Irvine, School of Information and Computer Sciences, 2017.