Algorithme espérance-maximisation

L'**algorithme espérance-maximisation** (en anglais *expectation-maximization algorithm*, souvent abrégé **EM**), proposé par Dempster et al. (1977)[1], est un algorithme itératif qui permet de trouver les paramètres de vraisemblance maximum d'un modèle probabiliste lorsque ce dernier dépend de variables latentes non observables. De nombreuses variantes ont par la suite été proposées, formant une classe entière d'algorithmes.

1 Usage

On utilise souvent l'algorithme EM pour la classification de données, l'apprentissage automatique, ou la vision artificielle. On peut également citer son utilisation en imagerie médicale dans le cadre de la reconstruction tomographique.

L'algorithme d'espérance-maximisation comporte :

- une étape d'évaluation de l'espérance (E), où l'on calcule l'espérance de la vraisemblance en tenant compte des dernières variables observées,
- une étape de maximisation (M), où l'on estime le maximum de vraisemblance des paramètres en maximisant la vraisemblance trouvée à l'étape E.

On utilise ensuite les paramètres trouvés en M comme point de départ d'une nouvelle phase d'évaluation de l'espérance, et l'on itère ainsi.

Pour résoudre le problème d'apprentissage des modèles de Markov cachés (HMM), c'est-à-dire la détermination des paramètres du modèle markovien, on utilise l'algorithme de Baum-Welch.

2 Principe de fonctionnement

En considérant un échantillon $\mathbf{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ d'individus suivant une loi $f(\boldsymbol{x}_i, \theta)$ paramétrée par $\boldsymbol{\theta}$, on cherche à déterminer le paramètre $\boldsymbol{\theta}$ maximisant la log-vraisemblance donnée par

$L(\mathbf{x}; \boldsymbol{\theta}) = \sum_{i=1}^{n} \log f(\boldsymbol{x}_i, \boldsymbol{\theta})$.

Cet algorithme est particulièrement utile lorsque la maximisation de L est très complexe mais que, sous réserve de connaître certaines données judicieusement choisies, on peut très simplement déterminer $\boldsymbol{\theta}$.

Dans ce cas, on s'appuie sur des données complétées par un vecteur $\mathbf{z} = (z_1, \ldots, z_n)$ inconnu. En notant $f(z_i|\boldsymbol{x}_i; \theta)$ la probabilité de z_i sachant $\boldsymbol{x}_i$ et le paramètre $\boldsymbol{\theta}$, on peut définir la log-vraisemblance complétée comme la quantité

$L((\mathbf{x}, \mathbf{z}); \boldsymbol{\theta}) = \sum_{i=1}^{n} (\log f(z_i|\boldsymbol{x}_i, \boldsymbol{\theta}) + \log f(\boldsymbol{x}_i; \boldsymbol{\theta}))$.

et donc,

$L(\mathbf{x}; \boldsymbol{\theta}) = L((\mathbf{x}, \mathbf{z}); \boldsymbol{\theta}) - \sum_{i=1}^{n} \log f(z_i|\boldsymbol{x}_i, \boldsymbol{\theta})$.

L'algorithme EM est une procédure itérative basée sur l'espérance des données complétées conditionnellement au paramètre courant. En notant $\boldsymbol{\theta}^{(c)}$ ce paramètre, on peut écrire

$E\left[L(\mathbf{x}; \boldsymbol{\theta})|\boldsymbol{\theta}^{(c)}\right] = E\left[L((\mathbf{x}, \mathbf{z}); \boldsymbol{\theta})|\boldsymbol{\theta}^{(c)}\right] - E\left[\sum_{i=1}^{n} \log f(z_i|\boldsymbol{x}_i, \boldsymbol{\theta})|\boldsymbol{\theta}^{(c)}\right]$, où l'espérance est prise sur $\mathbf{z}$.

ou encore

$L(\mathbf{x}; \boldsymbol{\theta}) = Q\left(\boldsymbol{\theta}; \boldsymbol{\theta}^{(c)}\right) - H\left(\boldsymbol{\theta}; \boldsymbol{\theta}^{(c)}\right)$, car $L(\mathbf{x}; \boldsymbol{\theta})$ ne dépend pas de $\mathbf{z}$,

avec $Q\left(\boldsymbol{\theta}; \boldsymbol{\theta}^{(c)}\right) = E\left[L((\mathbf{x}, \mathbf{z}); \boldsymbol{\theta})|\boldsymbol{\theta}^{(c)}\right]$ et $H\left(\boldsymbol{\theta}; \boldsymbol{\theta}^{(c)}\right) = E\left[\sum_{i=1}^{n} \log f(z_i|\boldsymbol{x}_i, \boldsymbol{\theta})|\boldsymbol{\theta}^{(c)}\right]$.

On montre que la suite définie par

$\boldsymbol{\theta}^{(c+1)} = \arg\max_{\boldsymbol{\theta}} \left(Q\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{(c)}\right)\right)$

fait tendre $L\left(\mathbf{x}; \boldsymbol{\theta}^{(c+1)}\right)$ vers un maximum local.

L'algorithme EM peut être défini par :

- Initialisation au hasard de $\boldsymbol{\theta}^{(0)}$
- c=0
- Tant que l'algorithme n'a pas convergé, faire
  - Evaluation de l'espérance (étape E) : $Q\left(\boldsymbol{\theta}; \boldsymbol{\theta}^{(c)}\right) = E\left[L((\mathbf{x}, \mathbf{z}); \boldsymbol{\theta}))|\boldsymbol{\theta}^{(c)}\right]$
  - Maximisation (étape M) : $\boldsymbol{\theta}^{(c+1)} = \arg\max_{\boldsymbol{\theta}} \left(Q\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{(\boldsymbol{c})}\right)\right)$
  - c=c+1
- Fin

En pratique, pour s'affranchir du caractère local du maximum atteint, on fait tourner l'algorithme EM un grand nombre de fois à partir de valeurs initiales différentes

de manière à avoir de plus grandes chances d'atteindre le maximum global de vraisemblance.

3 Exemple détaillé : application en classification automatique

Une des applications phares d'EM est l'estimation des paramètres d'une densité mélange en classification automatique dans le cadre des modèles de mélanges gaussiens. Dans ce problème, on considère qu'un échantillon $(x_1, \ldots, x_n)$ de $\mathbb{R}^p$, ie caractérisé par *p* variables continues, est en réalité issu de g différents groupes. En considérant que chacun de ces groupes G_k suit une loi f de paramètre θ_k , et dont les proportions sont données par un vecteur $(\pi_1, \ldots, \pi_g)$. En notant $\Phi = (\pi_1, \ldots, \pi_g, \theta_1, \ldots, \theta_g)$ le paramètre du mélange, la fonction de densité que suit l'échantillon est donnée par

$$g(x, \Phi) = \sum_{k=1}^{g} \pi_k f(x, \theta_k),$$

et donc, la log-vraisemblance du paramètre Φ est donnée par

$$L(x, \Phi) = \sum_{i=1}^{n} \log \left(\sum_{k=1}^{g} \pi_k f(x_i, \theta_k) \right).$$

La maximisation de cette fonction selon Φ est très complexe. Par exemple, si on souhaite déterminer les paramètres correspondant à 2 groupes suivant une loi normale dans un espace de dimension 3 (ce qui est peu), on doit optimiser une fonction non linéaire de $\mathbb{R}^{19}$!!!

Parallèlement, si on connaissait les groupes auxquels appartient chacun des individus, alors le problème serait un problème d'estimation tout à fait simple et très classique.

La force de l'algorithme EM est justement de s'appuyer sur ces données pour réaliser l'estimation. En notant z_{ik} la grandeur qui vaut 1 si l'individu x_i appartient au groupe G_k et 0 sinon, la log-vraisemblance des données complétée s'écrit

$$L(x, z, \Phi) = \sum_{i=1}^{n} \sum_{k=1}^{g} z_{ik} \log \left(\pi_k f(x_i, \theta_k) \right).$$

On obtient alors rapidement

$$Q\left(\Phi, \Phi^{(c)}\right) = \sum_{i=1}^{n} \sum_{k=1}^{g} E\left(z_{ik} | x, \Phi^{(c)}\right) \log \left(\pi_k f(x_i, \theta_k) \right)$$

En notant t_{ik} la quantité donnée par $t_{ik} = E\left(z_{ik} | x, \Phi^{(c)}\right)$, on peut séparer l'algorithme EM en deux étapes, qu'on appelle classiquement, dans le cas des modèles de mélanges, l'étape Estimation et l'étape Maximisation. Ces deux étapes sont itérées jusqu'à la convergence.

- Étape E : calcul de t_{ik} par la règle d'inversion de Bayes :

$$t_{ik} = \frac{\pi_k^{(c)} f(x_i, \theta_k^{(c)})}{\sum_{\ell=1}^{g} \pi_\ell^{(c)} f(x_i, \theta_\ell^{(c)})}$$

- Étape M : détermination de Φ maximisant

$$Q\left(\Phi, \Phi^{(c)}\right) = \sum_{i=1}^{n} \sum_{k=1}^{g} t_{ik} \log \left(\pi_k f(x_i, \theta_k) \right)$$

L'avantage de cette méthode est qu'on peut séparer le problème en g problèmes élémentaires qui sont, en général relativement simples. Dans tous les cas, les proportions optimales sont données par

$\pi_k = \frac{1}{n} \sum_{i=1}^{n} t_{ik}$

L'estimation des θ dépend par ailleurs de la fonction de probabilité f choisie. Dans le cas normal, il s'agit des moyennes μ_k et des matrices de variance-covariance Σ_k . Les estimateurs optimaux sont alors donnés par

$\mu_k = \frac{\sum_{i=1}^{n} t_{ik} x_i}{\sum_{i=1}^{n} t_{ik}}$

$\Sigma_k = \frac{\sum_{i=1}^{n} t_{ik} (x_i - \mu_k)(x_i - \mu_k)'}{\sum_{i=1}^{n} t_{ik}}$

Avec *M'* la matrice transposée de *M* et en supposant que les μ_k sont des vecteurs colonnes.

4 Variantes usuelles d'EM

L'algorithme EM allie, dans la plupart des cas, simplicité de mise en œuvre et efficacité. Néanmoins quelques cas problématiques ont donné lieu à des développements complémentaires. Parmi les variantes existantes de cet algorithme nous évoquerons l'algorithme *GEM* (generalized EM) qui permet de simplifier le problème de l'étape maximisation ; l'algorithme *CEM* (classification EM) permettant de prendre en compte l'aspect classification lors de l'estimation, ainsi que l'algorithme *SEM* (stochastic EM) dont l'objectif est de réduire le risque de tomber dans un optimum local de vraisemblance.

4.1 Algorithme GEM

GEM a été proposé en même temps qu'EM par Dempster et al. (1977) qui ont prouvé que pour assurer la

convergence vers un maximum local de vraisemblance, il n'est pas nécessaire de maximiser Q à chaque étape mais qu'une simple **amélioration de Q** est suffisante.

GEM peut donc s'écrire de la manière suivante :

- Initialisation au hasard de $\theta^{(0)}$
- $c = 0$
- Tant que l'algorithme n'a pas convergé, faire
  - choisir $\theta^{(c+1)}$ tel que $Q\left(\theta, \theta^{(c+1)}\right) > Q\left(\theta, \theta^{(c)}\right)$
  - $c = c + 1$
- Fin

4.2 Algorithme CEM

L'algorithme EM se positionne dans une optique *estimation*, c'est-à-dire qu'on cherche à maximiser la vraisemblance du paramètre θ , sans considération de la classification faite *a posteriori* en utilisant la règle de Bayes.

L'approche *classification*, proposée par Celeux et Govaert (1991)[2] consiste à optimiser, non pas la vraisemblance du paramètre, mais directement la vraisemblance complétée, donnée, dans le cas des modèles de mélange, par $L(x, z; \theta) = \sum_{i=1}^{n} \sum_{k=1}^{g} z_{ik} \log\left(\pi_k f(x, \theta_k)\right)$

Pour cela, il suffit de procéder de la manière suivante :

- Initialisation au hasard de $\theta^{(0)}$
- $c = 0$
- Tant que l'algorithme n'a pas convergé, faire
  - $z^{(c+1)} = \arg\max_z \left(L\left(x, z; \theta^{(c)}\right)\right)$
  - $\theta^{(c+1)} = \arg\max_\theta \left(L\left(x, z^{(c+1)}; \theta\right)\right)$
  - $c = c + 1$
- Fin

Lorsque les composantes du mélange appartiennent à la même famille exponentielle, en utilisant la bijection entre les divergences de Bregman et les familles exponentielles, on obtient l'algorithme k-MLE[3].

4.3 Algorithme SEM

Afin de réduire le risque de tomber dans un maximum local de vraisemblance, Celeux et Diebolt (1985)[4] proposent d'intercaler une étape stochastique de classification entre les étapes E et M. Après le calcul des probabilités $t_{ik}^{(c)}$, l'appartenance $z_{ik}^{(c)}$ des individus aux classes est tirée aléatoirement selon une loi multinomiale de paramètres $\mathcal{M}\left(1, t_{i1}^{(q)}, \ldots, t_{ig}^{(q)}\right)$.

Contrairement à ce qui se produit dans l'algorithme CEM, on ne peut considérer que l'algorithme a convergé lorsque les individus ne changent plus de classes. En effet, celles-ci étant tirées aléatoirement, la suite $\left(z^{(q)}, \theta^{(q)}\right)$ ne converge pas au sens strict. En pratique, Celeux et Diebolt (1985) proposent de lancer l'algorithme SEM un nombre de fois donné puis d'utiliser l'algorithme CEM pour obtenir une partition et une estimation du paramètre θ .

5 Voir aussi

- Partitionnement de données

6 Références

[1] (en) A.P. Dempster, N.M. Laird et Donald Rubin, « Maximum Likelihood from Incomplete Data via the EM Algorithm », *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, n° 1, 1977, p. 1–38 (JSTOR 2984875)

[2] (en) G. Celeux et G. Govaert, « A classification EM algorithm for clustering and two stochastic versions », *Computational Statistics Quarterly*, vol. 2, n° 1, 1991, p. 73–82

[3] (en) Frank Nielsen, « k-MLE : A fast algorithm for learning statistical mixture models », *arxiv (ICASSP 2012)*, 2012 (lire en ligne)

[4] (en) G. Celeux et G. Diebolt, « The sem algorithm : a probabilistic teacher algorithm derived from the em algorithm for the mixture problem », *Rapport de recherche RR-1364, Inria, Institut National de Recherche en Informatique et en Automatique*, 1985

-  Portail des probabilités et de la statistique
-  Portail de l'informatique théorique

7 Sources, contributeurs et licences du texte et de l'image

7.1 Texte

- **Algorithme espérance-maximisation** *Source :* https://fr.wikipedia.org/wiki/Algorithme_esp%C3%A9rance-maximisation?oldid=130744168 *Contributeurs :* Francis.sourd, MedBot, Phe-bot, Jef-Infojef, Gus2, Dake, Rama, Kipmaster, Chris93, Ripounet, Seb35, GôTô, EDUCA33E, YurikBot, Frelaur, Rilou2000, Sylenius, Liquid-aim-bot, Christophe Genolini, Rprl, NicoV, Jarfe, Le Pied-bot, Sebleouf, Babelborg, Numbo3, Nodulation, Chicobot, Joris.deguet, Gonzolito, Laddo, ZX81-bot, Ambigraphe, Love Sun and Dreams, Alecs.bot, Alexbot, ZetudBot, LaaknorBot, Luckas-bot, JackBot, Romainbrasselet, Jackverr, PAC2, EmausBot, Rezabot, Saimn, Pitoresk18, Nochnix, Zooky, JYBot, YFdyh-bot, Roll-Morton, Addbot, AméliorationsModestes, Everouane, Hypatie 64 et Anonyme : 28

7.2 Images

- **Fichier:Logo_proba_4.svg** *Source :* https://upload.wikimedia.org/wikipedia/commons/f/f7/Logo_proba_4.svg *Licence :* CC BY-SA 3.0 *Contributeurs :* Travail personnel *Artiste d'origine :* Ipipipourax
- **Fichier:Max-cut.svg** *Source :* https://upload.wikimedia.org/wikipedia/commons/c/cf/Max-cut.svg *Licence :* CC BY-SA 3.0 *Contributeurs :* Travail personnel *Artiste d'origine :* Miym

7.3 Licence du contenu

- Creative Commons Attribution-Share Alike 3.0