Estimation jointe et en ligne de modèles de mélanges avec les co-mélanges et les sacs de composantes

Olivier SCHWANDER[1], Frank NIELSEN[2]

[1]Groupe Viper, Laboratoire Vision par ordinateur et multimédia,
Université de Genève, Suisse

[2]Laboratoire d'Informatique (LIX, UMR 7161),
École Polytechnique, Palaiseau, France

olivier.schwander@unige.ch, nielsen@lix.polytechnique.fr

Résumé – Cet article introduit la notion nouvelle de co-mélange, un ensemble de modèles de mélange partageant les mêmes composantes, et présente plusieurs techniques pour apprendre et manipuler cette notion. Deux algorithmes pour estimer les paramètres d'un co-mélange sont proposés: l'un basé sur Espérance-Maximisation et l'autre, en ligne, utilisant un dictionnaire. Le partage des composantes est exploité dans une version rapide de l'approximation variationnelle de la divergence de Kullback-Leibler entre modèles de mélanges. Quelques résultats préliminaires pour l'évaluation des méthodes proposées sont présentés, en utilisant un jeu de données artificiel, généré à partir de modèles de mélange construits aléatoirement.

Abstract – This article introduces the novel notion of co-mixture, a set of mixture models sharing the same components, and presents several techniques to build and manipulate this new notion. Two algorithms for parameter estimation are proposed: a first one relying on Expectation-Maximization and the other one, working online, using a dictionary. The sharing between components is used for a fast version of the variational approximation of the Kullback-Leibler divergence between mixture models. A few preliminary results about the evaluation of the proposed methods are shown, using an artificial dataset, generated from random mixture models.

1 Introduction et motivation

L'estimation d'une densité de probabilité inconnue est un problème ancien et bien étudié et les modèles de mélange font parties des méthodes ayant le plus de succès. Beaucoup de travaux ont dès lors été consacrés à l'amélioration de la rapidité de l'estimation des paramètres des mélanges, ce qui est particulièrement intéressant pour des applications temps réel, comme le suivi d'objets dans des vidéos [SG99; Che+11; SN11].

La plupart des recherches sur les mélanges peuvent se diviser en trois catégories principales. D'abord, réduire le coût des algorithmes: par exemple k-MLE [Nie12] et cEM [CG92] remplacent les poids par des valeurs binaires pour accélérer le calcul. Ensuite, on peut travailler sur les données elles-même, en réduisant le nombre de points nécessaires pour estimer le modèle [BGP13; FFK11]. Enfin, on peut faire en sorte de stocker toutes les observations en utilisant un algorithme en ligne [SG99; CM08].

L'approche décrite ici présente un nouvel angle de vue: plutôt que s'intéresser à ce qu'on peut faire avec un jeu de données pour un unique mélange, on va étudier l'estimation jointe de plusieurs modèles de mélanges, sur plusieurs jeux de données. Pour permettre la manipulation efficace de ces mélanges, on impose la contrainte du partage des paramètres des composantes entre tous les mélanges, le seul degré de liberté étant donc le vecteur de poids de chaque mélange. Deux applications notables sont proposées: une version rapide, utilisant des précalculs permis par le partage, de l'approximation variationnelle de la divergence de Kullback-Leibler entre mélanges [HO07] et un algorithme en ligne pour apprendre des mélanges de ce type, utilisant un dictionnaire de composantes.

Les contributions sont les suivantes: d'abord on définit la notion de *co-mélange*, un ensemble de mélanges partageant les mêmes composantes; puis on décrit une variante d'Espérance-Maximisation (EM), baptisée *co-EM*, pour apprendre un co-mélange, donc pour estimer les paramètres des composantes partagées et les poids de chaque mélange; ensuite on propose la version rapide de la divergence Kullback-Leibler; et enfin on introduit une méthode en ligne pour les co-mélanges, baptisée *Sac de composantes*, qui utilise un dictionnaire fixé de composantes estimées à l'aide de co-EM.

2 Co-mélanges

Un co-mélange de familles exponentielles est un ensemble de mélanges partageant les mêmes paramètres pour leurs composantes:

$$\begin{cases} m_1(x;\omega_1^{(1)}\dots\omega_K^{(1)}) = \sum_{i=1}^K \omega_i^{(1)} p_F(x;\theta_i) \\ m_2(x;\omega_1^{(2)}\dots\omega_K^{(2)}) = \sum_{i=1}^K \omega_i^{(2)} p_F(x;\theta_i) \\ \dots \\ m_S(x;\omega_1^{(S)}\dots\omega_K^{(S)}) = \sum_{i=1}^K \omega_i^{(S)} p_F(x;\theta_i) \end{cases} \quad (1)$$

où p_F est la densité de la famille exponentielle de log-normaliseur F et où les $\theta_1 \dots \theta_K$ sont les paramètres des composantes. Les S vecteurs $\omega_1^{(s)} \dots \omega_K^{(s)}$ sont les poids de chaque mélange (donc positifs et normalisés à 1).

L'idée principale des co-mélanges est d'exploiter les similarités entre les différents jeux de donnés. On peut voir la construction d'un co-mélange comme une sorte d'apprentissage par transfert: par exemple pour pallier le manque d'observations dans l'un des jeux de données. On peut aussi directement étudier les poids pour mettre en lumière des similarités ou des différences: une composante avec un poids élevé pour tous les mélanges dénotera un point commun, tandis qu'une composante activée par peu de mélanges dénotera une différence.

Un autre avantage est évidemment la réduction de l'occupation mémoire, grâce à la réduction du nombre de paramètres à stocker.

3 Co-Espérance-Maximisation

L'approche Espérance-Maximisation [DLR77] est utilisable pour estimer les paramètres d'un co-mélange. Dans le cas d'un unique mélange, EM maximise (localement) la log-vraisemblance du mélange: ici, on va maximiser la moyenne des log-vraisemblances des différents éléments du co-mélange. On obtient un algorithme itératif en trois étapes principales.

Espérance La première étape estime la probabilité pour chaque observation de provenir de telle composante de tel mélange. Elle calcule donc la matrice de responsabilités de chaque jeu de données l ($1 \leq l \leq S$):

$$p^{(l)}(i,j) = \frac{\omega_j^{(l)} p_F(x_i, \eta_j)}{m(x_i^{(l)}|\omega^{(l)}, \eta)} \quad (2)$$

Maximisation (jeu de donnée par jeu de donnée) Dans une premier temps, on réalise la maximisation pour chaque jeu de donnée individuellement. En l'écrivant comme un barycentre Bregman [Ban+05] à l'aide d'une statistique suffisante t, on obtient:

$$\eta_j^{(l)} = \sum_i \frac{p^{(l)}(i,j)}{\sum_u p^{(l)}(u,j)} t(x_i^{(l)}) \quad (3)$$

Maximisation (agrégation) La finalisation de l'étape de maximisation s'écrit comme un barycentre des S paramètres construits précédemment:

$$\eta_j = \frac{1}{S} \sum_{l=1}^S \eta_j^{(l)} \quad (4)$$

Cet algorithme nécessite de stocker en mémoire les observations de tous les jeux de données mais se parallèlise facilement puisque les calculs sur chaque jeu peuvent se faire indépendemment: seule la dernière étape doit être centralisée (mais nécessite relativement peu de transferts mémoire, puisqu'on ne manipule que les $\eta_j^{(l)}$).

La Figure 1 présente un exemple de co-segmentation de plusieurs images réalisée avec co-EM.

4 Approximation rapide de Kullback-Leibler

La divergence de Kullback-Leibler [Kul97] est l'entropie de Shannon relative entre deux ditributions:

$$\mathrm{KL}(p\|q) = \int p(x) \log \frac{p(x)}{q(x)} \mathrm{d}x \quad (5)$$

$$= \int p(x) \log p(x) \mathrm{d}x - \int p(x) \log q(x) \mathrm{d}x \quad (6)$$

$$= H(p,q) - H(p) \quad (7)$$

où $H(p)$ est l'entropie de Shannon et $H(p,q) \geq H(p)$ est l'entropie croisée.

C'est une mesure de dissimilarité entre distributions très utilisée en pratique dans de nombreuses applications (par exemple [GGG03]), mais avec l'inconvénient de l'absence de forme close (connue) entre deux modèles de mélanges. Beaucoup de travaux ont donc été consacré à l'approximation efficace et précise de cette divergence [HO07; DTK12]. Parmi celles-ci, l'approche variationnelle [HO07] est la plus prometteuse:

$$\mathrm{KL}_{\mathrm{var}}(m_1\|m_2) = \sum_i \omega_i^{(1)} \log \frac{\sum_j \omega_j^{(1)} e^{-\mathrm{KL}(p_F(\cdot;\theta_i)\|p_F(\cdot;\theta_j))}}{\sum_j \omega_j^{(2)} e^{-\mathrm{KL}(p_F(\cdot;\theta_i)\|p_F(\cdot;\theta_j))}} \quad (8)$$

Le calcul de cette approximation nécessite de calculer les divergences entre toutes les paires de composantes pour les deux mélanges considérés, ce qui peut devenir prohibitif lorsque beaucoup d'évaluations de la divergence sont nécessaires.

Dans le cas d'un co-mélange, puisque les composantes sont partagées, on peut pré-calculer ces divergences entre composantes et les stocker dans la matrice:

$$D(i,j) = \mathrm{KL}(p_F(\cdot;\theta_i)\|p_F(\cdot;\theta_j)) \quad (9)$$

et l'approximation variationnelle devient:

$$\mathrm{KL}_{\mathrm{var}}(m_1\|m_2) = \sum_i \omega_i^{(1)} \log \frac{\sum_j \omega_j^{(1)} e^{-D_{ij}}}{\sum_j \omega_j^{(2)} e^{-D_{ij}}} \quad (10)$$

Figure 1: Segmentation avec EM et co-segmentation avec co-EM (sur des points RGBxy).

5 Algorithme en ligne

On peut facilement construire un algorithme en ligne, baptisé *Sac de composantes*, à partir du résultat de co-EM. En estimant un co-mélange et en oubliant les vecteurs de poids, on obtient un dictionnaire de composantes:

$$\mathcal{D} = \{\theta_1, \ldots, \theta_K\} \tag{11}$$

Après cette phase d'apprentissage (effectuée une unique fois), on peut construire une étape d'estimation où chaque observation est associée à l'élément du dictionnaire le plus probable:

$$\hat{\imath} = \arg\max_{\theta \in \mathcal{D}} p_F(x_j, \theta) \tag{12}$$

En comptant, le nombre de points associés à chaque composante, on peut alors reconstruire un vecteur de poids.

6 Expériences

Les expériences sont réalisées sur des jeux de données artificiels, générés à partir de mélanges de Gaussiennes construites aléatoirement. On tire d'abord un dictionnaire de composantes, puis des poids avec seulement 30% de valeurs non nulles. Dans les résultats qui suivent, 10 mélanges de Gaussiennes multivariées (à 10 composantes non-nulles tirées depuis un dictionnaire à 30 éléments) sont utilisées pour générer chacune 1000 points.

La Table 1 présente les temps de calcul pour les différentes étapes de l'estimation des paramètres des modèles. Les deux premières lignes montrent que le temps de calcul d'un co-mélange à 10 mélanges avec co-EM est sensiblement le même que le temps de calcul pour réaliser 10 EM pour 10 mélanges différents. Pour les calculs de la divergence de Kullback-Leibler, on observe une accélération d'environ 100 fois par rapport à la version classique (en tenant compte à la fois du pré-calcul et du calcul).

	Dim 2	Dim 10
$10 \times$ EM	210s	220s
co-EM	230s	200s
KL variationnel classique	2.1s	2.3s
Précalcul	0.012s	0.012s
KL variationnel rapide	0.10s	0.084s

Table 1: Temps de calcul pour les divergences en dimensions 2 et 10.

La Figure 2 étudie les temps de calcul et la qualité (en terme de log-vraisemblance relative $\frac{\mathrm{ll}_{\mathrm{BoC}} - \mathrm{ll}_{\mathrm{EM}}}{\mathrm{ll}_{\mathrm{BoC}}}$) de l'algorithme Sac de composantes, en fonction du nombre d'observations. Sans surprise, Sac de composantes est parfaitement linéaire, et EM est quasi-linéaire. Les temps de calcul relatifs peuvent paraître surprenant, puisque l'algorithme Sac de composantes semble beaucoup plus simple que EM, mais la recherche linéaire a en fait un coût du même ordre que l'étape E d'un EM: lorsque EM converge en peu d'itérations, on peut donc avoir un coût similaire à la méthode Sac de composantes. La qualité reste très proche de ce qu'on obtient avec EM, avec environ 3% d'écart de log-vraisemblance relative.

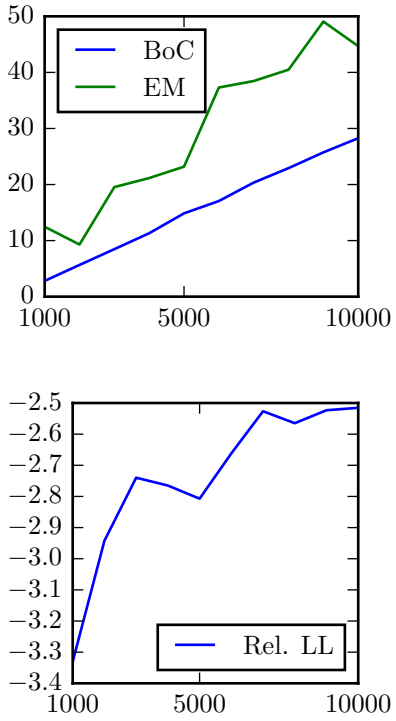

Figure 2: Temps de calcul pour EM et Sac de composantes (en secondes, à gauche) et log-vraisemblance relative (en pourcentage, à droite) par rapport aux nombres d'observations (entre 1000 et 10000, en dimension 5).

7 Conclusion

On a introduit la notion de co-mélange, un ensemble de modèles de mélange partageant les mêmes composantes et où seuls les poids changent d'un mélange à l'autre. Ces co-mélanges peuvent être appris à l'aide d'une variante de EM baptisée co-EM. Deux applications sont proposées: d'abord une version rapide de l'approximation variationnelle de Kullback-Leibler, précalculant des valeurs intermédiaires grâce au partage des composantes; puis un algorithme en ligne, qui utilise les composantes partagées comme un dictionnaire.

Ces travaux laissent la porte ouverte à de nombreuses perspectives. Dans un premier temps, il faudra évaluer les algorithmes et les méthodes sur une application réelle. Les co-mélanges devraient pouvoir également être utilisés pour de l'apprentissage par transfert. Enfin, les méthodes à dictionnaire pourront apporter des éléments pour améliorer l'algorithme en ligne.

References

[Ban+05] A. Banerjee, S. Merugu, I. S. Dhillon et J. Ghosh. "Clustering with Bregman divergences". In: *The Journal of Machine Learning Research* (2005).

[BGP13] Anthony Bourrier, Rémi Gribonval et Patrick Pérez. "Compressive Gaussian Mixture Estimation". In: *ICASSP - 38th International Conference on Acoustics, Speech, and Signal Processing*. 2013.

[CG92] Gilles Celeux et Gerard Govaert. "A Classification EM algorithm for clustering and two stochastic versions". In: *Comput. Stat. Data Anal.* 3 (1992).

[Che+11] Gang Chen, Zhezhou Yu, Qing Wen et Yangquan Yu. "Improved Gaussian Mixture Model for Moving Object Detection". English. In: *Artificial Intelligence and Computational Intelligence*. 2011.

[CM08] Olivier Cappé et Eric Moulines. "Online EM algorithm for latent data models". In: *Journal of the Royal Statistical Society* (2008).

[DLR77] Arthur P. Dempster, Nan M. Laird et Donald B. Rubin. "Maximum likelihood from incomplete data via the EM algorithm". In: *Journal of the Royal Statistical Society. Series B (Methodological)* (1977).

[DTK12] J.-L. Durrieu, J.-P. Thiran et F. Kelly. "Lower and upper bounds for approximation of the Kullback-Leibler divergence between Gaussian Mixture Models". In: *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2012.

[FFK11] Dan Feldman, Matthew Faulkner et Andreas Krause. "Scalable training of mixture models via coresets". In: *Advances in Neural Information Processing Systems*. 2011.

[GGG03] Jacob Goldberger, Shiri Gordon et Hayit Greenspan. "An efficient image similarity measure based on approximations of KL-divergence between two Gaussian mixtures". In: *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*. IEEE. 2003.

[HO07] J.R. Hershey et P.A. Olsen. "Approximating the Kullback Leibler Divergence Between Gaussian Mixture Models". In: *IEEE International Conference on Acoustics, Speech and Signal Processing, 2007. ICASSP 2007*. 2007.

[Kul97] Solomon Kullback. *Information theory and statistics*. 1997.

[Nie12] Frank Nielsen. "k-MLE: A fast algorithm for learning statistical mixture models". In: *CoRR* (2012).

[SG99] Chris Stauffer et W Eric L Grimson. "Adaptive background mixture models for real-time tracking". In: *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.* IEEE. 1999.

[SN11] Ronan Sicre et Henri Nicolas. "Improved Gaussian Mixture Model for the Task of Object Tracking". English. In: *Computer Analysis of Images and Patterns*. 2011.