Calcul de probabilités et estimation dans des modèles de Markov cachés graphiques

Jean-Baptiste Durand

Calcul de probabilités et estimation dans des modèles de Markov cachés graphiques

Jean-Baptiste Durand

Grenoble Universités – INRIA –
Laboratoire Jean Kuntzmann
51 rue des Mathématiques
B.P.53
38 041 Grenoble cedex 9

Résumé : nous présentons une classe de modèles de Markov cachés définis à partir de modèles graphiques, pour lesquels nous donnons des algorithmes efficaces et numériquement stables de calcul de probabilités, permettant le calcul des probabilités de lissage intervenant dans l'algorithme EM pour l'estimation des paramètres, et basés sur des quantités intermédiaires avec une interprétation probabiliste explicite.
Mots-clés : Modèles graphiques, Statistique des processus

Abstract : we show how to define a class of hidden Markov models based on graphical models. Efficient and numerically stable algorithms are derived, that allow to compute the smoothed probabilities involved in the EM algorithm for parameter estimation, and are based on intermediate quantities with explicit probabilistic interpretation.
Keywords : Graphical models, Statistics of random processes

1 Problématique du calcul de probabilités dans les modèles graphiques

Le formalisme des modèles probabilistes graphiques, bien que remontant au début du XXème siècle, a été établi dans les années 1980-1990, notamment à travers les travaux de Pearl (1982) et Lauritzen (1996). L'intérêt des modèles graphiques est de pemettre de définir des relations d'indépendance conditionnelle entre les variables aléatoires d'un processus indexé par un graphe. L'une des difficultés pour manipuler de tels modèles est d'arriver à concevoir des algorithmes efficaces de calcul de probabilités. Suivant la nature du graphe (non-orienté ou orienté sans circuit, OSC), la paramétrisation du processus a une interprétation probabiliste directe ou non. Dans le cas non-orienté, la paramétrisation canonique du modèle fait intervenir des «potentiels de clique» qui n'ont pas d'interprétation

probabiliste directe. Dans le cas orienté, le modèle est paramétré de manière naturelle par les probabilités de transition des sommets parents vers le descendant.

Dès les années 1990, deux algorithmes ont été proposés pour calculer des probabilités dans ces modèles. Tous deux se basent sur une hypothèse de graphe triangulé (quitte à ne pas tirer parti de certaines relations d'indépendance conditionnelle), puis sur un parcours d'arbre dit «de jonction» joignant les cliques[1] du graphe. L'existence d'un arbre de jonction est équivalente au caractère triangulé du graphe – voir par exemple Smyth *et al.* (1997).

L'algorithme de Lucke (1996), dédié aux modèles graphiques non-orientés, se base sur une paramétrisation du modèle par la loi jointe de chaque clique, avec des contraintes assurant la cohérence de cette paramétrisation vis-à-vis de la loi jointe du processus complet. Son avantage est que toutes les quantités calculées par l'algorithme sont des grandeurs probabilistes. L'un de ses inconvénients est la paramétrisation utilisée qui n'a pas d'intérêt pratique, et le fait que le cas des graphes OSC ne soit pas abordé.

L'algorithme de Jensen *et al.* (1990) s'applique en premier lieu aux graphes non-orientés, mais propose une méthode pour passer du cas des graphes OSC aux graphes non-orientés. Son inconvénient majeur est que les quantités intermédiaires calculées n'ont pas d'interprétation probabiliste. Il est donc utilisé comme une boîte noire : on donne une valeur $\boldsymbol{x}$ au processus $\boldsymbol{X}$ en entrée, puis on obtient $P(\boldsymbol{X} = \boldsymbol{x})$ en sortie, mais les calculs intermédiaires ne sont pas utilisables. Ainsi si l'on souhaite calculer $P(\boldsymbol{X} = \boldsymbol{x}')$ pour $\boldsymbol{x} \neq \boldsymbol{x}'$ on doit *a priori* tout recommencer. Par ailleurs on n'a pas accès à l'expression de $P(\boldsymbol{X} = \boldsymbol{x})$ en fonction des paramètres du modèle. Ceci exclut par exemple le calcul de scores, ou même de $E[Z|\boldsymbol{X}' = \boldsymbol{X}']$, où Z est l'une des variables aléatoires et $\boldsymbol{X}'$ un sous-ensemble de variables aléatoires de $\boldsymbol{X}$ – dans le cas où cette espérance est définie par une intégrale.

Enfin, ces deux algorithmes (dits «d'arbre de jonction») ont l'inconvénient de calculer directement des probabilités jointes, ce qui rend l'algorithme numériquement instable quand la taille du graphe est élevée. Si l'on peut disposer d'une interprétation probabiliste des quantités calculées par les algorithmes d'arbre de jonction, on pourra calculer des probabilités conditionnelles pour éviter de manipuler des probabilités jointes qui tendent vers zéro (voir Devijver (1985) dans le cas de chaînes de Markov cachées).

Le cadre des modèles graphiques est une approche privilégiée pour définir des modèles de Markov cachés – justement pour traduire l'indépendance conditionnelle, sachant ses voisins, d'un sommet et des autres sommets (aspect markovien). Ils combinent donc le problème de calcul de probabilités et celui de l'estimation, dans un contexte où une partie des variables aléatoires est manquante. Par ailleurs, l'étude de modèles de Markov cachés repose en général sur la possibilité de donner une valeur à ces variables manquantes, notamment en déterminant la valeur du processus caché de probabilité maximale sachant les observations (principe du Maximum A Posteriori ou MAP).

[1] Une clique est un sous-graphe complet et maximal.

Notre objectif est de proposer :
- un formalisme permettant de définir, à partir de modèles graphiques OSC, des modèles de Markov cachés avec une paramétrisation interprétable en termes probabilistes ;
- un algorithme numériquement stable qui permette, dans le cas de graphes triangulés, de calculer des probabilités de manière efficace, et d'interpréter et réutiliser les quantités intermédiaires calculées, notamment en disposant de leur expression explicite en fonction des paramètres ;
- une méthode pour estimer les paramètres par maximum de vraisemblance (basée sur l'algorithme EM de Dempster *et al.*, 1977) ;
- un algorithme du MAP pour restaurer les données manquantes.

2 Définition du modèle et notations

On définit un modèle de Markov caché graphique comme un processus $\boldsymbol{Y}$ tel qu'il existe un processus $\boldsymbol{S}$ (processus caché) à valeurs finies, vérifiant les propriétés suivantes :

1. le processus $\boldsymbol{X} = (\boldsymbol{Y}, \boldsymbol{S})$ est un modèle graphique OSC. En particulier, les variables aléatoires sont indexées par l'ensemble $\mathcal{U}$ des sommets d'un graphe $\mathcal{G} = (\mathcal{U}, \mathcal{E})$;
2. d'après Smyth *et al.* (1997), la loi de $\boldsymbol{X}$ se factorise comme suit :

$$P(\boldsymbol{X} = \boldsymbol{x}) = \prod_{u \in \mathcal{U}} P(X_u = x_u | \boldsymbol{X}_{\mathrm{pa}(u)} = \boldsymbol{x}_{\mathrm{pa}(u)}) \tag{1}$$

   où $\mathrm{pa}(u)$ est l'ensemble des parents de u et pour tout $U' \subset U$, $\boldsymbol{X}_{U'} = (X_u)_{u \in U'}$. On utilise la convention $P(X_u = x_u | \boldsymbol{X}_{\mathrm{pa}(u)} = \boldsymbol{x}_{\mathrm{pa}(u)}) = P(X_u = x_u)$ si $\mathrm{pa}(u) = \emptyset$. Ainsi, la loi $P_{\boldsymbol{Y}}$ est paramétrée par les probabilités de transition $p^{(u)}_{\boldsymbol{a},x} = P(X_u = x \mid \boldsymbol{X}_{\mathrm{pa}(u)} = \boldsymbol{a})$;
3. la loi de $\boldsymbol{Y}$ est définie par $P(\boldsymbol{Y} = \boldsymbol{y}) = \sum_{\boldsymbol{s} \in \mathcal{S}} P(\boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{S} = \boldsymbol{s})$, où $P(\boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{S} = \boldsymbol{s})$ est donné par l'équation (1), et où $\mathcal{S}$ désigne l'ensemble des valeurs du processus caché.

Pour simplifier les notations, on suppose que si une variable Y_u est continue, tous ses parents sont à valeurs finies. On fait alors l'hypothèse que la loi de Y_u sachant $\boldsymbol{X}_{\mathrm{pa}(u)} = \boldsymbol{a}$ est la loi $P_{\theta_{\boldsymbol{a}}}$, $(P_\theta)_{\theta \in \Theta}$ étant une famille de lois paramétrique, identifiable au sens des mélanges. De même que dans Lucke (1996), on suppose que le graphe $\mathcal{G}$ admet un arbre de jonction. Cette dernière hypothèse est assez restrictive, vu que $\mathcal{G}$ est déjà supposé orienté et sans circuit ; cependant, elle couvre plusieurs modèles d'intérêt, notamment les arbres de Markov cachés (voir Crouse *et al.*, 1998).

Un arbre de jonction est un arbre non orienté dont les sommets sont les cliques de $\mathcal{G}$ et tel que pour toute paire $\{\mathcal{C}, \mathcal{C}'\}$, l'intersection des cliques $\mathcal{C}$ et $\mathcal{C}'$ est contenue dans toute clique du chemin entre $\mathcal{C}$ et $\mathcal{C}'$ - voir Smyth *et al.* (1997).

Dans la perspective d'estimer les paramètres λ du modèle à partir d'une seule réalisation du graphe, on fait une hypothèse d'homogénéité du processus, c'est-à-dire qu'un paramètre $p^{(u)}_{\boldsymbol{a},x}$ est le même pour tous les sommets u dont les parents prennent leurs valeurs dans un même ensemble (on note $\bar{u}$ l'ensemble des sommets u vérifiant cette condition).

3 Algorithmes d'estimation des paramètres, de calcul de probabilités et du MAP

3.1 Estimation des paramètres

Pour estimer λ par maximum de vraisemblance à partir d'une réalisation $\boldsymbol{y}$ de $\boldsymbol{Y}$, il est naturel d'utiliser l'algorithme EM de Dempster *et al.* (1977), vu l'existence de variables inobservées $\boldsymbol{S}$. On construit ainsi une suite $(\lambda^{(m)})_{m\in\mathbb{N}}$ de vraisemblance croissante.

D'après Smyth *et al.* (1997), la formule de réestimation de $p_{\boldsymbol{a},i}$ à l'itération m de l'algorithme EM est :

$$p^{(m)}_{\boldsymbol{a},i} = \frac{\sum_{u\in\bar{u}} P_{\lambda^{(m-1)}}(X_u = i, \boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}} | \boldsymbol{Y} = \boldsymbol{y})}{\sum_{u\in\bar{u}} P_{\lambda^{(m-1)}}(\boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}} | \boldsymbol{Y} = \boldsymbol{y})}$$

où pour tout $U' \subset U$, $\boldsymbol{a}_{U'}$ désigne la restriction de $\boldsymbol{a}$ à U', $\mathcal{U}_{\boldsymbol{Y}}$ l'ensemble des sommets observés et $\mathcal{U}_{\boldsymbol{S}}$ l'ensemble des sommets cachés.

On montre par un principe analogue que $\theta_{\boldsymbol{a}}$ vérifie l'équation

$$\sum_{u\in\bar{u}} P_{\lambda^{(\eta-1)}}(\boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}} | \boldsymbol{Y} = \boldsymbol{y}) \frac{\nabla_{\theta_{\boldsymbol{a}}}[P_{\theta_{\boldsymbol{a}}}(y_u)]}{P_{\theta_{\boldsymbol{a}}}(y_u)} = 0$$

Il reste encore à établir un algorithme permettant de calculer les quantités
$P(X_u = i, \boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}} | \boldsymbol{Y} = \boldsymbol{y})$.

3.2 Calcul de probabilités et restauration des états

La méthode ci-dessous repose sur un lemme concernant les graphes OSC triangulés : il existe une unique clique $\mathcal{C}_0$ telle que tout sommet de cette clique a tous ses (éventuels) parents dans $\mathcal{C}_0$. On oriente alors l'arbre de jonction en choisissant $\mathcal{C}_0$ comme racine. Chaque arc $a = (\mathcal{C}_i, \mathcal{C}_j)$ de cette arborescence la sépare en deux composantes, dont une seule ne contient pas $\mathcal{C}_0$. Nous notons $\mathcal{T}_a$ l'ensemble des cliques comprises dans cette partie, l'ensemble des autres cliques étant noté $\mathcal{T}_a^c$. Nous noterons $\mathcal{T}_a^-$ l'ensemble des cliques de $\mathcal{T}_a$ privé de la clique $\mathcal{C}_j$ et $\mathcal{T}_a^{c-}$ l'ensemble des cliques de $\mathcal{T}_a^c$ privé de la clique $\mathcal{C}_i$. Les sommets des cliques de $\mathcal{T}_a$ engendrent un graphe dont l'ensemble des sommets est noté $\bar{\mathcal{K}}_a$. Nous notons $\mathcal{K}_a = \bar{\mathcal{K}}_a \setminus \mathcal{S}_a$ (où $\mathcal{S}_a$ désigne le graphe engendré par l'ensemble des sommets de $\mathcal{C}_i \cap \mathcal{C}_j$, ou par raccourci ses sommets eux-mêmes).

Nous proposons un algorithme récursif en trois étapes : dans un premier temps, on calcule la loi des cliques $\boldsymbol{X}_{\mathcal{C}_j}$ en fonction de celle de $\boldsymbol{X}_{\mathrm{pa}(\mathcal{C}_j)}$.
Dans un deuxième temps, on calcule les quantités

$$\beta_a(\boldsymbol{x}_{\mathbb{S}_a}) = \frac{P(\boldsymbol{X}_{\mathbb{S}_a} = \boldsymbol{x}_{\mathbb{S}_a} | \boldsymbol{Y}_{\mathcal{K}_a} = \boldsymbol{y}_{\mathcal{K}_a})}{P(\boldsymbol{X}_{\mathbb{S}_a} = \boldsymbol{x}_{\mathbb{S}_a})}$$

par un parcours de l'arbre de jonction, en commençant par les feuilles et en remontant vers la racine (récursion «arrière»). Les formules de propagation sont :

$$\begin{aligned}
\gamma_a(\boldsymbol{x}_{\mathbb{S}_a}) &= \sum_{\boldsymbol{s}_{\mathcal{C}_j \setminus \mathbb{S}_a} \in \mathcal{S}_{\mathcal{C}_j \setminus \mathbb{S}_a}} P(\boldsymbol{X}_{\mathcal{C}_j \setminus \mathbb{S}_a} = \boldsymbol{x}_{\mathcal{C}_j \setminus \mathbb{S}_a} | \boldsymbol{X}_{\mathbb{S}_a} = \boldsymbol{x}_{\mathbb{S}_a}) \prod_l \beta_{a_{j_l}}(\boldsymbol{x}_{\mathbb{S}_{a_{j_l}}}), \\
\mathcal{N}_a &= \sum_{\tilde{\boldsymbol{x}}_{\mathbb{S}_a}} P(\boldsymbol{X}_{\mathbb{S}_a} = \tilde{\boldsymbol{x}}_{\mathbb{S}_a}) \gamma_a(\tilde{\boldsymbol{x}}_{\mathbb{S}_a}), \\
\text{et } \beta_a(\boldsymbol{x}_{\mathbb{S}_a}) &= \frac{\gamma_a(\boldsymbol{x}_{\mathbb{S}_a})}{\mathcal{N}_a}.
\end{aligned}$$

Dans un troisième temps, on calcule les quantités dites de «lissage» $P(\boldsymbol{S}_{\mathcal{C}_j} | \boldsymbol{Y} = \boldsymbol{y})$ (qui interviennent dans l'algorithme EM) par un parcours de l'arbre de jonction, en commençant par la racine et en descendant vers les feuilles (récursion «avant»). La formule de propagation, qui fait intervenir les résultats de la récursion arrière, est :

$$\begin{aligned}
&P(\boldsymbol{S}_{\mathcal{C}_j} = \boldsymbol{s}_{\mathcal{C}_j} | \boldsymbol{Y} = \boldsymbol{y}) \\
&\quad = \left[\frac{P(\boldsymbol{X}_{\mathcal{C}_j \setminus \mathbb{S}_a} = \boldsymbol{x}_{\mathcal{C}_j \setminus \mathbb{S}_a} | \boldsymbol{X}_{\mathbb{S}_a} = \boldsymbol{x}_{\mathbb{S}_a})}{\mathcal{N}_a \beta_a(\boldsymbol{s}_{\mathbb{S}_a})} \prod_l \beta_{a_{j_l}}(\boldsymbol{s}_{\mathbb{S}_{a_{j_l}}}) \right] \sum_{\boldsymbol{s}_{\mathcal{C}_i \setminus \mathbb{S}_a} \in \mathcal{S}_{\mathcal{C}_i \setminus \mathbb{S}_a}} P(\boldsymbol{S}_{\mathcal{C}_i} = \boldsymbol{s}_{\mathcal{C}_i} | \boldsymbol{Y} = \boldsymbol{y})
\end{aligned}$$

La log-vraisemblance peut être calculée après la récursion arrière, par

$$\ln(P(\boldsymbol{Y} = \boldsymbol{y})) = \ln \left(\sum_{\boldsymbol{s}_{\mathcal{C}_0} \in \mathcal{S}_{\mathcal{C}_0}} P(\boldsymbol{X}_{\mathcal{C}_0} = \boldsymbol{x}_{\mathcal{C}_0}) \prod_l \beta_{a_l}(\boldsymbol{x}_{\mathbb{S}_{a_l}}) \right) + \sum_{a \in \mathcal{T}} \ln(\mathcal{N}_a)$$

Si on note $\mathcal{V}_{\mathcal{G}}$ l'ensemble des cliques de $\mathcal{G}$ et qu'on note, pour une clique C, taille(C) le nombre de valeurs prises par $C_{\mathcal{U}_{\boldsymbol{S}}}$, cet algorithme est de complexité (au pire) d'ordre $\mathcal{O}(\sum_{C \in \mathcal{V}_{\mathcal{G}}} \text{taille}(C))$, ce qui est la même complexité que celle de l'algorithme de Lucke (1996) (ou celui de Jensen *et al.*, 1990).

L'algorithme du MAP est similaire à une récursion arrière, où la somme serait remplacée par une maximisation vis-à-vis des états cachés.

4 Remarques de conclusion

Les récursions présentées généralisent en fait aux modèles de Markov cachés triangulés l'algorithme de Durand *et al.* (2004) concernant les arbres de Markov cachés.

L'hypothèse d'existence d'arbre de jonction pour un modèle graphique OSC est une hypothèse forte, qui permet de déterminer dans quel ordre effectuer les calculs récursifs. Elle permet aussi de décrire des graphes de manière générique en faisant croître le nombre de sommets. Ce problème d'ordre pratique demeure un obstacle à la description d'algorithmes d'inférence dans les modèles graphiques. Il existe en effet plusieurs modèles de Markov cachés graphiques OSC non-triangulés pour lesquels on sait concevoir des algorithmes récursifs de calcul de probabilités, et notamment les modèles M1-Mq de Muri (1997), ou les arbres de Markov cachés orientés des feuilles vers la racine. La difficulté d'unifier ces algorithmes d'estimation, qui sont pourtant similaires, est liée à la difficulté de décrire de manière générique la manière dont croissent les graphes eux-mêmes.

Bibliographie

[1] Crouse, M.S., Nowak, R.D. et Baraniuk, R.G. – Wavelet-Based Statistical Signal Processing Using Hidden Markov Models. *IEEE Transactions on Signal Processing*, vol. 46, n° 4, avril 1998, pp. 886–902.

[2] Dempster, A.P., Laird, N.M. et Rubin, D.B. – Maximum Likelihood from Incomplete Data via the EM Algorithm, (with discussion). *Journal of the Royal Statistical Society Series B*, vol. 39, 1977, pp. 1–38.

[3] Devijver, P. A. – Baum's forward-backward Algorithm Revisited. *Pattern Recognition Letters*, vol. 3, 1985, pp. 369–373.

[4] Durand, J.-B., Gonçalvès, P. et Guédon, Y. – Computational Methods for Hidden Markov Tree Models – An Application to Wavelet Trees. *IEEE Transactions on Signal Processing*, vol. 52, n° 9, septembre 2004, pp. 2551–2560.

[5] Jensen, F.V., Lauritzen, S.L. et Olesen, K.G. – Bayesian updating in causal probabilistic networks by local computations. *Computational Statistical Quaterly*, vol. 5, n° 4, 1990, pp. 269–282.

[6] Lauritzen, S.L. – *Graphical Models.* – Clarendon Press, Oxford, United Kingdom, 1996.

[7] Lucke, H. – Which Stochastic Models Allow Baum-Welch Training ? *IEEE Transactions on Signal Processing*, vol. 44, n° 11, novembre 1996, pp. 2746–2756.

[8] Muri, Florence. – *Comparaison d'algorithmes d'identification de chaînes de Markov cachées et application à la détection de régions homogènes dans les séquences d'ADN.* – Thèse de doctorat, Université Paris-V, octobre 1997.

[9] Pearl, J. – Reverend Bayes on Inference Engines : A Distributed Hierarchical Approach. *In : AAAI Conference on Artificial Intelligence*, 1982, pp. 133–136.

[10] Smyth, P., Heckerman, D. et Jordan, M.I. – Probabilistic Independence Networks for Hidden Markov Probability Models. *Neural Computation*, vol. 9, n° 2, 1997, pp. 227–270.