Introduction aux modèles graphiques **2007/2008**

Cours 6 — 07 novembre 2007

Enseignant: Francis Bach *Scribe: Odalric-Ambrym Maillard, Mathieu Pradel*

6.1 Vecteurs Gaussiens

6.1.1 Inversion de matrices

Proposition 6.1. *Si* $M = \begin{pmatrix} E & F \\ G & H \end{pmatrix}$, *alors on a la formule suivante pour l'inverse :*

$$\begin{pmatrix} E & F \\ G & H \end{pmatrix}^{-1} = \begin{pmatrix} I & 0 \\ -H^{-1}G & I \end{pmatrix} \begin{pmatrix} (E - FH^{-1}G)^{-1} & 0 \\ 0 & H^{-1} \end{pmatrix} \begin{pmatrix} I & -FH^{-1} \\ 0 & I \end{pmatrix}$$

de laquelle on obtient

$$(H - GE^{-1}F)^{-1} = H^{-1} + H^{-1}G(E - FH^{-1}G)^{-1}FH^{-1}$$

$$(H - GE^{-1}F)^{-1}GE^{-1} = H^{-1}G(E - FH^{-1}G)^{-1}$$

Ces formules d'inversion sont très pratiques (et permettent notamment de calculer le déterminant $|M| = |H| \times |H - GE^{-1}F|$. En particulier, si H est grande et facile à inverser, et E petite, alors $(H - GE^{-1}F)^{-1}$ se calcule facilement par la première formule. Par exemple, si $H = I$,$E = I$ et $U = G = F^T$, $(I + UU^T)^{-1} = I - U(I + U^TU)^{-1}U^T$.

En utilisant cette formule, on écrit donc :

$$\begin{aligned} p(x_1, x_2) &= \frac{(2\pi)^{-\frac{n_1}{2}}(2\pi)^{-\frac{n_2}{2}}}{|\Sigma_{22}|^{\frac{1}{2}}|\Sigma/\Sigma_{22}|^{\frac{1}{2}}} \\ &\quad \exp[-\frac{1}{2}\begin{pmatrix} x_1 - \mu_1 \\ x_2 - \mu_2 \end{pmatrix}^T \begin{pmatrix} I & 0 \\ -\Sigma_{22}^{-1}\Sigma_{11} & I \end{pmatrix} \\ &\quad \times \begin{pmatrix} (\Sigma/\Sigma_{22})^{-1} & 0 \\ 0 & \Sigma_{22}^{-1} \end{pmatrix} \begin{pmatrix} I & -\Sigma_{12}\Sigma_{22}^{-1} \\ O & I \end{pmatrix} \begin{pmatrix} x_1 - \mu_1 \\ x_2 - \mu_2 \end{pmatrix}] \end{aligned}$$

D'où :

$$\begin{aligned} p(x_1, x_2) &= \frac{(2\pi)^{-\frac{n_1}{2}}}{|\Sigma_{22}|^{\frac{1}{2}}|} \frac{(2\pi)^{-\frac{n_2}{2}}}{\Sigma/\Sigma_{22}|^{\frac{1}{2}}} \\ &\quad exp(-\frac{1}{2}(x_1 - \mu_1 - \Sigma_{12}\sigma_{22}^{-1}(x_2 - \mu_2))^T(\Sigma/\Sigma_{22})^{-1}(x_1 - \mu_1 - \Sigma_{12}\sigma_{22}^{-1}(x_2 - \mu_2)) \\ &\quad -\frac{1}{2}(x_2 - \mu_2)^T\Sigma_{22}^{-1}(x_2 - \mu_2)) \end{aligned}$$

On reconnait alors un produit de deux gaussiennes.

Ainsi, $p(x_1, x_2) = \mathcal{N}(x_2|\mu_2, \Sigma_{22})\mathcal{N}(x_1|\mu_{1|2}, \Sigma_{1|2})$, avec

- $\mu_{1|2} = \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(x_2 - \mu_2)$
- $\Sigma_{1|2} = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}$

Condition de factorisation dans un graphe

Proposition 6.2. *Soit $G = (V, E)$ un modèle graphique non orienté.*

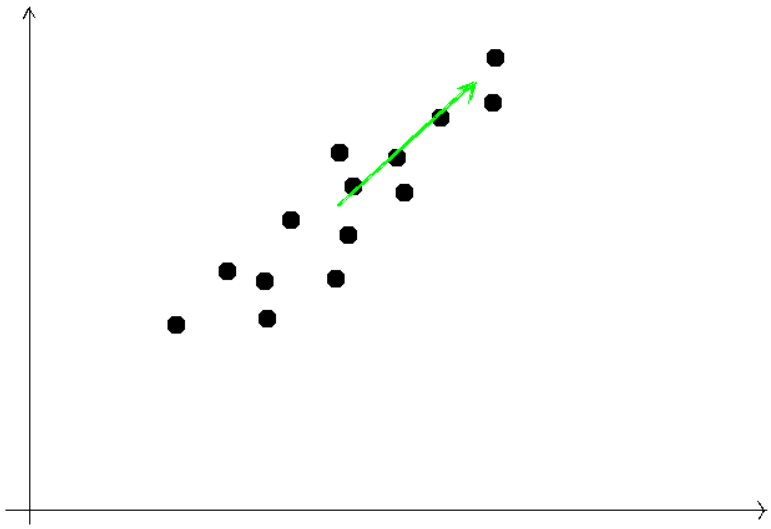
$$p(x) \in \mathcal{L}(G) \text{ si et seulement si } \forall (i,j) \in E \quad (\Sigma^{-1})_{i,j} = 0$$

Proof: En développant le produit matriciel, on constate que : $p(x) = C\Pi_{i,j} exp(-\frac{1}{2}(x_i - \mu_i)(x_j - \mu_j)(\Sigma^{-1})_{i,j})$. D'où la conclusion. □

Le DM3 montrera une CNS pour un résultat similaire dans le cas où G est un modèle graphique orienté.

6.2 Analyse factorielle

Cadre On suppose $X \sim \mathcal{N}(0, I) \in \mathbf{R}^q$, $Y \sim \mathcal{N}(\mu + \Lambda X, \Psi) \in \mathbf{R}^p$ avec $q < p$.

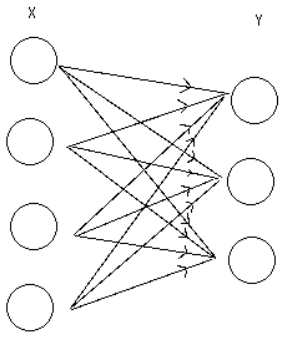


Ceci sert à modéliser le fait que des données sont proches d'un sous-espace (ici localisées autour de μ dans la direction Λ). Le modèle est la verison continue des modèles de mélange, et est proche du modèle ACP. On peut d'ailleurs montrer qu'il y a équivalence entre les deux modèles si Ψ a une certaine forme (voir M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. Journal of the Royal Statistical Society, 3(61), 1999. 4).

Comme dans tous les modèles à variables cachées, les deux tâches à affectuer sont :

- Inférence : $X|Y$
- Apprentissage des paramètres Λ, Ψ, μ.

$X \sim \mathcal{N}(0, I)$, donc X se factorise dans un modèle graphique trivial (sans arêtes). Ainsi le modèle graphique associé à l'analyse factorielle est :



Inférence On remarque que (X, Y) est nécessairement normale, on cherche donc à exprimer la moyenne $\tilde{\mu} \in \mathbb{R}^{p+q}$ et la covariance $\tilde{\Sigma}$ en utilisant les propriétés de l'espérance conditionnelle.

- $\mathbf{E}(X) = 0$
- $\mathbf{E}(Y) = \mathbf{E}(\mathbf{E}(Y|X)) = \mathbf{E}(\mu + \Lambda X) = \mu$
- $\mathbf{var}(X) = I$
- $\mathbf{var}(Y) = \mathbf{E}(\mathbf{var}(Y|X)) + \mathbf{var}(\mathbf{E}(Y|X)) = \Psi + \mathbf{var}(\mu + \Lambda X) = \Psi + \Lambda\Lambda^T$
- $\mathrm{Cov}(X, Y) = \mathrm{Cov}(X, \mathbf{E}(Y|X)) = \mathrm{Cov}(X, \mu + \Lambda X) = \Lambda^T$

ainsi $\tilde{\mu} = \begin{pmatrix} 0 \\ \mu \end{pmatrix}$ et $\tilde{\Sigma} = \begin{pmatrix} I & \Lambda \\ \Lambda^T & \Psi + \Lambda\Lambda^T \end{pmatrix}$

Enfin, on en déduit, d'après la section précédente,

$$\mathbf{E}(X|Y) = 0 + \Lambda^T(\Psi + \Lambda\Lambda^T)^{-1}(Y - \mu)$$

et

$$\mathbf{var}(X|Y) = I - \Lambda^T(\Psi + \Lambda\Lambda^T)^{-1}\Lambda$$

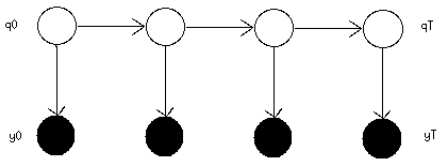
Estimation de μ, Λ, Ψ par EM Soient des données i.i.d. $Y_1, ...Y_n \in \mathbf{R}^p$. On construit la log-vraisemblance complete :

$$\begin{aligned} l_c &= \sum_{i=1}^{n} \log(\mathcal{N}(X_i|O, I)) + \sum_{i=1}^{n} \log(\mathcal{N}(Y_i|\mu + \Lambda X_i, \Psi)) \\ &= \sum_{i=1}^{n} \left[-(2\pi)^{q/2} - \frac{1}{2} X_i^T X_i - (2\pi)^{p/2} - \log|\Psi|^{1/2} - \frac{1}{2}(\mu + \Lambda X_i)^T \Psi^{-1}(\mu + \Lambda X_i) \right] \end{aligned}$$

On effectue alors l' E-step en remplaçant X_i par $< X_i >$ et $X_i X_i^T$ par $< X_i X_i^T >$ (où on prend l'espérance suivant $X_i|Y_i$ qui est une loi normale dont la moyenne et la variance ont été calculées ci dessus). Cf le polycopié pour plus de détails.

6.3 Modèles de Markov Cachés

Cadre On se donne le modèle suivant : On suppose avoir des états $(q_i)_{i=0..T}$ et une observation bruitée $(Y_i)_{i=O...T}$ des états, représentée par le modèle suivant :



Ceci montre qu'on suppose les bruits indépendants sachant les observations.

Applications Les applications de ce modèle (HMM) sont très vastes : Suivi de cibles, Traitement de la parole (q :mots prononcés), Bioinformatique (q :ADN), Musique (q :notes)...

On peut étendre naturellement les HMM à des grilles pour le traitement d'images, mais leur etude devient alors plus compliquée car la tâche d'infŕence a une complexité exponentielle.

Tâches On suppose que q_0 suit une loi multinomiale de paramètre π et que les q ne prennent que k valeurs. $q_{t+1}|q_t$ a pour loi la matrice $k \times k$ de transition $p(q_{t+1}|q_t)$.

Les tâches à accomplir sont :

- L'Inférence :
  - Le filtrage : $p(q_{t+1}|y_1, ..., y_t)$
  - Le lissage : $p(q_t|y_1, ..., y_T)$
  - Maximiser : $\max_q p(q|y)$
- L'apprentissage.

6.3.1 Application de l'algorithme somme-produit

Comme on a un arbre, on utilise l'algorithme somme-produit. Dans le cadre simple des HMM d'autres méthodes pourraient être envisagées, mais c'est en fait un bon prétexte pour apprendre à l'utiliser. D'ailleurs, pour des arbres compliqués, l'algorithme somme-produit marchera encore ce qui n'est pas nécessairement le cas des méthodes ad-hoc.

On identifie d'abord pour potentiels $p(q_0)$, $p(q_{t+1}|q_t)$ et $p(y_t|q_t)$, $t = 0, \ldots, T-1$.

Les messages sont envoyés selon le protocole : $y_i \to q_i, q_i \to q_{i+1}$ de $i = 0$ à $i = T-1$ et enfin $y_T \to q_T$. Ils valent :

- $m_{y_0,q_0}(q_0) = p(y_0|q_0)$
- $m_{q_0,q_1}(q_1) = \sum_{q_0} p(q_1|q_0) m_{y_0,q_O}(q_0)$

- ...
- $m_{y_t,q_t}(q_t) = p(y_t|q_t)$
- $m_{q_t,q_{t+1}}(q_{t+1}) = \sum_{q_t} p(q_{t+1}|q_t) m_{q_{t-1},q_t}(q_t) m_{y_t,q_t}(q_t)$

Proposition 6.3. *Les messages arrivant en $t+1$ vérifient* $m_{q_t,q_{t+1}}(q_{t+1}) m_{y_{t+1},q_{t+1}}(q_{t+1}) = p(y_0, ..y_{t+1}, q_{t+1})$

Pour obtenir ces formules, il suffit d'appliquer l'algorithme somme-produit au graphe obtenu en supprimant le futur.

Soit $\alpha_t(q_t) = m_{q_{t-1},q_t}(q_t) m_{y_t,q_t}(q_t)$, alors α_t est une probabilité égale à $p(y_0, ..y_{t+1}, q_{t+1})$ et on a la formule de récursion α suivante :

$$\boxed{\alpha_{t+1}(q_{t+1}) = p(y_{t+1}|q_{t+1}) \sum_{q_t} p(q_{t+1}|q_t) \alpha_t(q_t)}$$

avec pour initialisation $\alpha_0(q_0) = p(y_0|q_0)p(q_0)$.

On en déduit le théorème suivant :

Théorème 6.4.

$$p(y_1, ..., y_T) = \sum_{q_T} \alpha_t(q_t)$$

Le calcul de $p(y)$ se fait donc en $O(Tk^2)$ opérations (car chaque récursion est un produit matrice/vecteur). On effectue à présent la rétropropagation des messages. Soit $\beta_t(q_t) = m_{q_{t+1},q_t}(q_t)$, alors on a la formule de récursion suivante :

$$\boxed{\beta_t(q_t) = \sum_{q_{t+1}} p(q_{t+1}|q_t) \beta_{t+1}(q_{t+1}) p(y_{t+1}|q_{t+1})}$$

avec pour initialisation $\beta_T(q_T) = 1$.

Théorème 6.5. $p(q_t, y_0, ...y_T) = \alpha_t(q_t)\beta_t(q_t)$,

$p(q_t, q_{t+1}, y_0, ..., y_T) = p(q_{t+1}|q_t)\alpha_t(q_t)\beta_{t+1}(q_{t+1})p(y_{t+1}|q_{t+1})$.

Finalement, le calcul de $\max_q p(q|y)$ se fait en remplaçant les sommes par des max. La tâche d'inférence est donc effectuée.

Pratique En pratique, l'application directe de la formule de récursion pour le calcul de α_t ne marche pas pour un grand nombre d'états, car on atteint alors la précision machine de 10^{-16} pour les éléments de la somme. Il existe plusieurs solutions pour calculer efficacement les valeurs. Parmi celles-ci, citons le codage en *log* :

$$\log(\alpha_{t+1}) = \log p(y_{t+1}|q_{t+1}) + \log(\sum_{q_t} p(q_{t+1}|q_t) e^{\log(\alpha_t)})$$

On utilise alors la formule $\log(\sum e^{Y_i}) = \log(\sum e^{Y_i - \max Y_i}) + \max Y_i$ qui permet d'éviter les problèmes d'arrondi.

Estimation des paramètres

Supposons donnés :
- $p(q_0) = \pi_{q_0}$,
- $p(q_{t+1}|q_t) = A_{q_{t+1},q_t}$,
- $p(y_t|q_t) = f(y_t, q_t, B)$.

Alors on écrit la vraisemblance compl'ete :

$$\begin{aligned} l_c &= \log(p(q_0)\Pi_{t=0}^{T-1}p(q_{t+1}|q_t)\Pi_{t=0}^{T}p(y_t|q_t) \\ &= \log(\Pi_{i=1}^{k}\pi_i^{\delta(q_0=i)}\Pi_{t=0}^{T-1}\Pi_{i,j=1}^{k}A_{i,j}^{\delta(q_{t+1}=i,q_t=j)}\Pi_{t=0}^{T}\Pi_{i=1}^{k}f(y_t,i,B)^{\delta(q_t=i)}) \\ &= \sum_{i=1}^{k}\delta(q_0=i)log(\pi_i) + \sum_{t=0}^{T-1}\sum_{i,j=1}^{k}\delta(q_{t+1}=i,q_t=j)\log(A_{i,j}) + \sum_{t=0}^{T}\sum_{i=1}^{k}\delta(q_t=i)\log(f(q_t,i,B)) \end{aligned}$$

Pour l'étape E de l'algorithme EM, on calcule donc :
- $\mathbf{E}(\delta(q_0 = i)|y) = p(q_0 = i|y)$
- $\mathbf{E}(\delta(q_t = i)|y) = p(q_t = i|y)$
- $\mathbf{E}(\delta(q_{t+1} = i, q_t = j|y) = p(q_{t+1} = i, q_t = j|y)$

Il suffit alors de remplacer les variables cachées $\delta(q_0 = i)$,$\delta(q_t = i)$ et $\delta(q_{t+1} = i, q_t = j)$ par les quantités ci-dessus : puis on maximise la nouvelle log vraisemblance de la manière habituelle pour obtenir les estimateurs des paramètres.