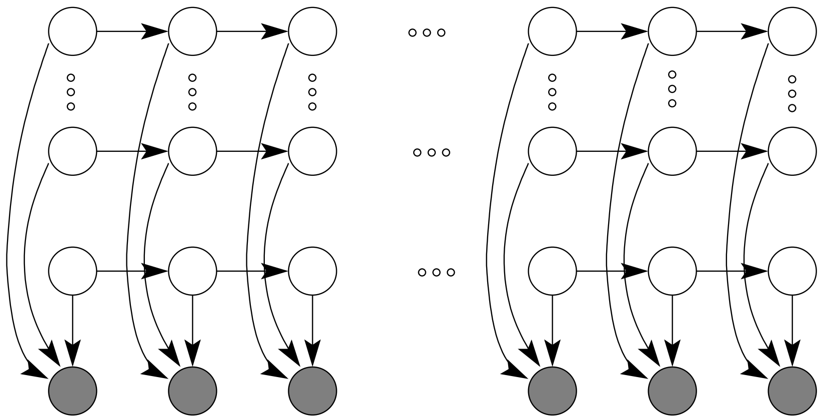
Mastere M2 MVA 2006 - Modèles graphiques

Exercices à rendre pour le 14 Décembre 2006.

Ces exercices doivent s'effectuer seul.

1 Modèles de Markov cachés factoriels

On considère le modèle de Markov caché factoriel suivant, avec M chaînes de Markov cachées de longueur T (les variables grisées sont toujours observées):



1. Quelle est la complexité de l'algorithme de l'arbre de jonction pour l'inférence? (on supposera que la i-ième chaîne a m_i états).
2. Quelle serait la complexité en utilisant (naivement) une seule chaîne de Markov avec $\prod_i m_i$ états? (si $x_1^t, \ldots, x_M^t$ sont les états des M chaînes à l'instant t, l'état de cette chaîne est $x^t = (x_1^t, \ldots, x_M^t)$).

2 Modèles orientés Gaussiens

Soient n variables aléatoires $X_1, \ldots, X_n$ de densité jointe Gaussienne de moyenne $\mu \in \mathbb{R}^n$ et de matrice de covariance $\Sigma \in \mathbb{R}^{n \times n}$.

1. Rappeler (sans démonstration) la condition nécessaire et suffisante pour que la loi Gaussienne $\mathcal{N}(\mu,\Sigma)$ se factorise dans un graphe non orienté. Le but de cet exercise est de déterminer une caractérisation similaire pour les graphes orientés.
2. On suppose que la loi $\mathcal{N}(\mu,\Sigma)$ se factorise dans un graphe orienté, caractérisé par les parents π_i de chaque sommet i, $i = 1, \ldots, n$. Montrer que la loi de X_i

sachant $X_{\pi_i} = x_{\pi_i}$ est une Gaussienne et exprimer sa moyenne et sa variance en fonction de x_{π_i}, μ et Σ.

3. Etant donnée une matrice symmétrique définie positive M, la décomposition de Cholesky est l'unique factorisation $M = RR^\top$ où R est triangulaire inférieure avec des éléments positifs sur la diagonal. La décomposition $LDL^\top$ est l'unique factorisation $M = LDL^\top$ où D est diagonale avec diagonale strictement positive et L triangulaire inférieure avec diagonale constante égale à 1 (on passe de l'une à l'autre par $R = LD^{1/2}$, où $D^{1/2}$ est la matrice diagonal des racines carrées positives des éléments diagonaux de D). Dans cet exercice, on considère les décompositions de $M = \Sigma^{-1}$.
   On suppose que l'ordre $(n, \ldots ,1)$ est topologique pour un graphe orienté donné $G = (V,E)$ (défini par l'ensemble des parents de chaque noeud), i.e., $\pi_i \subset \{i + 1, \ldots ,n\}$. Montrer que la loi $\mathcal{N}(\mu,\Sigma)$ se factorise dans le graphe orienté ssi
   $$\forall i \in \{1, \ldots ,n\},\ (j \neq i \text{ et } j \notin \pi_i) \Rightarrow R_{ji} = 0.$$
   ou, de manière équivalente,
   $$\forall i \in \{1, \ldots ,n\},\ (j \neq i \text{ et } j \notin \pi_i) \Rightarrow L_{ji} = 0.$$
   (i.e., les modèles graphiques orientés correspondent à une factorisation parcimonieuse de Σ^{-1}).
   INDICATION: on se ramènera à $\mu = 0$ et on utilisera la caractérisation suivante: si $(n, \ldots ,1)$ est un ordre topologique pour G, alors $p(x)$ se factorise dans G ssi $\forall i < n$, $p(x_i|x_{i+1}, \ldots ,x_n) = p(x_i|x_{\pi_i})$.
4. Montrer comment les résultats de la question 2.2 permettent de calculer directement L à partir de Σ, sous les hypothèses que la loi $\mathcal{N}(\mu,\Sigma)$ se factorise et que l'ordre $(n, \ldots ,1)$ est topologique.
5. BONUS: si la factorisation $LDL^\top$ de Σ^{-1} est connue, quelle est la complexité numérique des produits "matrice-vecteur" de la forme Σy et $\Sigma^{-1}y$, pour $y \in \mathbb{R}^n$ donné.
6. "DOUBLE" BONUS: si la loi $\mathcal{N}(\mu,\Sigma)$ se factorise dans un modèle graphique non orienté, et que Σ^{-1} est donnée, quelle est la complexité numérique des produits "matrice-vecteur" de la forme Σy et $\Sigma^{-1}y$, pour $y \in \mathbb{R}^n$ donné.

3 Apprentissage de la structure d'un arbre

Dans cet exercice, un algorithme pour apprendre la structure d'un modèle graphiques à partir de données i.i.d sera dérivé pour les arbres.

1. Préliminaire: Soient deux variables aléatoires discrètes X, Y à valeurs dans des ensembles finis $\mathcal{X}$ et $\mathcal{Y}$. Soit $\eta_{ij} = p(y = j|x = i)$. Soit un échantillon i.i.d (x^n,y^n)

de taille N de ces deux variables. On note $\hat{p}(x,y)$ la densité empirique, définie par $\hat{p}(x,y) = \frac{1}{N}\sum_{n=1}^{N}\delta(x^n = x)\delta(y^n = y)$. Montrer que la log-vraisemblance conditionnelle maximale $\max_\eta \sum_{n=1}^{N} \log p(y^n|x^n,\eta)$ est égale à:

$$\max_\eta \sum_{n=1}^{N} \log p(y^n|x^n,\eta) = N(H(X) - H(X,Y))$$

oú $H(X)$ est l'entropie empirique de X, définie par

$$H(X) = -\sum_{x\in\mathcal{X}} \hat{p}(x)\log\hat{p}(x)$$

et $H(X,Y)$ l'entropie empirique de (X,Y) définie par

$$H(X,Y) = -\sum_{y\in\mathcal{Y}}\sum_{x\in\mathcal{X}} \hat{p}(x,y)\log\hat{p}(x,y)$$

($\hat{p}(y)$ est la densité marginale correspondant à $\hat{p}(x,y)$).

2. Soient M variables aléatoires discrètes $X_1,\ldots,X_M$ se factorisant dans un arbre T orienté. Quelle est la paramétrisation la plus génerale pour les densités conditionnelles? Soit un échantillon i.i.d $(x_1^n,\ldots,x_M^n)$ de taille N de ces M variables. Calculer la valeur maximale de la log-vraisemblance des données pour une loi se factorisant dans T (i.e., maximiser par rapport à tous les paramètres) et exprimer le résultat en fonction des entropies empiriques $H(X_i)$ et $H(X_i,X_j)$, $i,j \in \{1,\ldots,n\}$.
3. Comment maximiser par rapport à l'arbre T l'expression précédente? Décrire un algorithme permettant d'apprendre la structure de l'arbre ayant le maximum de vraisemblance. Quelle est sa complexité?

4 Implémentation - HMM

On considère les mêmes données d'apprentissage que le devoir précédent, dans le fichier "EMGaussienne.dat", mais cette fois-ci en considérant la structure temporelle, i.e., les données sont de la forme $u_t = (x_t,y_t)$ où $u_t = (x_t,y_t) \in \mathbb{R}^2$, pour $t = 1,\ldots,T$. Le but de cet exercice est d'implémenter l'inférence dans les HMM ainsi que l'algorithme EM pour l'apprentissage des paramètres. Il est fortemenr conseillé d'utiliser le code du devoir précédent.

On considère le modèle HMM suivant avec une chaîne (q_t) à K=4 états et matrice de transition $a \in \mathbb{R}^{4\times 4}$, et des "probabilités d'émission" Gaussiennes: $u_t|q_t = i \sim \mathcal{N}(\mu_i,\Sigma_i)$.

1. Implémenter les récursions α et β vues en cours et dans le polycopié pour estimer $p(q_t|u_1,\ldots,u_T)$ et $p(q_t,q_{t+1}|u_1,\ldots,u_T)$.

2. Calculer les équations d'estimation de EM.
3. Implémenter l'algorithme EM pour l'apprentissage (on pourra initialiser les moyennes et les covariances avec celles trouvées dans le devoir précédent).
4. Implémenter l'inférence pour estimer la séquence d'états la plus probables, i.e. $\arg\max_q p(q_1, \ldots, q_T | y_1, \ldots, y_T)$, et représenter le résultat obtenu avec les données (pour le jeu de paramètres appris par EM).
5. Commenter les différents résultats obtenus avec ceux du devoir précédent. En particulier, comparer les log-vraisemblances, sur les données d'apprentissage, ainsi que sur les données de test (dans "EMGaussienne.test").

5 Projet de fin de cours

Décrire le projet en quelques lignes (thème, articles de référence, implémentations prévues, type de données, etc...).