Graphical models : Homework 3

Tous les graphiques demandés sont en annexe

COÏC Quentin
NAAMANE Chemsdin

3) Avec les notations de l'énoncé, on écrit la log-vraisemblance complète, $l_c(\theta)$

où $\theta = (\pi, A, \mu_k, \Sigma_k, k=1\dots4)$.

$$l_c(\theta) = \log\left(p(q_1)\prod_{t=1}^{T-1} p(q_{t+1}|q_t)\prod_{t=1}^{T} p(u_t|q_t)\right)$$

$$= \log(p(q_1)) + \sum_{t=1}^{T-1}\log(p(q_{t+1}|q_t)) + \sum_{t=1}^{T}\log(p(u_t|q_t))$$

$$= \sum_{i=1}^{4}\delta(q_1=i)\log(\pi_i) + \sum_{t=1}^{T-1}\sum_{i,i'=1}^{4}\delta(q_{t+1}=i, q_t=i')\log(A_{ii'})$$

$$+ \sum_{t=1}^{T}\sum_{i=1}^{4}\delta(q_t=i)\log(\mathcal{N}(u_t|\mu_i,\Sigma_i))$$

où δ est la fonction indicatrice et $\mathcal{N}(x|\mu,\Sigma)$ est la densité d'une gaussienne en x de moyenne μ et de matrice de variance-covariance Σ.

Pour déterminer les équations d'estimation des paramètres de l'algorithme EM, à l'étape k on doit maximiser l'espérance de $l_c(\theta)$ sous $\mathbb{P} = \mathbb{P}(q_1,\dots,q_T | u_1,\dots,u_T ; \theta^{(k-1)})$

où $\theta^{(k-1)}$ sont les valeurs des paramètres à l'étape $k-1$. On doit donc maximiser :

$$\mathbb{E}_{\mathbb{P}}[l_c(\theta)] = \sum_{i=1}^{4}\mathbb{P}(q_1=i|u)\log(\pi_i) + \sum_{t=1}^{T-1}\sum_{i,i'=1}^{4}\mathbb{P}(q_{t+1}=i, q_t=i'|u)\log(A_{ii'})$$

$$+ \sum_{t=1}^{T}\sum_{i=1}^{4}\mathbb{P}(q_t=i|u)\log(\mathcal{N}(u_t|\mu_i,\Sigma_i))$$

où $u = (u_1,\dots,u_T)$.

On s'intéresse d'abord à la partie dépendant de π : $f_1(\pi) = \sum_{i=1}^{4}\mathbb{P}(q_1=i|u)\log(\pi_i)$

On a la contrainte : $\sum_{i=1}^{4}\pi_i = 1$.

On écrit le Lagrangien : $L_1(\pi, \lambda) = \sum_{i=1}^{4} P(q_1 = i/u) \log(\pi_i) - \lambda \left(\sum_{i=1}^{4} \pi_i - 1 \right)$

On a un problème convexe (maximisation d'une fonction concave) et il existe π tel que $\forall i,\ \pi_i > 0$ et $\sum_{i=1}^{4} \pi_i = 1$ donc la contrainte de qualification de Slater est vérifiée et on a donc dualité forte. Il nous suffit alors de dériver le Lagrangien par rapport à π et de trouver le point π qui vérifie $\frac{\partial L_1}{\partial \pi} = 0$ et $\sum_{i=1}^{4} \pi_i = 1$ pour trouver le point qui maximise f_1.

$$\frac{\partial L_1}{\partial \pi_i}(\pi, \lambda) = \frac{P(q_1 = i/u)}{\pi_i} - \lambda \qquad \frac{\partial L_1}{\partial \pi_i}(\pi, \lambda) = 0 \iff \pi_i = \frac{P(q_1 = i/u)}{\lambda}$$

Avec la contrainte on a : $\sum_{i=1}^{4} \pi_i = 1 \iff \sum_{i=1}^{4} P(q_1 = i/u) = \lambda \iff \lambda = 1$

Ainsi, on a l'estimateur $\boxed{\hat{\pi}_i^{(u)} = P(q_1 = i/u) \quad \forall i = 1, \dots 4}$

On s'intéresse maintenant à la partie qui dépend de A : $f_2(A) = \sum_{t=1}^{T-1} \sum_{i,i'=1}^{4} P(q_{t+1} = i, q_t = i'/u) \log(A_{i,i'})$

On a la contrainte : $\sum_{i=1}^{4} A_{i,i'} = 1 \quad \forall i' = 1, \dots, 4$.

Comme pour π on va trouver l'estimateur à partir du Lagrangien (mêmes arguments).

$$L_2(A, \lambda) = \sum_{t=1}^{T-1} \sum_{i,i'=1}^{4} P(q_{t+1} = i, q_t = i'/u) \log(A_{i,i'}) - \sum_{i'=1}^{4} \lambda_{i'} \left(\sum_{i=1}^{4} A_{i,i'} - 1 \right)$$

$$\frac{\partial L_2}{\partial A_{i,i'}}(A, \lambda) = \sum_{t=1}^{T-1} \frac{P(q_{t+1} = i, q_t = i'/u)}{A_{i,i'}} - \lambda_{i'}$$

$$\frac{\partial L_2}{\partial A_{i,i'}}(A, \lambda) = 0 \iff A_{i,i'} = \frac{1}{\lambda_{i'}} \sum_{t=1}^{T-1} P(q_{t+1} = i, q_t = i'/u)$$

Avec la contrainte : $\sum_{i=1}^{4} A_{ii'} = 1 \Leftrightarrow \sum_{i=1}^{4} \sum_{t=1}^{T-1} \underbrace{\frac{P(q_{t+1}=i, q_t=i'/u)}{\lambda_{i'}}}_{\lambda_{i'}} = 1$

$$\Leftrightarrow \lambda_{i'} = \sum_{t=1}^{T-1} \sum_{i=1}^{4} P(q_{t+1}=i, q_t=i'/u) = \sum_{t=1}^{T-1} P(q_t=i'/u)$$

Ainsi, on a l'estimateur : $\boxed{\hat{A}_{ii'}^{(a)} = \frac{\sum_{t=1}^{T-1} P(q_{t+1}=i, q_t=i'/u)}{\sum_{t=1}^{T-1} P(q_t=i'/u)} \quad \forall i, i' = 1, \dots, 4}$

On s'intéresse ensuite à la partie qui dépend de μ_i, Σ_i, $i = 1, \dots 4$:

$$f_3(\mu, \Lambda) = \sum_{t=1}^{T} \sum_{i=1}^{4} P(q_t=i/u) \times \left[\frac{1}{2} \log(\det(\Lambda_i)) - \frac{1}{2} (u_t - \mu_i)^T \Lambda_i (u_t - \mu_i) \right]$$

$+$ constante

où $\Lambda_i = \Sigma_i^{-1} \quad \forall i = 1, \dots, 4$.

On va chercher ici les estimateurs de μ et Σ en égalisant à 0 les dérivées partielles de f_3 par rapport à μ et Σ :

$$\frac{\partial f_3}{\partial \mu_i}(\mu, \Lambda) = \sum_{t=1}^{T} P(q_t=i/u) \times \Lambda_i (u_t - \mu_i)$$

$$\frac{\partial f_3}{\partial \mu_i}(\mu, \Lambda) = 0 \Leftrightarrow \sum_{t=1}^{T} P(q_t=i/u) u_t = \sum_{t=1}^{T} P(q_t=i/u) \mu_i$$

On a donc l'estimateur, $\boxed{\hat{\mu}_i^{(a)} = \frac{\sum_{t=1}^{T} P(q_t=i/u) u_t}{\sum_{t=1}^{T} P(q_t=i/u)} \quad \forall i = 1, \dots, 4}$

$$\frac{\partial f_3}{\partial \Lambda_i}(\mu, \Lambda_i) = \frac{1}{2} \sum_{t=1}^{T} P(q_t=i/u) \Sigma_i - \frac{1}{2} \sum_{t=1}^{T} P(q_t=i/u) (u_t - \mu_i)(u_t - \mu_i)^T$$

$\frac{\partial f_3}{\partial \Lambda_i}(\mu_i, \Lambda) = 0 \Leftrightarrow \sum_{t=1}^{T} P(q_t = i | u)\, \Sigma_i = \sum_{t=1}^{T} P(q_t = i | u)\, (u_t - \mu_i)(u_t - \mu_i)^T$

On a donc l'estimateur :

$$\boxed{\hat{\Sigma}_i^{(q)} = \frac{\sum_{t=1}^{T} P(q_t = i | u)\, (u_t - \hat{\mu}_i)(u_t - \hat{\mu}_i)^T}{\sum_{t=1}^{T} P(q_t = i | u)} \quad \forall i = 1, \dots, 4}$$

Remarque : On a omis la dépendance des probabilités en $\theta^{(q-1)}$ pour alléger les notations.

5) On observe que la log-vraisemblance est toujours meilleure sur l'échantillon train que sur l'échantillon test, ce qui semble intuitif. De plus, on remarque que les deux log-vraisemblances ont une évolution similaire. En effet, les log-vraisemblances augmentent à chaque itération. C'est la première itération qui contribue le plus à cette augmentation et après cette dernière l'évolution se stabilise. En conclusion, l'algorithme EM sous cette modélisation semble être un choix pertinent.

6) On obtient avec les paramètres appris par l'EM du modèle HMM:

log-vraisemblance / échantillon	log-vraisemblance HMM	log-vraisemblance Gaussian mixture
train	-1896,79	-8458,89
test	-1955,55	-9500,02

(précision: 10^{-2})

Les log-vraisemblances sont meilleures sur l'échantillon train que sur l'échantillon test dans les deux cas, ce qui est intuitif. Cependant, les log-vraisemblances obtenues pour le modèle "Gaussian mixture" sont bien plus mauvaises que pour le modèle HMM. Ceci était attendu car les paramètres utilisés pour calculer

ces log-vraisemblances sont optimisées pour le modèle HMM. Comme le modèle HMM introduit une temporalité dans les données qui est absente du modèle "Gaussian mixture", ces deux modèles sont donc complètement différents et la comparaison avec les mêmes paramètres n'a ainsi pas de sens.

Dans le DM précédent on avait obtenu les log-vraisemblances suivantes : (après apprentissage par EM de chaque modèle respectivement)

modèle / échantillon	Gaussien mixture général	Gaussien mixture isotropique
train	-2327,75	-2645,54
test	-2409,05	-2692,81

Précision : 10^{-2}

On peut remarquer qu'on a de meilleurs résultats pour le modèle HMM que pour les modèles étudiés dans le DM précédent. Ici semble indiquer une présence de temporalité dans les données et pour vérifier cela nous avons d'abord mélangé seulement l'échantillon train et appliqué l'algorithme EM (HMM). On obtient alors une log-vraisemblance sur l'échantillon test qui diminue après la première itération alors que la log-vraisemblance sur l'échantillon train augmente jusqu'à -2324,14 ce qui est similaire à la performance du modèle "Gaussien mixture" général.

De plus, lorsque l'échantillon test est mélangé avec les mêmes permutations que pour l'échantillon train, on obtient une log-vraisemblance qui cette fois augmente similairement à la log-vraisemblance de l'échantillon train mais jusqu'à -2404,55. C'est là encore une performance similaire au modèle "Gaussien mixture" général.

Il est donc naturel d'accepter une hypothèse de temporalité sur les données car le mélange a une influence négative évidente sur les performances du modèle HMM qui sont alors comparables aux performances du modèle "Gaussien mixture" général.

2) On rappelle qu'on a avec les observations $\bar{u}_1, \dots, \bar{u}_T$ et les variables latentes $q_1, \dots, q_T$: $p(\bar{u}_1, \dots, \bar{u}_T, q_1, \dots, q_T) = p(q_1) \prod_{t=1}^{T-1} p(q_{t+1}|q_t) \prod_{t=1}^{T} p(u_t|q_t)^{\delta(u_t = \bar{u}_t)}$

où δ est la fonction indicatrice.

On utilise cette fois-ci au lieu d'un algorithme "sum-product", un algorithme "max-product" qui nous conduit aux messages :

$$\mu_{u_t \to q_t}(q_t) = \max_{u_t} p(u_t|q_t)\, \delta(u_t = \bar{u}_t) = p(\bar{u}_t|q_t) \qquad \forall t = 1, \dots, T$$

et $\mu_{q_{t-1} \to q_t}(q_t) = \max_{q_{t-1}} p(q_t|q_{t-1})\, \mu_{u_{t-1} \to q_{t-1}}(q_{t-1})\, \mu_{q_{t-2} \to q_{t-1}}(q_{t-1})$ $\forall t = 2, \dots, T$

avec $\mu_{q_0 \to q_1}(q_1) = p(q_1)$

On définit $w_t(q_t) = \mu_{u_t \to q_t}(q_t)\, \mu_{q_{t-1} \to q_t}(q_t) \qquad \forall t = 1, \dots, T$

On a : $w_t(q_t) = p(\bar{u}_t|q_t) \max_{q_{t-1}} p(q_t|q_{t-1})\, w_{t-1}(q_{t-1})$

$$\overset{\text{Par récurrence}}{= \dots =} \max_{q_1 \dots q_{t-1}} p(q_1) \prod_{i=1}^{t-1} p(q_{i+1}|q_i) \prod_{i=1}^{t} p(\bar{u}_i|q_i)$$

$$= \max_{q_1 \dots q_{t-1}} p(q_1, \dots, q_t, \bar{u}_1, \dots, \bar{u}_t)$$

Ainsi, afin de calculer $\arg\max_q p(q_1, \dots, q_T | \bar{u}_1, \dots, \bar{u}_T)$ avec l'algorithme de Viterbi, on effectue une première étape de type "forward" qui va consister à calculer $w_t(q_t)$ $\forall q_t$ et $\forall t = 1, \dots, T$ en partant de $w_1(q_1) = p(\bar{u}_1|q_1) p(q_1)$

On arrive donc à la fin de cette étape à $w_T(q_T) = \max_{q_1 \dots q_{T-1}} p(q_1, \dots, q_T, \bar{u}_1, \dots, \bar{u}_T)$

et comme $p(q_1, \dots, q_T | \bar{u}_1, \dots, \bar{u}_T) = \frac{p(q_1, \dots, q_T, \bar{u}_1, \dots, \bar{u}_T)}{p(\bar{u}_1, \dots, \bar{u}_T)} \propto p(q_1, \dots, q_T, \bar{u}_1, \dots, \bar{u}_T)$

par rapport à $q_1, \dots, q_T$

on peut donc trouver l'argument maximal en q_T de la séquence d'états la plus probable en cherchant $\arg\max_{q_T} w_T(q_T)$.

Ici ombre la seconde étape de l'algorithme où l'on récupère la séquence d'états la plus probable en cherchant ensuite pour $t = T-1, \dots, 1$ $\arg\max_{q_t} \left(p(\bar{u}_{t+1} | q_{t+1}^*) \, p(q_{t+1}^* | q_t) \, w_t(q_t) \right)$

pour q_{t+1}

en utilisant à chaque itération t l'argument maximal de l'étape précédente (q_{t+1}^*)

On recherche en fait l'argument maximal q_t qui nous a permis lors de l'étape "forward" de calculer $w_{t+1}(q_{t+1}^*)$. On aura alors juste à sélectionner cette valeur qu'on aura gardé en mémoire lors de la première étape.

On peut donc donner le pseudo-code suivant :

- Initialiser $w_1(q_1) = p(\bar{u}_1 | q_1) \, p(q_1)$ pour tout q_1
- Pour $t = 2, \dots, T$:

  calculer $w_t(q_t) = p(\bar{u}_t | q_t) \max_{q_{t-1}} \underbrace{p(q_t | q_{t-1}) \, w_{t-1}(q_{t-1})}_{(*)}$

  et garder en mémoire l'argument maximal en q_{t-1} de $(*)$ pour tout q_t

- calculer $\arg\max_{q_T} w_T(q_T)$ afin d'obtenir q_T^*

- Pour $t = T-1, \dots, 1$:

  sélectionner parmi les arguments q_t gardés en mémoire lors du calcul de $w_{t+1}(q_{t+1})$ celui qui correspond au calcul de $w_{t+1}(q_{t+1}^*)$ afin d'obtenir q_t^*

- Renvoyer $(q_1^*, \dots, q_T^*)$ la séquence d'états la plus probable.

Remarque : Comme pour l'algorithme "forward-backward", on calcule les $w_t(q_t)$, $t = 1, \dots, T$ sous forme logarithmique, i.e. : $\log(w_t(q_t)) = \log(p(\bar{u}_t | q_t)) + \max_{q_{t-1}} [\log(p(q_t | q_{t-1})) + \log(w_{t-1}(q_{t-1}))]$

car on peut intervertir le log et le max et la séquence d'états la plus probable reste la même.

11) On observe que l'algorithme de Viterbi et l'utilisation des probabilités marginales obtenues par l'algorithme "forward-backward" donnent exactement les mêmes résultats en ce qui concerne la séquence d'états la plus probable. Ceci n'est pas étonnant car on utilise le même modèle pour les deux algorithmes avec les mêmes paramètres (ceux appris par l'EM du modèle HMM). L'algorithme de Viterbi est cependant plus direct pour obtenir ce résultat.

12) Une première approche serait de prendre initialement pour l'algorithme "k-means" un nombre important d'états (k grand, par exemple $k = 20$) et de réduire de 1 le nombre de classe successivement tout en observant les effets sur la répartition des points afin de s'arrêter au bon moment. C'est-à-dire, on continue tant que retirer une classe ne perturbe pas trop la répartition des points.

Une autre approche peut-être à faire en premier serait de visualiser le nuage de points (les points sont dans $\mathbb{R}^2$ donc directement visualisables sans ACP).

Il existe également des critères de qualité adaptés à l'algorithme "k-means" pour sélectionner le nombre de classes tel que le critère de Davies-Bouldin : $DB = \frac{1}{C} \sum_{i=1}^{C} \max_{i \neq j} \frac{\sigma_i + \sigma_j}{d(G_i, G_j)}$

où C est le nombre de classes, σ_i est l'écart-type des distances des observations de la classe i et G_i est le centre de la classe i.

Ce critère est minimum lorsque le nombre de classes est optimal.