Algorithme Somme Produit et modèle de Markov Caché **2009/2010**

Cours 5 — 28 octobre 2009

Enseignant: Francis Bach *Scribe: Karen Randria, Pierrick Paillet*

5.1 Algorithme somme-produit

Sur un arbre non orienté (i.e., un graphe non orienté sans cycles), on considère une fonction $u(x)$ se factorisant dans G:

$$u(x) = \prod_{(i,j)\in E} \psi_{ij}(x_i x_j) \prod_{i\in V} \psi_i(x_i)$$

Le but est de calculer les marginales de $u(x)$.

Le graphe moralisé d'un DAG est un arbre uniquement si ce DAG ne contient pas de V-structures.

Définition 5.1 *Les lois marginales sont données par:*

$$\begin{aligned} u(x_i) &= \sum_{x_i, j\neq i} u(x) \\ u(x_i, x_j) &= \sum_{x_k, k\neq i, k\neq j} u(x) \end{aligned}$$

Le principe sous-jacent de cet algorithme est de remarquer que ce qui a été fait pour x_i n'a pas besoin d'être fait à nouveau pour $x_j, j < i$. On considère donc les messages suivants, auxquels on va attribuer un protocole de passage particulier:

Définition 5.2 *Pour un couple $(i,j) \in E$, le message m_{ij} de $i \to j$ (fonction de x_j) est égal à:*

$$m_{ij}(x_j) = \sum_{x_i} \psi_i(x_i)\psi_{ij}(x_i, x_j) \prod_{k\in\mathcal{N}(i)\setminus j} m_{ki}(x_i)$$

Pour calculer le message de i vers j, on prend donc en compte le potentiel lié à l'arête (i,j) et les messages envoyés par les voisins de i qui ne sont pas j ($\mathcal{N}(i) \setminus j$). Il faut donc avoir défini tous les messages précédents pour pouvoir calculer le message qui nous intéresse. On peut cependant remarquer que si i est une feuille de l'arbre, il n'a qu'un seul voisin (j). D'où le protocole de passage suivant, **valable seulement dans le cas de l'arbre**:

Remarque 5.3 *Pour un arbre à n nœuds, on compte $2(n-1)$ messages. En effet, il y a $n-1$ arêtes et on passe dans les deux sens.*

Construction d'un ordre de passage

Lorsque les messages sont passés en série, les ordres de passage respectant le protocole sont tous obtenus comme suit:

1. Définition d'une "racine" r (n'importe quel sommet)
2. Orientation du graphe
3. Choix d'un ordre topologique pour l'arbre orienté (sans perte de généralité, on considère que cet ordre est $(1, \ldots, n)$).

Nous pouvons alors distinguer 2 phases:

- La collecte (flèche en pointillés)

$$\begin{aligned} n &\rightarrow \pi_n \\ n-1 &\rightarrow \pi_{n-1} \\ &\vdots \\ 2 &\rightarrow \pi_2 = 1 \end{aligned}$$

  Ce qui permet d'obtenir les $n-1$ messages $m_{i,j}$ oËŸ $i\ j$

- la distribution (flèche en tirets)

$$\begin{aligned} \pi_2 &\rightarrow 2 \\ &\vdots \\ \pi_n &\rightarrow n \end{aligned}$$

  Ce qui permet d'obtenir les $n-1$ messages $m_{j,i}$ oËŸ $i\ j$

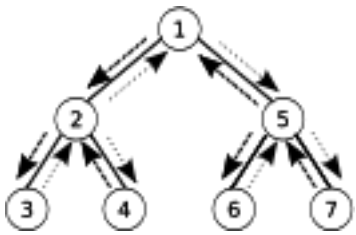


Figure 5.1. Collecte et Distribution des messages

Théorème 5.4 *Si le "protocole" de passage des messages est respecté, une fois les* $2(n-1)$ *messages passés, nous avons:*

- $\forall i, \quad u(x_i) = \psi_i(x_i) \prod_{k \in \mathcal{N}(i)} m_{kj}(x_i)$
- $\forall i, j \in E, \quad u(x_i, x_j) = \psi_i(x_i)\psi_j(x_j)\psi_{ij}(x_i, x_j) \prod_{k \in \mathcal{N}(i) \setminus j} m_{ki}(x_i) \prod_{k \in \mathcal{N}(j) \setminus i} m_{kj}(x_j)$

L'algorithme somme-produit n'est exact que pour les arbres. L'application aux graphes avec cycles (souvent appelée "loopy belief propagation") pose beaucoup de problèmes de convergence même si elle est souvent utilisée (voir chapitre sur les méthodes approchées d'inférence).

5.1.1 Exemple à 3 noeuds

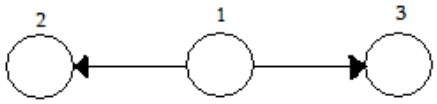


Figure 5.2. Modèle à trois noeuds

$$
\begin{aligned}
m_{31}(x_1) &= \sum_{x_3} \psi_3(x_3)\psi_{13}(x_1, x_3) \quad \text{attention } \psi_{13} = \psi_{31} \\
m_{21}(x_1) &= \sum_{x_2} \psi_2(x_2)\psi_{12}(x_1, x_2)
\end{aligned}
$$

on passe ensuite à la distribution : $m_{12}(x_2) = \sum_{x_1} \psi_1(x_1)\psi_{12}(x_1, x_2)m_{31}(x_1)$

$$m_{13}(x_3) = \sum_{x_1} \psi_1(x_1)\psi_{13}(x_1, x_3)m_{21}(x_1)$$

Au final

$$
\begin{aligned}
u(x_1) &= \sum_{x_2, x_3} \psi_1(x_1)\psi_2(x_2)\psi_{12}(x_1, x_2)\psi_3(x_3)\psi_{13}(x_1, x_3) \\
&= \psi_1(x_1) \sum_{x_2} \psi_2(x_2)\psi_{12}(x_1, x_2) \sum_{x_3} \psi_3(x_3)\psi_{13}(x_1, x_3) \\
&= \psi_1(x_1)m_{21}(x_1)m_{31}(x_1)
\end{aligned}
$$

de même, on obtient par exemple : $u(x_3) = \psi_3(x_3)m_{13}(x_3)$

5.1.2 Complexité totale

La complexité de l'algorithme de passage d'un message de $i \to j$ avec des variables prenant r valeurs vaut:

$$\mathcal{O}(r^2(\underbrace{\#(\mathcal{N}(i))}_{d_i = \text{ degré de } i} - 1))$$

On en déduit que la complexité totale est $O(r^2 n)$, i.e., linéaire en n, ce qui est plus rapide que l'élimination.

5.1.3 Risque d'underflow

On risque de travailler avec des valeurs tellement faibles que l'ordinateur peut renvoyer des erreurs. Par exemple :
pour k variables binaires $\longrightarrow p(x_1, x_2 ... x_n) = \dfrac{1}{2^k}$, donc si $k = 40$ la valeur est très faible.

La solution est de travailler avec des logarithmes. Par exemple :

$$\begin{aligned} a + b \;=\; c &\iff e^A + e^B \;=\; e^C \\ &\iff C \;=\; log(e^A + e^B) \\ &\iff C \;=\; A + log(1 + e^{B-A}) \;\text{ si } A > B \end{aligned}$$

Remarque 5.5 *Il existe un parallèle entre l'algorithme utilisé pour la FFT et l'algorithme somme-produit.*

5.1.4 Algorithme max-produit

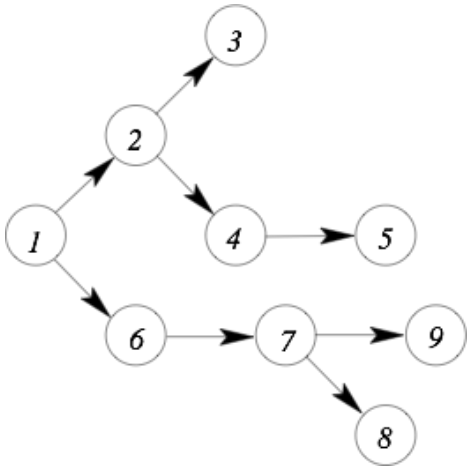
Dans le contexte du cours, seule la propriété que $(\mathbb{R}, +, \times)$ soit un semi-anneau commutatif est utilisée. En particulier, on a besoin de la distributivité du produit par rapport à la somme : $a.b + a.c = a(b + c)$. Cette distributivité est aussi présente sur le semi-anneau commutatif $(\mathbb{R}_+, \max, \times)$. En effet il suffit de remplacer $+$ par max et garder le produit. On peut donc faire une substitution entre les deux semi-anneaux et utiliser la nouvelle règle de distributivité $\max(a.b, a.c) = a \max(b, c)$.
Ainsi, toutes les procédures d'élimination et d'algorithmes somme-produit, qui ne sont basées que sur la distributivité s'appliquent. Par exemple, pour calculer $\max_x u(x)$, on fait passer les messages

$$m_{ij}(x_j) = \max_{x_i} \left(\psi_i(x_i)\psi_{ij}(x_i, x_j) \prod_{k \in \mathcal{N}(i) \setminus j} m_{ki}(x_i) \right)$$

5.2 Algorithme somme-produit - Preuve

On considère la procédure suivante (dite "en série") pour l'ordre de passage des messages: on définit une racine en choisissant un des sommet (au hasard); cette racine permet de définir un unique arbre orienté.

On considère un ordre topologique quelconque (i.e., tel qu'un noeud apparaît toujours après ses parents), et on suppose que les noeuds sont etiquettés en suivant l'ordre topologique, i.e., $V = \{1, \ldots, n\}$. Voir figure.



Les $2(n-1)$ messages sont passés dans l'ordre suivant:

$$n \to \pi_n, n-1 \to \pi_{n-1}, \ldots, 2 \to \pi_2, \pi_2 \to 2, \ldots, \pi_n \to n$$

Cette procédure respecte bien le protocole de passage des messages, i.e., un sommet n'envoie un message que quand il a recu un message de tous ses autres voisins.
Nous allons démontrer qu'une fois les $2(n-1)$ messages passés, les marginales sont égales à

$$\forall i \in V,\ u(x_i) = \psi_i(x_i) \prod_{k \in \mathcal{N}(i)} m_{k,i}(x_i), \tag{5.1}$$

$$\forall (i,j) \in E,\ u(x_i, x_j) = \psi_i(x_i)\psi_j(x_j)\psi_{ij}(x_i, x_j) \prod_{k \in \mathcal{N}(i)\setminus j} m_{k,i}(x_i) \prod_{k \in \mathcal{N}(j)\setminus i} m_{k,j}(x_j), \tag{5.2}$$

Le résultat se démontre par récurrence sur le nombre de sommets. L'hypothèse de récurrence est que pour tout arbre de taille n, toute famille de potentiels, toute racine, et tout ordre topologique correspondant, les équations (5.1) et (5.2) sont vraies.

5.2.1 $n = 2$

Si T a deux sommets 1 et 2, alors deux messages sont passés, de 1 vers 2: $m_{12}(x_2) = \sum_{x_1} \psi_1(x_1)\psi_{12}(x_1, x_2)$, puis de 2 vers 1: $m_{21}(x_1) = \sum_{x_2} \psi_2(x_2)\psi_{12}(x_1, x_2)$. D'autre part, on a par définition

$$\begin{aligned} u(x_1) &= \sum_{x_2} \psi_1(x_1)\psi_2(x_2)\psi_{12}(x_1, x_2) \\ &= \psi_1(x_1) \sum_{x_2} \psi_2(x_2)\psi_{12}(x_1, x_2) \\ u(x_1) &= \psi_1(x_1)m_{21}(x_1), \end{aligned}$$

et de faon analogue, $u(x_2) = \psi(x_2)m_{12}(x_2)$. Ainsi, 5.1 est vérifiée.

Enfin, on a directement par définition $u(x_1, x_2) = \psi(x_1)\psi(x_2)\psi(x_1, x_2)$. D'où 5.2 est également vérifiée.

On a donc montré que le résultat est vrai pour $n = 2$.

5.2.2 $n - 1 \longrightarrow n$

On suppose que le résultat est vrai pour les arbres de taille $n - 1$, et on considère un arbre de taille n, avec racine et ordre correspondant.

Comme le sommet n (dernier dans l'ordre topologique) n'a qu'un seul voisin π_n, le premier message passé, de n vers π_n est

$$m_{n\pi_n}(x_{\pi_n}) = \sum_{x_n} \psi_n(x_n)\psi_{n\pi_n}(x_n, x_{\pi_n})$$

Le dernier message passé, de π_n vers n est égal à:

$$m_{\pi_n n}(x_n) = \sum_{x_{\pi_n}} \psi_{\pi_n}(x_{\pi_n})\psi_{n\pi_n}(x_n, x_{\pi_n}) \prod_{k \in \mathcal{N}(\pi_n)\setminus\{n\}} m_{k\pi_n}(x_{\pi_n})$$

Nous allons construire un arbre $\tilde{T}$ de taille $n - 1$, ainsi qu'une famille de potentiels, de telle sorte que les $2(n-2)$ messages passés dans T (i.e., tous les messages exceptés le premier et le dernier) soient égaux aux $2(n - 2)$ messages passés dans $\tilde{T}$. On définit l'arbre et les potentiels comme suit:

- $\tilde{T} = (\tilde{V}, \tilde{E})$ avec $\tilde{V} = \{1, \ldots, n - 1\}$ et $\tilde{E} = E\setminus\{n, \pi_n\}$ (i.e., c'est le sous-arbre correspondant aux $n - 1$ premiers sommets).
- Les potentiels sont tous les mêmes que ceux de T, excepté le potentiel $\tilde{\psi}_{\pi_n}(x_{\pi_n}) = \psi_{\pi_n}(x_{\pi_n})m_{n\pi_n}(x_{\pi_n})$.

- La racine est inchangée et l'ordre topologique est aussi conservé.

Le produit des potentiels de l'arbre de taille $n-1$, est égale à:

$$\begin{aligned}
\tilde{u}(x_1,\ldots,x_{n-1}) &= \prod_{i=1}^{n-1}\psi_i(x_i)\left(\prod_{(i,j)\in E\setminus\{n,\pi_n\}}\psi_{ij}(x_i,x_j)\right)m_{n\pi_n}(x_{\pi_n}) \\
&= \prod_{i=1}^{n-1}\psi_i(x_i)\left(\prod_{(i,j)\in E\setminus\{n,\pi_n\}}\psi_{ij}(x_i,x_j)\right)\sum_{x_n}\psi_n(x_n)\psi_{n\pi_n}(x_n,x_{\pi_n}) \\
&= \sum_{x_n}\prod_{i=1}^{n}\psi_i(x_i)\prod_{(i,j)\in E}\psi_{ij}(x_i,x_j) \\
&= \sum_{x_n}u(x)
\end{aligned}$$

et est donc égal à la marginalisation de $u(x)$ aux $n-1$ premiers sommets.

Par construction, tous les messages passés dans $\tilde{T}$ correspondent aux messages passés dans T (hormis le premier et le dernier). Nous montrons maintenant que les équations (5.1) et (5.2) sont vraies.

Cas 1 : $i \neq n, i \neq \pi_n$

Les messages correspondant à ces noeuds sont les mêmes dans les deux arbres T et $\tilde{T}$, et comme les marginales sont les mêmes (car $\tilde{u}$ est elle-même la marginalisation de u sur les $n-1$ premiers sommets), les équations (5.1) et (5.2) sont vraies par hypothèse de récurrence. En particulier, (5.1) est vraie pour tout $i \notin \{n, \pi_n\}$ et (5.2) est vraie pour tout $i \notin \{n, \pi_n\}$ et j voisin de i.

Cas 2 : $i = \pi_n$

Dans ce cas, par hypothèse de récurrence,

$$\begin{aligned}
\tilde{u}(x_{\pi_n}) &= \tilde{\psi}_{\pi_n}(x_{\pi_n})\prod_{k\in\tilde{\mathcal{N}}(\pi_n)}m_{k\pi_n}(x_{\pi_n}) \quad \text{(produit sur les voisins de } \pi_n \text{ dans } \tilde{T}) \\
&= \tilde{\psi}_{\pi_n}(x_{\pi_n})\prod_{k\in\mathcal{N}(\pi_n)\setminus\{n\}}m_{k\pi_n}(x_{\pi_n}) \\
&= \psi(x_{\pi_n})m_{n\pi_n}(x_{\pi_n})\prod_{k\in\mathcal{N}(\pi_n)\setminus\{n\}}m_{k\pi_n}(x_{\pi_n}) \text{ par définition de } \tilde{\psi}_{\pi_n} \\
&= \psi(x_{\pi_n})\prod_{k\in\mathcal{N}(\pi_n)}m_{k\pi_n}(x_{\pi_n})
\end{aligned}$$

Comme $\tilde{u}(x_{\pi_n}) = u(x_{\pi_n})$, l'équation (5.1) est vérifiée.

Cas 3 : $i = n$

Il reste à montrer que les marginales sur n et (n, π_n) sont effectivement correctes. On a:

$$\begin{aligned} u(x_n, x_{\pi_n}) &= \sum_{\substack{x_i \\ i\neq n, i\neq \pi_n}} u(x) \\ &= \psi_n(x_n)\psi_{\pi_n}(x_{\pi_n})\psi_{n\pi_n}(x_n, x_{\pi_m}) \underbrace{\sum_{\substack{x_i \\ i\neq n, i\neq \pi_n}} \frac{u(x)}{\psi_n(x_n)\psi_{\pi_n}(x_{\pi_n})\psi_{n\pi_n}(x_n, x_{\pi_n})}}_{\alpha(x_{\pi_n})}, \end{aligned}$$

et par conséquent

$$\begin{aligned} u(x_{\pi_n}) &= \sum_{x_n} u(x_n, x_{\pi_n}) \\ &= \left(\sum_{x_n} \psi_n(x_n)\psi_{n\pi_n}(x_n, x_{\pi_n})\right) \psi_{\pi_n}(x_{\pi_n})\alpha(x_{\pi_n}) \\ &= m_{n\pi_n}(x_{\pi_n})\psi_{\pi_n}(x_{\pi_n})\alpha(x_{\pi_n}). \end{aligned}$$

On en déduit alors

$$\alpha(x_{\pi_n}) = \frac{1}{\psi_{\pi_n}(x_{\pi_n})} \frac{u(x_{\pi_n})}{m_{n\pi_n}(x_{\pi_n})}.$$

et

$$u(x_n, x_{\pi_n}) = \psi_n(x_n)\psi_{n\pi_n}(x_n, x_{\pi_n}) \frac{u(x_{\pi_n})}{m_{n\pi_n}(x_{\pi_n})}.$$

En utilisant le résultat montré pour le cas 2, on obtient:

$$\begin{aligned} u(x_n, x_{\pi_n}) &= \psi_n(x_n)\psi_{n\pi_n}(x_n, x_{\pi_n}) \frac{\psi_{\pi_n}(x_{\pi_n}) \prod_{k\in\mathcal{N}(\pi_n)} m_{k\pi_n}(x_{\pi_n})}{m_{n\pi_n}(x_{\pi_n})} \\ &= \psi_n(x_n)\psi_{\pi_n}(x_{\pi_n})\psi_{n\pi_n}(x_n, x_{\pi_n}) \prod_{k\in\mathcal{N}(\pi_n)\setminus n} m_{k\pi_n}(x_{\pi_n}) \end{aligned}$$

ce qui montre le résultat pour la probabilité jointe sur x_n, x_{π_n}. En sommant par rapport à x_{π_n}, on obtient immédiatement le résultat pour $u(x_n)$:

$$\begin{aligned} u(x_n) &= \sum_{x_{\pi_n}} u(x_n, x_{\pi_n}) \\ &= \sum_{x_{\pi_n}} \psi_n(x_n)\psi_{\pi_n}(x_{\pi_n})\psi_{n\pi_n}(x_n, x_{\pi_n}) \prod_{k\in\mathcal{N}(\pi_n)\setminus n} m_{k\pi_n}(x_{\pi_n}) \\ &= \psi_n(x_n) \sum_{x_{\pi_n}} \psi_{\pi_n}(x_{\pi_n})\psi_{n\pi_n}(x_n, x_{\pi_n}) \prod_{k\in\mathcal{N}(\pi_n)\setminus n} m_{k\pi_n}(x_{\pi_n}) \\ &= \psi_n(x_n) m_{\pi_n n}(x_n) \text{ par définition de } m_{\pi_n n} \end{aligned}$$

5.3 Chaîne de Markov Cachée (HMM)

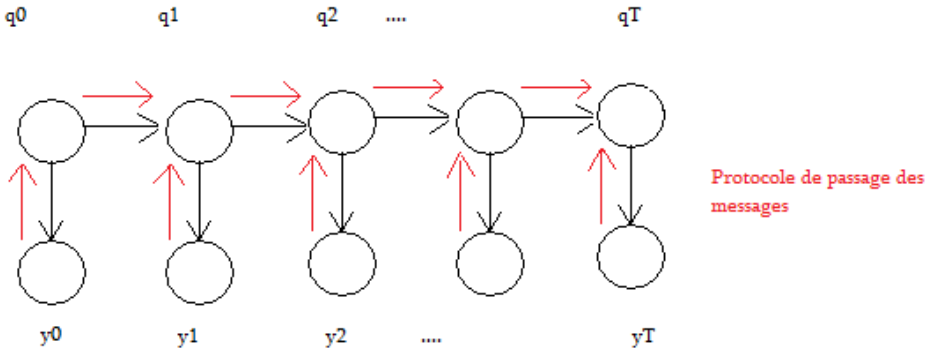


Figure 5.3. Modèle de Markov Cachés

On note $q_0...q_T$ les états discrets, et $y_0...y_T$ les observations discrètes ou continues.

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.

Les observations sont décrites par : $y_O^*...y_T^* \Longrightarrow \delta(y_t = y_t^*)$

On prend comme modèle $p(q, y) = p(q_0) \prod_{t=1}^{T} p(y_t|q_t) \prod_{t=1}^{T} p(q_t|q_{t-1})$

et comme fonction $u(q, y) = p(q, y) \prod_{t=1}^{T} \delta(y_t = y_t^*)$

5.3.1 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 :

$m_{y_0,q_0}(q_0) = p(y_0|q_0)$

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

...

$m_{y_t,q_t}(q_t) = p(y_t|q_t)$

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

Proprieté 5.6 *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 5.7

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

Le calcul de $p(y)$ se fait donc en $O(Tk^2)$ opérations (car chaque récursion est un produit matrice/vecteur). On effectue âĹŽ 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 5.8 $p(q_t, y_0, ...y_T) = \alpha_t(q_t)\beta_t(q_t)$,

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

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

Le graphe moralisé d'un DAG est un arbre uniquement si ce DAG ne contient pas de V-structures.

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 complète:

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

Pour l'étape E de l'algorithme EM, on calcule donc :

- $\mathbb{E}(\delta(q_0 = i)|y) = p(q_0 = i|y)$
- $\mathbb{E}(\delta(q_t = i)|y) = p(q_t = i|y)$
- $\mathbb{E}(\delta(q_{t+1} = i, q_t = j|y) = p(q_{t+1} = i, q_t = j|y)$

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.

5.4 Apprentissage dans les modèles graphiques orientés

Théorème 5.9
- *Soit G un DAG sur* $X_1, X_2, ...X_p$
- *Soient n observations complètes iid de* $X_1, X_2, ...X_p$
- *Supposons que chaque proba conditionnelle ne dépende que d'un seul paramètre :* $p(x_i|x_{\pi_i}, \theta_i)$

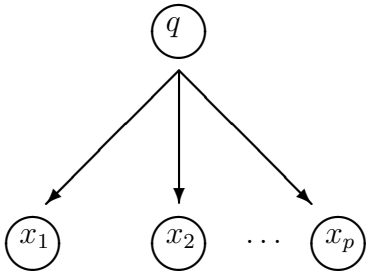
Alors le Maximum de Vraisemblance se découple.


Ce théorème ne s'applique que quand toutes le hypothèses sont satisfaites: (2) données complètes, (2) données i.i.d., (3) Modèle orienté, (4) paramétrisations découplées.

5.5 Naive Bayes

Cadre On considère un document texte q formé de mots pris dans un ensemble de mots de taille p (l'ordre de grandeur de p est 10^4). On définit une variable $x_i \in \{0, 1\}$ valant 1 si le i^{me} mot est dans le document et 0 sinon. On va chercher à classifier le document q parmi une ensemble de K classes de document.

Hypothèse Sachant la classe y, les mots sont indépendants.



Pratique On va evaluer $p(x_i = 1|q = k) = \eta_{ik}$ et ainsi par Bayes on pourra estimer $p(q = k|x_i$ pour $i \in \{1, p\})$ et ainsi classifier le document q.

5.6 Vecteurs Gaussiens

Cadre On ne considère plus des variables discretes mais gaussiennes. Posons $x = (x_1, ..., x_n) \in \mathbb{R}^n$ un vecteur gaussien de moyenne μ et de matrice de variance-covariance Σ.

Remarques préliminaires

- Supposer $\mu = 0$ ne change rien au problème
- $\Sigma_{i,j} = \mathbb{E}[x_i x_j]$
- $\Sigma_{i,j} = 0 \iff X_i \perp X_j$

5.6.1 Condition de factorisation dans un graphe

Proprieté 5.10 *Soit $G = (V, E)$ un modèle graphique non orienté.*
$p(x) \in \mathcal{L}(G) \iff \forall (i, j) \notin E \Rightarrow (\Sigma^{-1})_{i,j} = 0$

Démonstration En développant le produit matriciel, on constate que :

$$p(x) \propto \prod_{i,j} exp(-\frac{1}{2}(x_i - \mu_i)(x_j - \mu_j)(\Sigma^{-1})_{i,j})$$

or :

$$\prod_{i,j} exp(-\frac{1}{2}(x_i - \mu_i)(x_j - \mu_j)(\Sigma^{-1})_{i,j}) = \prod_{(i,j)\in E} exp(-\frac{1}{2}(x_i - \mu_i)(x_j - \mu_j)(\Sigma^{-1})_{i,j})$$

D'où la conclusion. ■