

## 3.1 Élimination de graphe

### 3.1.1 Algorithme d'élimination

Soit  $G = (V, E)$  un graphe non orienté avec  $V = \{1, \dots, n\}$ . On considère l'ordre d'élimination  $I = (n, n-1, \dots, 1)$ . A chaque étape, l'algorithme élimine le noeud qui suit dans l'ordre  $I$ , c'est-à-dire, supprime le noeud du graphe et connecte les voisins restants, i.e., cet algorithme construit une séquence de graphes  $G_n = G, G_{n-1}, \dots, G_0 = \emptyset$  tels que :

$$V_k = \{1, \dots, k\}$$

$$E_{k-1} = (E_k \cap V_{k-1} \times V_{k-1}) \cup N_{G_k}(k) \times N_{G_k}(k)$$

(où  $N_{G_k}(k)$  désigne l'ensemble des voisins du sommet  $k$  dans  $G_k$ ).

**Définition 3.1** *Le graphe reconstitué est le graphe  $\tilde{G} = (V, \tilde{E} = \cup_k E_k)$ , dont l'ensemble des arêtes contient l'ensemble  $E$  d'origine ainsi que les nouvelles arêtes créées au cours de l'exécution de l'algorithme.*



Ce graphe a des propriétés importantes étudiées plus tard dans le cours. La principale est qu'il est triangulé (cette procédure d'élimination est en fait un algorithme de triangulation de graphe).

**Définition 3.2** *On appelle clique d'élimination du noeud  $k$  l'ensemble des voisins d'un noeud après son élimination, le noeud lui-même y compris, i.e., l'ensemble  $\{k\} \cup N_{G_k}(k)$ .*

### 3.1.2 Élimination et marginalisation

On considère une fonction  $u(x)$  qui se factorise dans le graphe  $G$ , i.e., qui s'écrit

$$u(x) = \prod_{C \in \mathcal{C}} \psi_C(x_C)$$

où  $\mathcal{C}(G)$  désigne l'ensemble des cliques du graphe  $G$ .

**Proposition 3.3** *La marginale de  $u(x)$  sur  $\{x_1, \dots, x_k\}$*

$$u(x_1, \dots, x_k) = \sum_{x_i, i > k} u(x) = \prod_{C \in \mathcal{C}(\tilde{E}_k)} \tilde{\Psi}_C(x_C)$$

*se factorise dans  $G_k$ .*

**Proposition 3.4** *Passer de  $u(x_1, \dots, x_k)$  à  $u(x_1, \dots, x_{k-1})$  correspond à une somme sur  $\#\left(\{k\} \cup N_{G_k}(k)\right)$  éléments.*

### 3.1.3 Complexité

En supposant que les variables prennent toutes  $r$  valeurs, la version naïve de l'algorithme se fait en  $r^n$  opérations. L'algorithme d'élimination réduit ce nombre à

$$\sum_{k=1}^n r^{\#\left(\{k\} \cup N_{G_k}(k)\right)}$$

De manière générale, la complexité de l'algorithme est déterminée par le nombre de variables de la plus grande clique d'élimination, i.e., la complexité de l'algorithme de marginalisation par élimination est exponentielle dans la largeur arborescente. Réduire la complexité de l'algorithme d'élimination consiste à minimiser la taille de la plus grande clique d'élimination (que l'on note  $C(I, G)$ ) en jouant sur l'ordre d'élimination.

**Définition 3.5** *La largeur arborescente (treewidth) est égale à  $\min_I C(I, G) - 1$ .*

Le problème général de trouver le meilleur ordre d'élimination possible est un problème NP-difficile. Il y a un certain nombre de méthodes heuristiques efficaces qui permettent dans la pratique de trouver un bon ordre d'élimination. Beaucoup de graphes ont des ordres d'élimination optimaux évidents. Par exemple, pour un arbre, on élimine par les feuilles ( $tw = 1$ ).

### 3.1.4 Elimination et DAG

Pour les DAG, on applique l'algorithme d'élimination au graphe moralisé, car si une loi se factorise dans un DAG, elle se factorise aussi dans le graphe moralisé.

### 3.1.5 Conclusion

L'algorithme d'élimination est utile pour calculer des probabilités conditionnelles.

Soit un graphe  $G$  et  $p(x) \in L(G)$ . On observe les variables indexées par  $E$  (par convention, les noeuds sont grisés). On veut calculer  $p(x_R|x_E = \bar{x}_E)$  où  $R$  est un sous-ensemble disjoint de  $E$ .

Cas particulier :

$$p(x_1|\bar{x}_E) = \frac{p(x_1, \bar{x}_E)}{\sum_{x'_1} p(x'_1, \bar{x}_E)}$$

$$p(x_1, \bar{x}_E) = \sum_{x_2, \dots, x_n} p(x) \prod_{e \in E} \delta(x_e = \bar{x}_e)$$

est un produit de potentiel se factorisant et auquel on peut appliquer l'algorithme d'élimination.

On aimerait cependant calculer  $p(x_1|x_E = \bar{x}_E)$ ,  $p(x_2|x_E = \bar{x}_E)$ , ...,  $p(x_r|x_E = \bar{x}_E)$  ( $r \in R$ ) de manière jointe pour éviter les redondances de calcul. Par exemple, pour  $p(x_1|x_E = \bar{x}_E)$ ,  $p(x_2|x_E = \bar{x}_E)$ , toutes les étapes impliquant  $x_3, \dots, x_n$  seront répétées. La procédure générale permettant ce calcul s'inspire de l'algorithme somme-produit expliqué dans la partie suivante.

## 3.2 Arbre non orienté et 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  $u(x)$  définies par

**Définition 3.6** *Les lois marginales sont données par :*

$$u(x_i) = \sum_{x_{i,j} \neq i} u(x)$$

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

On considère les messages suivants, avec un protocole de passage particulier :

**Définition 3.7** *Pour un couple  $(i, j) \in E$ , le message  $m_{ij}$  de  $i \rightarrow j$  (fonction de  $x_j$ ) est égal à :*

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

**Définition 3.8** *Le protocole de passage des messages vérifie que le message de  $i \rightarrow j$  est passé après avoir reçu les messages de tous les voisins de  $i$ .*

Noter que ceci implique de commencer le passage des messages par les feuilles de l'arbre (i.e., des sommets avec un seul voisin). Lorsque les messages sont passés en série, les ordres de passage respectant le protocole sont tous obtenus comme suit :

1. définir une “racine”  $r$  (n'importe quel sommet)
2. construire l'unique arbre orienté avec  $r$  comme racine.
3. déterminer un ordre topologique pour l'arbre orienté (sans perte de généralité, on considère que cet ordre est  $(1, \dots, n)$ ).

Nous pouvons alors distinguer 2 phases :

- la collecte (flèche pointillée)

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

- la distribution (flèche en tiret)

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

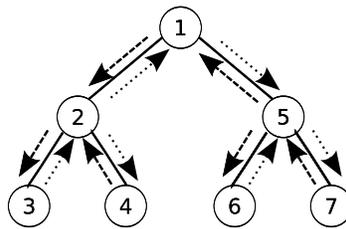


FIG. 3.1. Collecte et Distribution des messages

**Théorème 3.9** 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 méthodes approchées d'inférence).

### 3.2.1 Complexité

La complexité de l'algorithme de passage d'un message de  $i \rightarrow j$  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  $O(r^2n)$ , i.e., linéaire en  $n$ .

### 3.2.2 Algorithme max-produit

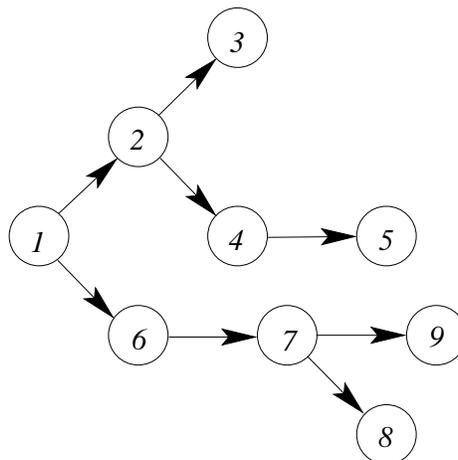
Dans le contexte du cours, celui d'un semi-anneau commutatif  $(\mathbb{R}, +, \times)$  la distributivité  $a.b + a.c = a(b + c)$  peut être réécrite  $\max(a.b, a.c) = a \max(b, c)$  sur le semi-anneau commutatif  $(\mathbb{R}_+, \max, \times)$ .

Du coup, 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

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

## 3.3 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 étiquetés en suivant l'ordre topologique, i.e.,  $V = \{1, \dots, n\}$ . Voir figure.



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

$$n \rightarrow \pi_n, n - 1 \rightarrow \pi_{n-1}, \dots, 2 \rightarrow \pi_2, \pi_2 \rightarrow 2, \dots, \pi_n \rightarrow n$$

Cette procédure respecte bien le protocole de passage des messages, i.e., un sommet n'envoie un message que quand il a reçu un message de tous ses autres voisins.

Nous allons démontrer qu'une fois les  $2(n - 1)$  messages passés, les marginales sont égales à

$$\forall i \in V, u(x_i) = \psi_i(x_i) \prod_{k \in \mathcal{N}(i)} m_{k,i}(x_i), \quad (3.1)$$

$$\forall (i, j) \in E, u(x_i, x_j) = \psi_i(x_i) \psi_j(x_j) \psi_{ij}(x_i, x_j) \prod_{k \in \mathcal{N}(i) \setminus \{j\}} m_{k,i}(x_i) \prod_{k \in \mathcal{N}(j) \setminus \{i\}} m_{k,j}(x_j), \quad (3.2)$$

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 (3.1) et (3.2) sont vraies.

### 3.3.1 $n = 2$

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 façon analogue,  $u(x_2) = \psi(x_2) m_{12}(x_2)$ . Ainsi, (3.1) 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 (3.2) est également vérifiée.

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

### 3.3.2 $n - 1 \rightarrow n$

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  $\pi_n$ , le premier message passé, de  $n$  vers  $\pi_n$  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  $\pi_n$  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  $\tilde{V} = \{1, \dots, n - 1\}$  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 :

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

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 (3.1) et (3.2) 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 (3.1) et (3.2) sont vraies par hypothèse de récurrence. En particulier, (3.1) est vraie pour tout  $i \notin \{n, \pi_n\}$  et (3.2) 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 (3.1) est vérifiée.

**Cas 3 :**  $i = n$

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

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

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

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

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  $x_n, x_{\pi_n}$ . En sommant par rapport à  $x_{\pi_n}$ , on obtient immédiatement le résultat pour  $u(x_n)$  :

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

### 3.4 Estimation de lois à partir de données

Sur un DAG, la loi  $p_\theta$  est définie par :

$$p_\theta(x) = \prod_{i=1}^m p_{\theta_i}(x_i | \pi_{x_i})$$

Le but de cette section et des cours suivant est de voir comment estimer  $\theta_j$  à partir de données, dans les cas simples (1 ou 2 sommets) puis dans le cas général des DAGs.



Attention aux indices : les données sont indexées par  $i$  et  $j$  avec  $i \in 1, \dots, n$  la taille de l'échantillon et  $j \in 1, \dots, m$  le nombre de variables.

#### 3.4.1 Cadre Bayésien

On étudie la loi  $p_\theta(x)$  avec en supposant que  $\theta$  une variable aléatoire. On définit alors la probabilité à priori :  $p(\theta)$  et la vraisemblance :  $p(x|\theta) = p_\theta(x)$ , ce qui permet d'en déduire la loi à postériori (par la règle de Bayes)

$$p(\theta|x) = \frac{p(x|\theta)p(\theta)}{p(x)}$$

### 3.4.2 Cadre fréquentiste

Il faut trouver un bon estimateur  $\hat{\theta}(x_1, \dots, x_k)$  et l'évaluer. L'estimateur utilisé dans ce cours sera le maximum de vraisemblance :  $\max_{\theta} p_{\theta}(x)$ , qui jouit de propriétés numériques (convexité) et statistiques (en théorie asymptotique) intéressantes.

## 3.5 Estimation de loi multinomiale

Soit une variable  $x$  prenant  $q$  valeurs  $\{1, \dots, q\}$ . La loi est paramétrée par un vecteur  $\pi \in \mathbb{R}^q$  tel que  $\pi \geq 0$  and  $\sum_i \pi_i = 1$ . Soit un échantillon  $x_1, \dots, x_n$  i.i.d. (indépendant et identiquement distribué). La vraisemblance est donnée par

$$\begin{aligned} p_{\pi}(x_1, \dots, x_n) &= \prod_{j=1}^n p_{\pi}(x_j) = \prod_{j=1}^n \prod_{i=1}^q \pi_i^{\delta(x_j=i)} \\ &= \prod_{i=1}^q \pi_i^{\sum_{j=1}^n \delta(x_j=i)} \\ &= \prod_{i=1}^q \pi_i^{n_i} \end{aligned}$$

où  $n_i = \sum_{j=1}^n \delta(x_j = i)$  est le nombre de valeurs  $i$  dans l'échantillon.

Le max de vraisemblance est donné par :

$$\max_{\pi \geq 0, \sum_i \pi_i = 1} \log \left( \prod_{i=1}^q \pi_i^{n_i} \right)$$

soit

$$\max_{\pi \geq 0, \sum_i \pi_i = 1} \left( \sum_{i=1}^q n_i \log \pi_i \right)$$

On introduit les multiplicateurs de Lagrange pour la contrainte  $\sum_i \pi_i = 1$  et le Lagrangien

$$\mathcal{L}(\pi, \lambda) = \sum_{i=1}^q n_i \log \pi_i - \lambda \left( \sum_i \pi_i - 1 \right).$$

On maximise par rapport à  $\pi \geq 0$  et on obtient :  $n_i/\pi_i = \lambda$ , i.e,  $p_i$  proportionnel à  $n_i$ , ce qui impose donc la solution suivante

$$\pi_i = \frac{n_i}{\sum_j n_j}.$$

NB : pour une introduction à l'optimisation convexe avec contraintes, voir par exemple le livre suivant : "Optimisation continue, Cours et problèmes corrigés", de Frédéric Bonnans (éd. Dunod).