4 Modèle de Markov caché

Un modèle de Markov caché (*hidden Markov model* — HMM) sur un alphabet $\mathcal{A}$ comprend les paramètres suivants. W(fr)

- ⋆ ensemble d'états $\mathcal{Q}$ de taille $|\mathcal{Q}| = N$
- ⋆ probabilités de transition entre états : $\tau\colon \mathcal{Q} \times \mathcal{Q} \to [0,1]$
- ⋆ probabilités d'émission à chaque état : $p\colon \mathcal{Q} \times \mathcal{A} \to [0,1]$
- ⋆ probabilités d'état initial : $\pi\colon \mathcal{Q} \to [0,1]$

Le modèle $M = (\mathcal{Q}, \tau, p, \pi)$ définit une distribution sur séquences d'une longueur fixe. En particulier, M produit une séquence $X[0..n-1]$ selon la procédure aléatoire suivante.

G1	choisir l'état initial $q[0]$ au hasard selon les probabilités π
G2	**for** $i \leftarrow 0, 1, \ldots, n-1$
G3	choisir $X[i]$ au hasard par les probabilités d'émission $p(q[i], \cdot)$
G4	choisir prochain état $q[i+1]$ au hasard par les probabilités de transition $p(q[i], \cdot)$

Questions fondamentales. On cherche la solution à trois questions fondamentales [Rabiner «A tutorial on hidden Markov models and selected applications in speech recognition», *Proceedings of the IEEE*, **77** :257–286, 1989] :

1. Si on obsèrve la séquence x, comment peut on calculer $\mathbb{P}\Big\{X = x \;\Big|\; M\Big\}$, la probabilité que M engèndre x ?
2. Si on obsèrve la séquence x, comment peut on trouver la séquence d'états $q[0..n-1]$ qui y correspond le mieux ?
3. Comment doit on choisir les paramètres τ, p, π ?

Vraisemblance. La vraisemblance selon M se définit par

$$L(M) = \mathbb{P}\Big\{X = x \;\Big|\; M\Big\} = \sum_{q[0..n-1] \in \mathcal{Q}^n} \begin{array}{ll} \Big(\pi\big(q[0]\big) \cdot p\big(q[0], x[0]\big) & \textit{(1er état+symbole)} \\ \times \tau\big(q[0], q[1]\big) \cdot p\big(q[1], x[1]\big) & \textit{(2e état+symbole)} \\ \times \cdots & \\ \times \tau\big(q[n-2], q[n-1]\big) \cdot p\big(q[n-1], x[n-1]\big)\Big). & \end{array} \tag{4.1}$$

Sommation sur N^n séquences d'états...

Probabilité de préfixe et de suffixe. On définit la vraisemblance sur les préfixes (*forward probabilities*)

$$\alpha_k(i) = \sum_{q[0..k]; q[k]=i} \mathbb{P}\Big\{X[0..k] = x[0..k] \;\Big|\; M\Big\},$$

ce qu'on peut calculer par programmation dynamique.

$$\alpha_0(q) = \pi(q) \tag{4.2a}$$

$$\alpha_k(i) = \Big(\sum_{j\in\mathcal{Q}} \alpha_{k-1}(j)\cdot\tau(j,i)\Big)\cdot p\big(i,x[k]\big) \qquad \{k>0\} \tag{4.2b}$$

On peut faire le même genre de calcul dans l'autre sens aussi, en définissant la vraisemblance de suffixes (*backward probabilities*) :

$$\beta_n(i) = 1 \qquad \{i\in\mathcal{Q}\} \tag{4.3a}$$

$$\beta_k(i) = \sum_{j\in\mathcal{Q}} \tau(i,j)\cdot p\big(j,x[k+1]\big)\cdot\beta_{k+1}(j) \qquad \{k<n-1\} \tag{4.3b}$$

On peut calculer tout α ou tout β en $O(N^2 n)$ temps.

Probabilité postérieure d'un état. Quel est l'état de la chaîne quand elle émet $x[k]$? La probabilité suivante considère toute séquence d'états avec $z[k] = i$ fixé :

$$\gamma_k(i) = \frac{\alpha_k(i)\cdot\beta_k(i)}{\sum_j \alpha_k(j)\cdot\beta_k(j)}. \tag{4.4}$$

La dénominateur sert à normaliser le probabilités : c'est la vraisemblance $L = \sum_j \alpha_k(j)\cdot\beta_k(j) = \sum_j \alpha_j[n-1]$. $\gamma_k(i)$ est la probabilité postérieure d'être en état i à position k.

W(fr)

Chemin Viterbi. On sait quel est l'état le plus probable (maximisant γ_k) en chaque position k. Mais quel est la *séquence* d'états la plus probable ou le **chemin Viterbi** ? Pour cela on définit la vraisemblance du meilleur préfixe, ce qu'on peut calculer par programmation dynamique :

$$\delta_0(i) = \pi(i)\cdot p\big(i,x[0]\big) \tag{4.5a}$$

$$\delta_k(i) = \Big(\max_{j\in\mathcal{Q}} \delta_{k-1}(j)\cdot\tau(j,i)\Big)\cdot p\big(i,x[k]\big) \qquad \{k>0\} \tag{4.5b}$$

On reconstruit le *chemin Viterbi* qui correspond à $\max_i \delta_{n-1}(i)$ par *backtracking* : il faut stocker le meilleur état précedent j à côté de $\delta_k(i)$ dans la récurrence de (4.5b).

Apprentissage de paramètres. On peut *entraîner* un modèle HMM à l'aide d'un échantillon de séquences (ou une seule séquence). En **apprentissage Viterbi**, on construit le chemin Viterbi, et on utilise les fréquences empiriques des transitions d'états, et celle des émissions pour estimer p et τ. Une telle procédure est utile en pratique, mais ne convèrge pas nécessairement. La méthode la plus appropriée est **l'apprentissage Baum-Welch**. L'idée est de se servir des probabilités forward-backward. Au lieu de compter les fréquences de transitions et émissions avec un seul chemin, on calcule plutôt des espérances incluant tous les chemins. Par exemple, le nombre de transitions entre états $i \to j$ en espérance égale

$$\hat\tau(i,j) = \frac{\sum_{k=1}^{n-1} \alpha_i(k-1)\tau(i,j)p\big(j,x[k]\big)\beta_j(k)}{\sum_{i',j'}\sum_{k=1}^{n-1} \alpha_{i'}(k-1)\tau(i,j)p\big(j',x[k]\big)\beta_{j'}(k)}.$$

Ou bien, on écrit $\hat\tau(i,j) \propto \sum_{k=1}^{n-1} \alpha_i(k-1)\tau(i,j)p\big(j,x[k]\big)\beta_j(k)$ avec le symbole de proportionnalité $\propto$.

Segmentation GC avec un HMM. Klein, Misulovin et Eddy [2002] on utilisé un HMM à deux états pour trouver des régions de taux GC élevée. Le HMM donne une distribution pour chaque segmentation z. Le modèle donc fournit la probabilité *a priori* des segmentations. Soit $z[0..n-1]$ un segmentation hypothétique. En séparant les facteurs en (4.1), on a

$$\log L(M) = \underbrace{\sum_{k=0}^{n-1} \log p\big(z[k], x[k]\big)}_{=\log \mathbb{P}\left\{ X=x \,\middle|\, Z=z \right\}} + \underbrace{\log \pi(z[0]) + \log \sum_{k=1}^{n-1} \log \tau(z[k-1], z[k])}_{=\log \mathbb{P}\{Z=z\}}$$

On retrouve de nouveau la vraisemblance de données avec une pénalisation de complexité.