4.2 Calculer la probsabilité d'émission d'une séquence

La formule

$$P(o_1 o_2 \dots o_n) = \sum_{s_1 s_2 \dots s_n} \pi(s_1) \prod_{i=1}^{n-1} T(s_i, s_{i+1}) \prod_{i=1}^{n} E(s_i, o_i)$$

ne permet pas de calculer pratiquement la probabilité d'une séquence d'observations. Elle nécessite en effet de considérer *toutes* les séquences d'états possibles $s_1 s_2 \dots s_n$, au nombre de $|S|^n$. Une technique de programmation dynamique permet d'effectuer ce calcul en un temps raisonnable. L'algorithme qui en résulte est appelé *algorithme Forward-Backward* : il peut se décliner, soit en réalisant le calcul de la gauche vers la droite (*forward*), soit de la droite vers la gauche (*backward*).

4.2.1 Algorithme Forward-Backward

Soit $o_1 o_2 \dots o_n$ une séquence d'observations.

Version Forward On définit

$$f_k(i) = Pr(o_1 \dots o_i, s_i = k)$$

la probabilité de générer la séquence $o_1 \dots o_i$ en terminant le processus de génération des états sur l'état k. On a, pour tout $1 \leq k \leq |S|$ et $1 \leq i < n$,

$$f_k(i+1) = \sum_{l=1}^{|S|} f_l(i) T(l,k) E(k, o_{i+1}) \text{ et } f_k(1) = \pi(k) E(k, o_1).$$

Ces formules permettent de calculer

$$P(o_1 o_2 \dots o_n) = \sum_k f_k(n)$$

en temps $O(n|S|^2)$.

Version Backward On définit

$$b_k(i) = Pr(o_{i+1} \dots o_n | s_i = k) \text{ pour } 1 \leq i \leq n-1,$$

la probabilité de générer la séquence $o_{i+1} \dots o_n$ à partir de l'état k. On pose $b_k(n) = 1$.

On a, pour tout $1 \leq k \leq |S|$ et $1 \leq i \leq n-1$,

$$b_k(i) = \sum_{l=1}^{|S|} T(k,l)E(l,o_i)b_l(i+1).$$

Ces formules permettent aussi de calculer

$$P(o_1o_2\dots o_n) = \sum_k \pi(k)E(k,o_1)b_k(1)$$

en temps $O(n|S|^2)$.

Exercice 12 Écrivez les méthodes `pfw(self,w)` et `pbw(self,w)` qui prennent en entrée une séquence `w` et calculent sa probabilité relativement au modèle courant. Remarque : on a toujours intérêt à utiliser les opérateurs linéaires proposés par `numpy` plutôt qu'à les reprogrammer. C'est en particulier le cas pour les produits de matrices et de vecteurs : `dot` pour les produits usuels et l'opérateur de multiplication ordinaire `*` pour le produit terme à terme. Ainsi, on peut réécrire la formule

$$f_k(i+1) = \sum_{l=1}^{|S|} f_l(i)T(l,k)E(k,o_{i+1})$$

en

```
F = np.dot(F,T)*E[:,o]
```

4.2.2 Prédire le prochain symbole

On considère le problème suivant : étant donné un HMM M et une séquence $w = o_1 \dots o_n$ générée à partir de M, prédire le prochain symbole qui sera émis.

Exemple 8

— Soit M_1 le HMM construit sur 1 état et 2 lettres tel que $E(0,0) = E(0,1) = 0.5$. Quelle que soit la séquence w, la probabilité d'émettre 0 est égale à celle d'émettre 1 : 0.5 dans les deux cas. On peut donc répondre indifféremment 0 ou 1 et le taux de succès moyen sera de 50%.

— Soit M_2, le HMM défini à la section précédente. Si après avoir généré w, le processus passe dans l'état E, la probabilité d'émettre 0 est égale à celle d'émettre 1 : il est indifférent de prédire 0 ou 1. En revanche, si le processus passe dans l'état B, la meilleure stratégie consiste à prédire 0 : probabilité de 0.7 contre 0.3. Globalement, il est donc optimal de toujours prédire 0. Comme les probabilités que le processus passe en E ou en B sont égales (puisque $\pi(E) = \pi(B)$ et $T[0,1] = T[1,0]$), le taux de succès moyen sera de 0.6. On ne peut pas faire mieux.

— Soit M_3 le HMM construit sur 2 lettres et deux états et vérifiant : $\pi(0) = 0.4, E[0,0] = 0.2, E[1,0] = 0.7, T[0,0] = 0.9, T[1,1] = 0.8$. La réponse ne paraît pas aussi immédiate.

On cherche à évaluer $P(o_{n+1} = o|w) = P(w,o)/P(w)$. La meilleure prédiction consiste à prédire le symbole o qui maximise $P(w,o)$. On peut donc, pour résoudre le problème, calculer $P(w,o)$ pour toutes les lettres o et choisir celle pour laquelle cette probabilité est maximale. Cette méthode directe peut être sujette à des problèmes d'arrondi pour des mots très longs : en effet, la probabilité $P(w)$ d'une séquence w décroît en général très rapidement vers 0, à mesure que la longueur de w augmente (pourquoi ?). Un calcul intermédiaire permet de contourner ce problème.

Soit $o_1, \ldots, o_n$ une suite d'observations. Posons $h_i(k) = P(s_{i+1} = k|o_1 \ldots o_i)$: probabilité qu'ayant observé $o_1 \ldots o_i$, l'état d'émission suivant soit k. On a

$$\begin{aligned}
h_i(k) &= \sum_{k'} P(s_{i+1} = k|o_1 \ldots o_i, s_i = k')P(s_i = k'|o_1 \ldots o_i) \\
&= \sum_{k'} T(k',k)P(o_i|o_1 \ldots o_{i-1}, s_i = k')\frac{P(s_i = k'|o_1 \ldots o_{i-1})}{P(o_i|o_1 \ldots o_{i-1})} \\
&\propto \sum_{k'} T(k',k)E(k',o_i)h_{i-1}(k')
\end{aligned}$$

où le signe $\propto$ signifie "égal à un coefficient de normalisation près". En remarquant que $h_0(k) = \pi(k)$ et que

$$\begin{aligned}
P(o_{n+1}|o_1 \ldots o_n) &= \sum_k P(o_{n+1}|o_1 \ldots o_n, s_{n+1} = k)P(s_{n+1} = k|o_1 \ldots o_n) \\
&= \sum_k E(k, o_{n+1})h_n(k)
\end{aligned}$$

on en déduit un manière plus robuste de calculer $P(o_{n+1} = o|w)$.

Exercice 13 Écrire une méthode `predit(self,w)` de la classe `HMM` qui retourne le symbole le plus probable après avoir observé w. Évaluez le taux de succès moyen du HMM M_2, sur une chaîne de longueur 1000 ?