Chapitre 12

L'apprentissage de modèles de Markov cachés

Quand les objets sur lesquels porte l'apprentissage sont des séquences d'événements, le concept extrait doit refléter à la fois la nature de ces événements et la manière dont ils s'enchaînent. On a déjà vu au chapitre 2 et au chapitre 3 des exemples d'objets de structure séquentielle. On a vu aussi au chapitre 7 des méthodes pour extraire des concepts sous forme de grammaires à partir d'exemples et de contre-exemples. Nous présentons dans ce chapitre un outil puissant pour induire un concept de nature statistique à partir seulement de séquences d'apprentissage appartenant à ce concept : les modèles de Markov cachés, ou HMM[1].

Par leur nature statistique, les HMM *se situent facilement dans le cadre de la décision bayésienne, qui a été présentée au chapitre 2. En particulier, le principe du maximum de vraisemblance a posteriori (MAP) prescrit d'attribuer une séquence inconnue à la classe qui a la plus grande probabilité de l'avoir engendrée. L'apprentissage consiste donc dans ce cadre à apprendre pour chaque classe de séquences le* HMM *le plus vraisemblable. En pratique, le problème revient à apprendre indépendamment un* HMM *par classe, sans tenir compte des contre-exemples.*

[1] En anglais : *Hidden Markov Models.*

Sommaire

Revenons sur l'étang où nagent des oies et des cygnes. L'ornithologue que nous avons connu débutant au commencement de ce livre est maintenant plus expérimenté. Ce matin, il arrive très tôt pour observer les oiseaux se poser. La veille, il était venu dans la matinée, quand tous les animaux étaient là et il avait observé une trentaine de cygnes et environ quatre-vingts oies. Il espère donc voir arriver une bonne centaine d'oiseaux.

De fait, les vols commencent et il y a bientôt sur le lac les premiers oiseaux. Mais pas dans la proportion attendue : vingt cygnes et quatre oies se sont d'abord posés. Dans les minutes qui suivent, une dizaine d'oiseaux se posent, moitié oies, moitié cygnes. Finalement, arrive le reste de la troupe des oies, soit environ soixante-dix éléments, et de temps en temps un cygne. Au total, ces derniers sont finalement une trentaine.

L'observateur comprend alors que les deux espèces n'ont pas les mêmes habitudes : les cygnes sont plus matinaux, ce qui explique que la proportion entre les deux espèces varie énormément dans le temps d'arrivée. En notant `O` l'arrivée d'une oie et `C` celle d'un cygne, et en mettant un intervalle entre les trois phases d'arrivée, la séquence observée peut se dénoter ainsi :

```
CCCCCCOCCCCOCCCOCCCCOCO      OCCOCOOOCC
OOOOCOOOOOOOOOOOOOOCOCOOOCOOOOOOOOOOOOOOOO
```

Comment décrire ce phénomène ? Attention : il ne s'agit pas seulement d'apprendre cette séquence par cœur. Il faut tenir compte du fait que l'ordre exact d'arrivée des oiseaux ne se reproduira pas exactement à l'identique chaque matin : certains oiseaux ne viendront pas, certains voleront plus ou moins vite, etc. Si l'ornithologue veut expliquer ce qu'il observe et prédire ce qu'un autre pourra observer, il doit donc faire l'apprentissage d'un concept qui décrive de manière satisfaisante les propriétés de telles séquences.

Si cet avimateur a lu le chapitre qui suit, il produira peut-être un concept exprimé sous la forme du tableau et du graphique (figure 12.1) qui sont donnés ci-dessous.

	Probabilité d'observer un cygne	Probabilité d'observer une oie
Période 1	0.8	0.2
Période 2	0.5	0.5
Période 3	0.1	0.9

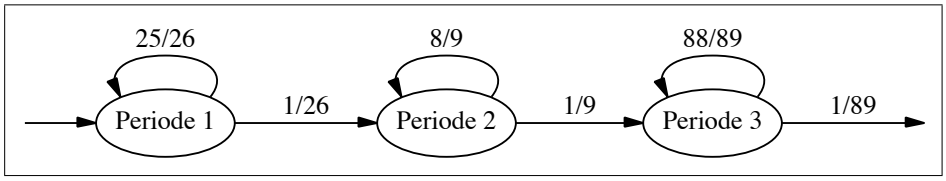

FIG. 12.1: *Comment les cygnes et les oies arrivent sur l'étang.*

Comment interpréter ce modèle ? Le cercle étiqueté « période 1 » correspond à la phase d'arrivée majoritaire des cygnes, celui étiqueté « période 2 » au moment où les populations sont en fréquence d'arrivée égale, et le dernier à l'arrivée massive des oies (avec quelques cygnes parmi elles). La succession temporelle se traduit par un parcours de gauche à droite en suivant les flèches, avec la règle que chaque passage dans un état correspond exactement à l'observation d'un oiseau, cygne ou oie. Quand on est dans un cercle (appelons-le désormais un état), on a deux solutions : soit y rester en faisant une boucle locale, soit passer au suivant. Le passage d'un état à lui-même ou au suivant est commandé par le chiffre situé sur la flèche, qui est une probabilité. Par exemple, dans l'état 1, la probabilité de passer à l'état 2 est de 1/26, celle de rester dans l'état 1 est de 25/26.

Et les oiseaux ? Leur observation est commandée par la table donnée au-dessus du graphe des états. Chaque passage dans l'état 1 correspond avec une probabilité de 0.8 à l'observation d'un cygne et donc de 0.2 à celle d'une oie. Dans l'état 2, l'observation est équiprobable. Quand on est dans l'état 3, il est 9 fois plus probable d'observer une oie qu'un cygne.

Effectuons maintenant un petit calcul. Combien d'oiseaux sont en moyenne observés pendant le séjour dans l'état 1 ? Environ 25, selon une formule simple du calcul des probabilités[2]. Par conséquent, compte tenu des proportions affectées par le tableau, on observera en moyenne $0.8 \times 25 = 20$ cygnes et $0.2 \times 25 = 5$ oies durant la première période représentée par cet état. Un calcul analogue montre que la durée moyenne de séjour dans l'état 2 est d'environ 8 : on y observera donc (en moyenne) 4 cygnes et 4 oies. Finalement, comme la probabilité de bouclage dans l'état 3 est la plus forte, on y reste en moyenne plus longtemps (le calcul donne 88) et on observe, toujours en moyenne, 80 oies et 8 cygnes.

Au total, la séquence moyenne engendrée par ce modèle comporte une trentaine de cygnes et presque trois fois plus d'oies, avec une proportion d'arrivées des cygnes beaucoup plus forte au début qu'à la fin.

Comme nous allons le voir, le concept décrit ci-dessus est un cas particulier de modèle de Markov caché (HMM). Dans un tel modèle, une séquence est donc considérée comme une suite temporelle gérée par ses états. À chaque instant, un nouvel événement de la séquence est analysé. La théorie des HMM décrit comment passer d'état en état à l'aide de probabilités de transitions et comment chaque élément de la séquence peut être émis par un état du HMM, à l'aide de probabilités d'observation par état. Il permet aussi de calculer la probabilité qu'une séquence donnée ait été émise par un HMM donné.

[2] Si x est la probabilité de boucler dans un état et $1 - x$ celle d'en sortir, la durée moyenne de séjour dans cet état vaut $\frac{x}{(1-x)}$.

Les méthodes Hmm sont robustes et fiables grâce à l'existence de bons algorithmes d'apprentissage ; de plus, la règle de décision est rapide à appliquer.

Notations utiles pour le chapitre

n	Le nombre d'états du modèle de Markov caché, ou Hmm
$S = \{s_1, s_2, \dots s_n\}$	Les états du Hmm
M	La taille de l'alphabet des observations quand celles-ci sont de nature discrète
$V = \{v_1, v_2, \dots, v_M\}$	L'alphabet des observations
A	La matrice des probabilités de transitions entre les états
a_{ij}, $i, j \in [1, n]$	Un élément de A
B	La matrice des probabilités d'observation des symboles de V
$b_j(k)$, $j \in [1, n]$,$k \in [1, M]$	Un élément de B
π	Le vecteur des probabilités initiales du Hmm
$\Lambda = (A, B, \pi)$	Un Hmm
T	La longueur d'une séquence observée
$O = O_1 \dots O_t \dots O_T$ avec $O_t \in V$	Une séquence observée
$O(i : j) = O_i \dots O_j$	Une sous-séquence de O
$q_1 \dots q_t \dots q_T$ avec $q_t \in S$	La suite des états qui a émis une séquence
$\mathbf{P}(O \mid \Lambda)$	La probabilité que le Hmm Λ ait émis la séquence O
$\mathcal{O} = O^1 \dots O^m$	Un ensemble d'apprentissage composé de m séquences
$\mathbf{P}(\Lambda \mid \mathcal{O})$	La probabilité que l'ensemble de séquences $\mathcal{O}$ ait été émis par le Hmm Λ.

1. Les modèles de Markov observables

Avant de décrire les Hmm proprement dits, nous présentons un modèle probabiliste plus simple pour l'observation de séquences : les *modèles de Markov observables*.

D'une manière générale, un *processus* ou *modèle stochastique observable* est un processus aléatoire qui peut changer d'état s_i, $i = 1, \dots, n$ au hasard, aux instants $t = 1, 2, \dots, T$. Le résultat observé est la suite des états dans lesquels il est passé. On peut aussi dire de ce processus qu'il *émet* des séquences d'états $S = s_1, s_2, \dots, s_T$. Chaque séquence est émise avec une probabilité[3] $\mathbf{P}(S) = \mathbf{P}(s_1, s_2, \dots, s_T)$. Pour calculer $\mathbf{P}(S)$, il faut se donner la probabilité initiale $\mathbf{P}(s_1)$ et les probabilités d'être dans un état s_t, connaissant l'évolution antérieure.

Un processus stochastique est *markovien*[4] (ou *de Markov*) si son évolution est entièrement déterminée par une probabilité initiale et des probabilités de transitions entre états. Autrement dit, en notant $(q_t = s_i)$ le fait que l'état observé à l'instant t est s_i

$$\forall t, \quad \mathbf{P}(q_t = s_i \mid q_{t-1} = s_j, q_{t-2} = s_k \dots)\mathbf{P}(q_t = s_i \mid q_{t-1} = s_j)$$

[3] Dans ce chapitre, nous étudions principalement des distributions de probabilité sur des ensembles finis.
[4] Au sens strict : markovien *d'ordre 1*.

d'où :

$$\mathbf{P}(q_1 \dots q_T) = \mathbf{P}(q_1) \times \mathbf{P}(q_2 \mid q_1) \times \dots \times \mathbf{P}(q_T \mid q_{T-1})$$

Nous supposons pour simplifier que les processus de Markov auquels nous avons affaire sont *stationnaires* c'est-à-dire que leurs probabilités de transition ne varient pas dans le temps. Cela autorise à définir une *matrice de probabilité de transitions* $A = [a_{ij}]$ telle que :

$$a_{ij} = \mathbf{P}(q_t = s_j \mid q_{t-1} = s_i) \quad 1 \leq i \leq n, \quad 1 \leq j \leq n$$

avec :

$$\forall i, j \;\; a_{ij} \geq 0, \quad \forall i \;\; \Sigma_{j=1}^{j=n} a_{ij} = 1$$

Nous appellerons maintenant pour simplifier *modèle de Markov observable* un processus stochastique observable, markovien et stationnaire.

Dans un tel modèle, il y a un lien direct à tout instant entre l'état où se trouve le processus et l'observation réalisée à cet instant, comme l'illustre la figure 12.2. C'est ce qui caractérise pour nous[5] le fait que ce processus soit observable. Nous allons maintenant voir comment nous débarasser de cette contrainte en présentant d'autres processus stochastiques : les modèles de Markov cachés. Ensuite, nous comparons leur puissance de modélisation sur un exemple.

2. Les modèles de Markov cachés (HMM)

2.1 Définition

Le modèle de Markov caché généralise le modèle de Markov observable car il produit une séquence en utilisant deux suites de variables aléatoires ; l'une cachée et l'autre observable.

- La suite cachée correspond à la suite des états $q_1, q_2, \dots, q_T$, notée $Q(1 : T)$, où les q_i prennent leur valeur parmi l'ensemble des n états du modèle $\{s_1, s_2, \dots, s_n\}$.
- La suite observable correspond à la *séquence des d'observations* $O_1, O_2, \dots, O_T$, notée $O(1 : T)$, où les O_i sont des lettres d'un alphabet de M *symboles observables* $V = \{v_1, v_2, \dots, v_M\}$.

Par conséquent, pour un HMM, un état n'est pas associé exclusivement à une lettre donnée qu'il émettrait à coup sûr : chaque lettre a désormais une certaine probabilité d'être émise par chaque état. En outre, ce ne sont pas les états qui sont observés, mais les lettres qu'ils émettent. Une conséquence importante est que l'on peut maintenant travailler avec des alphabets infinis. Une « lettre » est alors émise avec une certaine densité de probabilité, correspondant à une distribution propre à chaque état.

En pratique, on cherche à construire des HMM représentant des concepts dans l'espace de représentation des séquences. Nous prendrons ici pour simplifier des séquences construites sur un alphabet $V = \{v_1, v_2, \dots, v_M\}$ de taille finie. Mais la remarque ci-dessus doit être gardée à l'esprit : la taille de cet alphabet peut être infinie, ce qui signifie en pratique que chaque état peut émettre une variable continue ou un vecteur de $\mathbb{R}^d$.

2.2 Pourquoi faut-il des variables cachées ?

Montrons sur l'exemple de l'introduction la différence entre le modèle de Markov observable et le modèle de Markov caché. Quand on observe l'arrivée des oiseaux sur un étang, on obtient

[5] Si la même observation peut être affectée à plusieurs états, on peut améliorer la capacité de représentation des modèles observables. Nous ne discutons pas cette possibilité ici.

une suite sur l'alphabet $V = \{\texttt{O}, \texttt{C}\}$. Une séquence observée sera par exemple :

$$O = \texttt{O O C O C O}$$

Les probabilités *a priori* d'observer un cygne ou une oie peuvent être différentes. La construction de deux types de modèles de Markov pour modéliser les séquences sur V va nous conduire à préciser un certain nombre d'éléments relatifs à la nature des états, à leur nombre ainsi qu'aux probabilités de transition et d'observation. Un modèle de Markov observable pour ce problème est représenté dans la figure 12.2.

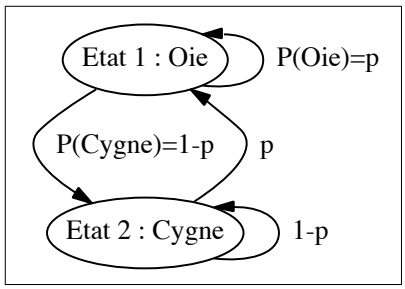

FIG. 12.2: *Le modèle de Markov observable qui modélise la suite des observations des oies et des cygnes.*

Il est composé de deux états ; chacun correspond directement à une observation possible : Oie (O) ou Cygne (C). Dans ce modèle, la suite d'états associée à une séquence observée est facile à déterminer : l'observation de O correspond à l'état 1 et l'observation de C correspond à l'état 2. Si la probabilité d'observer O à l'état 1 est $p = \mathbf{P}(Oie)$, alors la probabilité d'observer C à l'état 2 est $1 - p$. La probabilité d'observer la séquence $O(1:6) = \texttt{O O C O C O}$ se calcule facilement ; elle vaut :

$$p\ p\ (1-p)\ p\ (1-p)\ p = p^4\ (1-p)^2$$

Elle est par conséquent indépendante de l'ordre d'arrivée des oiseaux et ne tient compte que de leur nombre dans la séquence. Ce modèle n'exprime que les probabilités d'apparition *a priori* des observations.

La figure 12.3, accompagnée du tableau 12.1 définit un modèle de Markov *caché* (HMM) à deux états.

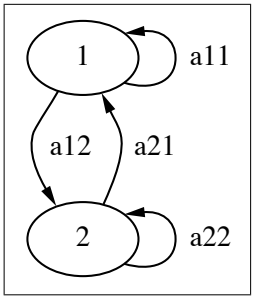

FIG. 12.3: *Le* HMM *à deux états.*

Etat	1	2
P(O)	p_1	p_2
P(C)	$1-p_1$	$1-p_2$

TAB. 12.1: *Les probabilités d'émission du* HMM *à deux états.*

Sans entrer encore dans les détails, on voit qu'un HMM est d'abord caractérisé par une probabilité a_{ij} de passer d'un état à un autre, ensuite qu'à chaque état est associée une probabilité de générer O ou C. À chaque instant, il y a, non pas un, mais deux tirages aléatoires : le premier pour tirer une lettre de l'alphabet des observations, le second pour changer d'état. L'observation d'une séquence de O et de C n'est donc plus directement liée à une suite unique d'états. Par exemple, comme on le voit sur la figure 12.3, la séquence O C C peut être engendrée avec une certaine probabilité (on verra plus loin comment on la calcule) par la suite d'états 1 2 2 ou la suite 2 2 2. Dans le modèle présenté, n'importe quelle suite d'états peut en réalité engendrer n'importe quelle suite d'observations avec une certaine probabilité.

Cette différence peut apparaître inutilement subtile. En réalité, elle est très importante. Précisons l'exemple pour mesurer la différence de puissance de modélisation entre un modèle de Markov observable et un HMM. Rappelons que la probabilité pour le modèle de Markov observable d'engendrer une séquence de longueur $2n$ comportant autant de O que de C est exactement $p^n(1-p)^n$, indépendamment de la répartition des O et des C dans cette séquence.

Dans le cas du HMM, en prenant a_{11}, p_1, a_{22} et p_2 proches de 1, il est intéressant de constater que la phrase O O C C aura une forte probabilité d'être émise, alors que la phrase C C O O aura une probabilité faible. Pourtant, ces deux phrases comportent le même nombre de O et de C. D'une manière générale, une phrase ayant plus de O dans sa première moitié aura une probabilité plus forte que sa symétrique d'être émise par ce HMM. Cet exemple peut convaincre que si le HMM est plus complexe que le modèle observable, il a en retour la possibilité de représenter des concepts plus élaborés. En l'occurence, avec deux états, il est capable de représenter qu'il y a une différence entre les instants d'arrivée des oies et ceux des cygnes[6]. On verra le développement de cet exemple au paragraphe 6.

Remarquons aussi ceci : bien que l'alphabet des séquences observables soit composé de deux lettres, le HMM n'a plus de raison d'avoir exactement deux états. La figure 12.4, associée au tableau 12.2, présente un HMM à trois états. Les remarques sur le HMM à deux états sont encore valables : n'importe quelle suite d'états de ce HMM peut engendrer n'importe quelle suite d'observations de O et C avec une certaine probabilité. Ajoutons la remarque suivante : puisqu'on n'associe pas dans un HMM un état à une observation, il est possible de définir des observations appartenant à un alphabet infini.

2.3 Notations

Un HMM est noté $\Lambda = (A, B, \pi)$ et se définit par :

- Ses états, en nombre n, qui composent l'ensemble $S = \{s_1, s_2, \ldots s_n\}$. L'état où se trouve le HMM à l'instant t est noté q_t ($q_t \in S$).

[6]Pour être complètement exact, un modèle observable pourrait aussi représenter une telle dépendance. Avec deux états, on peut en réalité représenter quatre probabilités différentes pour chaque séquence de deux observations (O O, O C), etc. et donc traduire une dépendance d'un événement avec l'événement précédent. En associant cette remarque à celle formulée précédemment en note de bas de page, on voit que le pouvoir d'expression des modèles observables peut être augmenté si on les sophistique... mais seulement sur des alphabets finis.

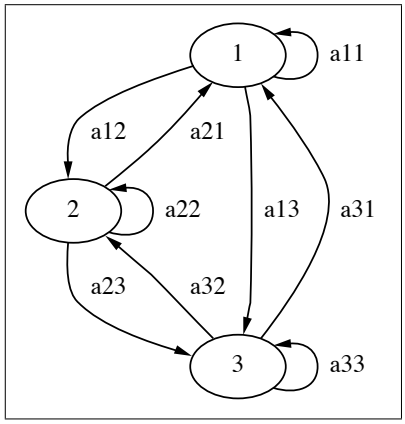

FIG. 12.4: *Le* HMM *à trois états.*

Etat	1	2	3
P(0)	p_1	p_2	p_3
P(C)	$1-p_1$	$1-p_2$	$1-p_3$

TAB. 12.2: *Les probabilités d'émission du* HMM *à trois états.*

- M symboles observables dans chaque état. L'ensemble des observations possibles (l'alphabet) est noté $V = \{v_1, v_2, \ldots, v_M\}$. $O_t \in V$ est le symbole observé à l'instant t.
- Une matrice A de *probabilités de transition* entre les états : a_{ij} représente la probabilité que le modèle évolue de l'état i vers l'état j :

$$a_{ij} = A(i,j) = \mathbf{P}(q_{t+1} = s_j \mid q_t = s_i)\ \forall i,j \in [1 \ldots n]\ \forall t \in [1 \ldots T]$$

avec :

$$a_{ij} \geq 0\ \ \forall i,j \qquad \text{et :} \qquad \sum_{j=1}^{n} a_{ij} = 1$$

- Une matrice B de *probabilités d'observation* des symboles dans chacun des états du modèle : $b_j(k)$ représente la probabilité que l'on observe le symbole v_k alors que le modèle se trouve dans l'état j, soit :

$$b_j(k) = \mathbf{P}(O_t = v_k \ \mid \ q_t = s_j)\ \ 1 \leq j \leq n,\ 1 \leq k \leq M$$

avec :

$$b_j(k) \geq 0\ \ \forall j,k \qquad \text{et :} \qquad \sum_{k=1}^{M} b_j(k) = 1$$

- Un vecteur π de *probabilités initiales* : $\pi = \{\pi_i\}_{i=1,2,\ldots,n}$. Pour tout état i, π_i est la probabilité que l'état de départ du HMM soit l'état i :

$$\pi_i = \mathbf{P}(q_1 = s_i) \quad 1 \leq i \leq n$$

avec :

$$\pi_i \geq 0\ \ \forall i \qquad \text{et :} \qquad \sum_{i=1}^{n} \pi_i = 1$$

- Un ou plusieurs *états finals*. Ici, nous supposons pour simplifier que le processus peut s'arrêter dans n'importe quel état, autrement dit que tout état est final.

2.4 Deux types de Hmm

En pratique, on utilise deux types de modèles de Markov cachés, le modèle *ergodique* et le modèle *gauche-droite*.

Le modèle ergodique est sans contrainte : toutes les transitions d'un état vers un autre sont possibles. Les exemples présentés précédemment sont de ce type.

Le modèle gauche-droite est un modèle contenant des contraintes résultant de la mise à zéro de certaines valeurs a_{ij}. Dans le modèle le plus utilisé, celui de la figure 12.5, l'état i n'est relié par une transition de probabilité non nulle qu'à trois états : lui-même, l'état $i+1$ et l'état $i+2$. D'où le nom de *modèle gauche-droite*[7].

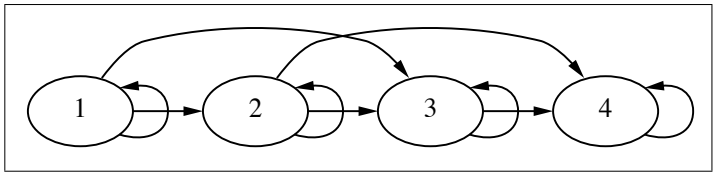

Fig. 12.5: *Le* Hmm *gauche-droite à quatre états.*

2.5 Comment un Hmm engendre-t'il une séquence ?

Un Hmm peut être vu comme un processus permettant d'engendrer une séquence ; inversement, on peut considérer une séquence comme une suite d'observations sur un Hmm en fonctionnement. En se plaçant du premier point de vue, la génération d'une séquence peut se décrire par l'algorithme 15 : c'est une procédure itérative gérée par des tirages aléatoires.

Algorithme 15 : Génération d'une séquence par un Hmm

début
  $t \leftarrow 1$
  Choisir l'état initial $q_1 = s_i$ avec la probabilité π_i
  tant que $t \leq T$ **faire**
    Choisir l'observation $o_t = v_k$ avec la probabilité $b_i(k)$
    Passer à l'état suivant $q_{t+1} = s_j$ avec la probabilité a_{ij}
    $t \leftarrow t+1$
  fin tant que
fin

Répétons ici qu'une séquence donnée peut en général être engendrée de plusieurs façons distinctes par un Hmm.

[7] Ou modèle de Bakis. Dans l'exemple d'introduction, le Hmm présenté est encore plus simple.

3. Les HMM comme règles de classification de séquences

3.1 Les trois problèmes des HMM

Les définitions précédentes ne sont utilisables que si l'on sait calculer la probabilité qu'une séquence soit engendrée par un HMM et surtout si l'on sait apprendre un HMM à partir d'exemples. On doit donc chercher des algorithmes pour résoudre les problèmes suivants.

- *L'évaluation de la probabilité de l'observation d'une séquence.* Étant donné la séquence d'observations O et un HMM $\Lambda = (A,\ B,\ \pi)$, comment évaluer la probabilité d'observation $\mathbf{P}(O \mid \Lambda)$? La réponse à cette question est importante : dans un problème de classification, on attribuera à une séquence la classe que modélise le HMM le plus probable étant donnée la séquence.
- *La recherche du chemin le plus probable.* Étant donnés la suite d'observations O et un HMM Λ, comment trouver une suite d'états $Q = q_1, q_2, \ldots q_T$ qui maximise la probabilité d'observation de la séquence ?
- *L'apprentissage.* Comment ajuster les paramètres $(A,\ B,\ \pi)$ d'un HMM Λ pour maximiser

$$\mathbf{P}(\mathcal{O} \mid \Lambda) = \prod_{O \in \mathcal{O}} \mathbf{P}(O \mid \Lambda)$$

  à partir d'un ensemble $\mathcal{O}$ de séquences d'apprentissage ?

Notons que la résolution du second problème n'est pas indispensable à l'utilisation des HMM en décision bayésienne. On reviendra sur son utilité au paragraphe 7.

3.2 Les HMM et la classification bayésienne

Le principe est d'apprendre un HMM par classe à partir des exemples de cette classe. L'apprentissage d'un HMM s'effectue à partir d'un modèle initial ; le HMM se modifie, mais en gardant jusqu'à sa convergence certaines caractéristiques du modèle initial (une certaine *architecture*) :

- le nombre d'états reste inchangé ;
- une transition de probabilité nulle entre deux états du modèle initial garde toujours une valeur nulle.

Le mieux est de prendre pour chaque classe un modèle initial ayant la même architecture : par exemple un modèle ergodique ou un modèle de Bakis. Pour chaque classe, le modèle initial peut simplement être pris avec le même nombre d'états[8].

Après C apprentissages indépendants, on dispose donc de C HMM, que l'on peut noter $\Lambda_1, \cdots, \Lambda_C$
Étant donnée une séquence quelconque O, on a pour la classe de rang k :

$$\mathbf{P}(\Lambda_k \mid O) = \frac{\mathbf{P}(O \mid \Lambda_k).\mathbf{P}(\Lambda_k)}{\mathbf{P}(O)}$$

Le modèle qui doit être choisi par la règle bayésienne est celui qui maximise $\mathbf{P}(\Lambda_k \mid O)$ (règle MAP : maximum *a posteriori*), ou si l'on suppose les classes équiprobables, celui qui maximise $\mathbf{P}(O \mid \Lambda_k)$ (maximum de vraisemblance), comme indiqué au chapitre 2.

On doit donc être capable de calculer cette dernière valeur pour tout i. Cela nécessite un algorithme capable d'évaluer la probabilité qu'une phrase soit émise par un HMM. C'est le sujet que nous allons développer au paragraphe suivant.

[8] Cette simplification n'est nullement nécessaire à l'application du principe MAP, mais elle est utilisée en l'absence de connaissances qui pourraient la mettre en cause.

4. L'évaluation de la probabilité d'observation

L'évaluation directe

Remarquons d'abord que la probabilité de la suite d'observations O, étant donné le modèle Λ, est égale à la somme sur toutes les suites d'états possibles Q des probabilités conjointes de O et de Q :

$$\mathbf{P}(O \mid \Lambda) = \sum_Q \mathbf{P}(O, Q \mid \Lambda) = \sum_Q \mathbf{P}(O \mid Q, \Lambda)\mathbf{P}(Q \mid \Lambda)$$

Or, on a les relations :

$$\mathbf{P}(Q \mid \Lambda) = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} \dots a_{q_{T-1} q_T}$$
$$\mathbf{P}(O \mid Q, \Lambda) = b_{q_1}(O_1) b_{q_2}(O_2) \dots b_{q_T}(O_T)$$

On déduit donc des formules précédentes, en réarrangeant les termes :

$$\mathbf{P}(O \mid \Lambda) = \sum_{Q=q_1,q_2,\dots,q_T} \pi_{q_1} b_{q_1}(O_1) a_{q_1 q_2} b_{q_2}(O_2) \dots a_{q_{T-1} q_T} b_{q_T}(O_T)$$

Cette formule directe nécessite d'énumérer toutes les suites d'états de longueur T, soit une complexité en $\Theta(n^T)$. Il existe heureusement une méthode plus rapide.

L'évaluation par les fonctions forward-backward.

Dans cette approche [Bau72], on remarque que l'observation peut se faire en deux temps : d'abord, l'émission du début de l'observation $O(1:t)$ en aboutissant à l'état q_i au temps t, puis, l'émission de la fin de l'observation $O(t+1:T)$ sachant que l'on part de q_i au temps t. Ceci posé, la probabilité de l'observation est donc égale à :

$$\mathbf{P}(O \mid \Lambda) = \sum_{i=1}^{n} \alpha_t(i)\beta_t(i)$$

où $\alpha_t(i)$ est la probabilité d'émettre le début $O(1:t)$ et d'aboutir à q_i à l'instant t, et $\beta_t(i)$ est la probabilité d'émettre la fin $O(t+1:T)$ sachant que l'on part de q_i à l'instant t. Le calcul de α s'effectue avec t croissant tandis que le calcul de β est réalisé avec t décroissant, d'où l'appellation *forward-backward*.

Le calcul de α

On a :

$$\alpha_t(i) = \mathbf{P}(O_1 O_2 \dots O_t,\ q_t = s_i \mid \Lambda)$$

$\alpha_t(i)$ se calcule par l'algorithme 16, qui exprime que pour émettre le début de l'observation $O(1:t+1)$ et aboutir dans l'état s_j au temps $t+1$, on doit nécessairement être dans l'un des états s_i à l'instant t. Cette remarque permet d'exprimer $\alpha_{t+1}(j)$ en fonction des $\alpha_t(i)$ et d'utiliser un algorithme de programmation dynamique pour le calcul de tous les $\alpha_t(i)$ pour tout i, puis des $\alpha_{t+1}(i)$ pour tout i, etc.

Ce calcul a une complexité en $\Theta(n^2 T)$.

Le calcul de β

De manière analogue, $\beta_t(i)$ se calcule par l'algorithme 17.

Algorithme 16 : **Calcul de la fonction *forward* α**

début
  pour $i = 1, n$ **faire** $\alpha_1(i) \leftarrow \pi_i b_i(O_1)$
  $t \leftarrow 1$
  tant que $t < T$ **faire**
    $j \leftarrow 1$
    tant que $j \leq n$ **faire**
      $\alpha_{t+1}(j) \leftarrow [\sum_{i=1}^{n} \alpha_t(i) a_{ij}] b_j(O_{t+1})$
      $j \leftarrow j + 1$
    fin tant que
    $t \leftarrow t + 1$
  fin tant que
  $\mathsf{P}(O \mid \Lambda) \leftarrow \sum_{i=1}^{n} \alpha_T(i)$
fin

Algorithme 17 : **Calcul de la fonction *backward* β**

début
  pour $i = 1, n$ **faire** $\beta_T(i) \leftarrow 1$
  $t \leftarrow T - 1$
  tant que $t \geq 1$ **faire**
    $i \leftarrow 1$
    tant que $i \leq n$ **faire**
      $\beta_t(i) \leftarrow \sum_{j=1}^{n} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)$
      $i \leftarrow i + 1$
    fin tant que
    $t \leftarrow t - 1$
  fin tant que
  $\mathsf{P}(O \mid \Lambda) \leftarrow \sum_{i=1}^{n} \beta_1(i)$
fin

Le calcul de β est lui aussi en $\Theta(n^2 T)$.

Le calcul de la probabilité d'observation

Finalement, la probabilité d'observation d'une séquence est obtenue en prenant les valeurs de α et de β à un instant t quelconque : $\mathsf{P}(O \mid \Lambda) = \sum_{i=1}^{n} \alpha_t(i)\beta_t(i)$. Cependant, on utilise le plus souvent les valeurs obtenues pour deux cas particuliers $(t = 0)$ ou $(t = T)$, ce qui donne :

$$\mathsf{P}(O \mid \Lambda) = \sum_{i=1}^{n} \alpha_T(i) = \sum_{i=1}^{n} \pi_i \beta_0(i)$$

— Exemple —

Soit le modèle $\Lambda = (A, B, \pi)$ (figure 12.6) comprenant trois états 1, 2, 3 chacun permettant d'observer un symbole de l'alphabet $V = \{a, b\}$.

$$A = \begin{pmatrix} 0.3 & 0.5 & 0.2 \\ 0 & 0.3 & 0.7 \\ 0 & 0 & 1 \end{pmatrix} \quad B = \begin{pmatrix} 1 & 0 \\ 0.5 & 0.5 \\ 0 & 1 \end{pmatrix} \quad \pi = \begin{pmatrix} 0.6 \\ 0.4 \\ 0 \end{pmatrix}$$

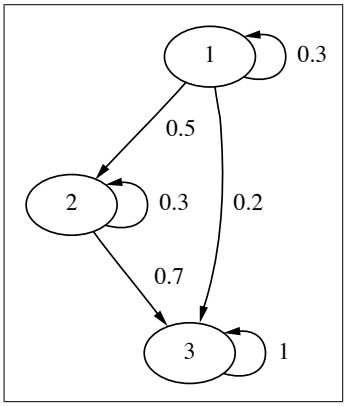

FIG. 12.6: *Un exemple de* HMM.

Etat	1	2	3
P(a)	1	0.5	0
P(b)	0	0.5	1

TAB. 12.3: *La matrice B de ce* HMM.

La figure 12.7 illustre le calcul de α pour la suite d'observations : *a a b b*.

$$
\begin{array}{ll}
\alpha_1(1) & = \pi_1 b_1(a) = 0.6 \times 1 = 0.6 \\
\alpha_1(2) & = \pi_2 b_2(a) = 0.4 \times 0.5 = 0.2 \\
\alpha_1(3) & = \pi_3 b_3(a) = 0 \times 0 = 0 \\
\alpha_2(1) & = (\alpha_1(1)a_{11} + \alpha_1(2)a_{21} + \alpha_1(3)a_{31})b_1(a) \\
 & = (0.6 \times 0.3 + 0.2 \times 0 + 0 \times 0) \times 1 \\
 & = (0.18) \times 1 = 0.18 \\
\alpha_2(2) & = (\alpha_1(1)a_{12} + \alpha_1(2)a_{22} + \alpha_1(3)a_{32})b_2(a) \\
 & = (0.6 \times 0.5 + 0.2 \times 0.3 + 0 \times 0) \times 0.5 \\
 & = (0.36) \times 0.5 = 0.18 \\
\ldots & \ldots \\
\mathbf{P}(a\ a\ b\ b\ \mid \Lambda) & = \sum_{q_i} \alpha_4(i) = 0.2228
\end{array}
$$

5. Le calcul du chemin optimal : l'algorithme de Viterbi

Il s'agit maintenant de déterminer le meilleur chemin correspondant à l'observation, c'est-à-dire de trouver dans le modèle Λ la *meilleure suite d'états* Q, qui maximise la quantité :

$$\mathbf{P}(Q, O \mid \Lambda)$$

Pour trouver $Q = (q_1, q_2, \ldots, q_T)$ pour une séquence d'observations $O = (O_1, O_2, \ldots, O_T)$, on définit la variable intermédiaire $\delta_t(i)$ comme la probabilité du meilleur chemin amenant à l'état s_i à l'instant t, en étant guidé par les t premières observations :

$$\delta_t(i) = \underset{q_1,\ldots,q_{t-1}}{Max}\ \mathbf{P}(q_1, q_2, \ldots, q_t = s_i, O_1, O_2, \ldots, O_t \mid \Lambda)$$

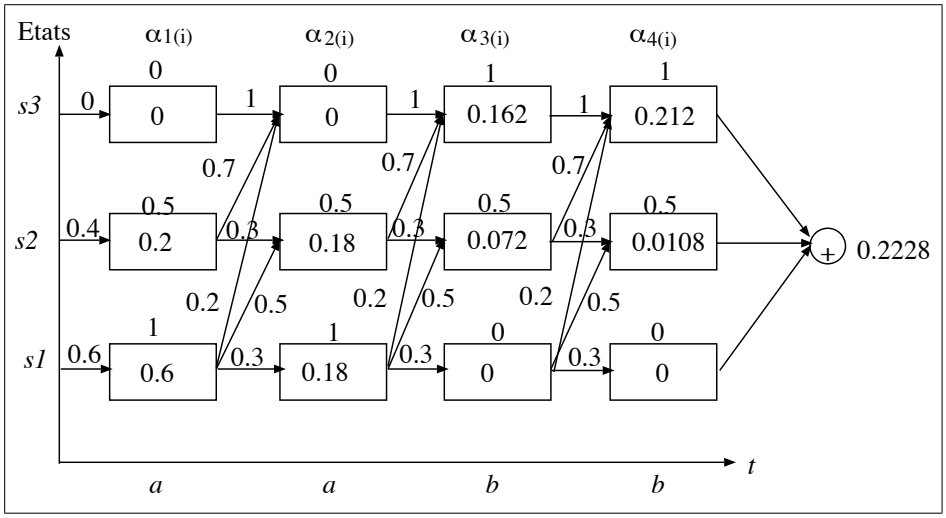

FIG. 12.7: *Calcul de α pour la suite d'observations aabb.*

Par récurrence, on calcule

$$\delta_{t+1}(j) = [\underset{i}{\text{Max}}\ \delta_t(i)\, a_{ij}]\ b_j(O_{t+1})$$

en gardant trace, lors du calcul, de la suite d'états qui donne le meilleur chemin amenant à l'état s_i à t dans un tableau ψ.

On utilise une variante de la programmation dynamique, l'*algorithme de Viterbi* (algorithme 18) pour formaliser cette récurrence. Il fournit en sortie la valeur $\mathbf{P}^*$ de la probabilité de l'émission de la séquence par la meilleure suite d'états $(q_1^*, \cdots, q_T^*)$.

La fonction *Argmax* permet de mémoriser l'indice i, entre 1 et n, avec lequel on atteint le maximum des quantités $(\delta_{t-1}(i)a_{ij})$. Le coût des opérations est également en $\Theta(n^2T)$.

—— EXEMPLE

D'après [BB92].

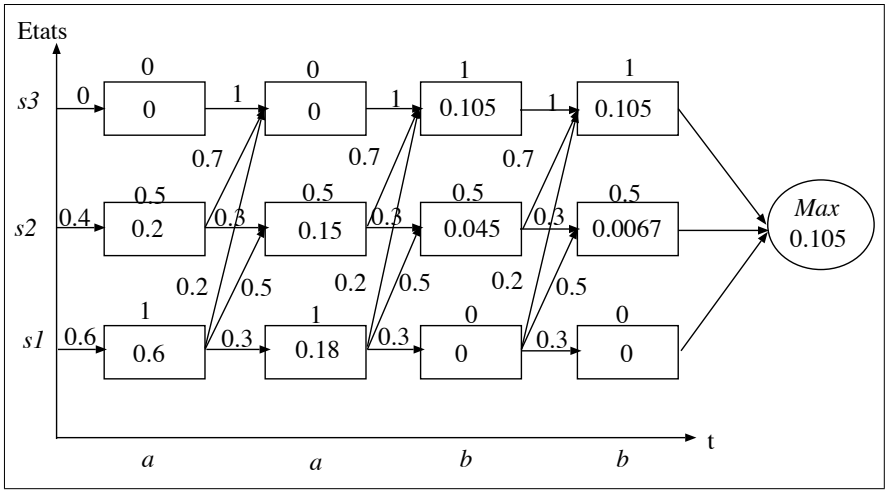

FIG. 12.8: *Calcul de δ pour la suite d'observations aabb.*

Algorithme 18 : Algorithme de Viterbi

début
  pour $i = 1, n$ **faire**
    $\delta_1(i) \leftarrow \pi_i b_i(O_1)$
    $\psi_1(i) \leftarrow 0$
  fin pour
  $t \leftarrow 2$
  tant que $t \leq T - 1$ **faire**
    $j \leftarrow 1$
    tant que $j \leq n$ **faire**
      $\delta_t(j) \leftarrow \text{Max}_{1 \leq i \leq n} \left[\delta_{t-1}(i)\, a_{ij}\right] b_j(O_t)$
      $\psi_t(j) \leftarrow \text{ArgMax}_{1 \leq i \leq n} \left[\delta_{t-1}(i)\, a_{ij}\right]$
      $j \leftarrow j + 1$
    fin tant que
    $t \leftarrow t + 1$
  fin tant que
  $\mathbf{P}^* \leftarrow \text{Max}_{1 \leq i \leq n} \left[\delta_T(i)\right]$
  $q_T^* \leftarrow \text{ArgMax}_{1 \leq i \leq n} \left[\delta_T(i)\right]$
  $t \leftarrow T$
  tant que $t \geq 1$ **faire**
    $q_t^* \leftarrow \psi_{t+1}(q_{t+1}^*)$
    $t \leftarrow t - 1$
  fin tant que
fin

À partir de la figure 12.8 qui illustre le calcul de δ, on peut calculer les quantités δ, ψ et q^*comme suit :

$$
\begin{array}{lll}
\delta_1(1) & = \pi_1 b_1(a) = 0.6 \times 1 = 0.6 & \psi_1(1) = 0, \\
\delta_1(2) & = \pi_2 b_2(a) = 0.4 \times 0.5 = 0.2 & \psi_1(2) = 0, \\
\delta_1(3) & = \pi_3 b_3(a) = 0 \times 0 = 0 & \psi_1(3) = 0, \\
\delta_2(1) & = max_{1 \leq i \leq n}(\delta_1(i) a_{i1}) b_1(a) & \\
 & = max \left\{ \begin{array}{l} \delta_1(1) a_{11} \\ \delta_1(2) a_{21} \\ \delta_1(3) a_{31} \end{array} \right\} \times b_1(a) & \\
 & = max \left\{ \begin{array}{l} \underline{0.6 \times 0.3} \\ 0 \\ 0 \end{array} \right\} \times 1 = 0.18 & \psi_2(1) = 1, \\
\ldots & &
\end{array}
$$

Finalement :

$$
\begin{array}{llll}
\underline{\psi_1(1) = 0} & \psi_2(1) = 1 & \psi_3(1) = 1 & \psi_4(1) = 1 \\
\underline{\psi_1(2) = 0} & \underline{\psi_2(2) = 1} & \psi_3(2) = 1 & \psi_4(2) = 2 \\
\psi_1(3) = 0 & \underline{\psi_2(3) = 2} & \underline{\psi_3(3) = 2} & \underline{\psi_4(3) = 3},
\end{array}
$$

$$q_4^* = max \begin{pmatrix} \delta_4(1) \\ \delta_4(2) \\ \delta_4(3) \end{pmatrix} = 3,$$

$$\begin{aligned} q_3^* &= \psi_4(3) = 3, \\ q_2^* &= \psi_3(3) = 2, \\ q_1^* &= \psi_2(2) = 1. \end{aligned}$$

On déduit donc de ce calcul que la *meilleure suite d'états*, celle qui engendre la phrase *a a b b* avec la plus forte probabilité, est : *1 2 3 3*.

---

6. L'apprentissage

Principe

Supposons disposer d'un ensemble de séquences $\mathcal{O} = \{O^1, \cdots, O^m\}$, dont l'élément courant est noté O^k. Le but de l'apprentissage est de déterminer les paramètres d'un HMM d'architecture fixée $\Lambda = (A, B, \pi)$, qui maximisent la probabilité $\mathbf{P}(\mathcal{O} \mid \Lambda)$. Comme on suppose les séquences d'apprentissages tirées indépendamment, on cherche donc à maximiser :

$$\mathbf{P}(\mathcal{O} \mid \Lambda) = \prod_{k=1}^{m} \mathbf{P}(O^k \mid \Lambda)$$

L'idée est d'utiliser une procédure de réestimation qui affine le modèle petit à petit selon les étapes suivantes :

- choisir un ensemble initial Λ_0 de paramètres ;
- calculer Λ_1 à partir de Λ_0, puis Λ_2 à partir de Λ_1, etc.
- répéter ce processus jusqu'à un critère de fin.

Pour chaque étape p d'apprentissage, on dispose de Λ_p et on cherche un Λ_{p+1} qui doit vérifier :

$$\mathbf{P}(\mathcal{O} \mid \Lambda_{p+1}) \geq \mathbf{P}(\mathcal{O} \mid \Lambda_p)$$

soit :

$$\prod_{k=1}^{m} \mathbf{P}(O^k \mid \Lambda_{p+1}) \;\geq\; \prod_{k=1}^{m} \mathbf{P}(O^k \mid \Lambda_p)$$

Λ_{p+1} doit donc améliorer la probabilité de l'émission des observations de l'ensemble d'apprentissage. La technique pour calculer Λ_{p+1} à partir de Λ_p consiste à utiliser l'algorithme EM. Pour cela, on effectue un comptage de l'utilisation des transitions A et des distributions B et π du modèle Λ_p quand il produit l'ensemble $\mathcal{O}$. Si cet ensemble est assez important, ces fréquences fournissent de bonnes approximations *a posteriori* des distributions de probabilités A,B et π et sont utilisables alors comme paramètres du modèle Λ_{p+1} pour l'itération suivante.

La méthode d'apprentissage EM consiste donc dans ce cas à regarder comment se comporte le modèle défini par Λ_p sur $\mathcal{O}$, à réestimer ses paramètres à partir des mesures prises sur $\mathcal{O}$, puis à recommencer cette réestimation jusqu'à obtenir une convergence. L'annexe 7 donne quelques détails sur cette méthode.

Dans les calculs qui suivent, on verra apparaître en indice supérieur la lettre k quand il faudra faire référence à la séquence d'apprentissage concernée. L'indice p, qui compte les passes d'apprentissage, sera omis : on partira d'un modèle noté simplement Λ et on calculera celui qui s'en déduit.

Les formules de réestimation

On définit $\xi_t^k(i,j)$ comme la probabilité, étant donnés une phrase O^k et un HMM Λ, que ce soit l'état s_i qui ait émis la lettre de rang t de O^k et l'état s_j qui ait émis celle de rang $t+1$. Donc :

$$\xi_t^k(i,j) = \mathbf{P}(q_t = s_i,\ q_{t+1} = s_j \ \mid O^k, \Lambda)$$

Ce qui se récrit :

$$\xi_t^k(i,j) = \frac{\mathbf{P}(q_t = s_i,\ q_{t+1} = s_j,\ O^k \mid \Lambda)}{\mathbf{P}(O^k \mid \Lambda)}$$

Par définition des fonctions *forward-backward*, on en déduit :

$$\xi_t^k(i,j) = \frac{\alpha_t^k(i) a_{ij} b_j(O_{t+1}^k) \beta_{t+1}^k(j)}{\mathbf{P}(O^k \mid \Lambda)}$$

On définit aussi la quantité $\gamma_t^k(i)$ comme la probabilité que la lettre de rang t de la phrase O^k soit émise par l'état s_j.

$$\gamma_t^k(i) = \mathbf{P}(q_t = s_i \mid O^k, \Lambda)$$

Soit :

$$\gamma_t^k(i) = \sum_{j=1}^{n} \mathbf{P}(q_t = s_i,\ q_{t+1} = s_j \mid O^k, \Lambda) = \frac{\sum_{j=1}^{n} \mathbf{P}(q_t = s_i,\ q_{t+1} = s_j, O^k \mid \Lambda)}{\mathbf{P}(O^k \mid \Lambda)}$$

On a la relation :

$$\gamma_t^k(i) = \sum_{j=1}^{n} \xi_t(i,j) \frac{\alpha_t^k(i) \beta_t^k(i)}{\mathbf{P}(O^k \mid \Lambda)}$$

Le nouveau modèle HMM se calcule à partir de l'ancien en réestimant π, A et B par comptage sur la base d'apprentissage. On mesure les fréquences :

$$\overline{a}_{ij} = \frac{\text{nombre de fois où la transition de } s_i \text{ à } s_j \text{ a été utilisée}}{\text{nombre de transitions effectuées à partir de } s_i}$$

$$\overline{b}_j(l) = \frac{\text{nombre de fois où le HMM s'est trouvé dans l'état } s_j \text{ en observant } v_l}{\text{nombre de fois où le HMM s'est trouvé dans l'état } s_j}$$

$$\overline{\pi}_i = \frac{\text{nombre de fois où le HMM s'est trouvé dans l'état } s_i \ \ldots}{\text{nombre de fois où le HMM } \ldots}$$

$$\frac{\ldots \text{ en émettant le premier symbole d'une phrase}}{\ldots \text{ a émis le premier symbole d'une phrase}}$$

Compte tenu de ces définitions :

$$\overline{\pi}_i = \frac{1}{m} \sum_{k=1}^{m} \gamma_1^k(i)$$

$$\overline{a}_{ij} = \frac{\sum_{k=1}^{m} \sum_{t=1}^{|O^k|-1} \xi_t^k(i,j)}{\sum_{k=1}^{N} \sum_{t=1}^{|O^k|-1} \gamma_t^k(i)}$$

$$\overline{b}_j(l) = \frac{\sum_{k=1}^{m} \sum_{\substack{t=1 \\ \text{avec } O_t^k = v_l}}^{|O^k|-1} \gamma_t^k(j)}{\sum_{k=1}^{m} \sum_{t=1}^{|O^k|-1} \gamma_t^k(j)}$$

Ces formules ont été établies par Baum [Bau72], comme une application de la procédure EM à l'apprentissage des paramètres HMM. La suite des modèles construits par l'algorithme de Baum-Welsh [RJ93] vérifie la relation cherchée :

$$\mathbf{P}(\mathcal{O} \mid \Lambda_{p+1}) \geq \mathbf{P}(\mathcal{O} \mid \Lambda_p)$$

Algorithme 19 : Algorithme de Baum-Welch

début
  Fixer des valeurs initiales (A, B, π)
  On définit le HMM de départ comme $\Lambda_0 = (A, B, \pi)$.
  $p \leftarrow 0$
  tant que *la convergence n'est pas réalisée* **faire**
    On possède le HMM Λ_p
    On calcule pour ce modèle, sur l'ensemble d'apprentissage, les valeurs :
      $\xi(i,j), \quad \gamma_t(i) \quad 1 \leq i,j \leq n \qquad 1 \leq t \leq T-1$
    On en déduit $\overline{\pi}$, $\overline{A}$, $\overline{B}$ en utilisant les formules de réestimation.
    Le HMM courant est désormais défini par $\Lambda_{p+1} = (\overline{\pi}, \overline{A}, \overline{B})$
    $p \leftarrow p + 1$
  fin tant que
fin

Remarques

- Le choix du modèle initial influe sur les résultats ; par exemple, si certaines valeurs de A et B sont égales à 0 au départ, elles le resteront jusqu'à la fin de l'apprentissage. Ceci permet en particulier de garder la structure dans les modèles gauches-droits.
- L'algorithme converge vers des valeurs de paramètres qui assurent un *maximum local* de $\mathbf{P}(O \mid \Lambda)$. Il est donc important, si l'on veut être aussi près que possible du minimum global, de bien choisir la structure et l'initialisation.
- Le nombre d'itérations est fixé empiriquement. L'expérience prouve que, si le point précédent a été correctement traité, la stabilisation des paramètres ne correspond pas à un surapprentissage : il n'y a donc en général pas besoin de contrôler la convergence par un ensemble de validation. Mais cette possibilité est évidemment toujours à disposition.

— Exemple —

En partant du Hmm Λ_0 défini par les paramètres suivants :

$$A = \begin{pmatrix} 0.45 & 0.35 & 0.20 \\ 0.10 & 0.50 & 0.40 \\ 0.15 & 0.25 & 0.60 \end{pmatrix} \quad B = \begin{pmatrix} 1.0 & 0.0 \\ 0.5 & 0.5 \\ 0.0 & 1.0 \end{pmatrix} \quad \pi = \begin{pmatrix} 0.5 \\ 0.3 \\ 0.2 \end{pmatrix}$$

on peut calculer que, s'il émet sur l'alphabet à deux lettres $V = \{a, b\}$, on a :

$$\mathbf{P}(a\ b\ b\ a\ a \mid \Lambda_0) = 0.0278$$

Si on prend comme ensemble d'apprentissage cette seule phrase, l'application de l'algorithme de Baum-Welsh doit augmenter sa probabilité de reconnaissance.
Après une réestimation[9], on trouve le Hmm Λ_1 :

$$A = \begin{pmatrix} 0.346 & 0.365 & 0.289 \\ 0.159 & 0.514 & 0.327 \\ 0.377 & 0.259 & 0.364 \end{pmatrix} \quad B = \begin{pmatrix} 1.0 & 0.0 \\ 0.631 & 0.369 \\ 0.0 & 1.0 \end{pmatrix} \quad \pi = \begin{pmatrix} 0.656 \\ 0.344 \\ 0.0 \end{pmatrix}$$

$$\mathbf{P}(a\ b\ b\ a\ a \mid \Lambda_1) = 0.0529$$

Après quinze itérations :

$$A = \begin{pmatrix} 0.0 & 0.0 & 1.0 \\ 0.212 & 0.788 & 0.0 \\ 0.0 & 0.515 & 0.485 \end{pmatrix} \quad B = \begin{pmatrix} 1.0 & 0.0 \\ 0.969 & 0.031 \\ 0.0 & 1.0 \end{pmatrix} \quad \pi = \begin{pmatrix} 1.0 \\ 0.0 \\ 0.0 \end{pmatrix}$$

$$\mathbf{P}(a\ b\ b\ a\ a \mid \Lambda_{15}) = 0.2474$$

Après cent cinquante itérations, la convergence est réalisée. La figure 12.9 et le tableau 12.4 décrivent le résultat, que l'on peut donner aussi sous la forme suivante :

$$A = \begin{pmatrix} 0.0 & 0.0 & 1.0 \\ 0.18 & 0.82 & 0.0 \\ 0.0 & 0.5 & 0.5 \end{pmatrix} \quad B = \begin{pmatrix} 1.0 & 0.0 \\ 1.0 & 0.0 \\ 0.0 & 1.0 \end{pmatrix} \quad \pi = \begin{pmatrix} 1.0 \\ 0.0 \\ 0.0 \end{pmatrix}$$

$$\mathbf{P}(a\ b\ b\ a\ a \mid \Lambda_{150}) = 0.2500$$

Etat	1	2	3
P(a)	1	1	0
P(b)	0	0	1

Tab. 12.4: *La matrice B de ce* Hmm.

Il peut paraître curieux que ce Hmm ne génère pas son unique phrase d'apprentissage avec la probabilité 1 et toutes les autres séquences avec la probabilité 0. Ceci vient du fait que le nombre des états est trop petit pour réaliser un apprentissage par cœur. Mais si l'on part d'un Hmm initial à cinq états, il converge en revanche vers un Hmm Λ défini par :

[9] Les calculs sont très pénibles à faire à la main, même sur un exemple simple comme celui-ci.

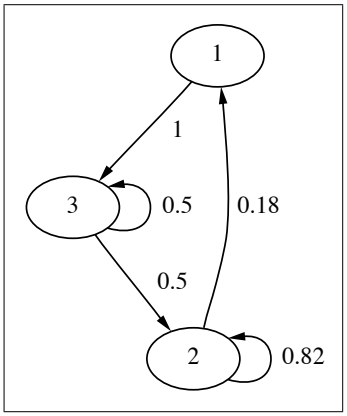

FIG. 12.9: *Le* HMM *entraîné sur une seule phrase, après convergence. Le seul état initial possible est l'état 1.*

$$A = \begin{pmatrix} 0.0 & 1.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 1.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 1.0 \end{pmatrix} \quad B = \begin{pmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \\ 0.0 & 1.0 \\ 1.0 & 0.0 \\ 1.0 & 0.0 \end{pmatrix} \quad \pi = \begin{pmatrix} 1.0 \\ 0.0 \\ 0.0 \\ 0.0 \\ 0.0 \end{pmatrix}$$

et l'on a :

$$\mathbf{P}(a\ b\ b\ a\ a \mid \Lambda) = 1.0$$

Ce HMM est en réalité une variante d'un modèle observable : à chaque état est associée l'émission certaine d'une lettre. La différence avec les modèles présentés en début de ce chapitre est que la même lettre peut être associée à plusieurs états. Ici, par exemple, la lettre a aux états 1, 4 et 5.

EXEMPLE

Reprenons maintenant l'exemple du paragraphe 2.2. Nous partons du HMM suivant, dont les paramètres ont été tirés aléatoirement :

$$A = \begin{pmatrix} 0.40 & 0.60 \\ 0.52 & 0.48 \end{pmatrix} \quad B = \begin{pmatrix} 0.49 & 0.51 \\ 0.40 & .60 \end{pmatrix} \quad \pi = \begin{pmatrix} 0.31 \\ 0.69 \end{pmatrix}$$

Nous lui faisons subir deux apprentissages sur deux ensembles de séquences : le premier est composé d'éléments ayant à peu près autant de a que de b, ces derniers étant situés en majorité dans la seconde partie ; l'autre ensemble d'apprentissage est composé de phrases de type symétrique.

$$\mathcal{O}_1 = \{aaabb, abaabbb, aaababb, aabab, ab\}$$

$$\mathcal{O}_2 = \{bbbaa, babbaa, bbbabaa, bbabba, bbaa\}$$

Après convergence, on obtient deux HMM différents donnés sur la figure 12.10.
Λ_1 est défini par :

$$A = \begin{pmatrix} 1.0 & 0.0 \\ 0.69 & 0.31 \end{pmatrix} \quad B = \begin{pmatrix} 0.36 & 0.64 \\ 1.0 & 0.0 \end{pmatrix} \quad \pi = \begin{pmatrix} 0.0 \\ 1.0 \end{pmatrix}$$

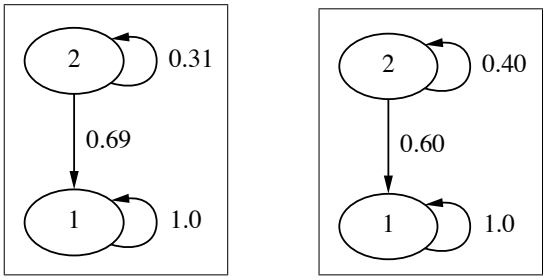

FIG. 12.10: *Les* HMM Λ_1 *et* Λ_2.

Son unique état de départ est l'état 2 qui émet a avec la probabilité 1. La probabilité d'émettre b par l'état 1 est de 0.64.
Λ_2 est défini par :

$$A = \begin{pmatrix} 1.0 & 0.0 \\ 0.60 & 0.40 \end{pmatrix} \quad B = \begin{pmatrix} 0.65 & 0.34 \\ 0.0 & 1.0 \end{pmatrix} \quad \pi = \begin{pmatrix} 0.0 \\ 1.0 \end{pmatrix}$$

Son unique état de départ est l'état 2 qui émet b avec la probabilité 1. La probabilité d'émettre a par l'état 1 est de 0.65.
Les deux HMM appris sont dont assez semblables, quand on permute a et b. On remarque qu'ils ne font pas jouer un rôle symétrique à leurs états.
Sur deux phrases n'ayant pas participé à l'apprentissage, on obtient le résultat attendu :

$$\mathbf{P}(a\ a\ b\ a\ b\ b\ b \mid \Lambda_1) = 0.0437$$

$$\mathbf{P}(a\ a\ b\ a\ b\ b\ b \mid \Lambda_2) = 0.000$$

$$\mathbf{P}(b\ b\ a\ b\ a\ a\ a \mid \Lambda_1) = 0.000$$

$$\mathbf{P}(b\ b\ a\ b\ a\ a\ a \mid \Lambda_2) = 0.0434$$

7. Approfondissements

Comme on l'a dit plus haut, il est possible de définir des HMM produisant des séquences de valeurs continues et même des séquences de vecteurs de valeurs continues. Dans ce cas, elles ne sont évidemment plus construites sur un alphabet fini. Il faut alors remplacer la matrice B par un ensemble de distributions éventuellement multidimensionnelles de probabilités ; les calculs précédents restent valables, mais les formules, en particulier celles de la réestimation pour l'apprentissage doivent porter sur des paramètres caractéristiques des distributions de probabilité en question. Généralement, on suppose celles-ci gaussiennes, ou multigaussiennes.

C'est souvent le cas en reconnaissance de la parole : prenons l'exemple de la reconnaissance du vocabulaire des dix chiffres. L'échantillon d'apprentissage permet de créer dix ensembles d'exemples, chacun supervisé par un chiffre différent. Pour chaque chiffre, on va apprendre un HMM. En phase de reconnaissance, un son inconnu sera classé comme le chiffre associé au HMM qui peut l'émettre avec la plus forte probabilité. Qu'est-ce qu'une séquence représentant la prononciation d'un son ? Au départ, un signal échantillonné, soit huit ou seize mille valeurs réelles

par seconde. Ce signal est transformé par des techniques de type transformation de Fourier pour en extraire ses caractéristiques fréquencielles. Au final, on dispose en général d'un codage de chaque centième de seconde de parole émise par un vecteur d'une dizaine de valeurs réelles. La prononciation d'un chiffre est donc une séquence dont la longueur est de l'ordre de quelques dizaines et dont chaque élément est un vecteur de $\mathbb{R}^{10}$.

Il est dès lors impossible de définir la matrice B puisque dans chaque terme $b_j(k)$, k devrait parcourir tous les vecteurs différents représentant un centième de seconde de parole et donc prendre un nombre infini de valeurs. Le plus simple est d'éffectuer une estimation paramétrique de la distribution des composantes du vecteur émis par chaque état, comme au chapitre 15. On supposera par exemple que ce sont des tirages aléatoires d'une gaussienne de dimension 10 : dans ce cas, pour chaque état, il faudra estimer la moyenne et la covariance de la densité de probabilité d'émission d'un vecteur de dimension 10. L'algorithme de Baum-Welsh s'adapte facilement à cette situation.

Des approfondissements et des alternatives ont été proposés : il est commun de supposer que les vecteurs émis sont des *sommes pondérées* de plusieurs distributions gaussiennes multidimensionnelles. L'algorithme EM permet encore de calculer les moyennes et les covariances de chacune, ainsi que les coefficients de pondération (voir le chapitre 18). Il a été aussi proposé d'estimer les densités d'émission par la méthode non paramétrique des k plus proches voisins qui est présenté au chapitre 15 ou par des réseaux connexionnistes (chapitre 10).

8. Applications

Les Hmm sont actuellement les outils d'apprentissage les plus efficaces pour la classification des séquences : ils ne réclament que peu de connaissances *a priori* ; à condition de disposer de suffisamment de données d'apprentissage, ils sont très efficaces. Un grand nombre de raffinements leur ont été apportés, en particulier pour résoudre des problèmes aussi complexes que celui de la reconnaissance de la parole ou de l'écriture manuscrite. Ces outils sont également très employés dans les séquences biologiques, en particulier pour la prédiction des structures secondaires et tri-dimensionnelles des protéines. On les utilise aussi en fouille de données, pour la recherche approchée de séquences dans des textes ou dans des masses de données de bio-séquences.

À l'heure actuelle, par exemple, presque tous les systèmes de reconnaissance de la parole sont construits à base de Hmm, parfois hybridés de réseaux connexionnistes. Dans la plupart des cas, les informations phonétiques, lexicales et syntaxiques sont « compilées » dans un Hmm de plusieurs centaines d'états, dont chacun vise à possèder une signification linguistique ; l'apprentissage se fait sur des très grandes quantités de données. La reconnaissance est effectuée en récupérant la séquence d'états par l'algorithme de Viterbi. En effet, les états ont dans ce cas une signification linguistique. Ce n'est pas tant la probabilité finale qui est intéressante que le chemin par lequel passe la meilleure façon d'engendrer le signal.

Notes historiques et sources bibliographiques

L'algorithme de Baum-Welsh [Bau72] est une application aux modèles de Markov de la technique générale EM d'estimation statistique de paramètres cachés. L'annexe 7 qui traite de cette

technique est en partie reprise du document [Ros97]. L'utilisation en apprentissage vient de la communauté de la reconnaissance de la parole. Les premières références que l'on peut relier à cette technique datent des années 1970 [Jel76, Bak75]. L'utilisation en reconnaissance bayésienne pour des séquences, comme présenté dans ce chapitre, est surtout fondée sur les travaux de Rabiner et de son équipe, qui ont démarré vers 1983 [LRS83]. L'article [Rab89] est un exposé incontournable sur les HMM et leur application à la parole. Les ouvrages [Jel97] et surtout [RJ93, JM00] et donnent un panorama de l'utilisation des HMM en reconnaissance de la parole. Pour leurs applications au traitement des séquences biologiques et des images, des références intéressantes sont [DEKM98, BB01] et [BS97, MS01]. L'exemple 12.6 et les figures associées sont empruntés, avec l'aimable accord des auteurs, à [BB92]. Ce dernier ouvrage (entre autres mérites) présente une bonne introduction à ces techniques d'apprentissage et à leur application pour la reconnaissance de l'écriture manuscrite ; ce chapitre lui a fait quelques autres emprunts.

Résumé

- Les HMM sont des modèles probabilistes d'émission de séquences, discrètes ou continues (et dans ce cas, éventuellement vectorielles).
- En version de base, ils sont utilisés en classification bayésienne.
- L'algorithme *forward-backward* permet de connaître la probabilité qu'un HMM ait émis une séquence.
- L'algorithme de *Viterbi* permet de connaître la suite des états du HMM qui a la plus forte probabilité d'avoir émis une séquence.
- L'algorithme de *Baum-Welsh* permet d'ajuster les paramètres d'un HMM au maximum de vraisemblance à partir d'un ensemble de séquences d'apprentissage.