APPRENTISSAGE A L'AIDE DES MODÈLES DE MARKOV CACHÉS ET APPLICATIONS

Sylvain ILOGA[1,2,3]
[1]Université de Maroua, ENS, département d'informatique, Cameroun
[2]IRD UMI 209, Laboratoire UMMISCO, Cameroun
[3]UMR 8051, Laboratoire ETIS, CNRS, ENSEA, France

École de Mathématiques Africaine (EMA), Juillet 2019

Table des matières

Chapitre 1

Généralités sur les MMCs

1.1 Notion d'observation

Le concept de base sur lequel reposent les MMCs est le concept d'observation. Afin de le définir, nous devons présenter les deux éléments atomiques qui caractérisent une observation, à savoir :

1. Les symboles : Un symbole est simplement quelque chose ou quelqu'un que l'on peut '*observer*', c'est-à-dire que l'on peut voir, identifier, toucher, sentir, écouter, etc. Il peut par exemple s'agir d'objets (table, écran, couleur, etc.), d'activités (dormir, manger, courrir, danser, etc.), d'êtres vivants (animal, plante, personne, etc), de fonctions (enseignant, chauffeur, gardien, footbaleur, etc) ou même de concepts abstraits (genre musical, odeur, saveur, etc.)
2. Les états : Un symbole ne peut être observé qu'à partir d'un point d'observation précis. Un état est considéré comme un point d'observation à partir duquel des symboles peuvent être observés. Il peut par exemple s'agir de lieux (hôpital, école, maison, etc), de moments de la journée (matin, midi, soir, etc.), de saisons climatiques (hiver, été, automne, etc), d'états physiologiques (malade, fatigué, En forme, etc).

Étant donnés un ensemble fini S de symboles et un ensemble fini E d'états, nous pouvons maintenant énoncer la définition formelle d'une observation ainsi qu'il suit :

Définition 1.1. *Une observation* $\begin{smallmatrix} s \\ \uparrow \\ e \end{smallmatrix}$ *matérialise le fait que le symbole* $s \in S$ *est observé dans l'état* $e \in E$.

En fonction de vos connaissances ou de votre ressentit, certaines observations vous sembleront plus probables que d'autres.

Exemple 1.1. *Dans cet exemple, nous considérons que les états sont des états physiologiques et que les symboles sont des activités. Si nous posons* $E_1 = \{Fatigue, EnForme\}$ *et que nous posons également* $S_1 = \{Dormir, manger, Visionner, Jouer, Travailler, Danser\}$, *alors quelles probabilités accordez-vous aux observations suivantes ? :*

Observ.	$Dormir \uparrow Fatigue$	$Manger \uparrow Fatigue$	$Visionner \uparrow Fatigue$	$Jouer \uparrow EnForme$	$Travailler \uparrow EnForme$	$Danser \uparrow EnForme$
Proba (%)						

Que pouvez-vous dire de ces observations de manière générale ?

Accorderiez-vous des probabilités similaires aux observations ci-dessous ? Pourquoi ?

Quelles probabilités leurs accorderiez-vous donc ?

Observ.	$Jouer \uparrow Fatigue$	$Travailler \uparrow Fatigue$	$Danser \uparrow Fatigue$	$Dormir \uparrow EnForme$	$Manger \uparrow EnForme$	$Visionner \uparrow EnForme$
Proba (%)						

Exemple 1.2. *Dans ce deuxième exemple, nous considérons plutôt que les états sont des lieux et que les symboles restent des activités. Si nous posons* $E_2 = \{Domicile, Eglise, Bar, Ecole\}$ *et* $S_2 = \{Boire, Prier, Etudier, Danser, Dormir\}$*, alors quelles probabilités accordez-vous aux observations suivantes ?*

Que pouvez-vous dire de la probabilité rattachée à toute observation réalisée à partir de l'état Domicile ? Pourquoi ?

Quelle remarque faites-vous au sujet des observations suivantes ? Quelles probabilités leurs accorderiez-vous ?

Observ.	$Prier \uparrow Eglise$	$Etudier \uparrow Ecole$	$Boire \uparrow Bar$
Proba (%)			

Accorderiez-vous des probabilités similaires aux observations ci-dessous ? Pourquoi ?

Quelles probabilités leurs accorderiez-vous donc ?

Observ.	$Etudier \uparrow Bar$	$Prier \uparrow Bar$	$Danser \uparrow Ecole$	$Dormir \uparrow Ecole$
Proba (%)				

1.2 Notion de transition d'états

Afin d'avoir une vue d'ensemble d'un évènement, un caméraman est très souvent obligé de se déplacer (de transiter) d'une position à une autre pour avoir des prises de vues différentes. Cette allégorie nous permet intuitivement d'énoncer la définition d'une transition d'états.

Définition 1.2. *Une transition d'états $e_i \to e_j$ matérialise un changement d'état de l'état $e_i \in E$ vers l'état $e_j \in E$.*

Comme cela était précédemment le cas pour les observations, certaines transitions d'états vous sembleront plus probables que d'autres en fonction de vos connaissances ou de votre ressentit.

Exemple 1.3. *Dans les conditions de l'exemple 1.2,*

Que pouvez-vous dire de la probabilité rattachée à toute transition partant de l'état Domicile ? Pourquoi ?

Que pensez-vous de la transition Eglise → Bar ? Pourquoi ? Quelle probabilité lui attriburiez-vous ?

Cette transition est-elle d'après vous équivalente à la transition Bar → Eglise ? Pourquoi ? Quelle probabilité lui attribueriez-vous ?

Mêmes questions pour les transitions Bar → Ecole et Bar → Ecole

---

Quelle propriété relative aux transitions d'états pouvez-vous en déduire ?

Quelle probabilité attribueriez-vous à la transition récursive Bar → Bar ? En d'autres termes, quelle est d'après vous la probabilité qu'un individu soit dans un bar à l'instant $t+1$, sachant qu'il y était déjà à l'instant t ? Pourquoi ?

Mêmes questions pour les transitions Ecole → Ecole et Eglise → Eglise.

1.3 Notion de chaîne de Markov

Maintenant que les concepts de symbole, d'état, d'observation et de transition d'états sont bien assimilés, nous pouvons donner une définition formelle de ce qu'est une chaîne de Markov.

Définition 1.3. *Une chaîne de Markov de longueur T est une suite ordonnées de T observations consécutives déclenchant de manière implicite des transitions d'états lorsque l'on passe de la i^e à la j^e observation.*

Sur la base de cette définition, il est possible de visualiser une chaîne de Markov. En effet, soit $E = \{e_1, \dots, e_N\}$ un ensemble d'états et $S = \{s_1, \dots, s_M\}$ un ensemble de symboles. Une chaîne de Markov δ de longueur $|\delta| = T$ est donnée par l'équation 1.1 où les $e_{i_j} \in E$ et les $s_{i_j} \in S$

$$\begin{array}{cccccccc} \text{(Symboles)} & s_{i_1} & & s_{i_2} & & \dots & & s_{i_T} \\ & \uparrow & & \uparrow & & \vdots & & \uparrow \\ \text{(Etats)} & e_{i_1} & \rightarrow & e_{i_2} & \rightarrow & \dots & \rightarrow & e_{i_T} \end{array} \tag{1.1}$$

Dans l'équation 1.1, l'état e_{i_1} est appelé état initial. La suite de symboles $s_{i_1} s_{i_2} \dots s_{i_T}$ est appelée séquence et la suite $e_{i_1} e_{i_2} \dots e_{i_T}$ est appelée chemin.

Remarque 1.1. *En résumé, construire une chaîne de Markov, consiste à parcourir un chemin composé d'états en observant un symbole particulier dans chaque état.*

Exemple 1.4. *Dans les conditions de l'exemple 1.1, nous pouvons avoir les chaînes de Markov δ_1 et δ_2 présentées respectivement dans les équations 1.2 et 1.3.*

$$\begin{array}{cccccccc} \delta_1 : & \textit{Boire} & & \textit{Travailler} & & \textit{Jouer} & & \textit{Dormir} \\ & \uparrow & & \uparrow & & \uparrow & & \uparrow \\ & \textit{Fatigue} & \rightarrow & \textit{EnForme} & \rightarrow & \textit{EnForme} & \rightarrow & \textit{Fatigue} \end{array} \tag{1.2}$$

$$\begin{array}{cccccc} \delta_2 : & \textit{Visionner} & & \textit{Dormir} & & \textit{Danser} \\ & \uparrow & & \uparrow & & \uparrow \\ & \textit{EnForme} & \rightarrow & \textit{Fatigue} & \rightarrow & \textit{EnForme} \end{array} \tag{1.3}$$

Que valent $|\delta_1|$ et $|\delta_2|$?

Quelles probabilités attribueriez-vous à δ_1 et à δ_2 ? Pourquoi ?

Selon vous, de quels paramètres formels dépendent les valeurs de ces probabilités ?

Exemple 1.5. *Dans les conditions de l'exemple 1.2, nous pouvons aussi avoir les chaînes de Markov δ_3 et δ_4 présentées respectivement dans les équations 1.4 et 1.5.*

$$\begin{array}{cccccccc} \delta_3 : & \textit{Dormir} & & \textit{Etudier} & & \textit{Prier} & & \textit{Boire} \\ & \uparrow & & \uparrow & & \uparrow & & \uparrow \\ & \textit{Domicile} & \rightarrow & \textit{Ecole} & \rightarrow & \textit{Eglise} & \rightarrow & \textit{Bar} \end{array} \tag{1.4}$$

$$\begin{array}{cccccccc} \delta_4 : & \textit{Prier} & & \textit{Boire} & & \textit{Boire} & & \textit{Dormir} & & \textit{Danser} \\ & \uparrow & & \uparrow & & \uparrow & & \uparrow & & \uparrow \\ & \textit{Bar} & \rightarrow & \textit{Eglise} & \rightarrow & \textit{Eglise} & \rightarrow & \textit{Ecole} & \rightarrow & \textit{Ecole} \end{array} \tag{1.5}$$

Que valent $|\delta_3|$ et $|\delta_4|$?

Quelles probabilités attriburiez-vous à δ_3 et à δ_4 ? Pourquoi ?

Proposez maintenant trois chaînes de Markov toutes de longueur 7 qui selon vous correspondent respectivement le mieux à un ivrogne, à un homme pieux et à un élève irresponsable. Quelle est selon-vous la probabilité d'observer chacune d'elle dans la société ?

1.4 Influence de l'état initial

Dans les conditions de l'exemple 1.1, Si nous supposons que les observations constituant chaque chaîne de Markov sont capturées à des intervalles réguliers de 2h et que la toute première observation est toujours capturée à 5h du matin :

Quelle est d'après vous la probabilité que l'état initial soit *Fatigue* ? *EnForme* ? Pourquoi ?

Vos points de vues resteraient-ils invariables si la toute première observation était plutôt capturée à 9h ? 15h ? 23h ? Pourquoi ?

Dans le même ordre d'idées, mettez-vous maintenant dans les conditions de l'exemple 1.2 en supposant que les observations constituant chaque chaîne de Markov sont capturées à des intervalles réguliers de 2h et que la toute première observation est toujours capturée à 8h du matin.

Quelle est d'après vous la probabilité que l'état initial soit *Ecole* ? *Eglise* ? Pourquoi ?

Que pouvez-vous dire d'un individu dont la chaîne de Markov a pour état initial *Bar* ? Pourquoi ?

Quel est selon vous l'impact d'un tel état initial sur le reste des observations de la chaîne de Markov ?

Que pouvez-vous en déduire sur l'influence de l'état initial d'une chaîne de Markov sur la probabilité d'observer cette dernière ?

1.5 Définition d'un MMC

Soit $E = \{e_1, \ldots, e_N\}$ un ensemble d'états et $S = \{s_1, \ldots, s_M\}$ un ensemble de symboles. Il est facile de remarquer que pour répondre à presque toutes les questions posées jusqu'ici, il était nécessaire de faire usage de vos connaissances ou de votre ressentit afin :

1. D'attribuer une probabilité à chaque observation $\begin{matrix} s_j \\ \uparrow \\ e_i \end{matrix}$ pour tout $s_j \in S$ et $e_i \in E$.
2. D'attribuer une probabilité à chaque transition d'états $e_i \rightarrow e_j$ pour tout $(e_i, e_j) \in E^2$.
3. D'estimer la probabilité que l'état e_i soit l'état initial

Un MMC peut être simplement vu comme une structure de sauvegarde ou un support de stockage dans lequel vous enregistrez toutes les probabilités précédentes en vue d'un usage ultérieur. Plus formellement :

Définition 1.4. *Une modèle de Markov caché [1] λ est un 5-upplet composé :*

1. *D'un ensemble $E = \{e_1, \ldots, e_N\}$ d'états.*
2. *D'un ensemble $S = \{s_1, \ldots, s_M\}$ de symboles.*
3. *D'une matrice $N \times N$ de transition d'états notée $A = \{a_{ij}\}$ vérifiant $a_{ij} = Pr(e_i \rightarrow e_j)$*
4. *D'une matrice $N \times M$ d'observations notée $B = \{b_j(k)\}$ vérifiant $b_j(k) = Pr(\begin{matrix} s_k \\ \uparrow \\ e_j \end{matrix})$*
5. *D'un vecteur de taille N des états initiaux noté $\pi = \{\pi(i)\}$ vérifiant $\pi(i) = Pr(e_i \text{ soit un état initial})$*

De par leurs définitions, on constate que les composantes A, B et π d'un MMC sont des ***distributions de probabilités***. Ainsi la somme des éléments de chacune de leurs lignes vaut soit 1. Dans la littérature, les notations condensées $\lambda = \{A, B, \pi\}$ ou $\lambda = (A, B, \pi)$ sont très souvent utilisées. Dans ce cours, c'est la deuxième notation qui est adoptée.

Remarque 1.2. *Les modèles de Markov cachés sont en réalité des modèles de Markov '**à états cachés**'. Le qualificatif '**cachés**' rattaché aux états fait ici référence au fait que dans la pratique, on s'intéresse uniquement aux séquences (composées de symboles) observées par ces modèles, peu importe les chemins (composés d'états) par lesquels on passe pour observer ces séquences. Les états sont donc cachés, sans toutefois être inutiles.*

Exercice 1.1. *En vous servant de vos réponses aux questions posées jusqu'à ce niveau du cours, proposez un MMC $\lambda_1 = (A_1, B_1, \pi_1)$ associé à l'exemple 1.1. Complétez par vous-mêmes les probabilités manquantes. Les valeurs de probabilités nulles sont autorisées.*

$\boldsymbol{A_1}$	*Fatigue*	*EnForme*
Fatigue		
EnForme		

$\boldsymbol{B_1}$	*Dormir*	*Manger*	*Visionner*	*Jouer*	*Travailler*	*Danser*
Fatigue						
EnForme						

	Fatigue	*EnForme*
$\boldsymbol{\pi_1}$		

Exercice 1.2. *Proposez également un MMC $\lambda_2 = (A_2, B_2, \pi_2)$ associé à l'exemple 1.2 en vous servant de vos réponses aux questions posées jusqu'à ce niveau du cours. Complétez par vous-mêmes les probabilités manquantes. Les valeurs de probabilités nulles sont aussi autorisées.*

$\boldsymbol{A_2}$	*Domicile*	*Eglise*	*Bar*	*Ecole*
Domicile				
Eglise				
Bar				
Ecole				

$\boldsymbol{B_2}$	*Boire*	*Prier*	*Etudier*	*Danser*	*Dormir*
Domicile					
Eglise					
Bar					
Ecole					

	Domicile	*Eglise*	*Bar*	*Ecole*
$\boldsymbol{\pi_2}$				

1.6 Représentation graphique d'un MMC

De manière générale, seule la matrice A d'un MMC est représentée graphiquement par un graphe orienté particulier dans lequel :

1. Les états du MMC sont les noeuds du graphe.
2. Il existe un arc du noeud e_i vers le noeud e_j du graphe si la valeur a_{ij} du MMC est non nulle. Le poids de cet arc est la valeur de a_{ij}.

La particularité de ce graphe vient du fait que dans certains documents, il arrive que la matrice B apparaisse aussi sur le graphe. Dans ce cas, un arc en pointillés sortant de chaque état e_i est rajouté sur le graphe et la flèche de cet arc pointe vers la i^e ligne de la matrice B. Le vecteur π n'apparaît en général pas sur la représentation graphique.

Exemple 1.6. *La figure 1.1 est représente le MMC $\lambda_3 = (A_3, B_3, \pi_3)$ pour lequel :*

1- L'ensemble des états est $E = \{a, b, c, d\}$

2- L'ensemble des symboles est $S = \{x, y\}$

3- La distribution A_3 est donnée par :

A_3	a	b	c	d
a	0.7	0.1	0	0.2
b	0	0	1	0
c	0.5	0	0.3	0.2
d	0.6	0	0.4	0

4- La distribution B_3 est donnée par :

B_3	x	y
a	0.8	0.2
b	0.3	0.7
c	0.6	0.4
d	0.2	0.8

5- La distribution π_3 est donnée par :

π_3	a	b	c	d
	0.4	0.3	0.2	0.1

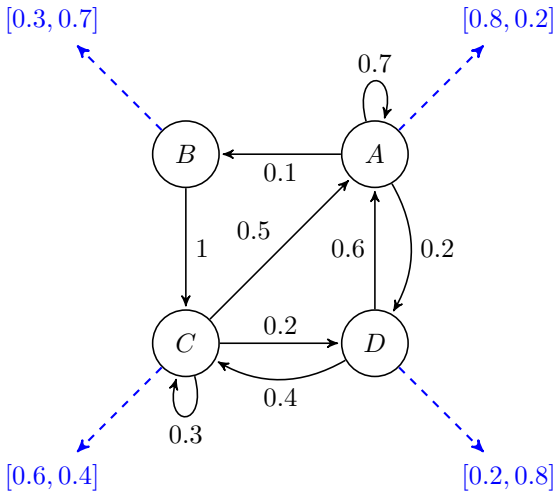

FIGURE 1.1 – Graphe associé au MMC λ_3

Exercice 1.3. *Réalisez les représentations graphiques des MMCs λ_1 et λ_2 que vous avez respectivement initialisés aux exercices 1.1 et 1.2.*

Représentation de λ_1

Représentation de λ_2

1.7 Les 3 principaux problèmes liés aux MMC

Les MMCs sont de manière générale utilisés pour résoudre les trois problèmes suivants :

1. **L'évaluation** de la probabilité d'observation d'une séquence O de symboles sachant le modèle λ. Cette évaluation doit être faite peu importe le chemin emprunté, car les états constituant ces chemins sont cachés. Il est donc question de calculer $Pr[O|\lambda]$. Le chapitre 2 présente les solutions à ce problème parmi lesquelles l'algorithme Forward-Backward.
2. **Le calcul du chemin optimal** pour l'observation d'une séquence O de symbole sachant le modèle λ. Il est question de déterminer le chemin idéal permettant de maximiser la valeur de $Pr[O|\lambda]$. La solution à ce problème est donnée par l'algorithme de Viterbi, mais cet algorithme ne sera pas présenté dans la suite de ce cours car il est très peu utilisé dans la pratique.
3. **L'entraînement** d'un modèle λ pour l'observation optimale d'une séquence de symboles O. Il est question ici d'optimiser les paramètres (A, B, π) de λ afin d'obtenir un nouveau modèle $\bar{\lambda}$ tel que $Pr[O|\bar{\lambda}]$ soit optimale. L'algorithme de Baum-Welch présenté au chapitre 3 permet de résoudre ce problème.

Chapitre 2

Le problème d'évaluation

2.1 Solution basique

Soit $E = \{e_1, \ldots, e_N\}$ un ensemble d'états et $S = \{s_1, \ldots, s_M\}$ un ensemble de symboles. Dans la suite de cours, q_t sera utilisé pour désigné l'état du modèle à l'instant t, et o_t désignera le symbole observé par le modèle dans l'état q_t. Considérez une séquence $O = o_1 o_2 \ldots o_T$ et un MMC $\lambda = (A, B, \pi)$. Nous voulons calculer la probabilité de pouvoir observer O peu importe le chemin emprunté sachant qu'on utilise le contenu de λ, c'est-à-dire $Pr[O|\lambda]$. Cela revient à évaluer pour chaque chemin possible Q, la probabilité d'observer O en empruntant le chemin Q sachant qu'on utilise le contenu de λ, c'est-à-dire $Pr[O, Q|\lambda]$. Soit $Q = q_1 q_2 \ldots q_T$ l'un de ces chemins possibles, calculer $Pr[O, Q|\lambda]$ revient à évaluer la probabilité :

1. d'emprunter le chemin Q sachant qu'on utilise le contenu de $\lambda \Rightarrow Pr[Q|\lambda]$
2. puis d'observer O sachant qu'on a emprunter ce chemin Q et qu'on utilise le contenu de $\lambda \Rightarrow Pr[O|Q, \lambda]$

Ainsi, $Pr[O, Q|\lambda] = Pr[Q|\lambda] \times Pr[O|Q, \lambda]$. Or :

1. $\begin{aligned} Pr[O|Q,\lambda] &= b_{q_1}(o_1) \times b_{q_2}(o_2) \times \ldots \times b_{q_T}(o_T) \\ &= \left(\prod_{t=1}^{T} b_{q_t}(o_t)\right) \end{aligned}$

2. $\begin{aligned} Pr[Q|\lambda] &= \pi_{q_1} \times a_{q_1 q_2} \times a_{q_2 q_3} \times \ldots \times a_{q_{T-1} q_T} \\ &= \pi_{q_1} \times \left(\prod_{t=1}^{T-1} a_{q_{t-1} q_t}\right) \end{aligned}$

Lorsqu'on prend en compte tous les chemins $Q = q_1 q_2 \ldots q_T$ possibles, on obtient :

$$\begin{aligned} Pr[O|\lambda] &= \Sigma_{\forall Q} Pr[O, Q|\lambda] \\ &= \Sigma_{\forall Q}(Pr[Q|\lambda]) \times (Pr[O|Q, \lambda]) \\ &= \Sigma_{\forall q_1 q_2 \ldots q_T} \left(\pi_{q_1} \times \prod_{t=1}^{T-1} a_{q_{t-1} q_t}\right) \times \left(\prod_{t=1}^{T} b_{q_t}(o_t)\right) \end{aligned} \tag{2.1}$$

Étant donné un chemin Q, combien d'opérations arithmétiques sont nécessaires pour calculer $Pr[Q|\lambda]$?

En considérant ce chemin Q, combien d'opérations arithmétiques sont nécessaires pour calculer $Pr[O|Q,\lambda]$?

Déterminez à partir de ces deux résultats le nombre d'opérations arithmétiques requises par le calcul de $Pr[O,Q|\lambda]$ pour ce même chemin Q ?

Combien de chemins Q distincts peuvent être empruntés pour observer la séquence O ?

Déduisez de ce qui précède la complexité du calcul de $Pr[O|\lambda]$ en opérations arithmétiques. Quel est l'ordre de grandeur de cette complexité ? Que pouvez-vous en conclure ?

Exercice 2.1. *En considérant le MMC λ_1 dont vous avez fixé les paramètres à l'exercice 1.1, calculez $Pr[\ (Visionner\ Dormir\ Danser)\ |\lambda_1]$. Vous supposerez que l'état Fatigue a pour index 1 et que l'état EnForme a pour index 2.*

Résolution de l'exercice 2.1

2.2 Algorithme Forward-Backward

Une alternative plus réaliste pour le calcul de $Pr[O|\lambda]$ est donnée par l'algorithme Forward-Backward [1] qui utilise les variables Forward et les variables Backward.

2.2.1 Les variables Forward

Considérez un MMC λ ayant pour ensemble d'états $E = \{e_1, \ldots, e_N\}$ et pour ensemble de symboles $S = \{s_1, \ldots, s_M\}$. Soit $O = o_1 o_2 \ldots o_T$ une séquence de symboles.

Définition 2.1. *La variable Forward d'index j à l'instant t qu'on note $\alpha_t(j)$ est la probabilité d'observer la sous-séquence $o_1 o_2 \ldots o_t$ peu importe le chemin emprunté et qu'à l'instant t on est dans l'état e_j* ***sachant*** *qu'on utilise le contenu de λ. Soit que $\alpha_t(j) = Pr[o_1 o_2 \ldots o_t, q_t = e_j|\lambda]$ comme cela est décrit ci-dessous.*

$$\begin{array}{c} \overbrace{o_1 \; o_2 \; \ldots \; o_{t-1} \; o_t}^{\alpha_t(j)} \\ \uparrow \\ e_j \end{array}$$

Afin de calculer $\alpha_t(j)$ pour tout $t = 1, 2, \ldots, T$, nous allons procéder de manière récursive en calculant dans un premier temps $\alpha_1(j)$, puis en montrant comment calculer $\alpha_{t+1}(j)$ à partir de $\alpha_t(j)$.

$$\begin{array}{lrcl} 1) & \alpha_1(j) & = & Pr[o_1, (q_1 = e_j)|\lambda] \\ & & = & Pr[(q_1 = e_j)|\lambda] \times Pr[o_1|(q_1 = e_j), \lambda] \\ & & = & \pi_j \times b_j(o_1) \end{array}$$

$$2) \quad \alpha_{t+1}(j) = Pr[o_1 o_2 \ldots o_t o_{t+1}, q_{t+1} = e_j|\lambda]$$

Cela se traduit par N chaînes de Markov différentes comme le démontre la représentation graphique ci-dessous lorsque que i varie de 1 à N :

$$\begin{array}{c} \overbrace{o_1 \; o_2 \; \ldots \; o_t \; o_{t+1}}^{\alpha_{t+1}(j)} \\ \uparrow \\ e_j \end{array} \quad \Longrightarrow \quad \begin{array}{cc} \overbrace{o_1 \; o_2 \; \ldots \; o_t}^{\alpha_t(i), \forall i} & o_{t+1} \\ \uparrow & \uparrow \\ e_i & e_j \end{array}$$

Finalement, on obtient : $\begin{array}{lrcl} & \alpha_{t+1}(j) & = & \sum_{i=1}^{N} \alpha_t(i) \times a_{ij} \times b_j(o_{t+1}) \\ & & = & b_j(o_{t+1}) \times \left(\sum_{i=1}^{N} \alpha_t(i) \times a_{ij} \right) \end{array}$

En résumé, les variables Forward se calculent avec l'équation 2.2.

$$\begin{cases} \alpha_1(j) = \pi_j \times b_j(o_1) \\ \alpha_{t+1}(j) = b_j(o_{t+1}) \times \left(\sum_{i=1}^{N} \alpha_t(i) \times a_{ij} \right) \end{cases} \tag{2.2}$$

Exercice 2.2. *Calculez tous les index des variables Forward associées à la séquence (Visionner Dormir Danser) en considérant votre MMC λ_1 de l'exercice 1.1.*

Résolution de l'exercice 2.2

$\boldsymbol{\alpha_t(j)}$	$t=1$	$t=2$	$t=3$
$j=1$			
$j=2$			

2.2.2 Les variables Backward

Définition 2.2. *La variable Backward d'index i à l'instant t qu'on note $\beta_t(i)$ est la probabilité d'observer la sous-séquence $o_{t+1}o_{t+2}\ldots o_T$ peu importe le chemin emprunté* ***sachant*** *qu'à l'instant t on était dans l'état e_i et qu'on utilise le contenu de λ. Cela voudrait formellement dire que $\beta_t(i) = Pr[o_{t+1}o_{t+2}\ldots o_T|q_t = e_i, \lambda]$ comme cela est décrit ci-dessous.*

$$\begin{array}{l} o_t \overbrace{o_{t+1}\ o_{t+2}\ \ldots\ o_{T-1}\ o_T}^{\beta_t(i)} \\ \uparrow \\ e_i \end{array}$$

De manière analogue à ce qui a été fait pour le calcul précédent, nous allons calculer récursivement les $\beta_t(i)$ en calculant dans un premier temps $\beta_T(i)$, puis en montrant comment calculer $\beta_{t-1}(i)$ à partir de $\beta_t(i)$.

1) $\beta_T(i) = Pr[o_{T+1}\ldots o_T|q_T = e_i, \lambda]$ ce qui n'a pas de sens !!!
On fixe donc par covention $\beta_T(i) = 1$

2) $\beta_{t-1}(i) = Pr[o_t o_{t+1}\ldots o_T|q_{t-1} = e_i, \lambda]$

Cela se traduit par N chaînes de Markov différentes comme le démontre la représentation graphique ci-dessous lorsque que j varie de 1 à N :

$$\begin{array}{l} o_{t-1} \overbrace{o_t\ o_{t+1}\ \dots\ o_T}^{\beta_{t-1}(i)} \\ \uparrow \\ e_i \end{array} \quad \Longrightarrow \quad \begin{array}{l} o_{t-1} \quad o_t \overbrace{o_{t+1}\ \dots\ o_T}^{\beta_t(j),\forall j} \\ \uparrow \quad\quad \uparrow \\ e_i \ \rightarrow \ e_j \end{array}$$

Finalement, on obtient : $\beta_{t-1}(i) = \sum_{j=1}^{N} a_{ij} \times b_j(o_t) \times \beta_t(j)$

En résumé, les variables Backward se calculent avec l'équation 2.3.

$$\begin{cases} \beta_T(i) = 1 \\ \beta_{t-1}(i) = \sum_{j=1}^{N} a_{ij} \times b_j(o_t) \times \beta_t(j) \end{cases} \tag{2.3}$$

Exercice 2.3. *Calculez tous les index des variables Backward associées à la séquence (Visionner Dormir Danser) en considérant votre MMC λ_1 de l'exercice 1.1.*

Résolution de l'exercice 2.3

$\boldsymbol{\beta_t(i)}$	$t=1$	$t=2$	$t=3$
$i=1$			
$i=2$			

2.2.3 Forward-Backward

Considérons une séquence $O = o_1 o_2 \dots o_T$ et un MMC λ. Lorsque les représentations graphiques des variables Forward et Backward respectivement présentées dans les définitions 2.1 et 2.2 sont combinées, nous obtenons la représentation globale ci-dessous pour une valeur arbitraire $\bar{t}$ de t sélectionnée entre 1 et T :

$$\begin{array}{l} \overbrace{o_1\ o_2\ \dots\ o_{\bar{t}-1}\ o_{\bar{t}}}^{\alpha_{\bar{t}}(j)} \quad \overbrace{o_{\bar{t}+1}\ o_{\bar{t}+2}\ \dots\ o_{T-1}\ o_T}^{\beta_{\bar{t}}(j)} \\ \uparrow \\ e_j \end{array} \tag{2.4}$$

En vous référant à l'équation 2.4, que représente selon vous le produit $\alpha_{\bar{t}}(j) \times \beta_{\bar{t}}(j)$?

Dans ces conditions, que représente alors $\sum_{j=1}^{N} \alpha_{\bar{t}}(j) \times \beta_{\bar{t}}(j)$

La valeur de $\bar{t}$ influence t-elle la réponse à la question précédente ? Pourquoi ?

Déduisez de ce qui précède les étapes d'un algorithme faisant usage des variables Forward et Backward pour calculer $Pr[O|\lambda]$.

Exercice 2.4. *En vous servant de votre réponse à la question précédente, complétez le pseudo-code de l'algorithme Forward-Backward ci-dessous afin de calculer* $Pr[O|\lambda]$.

Algorithme 1 ForwardBackward($O = o_1 o_2 \ldots o_T, \lambda = (A, B, \pi)$)

```
1: Choisir t̄
2: Pour (j = 1 ; (j≤N) ; j++) Faire
3:     α_1(j) =
4:     β_T(j) =
5: FinPour
6: Pour (t = 1 ; (t<t̄) ; t++) Faire
7:     Pour (j = 1 ; (j≤N) ; j++) Faire
8:         α_{t+1}(j) =
9:     FinPour
10: FinPour
11: Pour (t = T ; (t>t̄) ; t−−) Faire
12:     Pour (i = 1 ; (i≤N) ; i++) Faire
13:         β_{t−1}(i) =
14:     FinPour
15: FinPour
16: Renvoyer P[O|λ] =
```

Devoir 2.1. *Il vous est demandé :*

1. *D'écrire un programme en langage C qui charge les paramètres d'un MMC* $\lambda = (A, B, \pi)$ *depuis un fichier texte suivant le formatage de votre choix et qui lit une séquence* O *de longueur* T *au clavier, puis calcule et affiche la valeur de* $Pr[O|\lambda]$ *suivant le principe de l'algorithme Forward-Backward.*
2. *D'utiliser votre programme pour calculer* $Pr[O|\lambda_2]$ *où* λ_2 *est le MMC dont vous avez fixé les paramètres à l'exercice 1.2 et* $O = (Prier\ Boire\ Boire\ Dormir\ Danser)$.
3. *D'utiliser votre programme pour calculer* $Pr[O|\lambda_3]$ *où* λ_3 *est le MMC de l'exemple 1.6 et* $O = (xxxyyxy)$.

Devoir 2.2. *Il vous est demandé de :*

1. *Calculer la complexité en nombre d'opérations arithmétiques de l'algorithme 1 dont vous avez complété le pseudo-code à l'exercice 2.4.*
2. *Montrer que cette complexité est de l'ordre de $\theta(T.N^2)$.*
3. *Comparer cette complexité à celle de la solution proposée à la section 2.1.*

Exercice 2.5. *En considérant le MMC λ_1 de l'exercice 1.1 ainsi que les variables Forward et Backward que vous avez respectivement calculé aux exercices 2.2 et 2.3, calculez la valeur de $Pr[O|\lambda_1]$ avec $O = (Visionner\ Dormir\ Danser)$ pour chaque valeur de $\bar{t}$ prise entre 1 et 3.*

Résolution de l'exercice 2.5

	$\bar{t}=1$	$\bar{t}=2$	$\bar{t}=3$
$\boldsymbol{Pr[O\|\lambda_1]}$			

Chapitre 3

Le problème d'entraînement d'un MMC

Soit $E = \{e_1, \ldots, e_N\}$ un ensemble d'états et $S = \{s_1, \ldots, s_M\}$ un ensemble de symboles. Considérez une séquence $O = o_1 o_2 \ldots o_T$ et un MMC $\lambda = (A, B, \pi)$. Nous voulons optimiser les paramètres (A, B, π) de λ afin d'obtenir un nouveau modèle $\bar{\lambda}$ tel que $Pr[O|\bar{\lambda}]$ soit optimale. Cette optimisation est nécessaire car les paramètres du modèle initial λ sont très souvent arbitrairement fixés et ne tiennent généralement pas compte du contenu de la séquence O, ni des éventuels chemins pouvant être empruntés pour observer O. Pour parvenir à nos fins, nous allons commencer par définir de nouvelles variables nommées Xi.

3.1 Les variables Xi

Définition 3.1. *La variable Xi rattachée aux index i et j à l'instant t notée $\xi_t(i,j)$ est la probabilité d'avoir emprunté un chemin par lequel de l'instant t à l'instant $t+1$, il y a une transition $e_i \rightarrow e_j$ **sachant** qu'on a observé la séquence O et qu'on a utilisé le contenu de λ. Soit que $\xi_t(i,j) = Pr[q_t = e_i, q_{t+1} = e_j|O, \lambda]$ comme cela est décrit ci-dessous.*

$$\begin{array}{ccccccccc} o_1 & o_2 & \ldots & o_t & & o_{t+1} & o_{t+2} & \ldots & o_T \\ & & & \uparrow & & \uparrow & & & \\ & & & e_i & \rightarrow & e_j & & & \end{array}$$

Ainsi, $\xi_t(i,j)$ peut simplement être perçue comme la probabilité qu'une transition $e_i \rightarrow e_j$ soit déclenchée à l'instant t durant l'observation de O. Comme vous pouvez le constater dans cette définition, les variables ξ se focalisent sur une spécificité du chemin emprunté en supposant qu'on a déjà observé l'intégralité de la séquence O contrairement aux variables α et β qui elles se focalisent plutôt sur la possibilité d'observer une sous-séquence de O. Commençons par remarquer que les chemins pris en compte dans le calcul de $\xi_t(i,j)$ ne sont qu'un sous-ensemble des chemins pris en compte dans le calcul de $P[O|\lambda]$. Cette analyse nous permet de déduire que $\xi_t(i,j)$ peut se mettre sous la forme d'une fraction $\frac{X}{Y}$ où :

1. $X = Pr[O, q_t = e_i, q_{t+1} = e_j|\lambda]$ est le nombre de cas favorables, c'est-à-dire la probabilité de pouvoir observer O en passant par un chemin ayant les propriétés souhaitées sachant qu'on utilise le contenu de λ.

2. $Y = P[O|\lambda]$ est le nombre de cas possibles, c'est-à-dire la probabilité de pouvoir observer O peu importe le chemin emprunté sachant qu'on utilise le contenu de λ.

La valeur de Y s'obtient par le biais de l'algorithme Forward-Backward (Cf. algorithme 1). Donc, il nous reste seulement à calculer la valeur de X. Le calcul de X aussi peut se déduire des variables α et β comme le montre la figure ci-dessous :

$$\begin{array}{cccc} \overbrace{o_1\ o_2\ \ldots\ o_t}^{\alpha_t(i)} & & o_{t+1}\ \overbrace{o_{t+2}\ \ldots\ o_T}^{\beta_{t+1}(j)} \\ \uparrow & & \uparrow \\ e_i & \rightarrow & e_j \end{array}$$

De cette représentation, on peut donc déduire que :

$$\begin{aligned} X &= Pr[O, q_t = e_i, q_{t+1} = e_j|\lambda] \\ &= Pr[o_1 \ldots o_t, q_t = e_i|\lambda] \times Pr[e_i \rightarrow e_j|\lambda] \times Pr[\ \overset{o_{t+1}}{\underset{e_j}{\uparrow}}\ |\lambda] \times Pr[o_{t+2} \ldots o_T, q_{t+1} = e_j|\lambda] \\ &= \alpha_t(i) \times a_{ij} \times b_j(o_{t+1}) \times \beta_{t+1}(j) \end{aligned}$$

Finalement, l'équation 3.1 résume le calcul de $\xi_t(i,j)$ avec $\bar{t} \in \{1, 2, \ldots, T\}$.

$$\xi_t(i,j) = \left(\frac{\alpha_t(i) \times a_{ij} \times b_j(o_{t+1}) \times \beta_{t+1}(j)}{P[O|\lambda]} \right) = \left(\frac{\alpha_t(i) \times a_{ij} \times b_j(o_{t+1}) \times \beta_{t+1}(j)}{\sum_{k=1}^{N} \alpha_{\bar{t}}(k) \times \beta_{\bar{t}}(k)} \right) \tag{3.1}$$

Exercice 3.1. *Calculez toutes les variables Xi rattachées à tous les couples d'index en considérant la séquence (Visionner Dormir Danser) et votre MMC λ_1 de l'exercice 1.1. Vous utiliserez pour cela les variables Forward et Backward respectivement calculées aux exercices 2.2 et 2.3 ainsi que la probabilité calculée à l'exercice 2.5*

Résolution de l'exercice 3.1

$\xi_t(i,j)$		$t=1$	$t=2$	$t=3$
$i=1$	$j=1$			
$i=1$	$j=2$			
$i=2$	$j=1$			
$i=2$	$j=2$			

Devoir 3.1. *Il vous est demandé :*

1. *D'écrire un programme en langage C qui va charger les paramètres d'un MMC* $\lambda = (A, B, \pi)$ *depuis un fichier texte suivant le formatage de votre choix et lire une séquence* O *de longueur* T *au clavier, puis il va ranger dans un autre fichier texte toutes les variables Xi pour tous les couples d'index.*
2. *De tester votre programme en considérant le MMC* λ_2 *dont vous avez fixé les paramètres à l'exercice 1.2 et la séquence* $O = (Prier\ Boire\ Boire\ Dormir\ Danser)$.
3. *De tester votre programme en considérant le MMC* λ_3 *de l'exemple 1.6 et la séquence* $O = (xxxyyxy)$.

3.2 Optimisation des paramètres d'un MMC

3.2.1 Optimisation de π

L'idée ici est d'optimiser la valeur de $\pi(i)$ en la remplaçant par $\bar{\pi}(i) = Pr[q_1 = e_i, O|\lambda]$ qui est la probabilité d'emprunter un chemin dans lequel l'état e_i est l'état initial durant l'observation de O. Il s'agit d'une réelle amélioration car contrairement à $\pi(i) = Pr[q_1 = e_i|\lambda]$ qui est juste la probabilité que l'état e_i soit l'état initial, peu importe la séquence à observer, $\bar{\pi}(i)$ s'assure que e_i a été l'état initial durant l'observation la séquence O. Comment calculer $\bar{\pi}_i$?

Si on pose $\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j)$, quel sens donnez-vous cette variable ?

Dans ces conditions, que représente alors $\gamma_1(i)$?

Déduisez de ce qui précède que pour optimiser le vecteur π, il suffit de fixer $\bar{\pi}(i) = \gamma_1(i)$.

Exercice 3.2. *En vous servant des variables Xi calculées à l'exercice 3.1, calculez le vecteur* $\bar{\pi}_1$ *associé à la séquence* $(Visionner\ Dormir\ Danser)$ *en considérant le MMC* λ_1 *de l'exercice 1.1.*

Résolution de l'exercice 3.2

	$i = 1$	$i = 2$
$\bar{\pi}_1(i)$		

3.2.2 Optimisation de A

L'idée ici est d'optimiser la valeur de a_{ij} en la remplaçant par $\bar{a}_{ij} = Pr[e_i \to e_j, O|\lambda]$ qui est la probabilité de rencontrer une transition $e_i \to e_j$ durant l'observation de O, peu importe l'instant. Il s'agit là aussi d'une réelle amélioration car contrairement à $a_{ij} = Pr[e_i \to e_j|\lambda]$ qui est juste la probabilité d'effectuer une transition $e_i \to e_j$, peu importe la séquence à observer et peu importe l'instant, $\bar{a}_{ij}$ s'assure que cette transition a lieu durant l'observation de O. Comment calculer $\bar{a}_{ij}$?

Nous devons commencer par remarquer que les chemins pris en compte dans le calcul de $\bar{a}_{ij}$ ne sont qu'un sous ensemble de tous les chemins dans lesquels on rencontre une transition partant de e_i, peu importe l'état cible et peu importe l'instant, durant l'observation de O. On en déduit que $\bar{a}_{ij}$ peut s'écrire sous la forme $\frac{X}{Y}$ où :

1. $X = Pr[e_i \to e_j|O, \lambda]$ est le nombre de cas favorables, c'est-à-dire la probabilité d'avoir observé O en empruntant un chemin dans lequel on rencontre une transition $e_i \to e_j$, peu importe l'instant.
2. $Y = Pr[e_i \to ?|O, \lambda]$ est le nombre de cas possibles, c'est-à-dire la probabilité d'avoir observé O en empruntant un chemin dans lequel on rencontre une transition partant de e_i, peu importe l'état cible et peu importe l'instant.

Quel sens donnez-vous à la somme $\sum_{t=1}^{T-1} \xi_t(i,j)$?

De même, quel sens donnez-vous à la somme $\sum_{t=1}^{T-1} \gamma_t(i)$?

En déduire les valeurs de X et de Y permettant optimiser le matrice A en fixant $\bar{a}_{ij} = \frac{X}{Y}$

Exercice 3.3. *Utilisez les variables Xi calculées à l'exercice 3.1 pour calculer la matrice $\bar{A}_1$ associée à la séquence (Visionner Dormir Danser) en considérant votre MMC λ_1 de l'exercice 1.1.*

Résolution de l'exercice 3.3

$\bar{A}_1$	$j=1$	$j=2$
$i=1$		
$i=2$		

3.2.3 Optimisation de B

De manière analogue aux deux précédentes optimisations, l'idée ici est d'optimiser la valeur de $b_j(l) = Pr[\overset{s_l}{\underset{e_j}{\uparrow}} |\lambda]$ en la remplaçant par $\bar{b}_j(l) = Pr[\overset{s_l}{\underset{e_j}{\uparrow}}, O|\lambda]$ qui est la probabilité d'observer le symbole s_l étant dans l'état e_i durant l'observation de O, peu importe l'instant. Comme cela a été le cas pour les deux précédentes optimisations, $\bar{b}_j(l)$ peut s'écrire sous la forme $\frac{X}{Y}$ où :

1. $X = Pr[\overset{s_k}{\underset{e_j}{\uparrow}} |O, \lambda]$ est le nombre de cas favorables, c'est-à-dire la probabilité d'avoir observé O en observant le symbole s_k étant dans l'état e_i, peu importe l'instant.
2. $Y = Pr[\overset{?}{\underset{e_j}{\uparrow}} |O, \lambda]$ est le nombre de cas possibles, c'est-à-dire la probabilité d'avoir observé O en passant par l'état e_i, peu importe le symbole observé dans cet état et peu importe l'instant.

Soit $U_l = \{t \mid o_t = s_l\}$ l'ensemble des instants où le symbole courant dans la séquence O est s_l. Quel sens donnez-vous à la somme $\sum_{t \in U_l} \gamma_t(j)$?

Rappelez le sens que vous aviez donné à la somme $\sum_{t=1}^{T-1} \gamma_t(j)$?

En déduire les valeurs de X et de Y permettant optimiser le matrice B en fixant $\bar{b}_j(l) = \frac{X}{Y}$

Exercice 3.4. *Utilisez les variables Xi calculées à l'exercice 3.1 afin de calculer la matrice $\bar{B}_1$ associée à la séquence (Visionner Dormir Danser) en considérant le MMC λ_1 de l'exercice 1.1.*

Résolution de l'exercice 3.4

Suite de l'exercice 3.4

$\bar{B}_1$	*Dormir*	*Manger*	*Visionner*	*Jouer*	*Travailler*	*Danser*
$j = 1$						
$j = 2$						

Devoir 3.2. *Il vous est demandé :*

1. *D'utiliser le programme écrit dans le devoir 2.1 pour calculer* $Pr[O|\bar{\lambda}_1]$ *avec* $O = (Visionner\ Dormir\ Danser)$ *où* $\bar{\lambda}_1 = (\bar{A}_1, \bar{B}_1, \bar{\pi}_1)$ *est le MMC dont vous avez calculé les paramètres aux exercices 3.3, 3.4 et 3.2.*
2. *De comparer la valeur de* $Pr[O|\bar{\lambda}_1]$ *que vous venez de calculer avec celle de* $Pr[O|\lambda_1]$ *que vous aviez calculé à l'exercice 2.5 ? Que remarquez-vous ?*
3. *D'appliquer une nouvelle fois le même principe d'optimisation pour calculer* $Pr[O|\bar{\bar{\lambda}}_1]$ *avec* $O = (Visionner\ Dormir\ Danser)$ *sachant que* $\bar{\bar{\lambda}}_1 = (\bar{\bar{A}}_1, \bar{\bar{B}}_1, \bar{\bar{\pi}}_1)$.
4. *De comparer* $Pr[O|\lambda_1]$, $Pr[O|\bar{\lambda}_1]$ *et* $Pr[O|\bar{\bar{\lambda}}_1]$. *Quelle remarque faites-vous ? Si vous répétez le même processus un grand nombre de fois, que se passera t-il d'après-vous ?*
5. *De déduire un principe algorithmique pour l'entraînement d'un MMC.*

3.3 Algorithme de Baum-Welch pour une séquence

Partant d'un modèle initial $\lambda = (A, B, \pi)$, le principe de l'algorithme de Baum-Welch [1] est d'optimiser de manière itérative les trois paramètres de λ afin d'obtenir un meilleur modèle $\bar{\lambda} = (\bar{A}, \bar{B}, \bar{\pi})$ puis de remplacer à chaque itération λ par $\bar{\lambda}$ tant que l'une des deux conditions ci-dessous est vérifiée :

1. $Pr[O|\bar{\lambda}] - P[O|\lambda] > \varepsilon$ où ε est un seuil minimal de précision fixé par l'utilisateur.
2. Le nombre d'itérations actuel est inférieur à un seuil $MaxIter$ fixé par l'utilisateur.

L'algorithme va itérer ainsi jusqu'à un point de convergence où $\lambda \approx \bar{\lambda}$. La précision de cette approximation dépend de la valeur de ε. Afin d'obtenir d'espérer atteindre un point de convergence parfaite, c'est-à-dire que $\lambda = \bar{\lambda}$, il suffit de fixer $\varepsilon = 0$ et de ne pas mettre de condition sur le nombre d'itérations. L'algorithme 2 résume ce principe.

Algorithme 2 Baum-Welch($O = o_1 o_2 \ldots o_T, \lambda = (A, B, \pi)$, $MaxIter$, ε, $\bar{t}$)

Calculer les $\alpha_t(i)$ et les $\beta_t(i)$ en utilisant λ
$Fin = 0$
$Iter = 0$
Repéter
  Calculer les $\xi_t(i,j)$ en utilisant les $\alpha_t(i)$, les $\beta_t(i)$, $\bar{t}$ et λ
  Calculer les $\gamma_t(j)$ en utilisant les $\xi_t(i,j)$
  Calculer les $\bar{a}_{ij}$ en utilisant les $\xi_t(i,j)$ et les $\gamma_t(j)$
  Calculer les $\bar{b}_j(l)$ en utilisant les $\gamma_t(j)$
  Calculer les $\bar{\pi}_i$ en utilisant les $\gamma_t(j)$
  $\bar{\lambda} = (\{\bar{a}_{ij}\}, \{\bar{b}_j(l)\}, \{\bar{\pi}_i\})$
  Calculer les $\bar{\alpha}_t(i)$ et les $\bar{\beta}_t(i)$ en utilisant les $\bar{\lambda}$
  $Pr[O|\lambda] = \sum_{i=1}^{N} \alpha_{\bar{t}}(i) \times \beta_{\bar{t}}(i)$
  $Pr[O|\bar{\lambda}] = \sum_{i=1}^{N} \bar{\alpha}_{\bar{t}}(i) \times \bar{\beta}_{\bar{t}}(i)$
  $Iter = Iter + 1$
  Si $(Pr[O|\bar{\lambda}] - Pr[O|\lambda] \leq \varepsilon)$ **ou** $(Iter > MaxIter)$ **Alors**
    $Fin = 1$
  Sinon
    $\lambda = \bar{\lambda}$
    $\alpha_t(i) = \bar{\alpha}_t(i)$
    $\beta_t(i) = \bar{\beta}_t(i)$
  FinSi
Jusqu'à $(Fin == 1)$
Renvoyer $\bar{\lambda}$

Devoir 3.3. *En vous appuyant sur le contenu de ce chapitre :*

1. *Calculez la complexité en nombre d'opérations arithmétiques :*
   (a) *Du calcul de tous les* $\xi_t(i,j)$ *sachant* λ
   (b) *Du calcul de tous les* $\gamma_t(j)$ *sachant* λ
   (c) *Du calcul de tous les* $\bar{a}_{ij}$ *sachant* λ
   (d) *Du calcul de tous les* $\bar{b}_j(k)$ *sachant* λ
   (e) *Du calcul de tous les* $\bar{\pi}_i$ *sachant* λ
2. *Rappelez l'ordre de grandeur de la complexité en nombre d'opérations arithmétiques de l'algorithme Forward-Backward que vous avez calculé dans le devoir 2.2.*
3. *Déduisez l'ordre de grandeur de la complexité d'une itération de l'algorithme 2.*
4. *Soit Iter le nombre exact d'itérations réalisées par une exécution de l'algorithme 2. Déterminez en fonction de Iter, T et N l'ordre de grandeur de la complexité de l'algorithme de Baum-Welch.*

Devoir 3.4. *Il vous est demandé :*

1. *D'écrire un programme en langage C qui charge les paramètres d'un MMC* $\lambda = (A, B, \pi)$ *depuis un fichier texte suivant le formatage de votre choix et qui lit une séquence* O *de longueur* T *au clavier, puis range dans un autre fichier texte le MMC* $\bar{\lambda} = (\bar{A}, \bar{B}, \bar{\pi})$ *suivant le principe de l'algorithme 2. Vous fixerez* $\varepsilon = 0$, $MaxIter = 100$ *et* $\bar{t} = 1$.
2. *D'utiliser votre programme pour calculer* $\bar{\lambda}_1 = (\bar{A}_1, \bar{B}_1, \bar{\pi}_1)$ *où* λ_1 *est le MMC dont vous avez fixé les paramètres à l'exercice 1.1 et* $O = (Visionner\ Dormir\ Danser)$. *Comparez* $Pr[O|\bar{\lambda}_1]$ *et* $Pr[O|\lambda_1]$. *Que remarquez-vous ?*
3. *D'utiliser votre programme pour calculer* $\bar{\lambda}_2 = (\bar{A}_2, \bar{B}_2, \bar{\pi}_2)$ *où* λ_2 *est le MMC dont vous avez fixé les paramètres à l'exercice 1.2 et* $O = (Prier\ Boire\ Boire\ Dormir\ Danser)$. *Comparez* $Pr[O|\bar{\lambda}_2]$ *et* $Pr[O|\lambda_2]$. *Que remarquez-vous ?*
4. *D'utiliser votre programme pour calculer* $\bar{\lambda}_3 = (\bar{A}_3, \bar{B}_3, \bar{\pi}_3)$ *où* λ_3 *est le MMC de l'exemple 1.6 et* $O = (xxxyyxy)$. *Comparez* $Pr[O|\bar{\lambda}_3]$ *et* $Pr[O|\lambda_3]$. *Que remarquez-vous ?*

3.4 Algorithme de Baum-Welch multi-séquences

Dans la section 3.3, vous avez vu comment un MMC λ peut '*apprendre*' une séquence O en optimisant ses paramètres pour l'observation de O. Dans le contexte de l'exercice 1.2, la séquence O traduit la suite d'actions consécutives posées par un seul individu v. Par exemple : $O = (Prier\ Boire\ Boire\ Dormir\ Danser)$. L'algorithme de Baum-Welch précédent permet alors de capturer le comportement de l'individu v au regard de ses actions consécutives. Cependant, il peut arriver que l'on ne souhaite pas seulement étudier le comportement d'un seul individu mais plutôt celui d'un ensemble $V = \{v_1, v_2, \ldots v_k\}$ composé de K individus ayant posés les séquences d'actions $O = \{O^{(1)}, O^{(2)}, \ldots, O^{(K)}\}$ sachant que l'individu v_k est l'auteur de la séquence $O^{(k)}$de longueur $T^{(k)}$ avec $k \in \{1, 2, \ldots, K\}$. L'un des atouts majeurs des MMCs par rapports aux autres modèles mathématiques est qu'ils offrent la ***possibilité d'apprendre le contenu d'un ensemble fini de séquences***. C'est la version multi-séquences de l'algorithme de Baum-Welch.

Soit $\lambda = (\{a_{ij}\}, \{b_j(l)\}, \{\pi(i)\})$ le MMC initial associé à l'ensemble $O = \{O^{(1)}, O^{(2)}, \ldots, O^{(K)}\}$. Si à chaque séquence $O^{(k)}$ on associe les variables $\alpha_t^{(k)}(i)$, $\beta_t^{(k)}(i)$, $\xi_t^{(k)}(i,j)$, $\gamma_t^{(k)}(j)$ et l'ensemble $U_l^{(k)} = \{t \mid o_l^{(k)} = s_l\}$, alors :

Quelle interprétation faites-vous de la somme $Y_1 = \left(\sum_{k=1}^{K} \gamma_1^{(k)}(i)\right)$?

Quel sens attribuez-vous à la somme $Y_2 = \sum_{k=1}^{K} \left(\sum_{t=1}^{T^{(k)}-1} \xi_t^{(k)}(i,j)\right)$?

Comment interprétez-vous la somme $Y_3 = \sum_{k=1}^{K} \left(\sum_{t=1}^{T^{(k)}-1} \gamma_t^{(k)}(j)\right)$?

Enfin, comment interprétez-vous la somme $Y_4 = \sum_{k=1}^{K} \left(\sum_{t \in U_l^{(k)}} \gamma_t^{(k)}(j)\right)$?

Servez-vous de vos interprétations précédentes pour déduire en fonction de Y_1, Y_2, Y_3 et Y_4 comment procéder pour optimiser λ en calculant les paramètres du MMC optimisé $\bar{\lambda} = (\{\bar{a}_{ij}\}, \{\bar{b}_j(l)\}, \{\bar{\pi}(i)\})$

Soit ρ l'ordre de grandeur de la complexité de l'algorithme de Baum-Welch mono-séquence calculé en dernière question du devoir 3.3. Déterminez en fonction de ρ et de K l'ordre de grandeur de la complexité de l'algorithme de Baum-Welch pour l'apprentissage de K séquences.

Au regard de votre réponse précédente, qu'adviendra-t-il de cette complexité pour de grandes valeurs de K ?

Quelle solution naturelle est alors envisageable pour de grandes valeurs de K ?

Devoir 3.5. *Il vous est demandé :*

1. *De proposer le pseudo-code de la version multi-séquences de l'algorithme Baum-Welch en vous appuyant sur vos réponses précédentes.*
2. *D'écrire un programme en langage C qui charge les paramètres d'un MMC* $\lambda = (A, B, \pi)$ *depuis un fichier texte suivant le formatage de votre choix et qui lit un entier* K *puis* K *séquences* $\{O^{(1)}, O^{(2)}, \ldots, O^{(K)}\}$ *de longueurs éventuellement différentes au clavier, avant de ranger dans un autre fichier texte le MMC* $\bar{\lambda} = (\bar{A}, \bar{B}, \bar{\pi})$ *suivant le principe de votre pseudo-code. Vous supposerez que l'unique condition d'arrêt ici est celle relative au nombre d'itérations. Vous fixerez* $MaxIter = 100$ *et* $\bar{t} = 1$.
3. *D'utiliser votre programme pour calculer* $\bar{\lambda}_1 = (\bar{A}_1, \bar{B}_1, \bar{\pi}_1)$ *où* λ_1 *est le MMC dont vous avez fixé les paramètres à l'exercice 1.1 avec les 3 séquences d'entraînement suivantes :* $O^{(1)} = (Visionner\ Dormir\ Danser)$, $O^{(2)} = (Manger\ Manger\ Dormir\ Manger)$ *et* $O^{(3)} = (Danser\ Jouer\ Dormir\ Danser)$.
4. *D'utiliser votre programme pour calculer* $\bar{\lambda}_2 = (\bar{A}_2, \bar{B}_2, \bar{\pi}_2)$ *où* λ_2 *est le MMC dont vous avez fixé les paramètres à l'exercice 1.2 avec les 2 séquences d'entraînement suivantes :* $O^{(1)} = (Prier\ Boire\ Boire\ Dormir\ Danser)$ *et* $O^{(2)} = (Dormir\ Boire\ Prier\ Prier)$.
5. *D'utiliser votre programme pour calculer* $\bar{\lambda}_3 = (\bar{A}_3, \bar{B}_3, \bar{\pi}_3)$ *où* λ_3 *est le MMC de l'exemple 1.6 avec les 6 séquences d'entraînement suivantes :* $O^{(1)} = (xxxyxxy)$, $O^{(2)} = (xxyy)$, $O^{(3)} = (xxxxyy)$, $O^{(4)} = (yxx)$, $O^{(5)} = (xxyyxxy)$ *et* $O^{(6)} = (xxxy)$. *Vous mesurerez le temps d'exécution pour ce cas particulier.*
6. *D'utiliser le programme écrit dans le devoir 2.1 pour calculer* $Pr[O^{(k)}|\lambda]$ *et* $Pr[O^{(k)}|\bar{\lambda}]$ *avec chacun des MMC et chacune des séquences d'entraînement précédents. Que remarquez-vous ?*

3.5 Baum-Welch multi-séquences en parallèle

La raison d'être d'une version parallèle de l'algorithme de Baum-Welch multi-séquences réside dans sa complexité explosive lorsque K est très grand. Comment donc réaliser cette parallélisation sans toutefois altérer les résultats ?

La remarque fondamentale ici est qu'à chaque itération de l'algorithme, les expressions Y_1, Y_2, Y_3 et Y_4 définies dans la Section 3.4 ont toutes en commun le constructeur de somme $\sum_{k=1}^{K}$. Si l'on dispose de P processeurs organisés dans un réseau en étoile, ce constructeur de somme peut être éclaté en P constructeurs de sommes partielles, chaque somme partielle étant affectée à un processeur. Le processeur central se chargera à la fin de collecter les résultats des processeurs périphériques pour calculer les valeurs de Y_1, Y_2, Y_3 et Y_4.

Considérez un réseau en étoile composé de P processeurs numérotés de 1 à P, le processeur central étant le processeur 1. Considérez aussi l'ensemble $O = \{O^{(1)}, O^{(2)}, \ldots, O^{(K)}\}$ composé de K séquences, chaque séquence d'index k étant de longueur $T^{(k)}$. L'analyse précédente permet de paralléliser l'algorithme [2] ainsi qu'il suit :

1. Le processeur 1 partitionne l'ensemble $I = \{1, 2, \ldots, K\}$ des index des séquences en P partitions $V_1, V_2, \ldots, V_P$. En d'autres termes : $I = \left(\cup_{p=1}^{P} V_p\right)$ et $\left(\cap_{p=1}^{P} V_p\right) = \emptyset$. Pour plus defficacité, il est souhaitable que les partitions soient de même taille lorsque cela est possible.
2. Le processeur 1 envoie à chaque processeur d'index $p = 2, 3, \ldots, P$ la valeur de $MaxIter$ puis les séquences $O^{(k)}$ pour chaque index $k \in V_p$.

3. Chaque processeur d'index $p = 1, 2, \ldots, P$ initialise sa variable locale $Iter$ à 0.
4. Le processeur 1 envoie à chaque processeur d'index $p = 2, 3, \ldots, P$ le modèle λ
5. Chaque processeur d'index $p = 1, 2, \ldots, P$ utilise λ pour calculer les valeurs partielles $Y_1^{(p)}, Y_2^{(p)}, Y_3^{(p)}$ et $Y_4^{(p)}$ ci-dessous puis incrémente sa variable locale $Iter$. (**ces calculs sont réalisés en parallèle**) :

$$Y_1^{(p)} = \sum_{k \in V_p} \gamma_1^{(k)}(i) \qquad Y_2^{(p)} = \sum_{k \in V_p} \left(\sum_{t=1}^{T^{(k)}-1} \xi_t^{(k)}(i,j) \right)$$

$$Y_3^{(p)} = \sum_{k \in V_p} \left(\sum_{t=1}^{T^{(k)}-1} \gamma_t^{(k)}(j) \right) \qquad Y_4^{(p)} = \sum_{k \in V_p} \left(\sum_{t \in U_l^{(k)}} \gamma_t^{(k)}(j) \right)$$

6. Chaque processeur d'index $p = 2, 3, \ldots, P$ envoie ses résultats partiels $Y_1^{(p)}, Y_2^{(p)}, Y_3^{(p)}$ et $Y_4^{(p)}$ au processeur 1 puis teste si $Iter > MaxIter$. Si c'est le cas, le processeur p arrête de fonctionner.
7. Le processeur 1 calcule alors les valeurs de Y_1, Y_2, Y_3 et Y_4 de la manière suivante :

$$Y_1 = \left(\sum_{p=1}^{P} Y_1^{(p)} \right) \qquad Y_2 = \left(\sum_{p=1}^{P} Y_2^{(p)} \right)$$

$$Y_3 = \left(\sum_{p=1}^{P} Y_3^{(p)} \right) \qquad Y_4 = \left(\sum_{p=1}^{P} Y_4^{(p)} \right)$$

8. Le processeur 1 calcule les paramètres du MMC optimisé $\bar{\lambda} = (\{\bar{a}_{ij}\}, \{\bar{b}_j(l)\}, \{\bar{\pi}(i)\})$ en fonction des valeurs de Y_1, Y_2, Y_3 et Y_4.
9. Le processeur 1 teste si $Iter > MaxIter$, si c'est le cas il range le résultat et arrête de fonctionner. Dans le cas contraire, il fait $\lambda = \bar{\lambda}$ et on retourne à l'étape 4.

Devoir 3.6. *Il vous est demandé :*

1. *D'évaluer de manière théorique l'accélération du principe parallèle précédent avec le principe séquentiel de la Section 3.4.*
2. *D'écrire un programme parallèle en langage C utilisant P processeurs qui charge les paramètres d'un MMC $\lambda = (A, B, \pi)$ depuis un fichier texte suivant le formatage de votre choix et qui lit un entier K puis K séquences $\{O^{(1)}, O^{(2)}, \ldots, O^{(K)}\}$ de longueurs éventuellement différentes au clavier, avant de ranger dans un autre fichier texte le MMC $\bar{\lambda} = (\bar{A}, \bar{B}, \bar{\pi})$ **suivant le principe parallèle précédent**. Vous utiliserez la bibliothèque '`mpi.h`'. Vous supposerez aussi que l'unique condition d'arrêt ici est celle relative au nombre d'itérations. Enfin, vous fixerez $MaxIter = 100$ et $\bar{t} = 1$.*
3. *De tester votre programme avec 3 processeurs pour calculer $\bar{\lambda}_3 = (\bar{A}_3, \bar{B}_3, \bar{\pi}_3)$ où λ_3 est le MMC de l'exemple 1.6 avec les 6 séquences d'entraînement suivantes : $O^{(1)} = (xxxyxxy)$, $O^{(2)} = (xxyy)$, $O^{(3)} = (xxxxyy)$, $O^{(4)} = (yxx)$, $O^{(5)} = (xxyyxxx)$ et $O^{(6)} = (xxxy)$. Vous mesurerez le temps d'exécution de votre programme.*
4. *Comparez ce temps avec celui obtenu par la version séquentielle à la question 5 du devoir 3.5. Que remarquez-vous ?*

3.6 Convergence de l'algorithme de Baum-Welch

Comme cela a déjà été vu dans ce chapitre, l'algorithme de Baum-Welch permet d'entraîner un modèle initial λ avec une séquence ou plusieurs séquences afin d'obtenir un modèle final $\bar{\lambda}$ tel la probabilité d'observer la ou les séquences d'entraînement sachant $\bar{\lambda}$ soit optimale. Cependant l'algorithme de Baum-Welch ne garanti pas d'atteindre l'optimum global, c'est-à-dire le modèle final permettant d'obtenir la meilleure probabilité d'observation des séquences d'entraînement, peu importe le modèle initial. La seule certitude est celle de s'approcher de optimum local, c'est-à-dire le modèle final permettant d'obtenir la meilleure probabilité d'observation des séquences d'entraînement, sachant que le modèle initial est λ. Tout dépend donc du modèle initial.

3.6.1 Influence du modèle initial

Pour un même ensemble de séquences, des modèles initiaux distincts aboutiront à des modèles finaux distincts et donc, à des probabilités distinctes. Ainsi, il ne suffit pas seulement d'appliquer l'algorithme de Baum-Welch pour obtenir '*le bon modèle final*' (optimum global), mais il faut surtout choisir le '*bon modèle initial*'. En résumé :

λ mal paramétré $\implies$ $\bar{\lambda}$ éloigné de l'optimum global
λ bien paramétré $\implies$ $\bar{\lambda}$ proche de l'optimum global

Étant donné un ensemble de séquences O, le choix du modèle initial λ reste à ce jour un problème ouvert et délicat car la convergence vers l'optimum global en dépend. En effet :

1. $\bar{\lambda}$ héritera inévitablement de nombreuses propriétés de son ancêtre λ. Par exemple, toute probabilité nulle dans λ restera nulle dans $\bar{\lambda}$, peu importe le nombre d'itérations de l'algorithme de Baum-Welch. Par contre, une probabilité non nulle dans λ peut devenir nulle dans $\bar{\lambda}$. C'est pour cette raison qu'il est conseillé d'éviter de mettre des probabilités nulles dans le modèle initial quand on n'en a pas la certitude, au risque de biaiser les résultats finaux. Vaudrait mieux inclure uniquement des probabilités non nulles λ, l'algorithme de Baum-Welch annulera celles pour lesquelles cela est nécessaire.
2. La durée de l'entraînement dépend aussi λ. Si λ est bien paramétré, la convergence aura lieu très rapidement car on passera par peu de MMC intermédiaires et tous seront pertinents (chemin de convergence optimal). Par contre si λ est mal paramétré, l'algorithme de Baum-Welch transitera par de nombreux MMC intermédiaires non pertinents, rallongeant énormément le temps d'entraînement sans pour autant s'approcher de l'optimum global.

3.6.2 Choix du modèle initial

Dans la pratique, l'utilisateur dispose seulement d'un ensemble de séquences d'entraînement et il doit fixer un modèle initial avant de l'entraîner. Dans ces conditions, les choix du nombre M de symboles et de l'ensemble $S = \{s_1, \ldots, s_M\}$ des symboles sont évidents car il peut directement les extraire des séquences d'entraînement. Les difficultés relatives à la construction de λ sont plutôt liées aux choix des autres paramètres de λ :

1. Le choix du nombre N d'états est par contre très difficile à opérer. Cela est lié au fait que seules les séquences d'entraînement (ne contenant aucune information sur les états) sont utilisées pour construire le modèle initial. De plus, comme les états sont supposés être cachés, aucun sens ne leurs est généralement attribué. Et pourtant, '*états cachés*' ne voudrait pas dire '*états inconnus*' ou '*états dépourvus de sens*'. Ce choix est pourtant évident lorsqu'un sens est attribué aux état. C'est le cas dans tous les exercices liés aux exemples 1.1 et 1.2 de ce cours dans lesquels les états ont un sens clair, bien-qu'ils soient cachés lorsqu'on réalise l'entraînement. Quand aucun sens clair n'est donné aux états, il est compliqué de choisir formellement le nombre d'états. Dans la pratique, le nombre N d'états est généralement fixé aléatoirement. Ce qui est dangereux car :
   N trop petit $\Rightarrow$ Peu de points d'observations $\Rightarrow$ Vue d'ensemble limitée
   N trop grand $\Rightarrow$ Trop de points d'observations $\Rightarrow$ Vue d'ensemble biaisée

2. Pour les mêmes raisons, le choix de l'ensemble $E = \{e_1, \ldots, e_N\}$ des états est tout autant difficile. Dans la pratique, une fois que N est fixé aléatoirement, l'ensemble E des états est très souvent réduit à $E = \{1, 2, \ldots, N\}$ sans qu'aucun sens propre ne leur soit attribué. Cette manière de procéder est problématique car elle ne facilite pas l'interprétation des données et des résultats. En effet, il était aisé pour vous de répondre aux nombreuses questions de ce cours relatives aux exemples 1.1 et 1.2 car le sens attribué aux états permettaient une interprétation fluide. cela n'aurait pas été possible si les états n'avaient aucun sens.
3. De manière analogue à ce qui précède, les choix des contenus des distributions A, B et π du modèle initial sont compliqués lorsque seules les séquences d'entraînement sont utilisées, puisqu'elles ne contiennent aucune information relative aux états. En général, même si A et π ne sont pas très bien choisis, l'algorithme de Baum-Welch fini toujours par s'approcher de l'optimum local. Mais si B n'est pas bien choisi, même la proximité avec l'optimum local n'est plus garantie. Dans la pratique, ces distributions sont parfois initialisées aléatoirement, ce qui est à proscrire si on souhaite atteindre l'optimum local. Néanmoins, il existe des approches '*Bayésiennes*' formelles permettant de définir ces distributions en se rapprochant autant que faire se peut d'un modèle initial pouvant conduire à l'optimum local. Malheureusement, cette solution n'est efficace que si le nombre et l'ensemble des états sont bien choisis. Sa mise en oeuvre nécessite généralement plusieurs tests avec diverses valeurs de N afin de conserver celle permettant d'obtenir la probabilité la plus élevée.

3.6.3 Convergence vers l'optimum global

L'analyse précédente permet de se rendre compte que si l'on souhaite que l'algorithme de Baum-Welch converge vers l'optimum global, il faut que tous les paramètres du modèle initial reflètent statistiquement le contenu des données d'entraînement. Pour y parvenir, il est donc nécessaire :

1. De donner un sens clair aux états. Cela permettra logiquement de choisir le bon nombre d'états et de faire des interprétations aisées.
2. De construire le modèle initial en se servant de chaînes de Markov d'entraînement et non pas seulement des séquences d'entraînement. En effet, contrairement aux séquences qui ne contiennent que les informations sur la séquentialité des symboles (insuffisantes pour initialiser A, B et π), les chaînes de Markov contiennent Les informations sur :

(a) La séquentialité des états permettant d'initialiser A.

(b) Le symbole observé dans chaque état permettant d'initialiser B.

(c) Les états initiaux permettant d'initialiser π.

Devoir 3.7. *Soit $E = \{e_1, \ldots, e_N\}$ un ensemble d'états et $S = \{s_1, \ldots, s_M\}$ un ensemble de symboles. En supposant que vous disposez d'un ensemble $\Delta = \{\delta^{(1)}, \delta^{(2)}, \ldots, \delta^{(K)}\}$ contenant K chaînes de Markov, chaque chaîne de Markov $\delta^{(k)}$ réalisant l'observation de la séquence de symboles $O^{(k)}$.*

1. *Expliquez comment vous pouvez capturer le contenu statistique de Δ pour initialiser les paramètres A, B et π du MMC initial λ associé à l'ensemble de séquences $O = \{O^{(1)}, O^{(2)}, \ldots, O^{(K)}\}$.*
2. *Utilisez le principe que vous venez de proposer pour construire un nouveau MMC $\lambda'_3 = (A'_3, B'_3, \pi'_3)$ qui doit capturer le contenu statistique de l'ensemble $\Delta = \{\delta^{(1)}, \ldots, \delta^{(6)}\}$ contenant les 6 chaînes de Markov ci-dessous, chaque chaîne de Markov $\delta^{(k)}$ réalisant l'observation de la séquence $O^{(k)}$ utilisée à la question 5 du devoir 3.5.*
3. *Utilisez le programme écrit à la question 2 du devoir 3.5 pour calculer le modèle $\bar{\lambda}'_3 = (\bar{A}'_3, \bar{B}'_3, \bar{\pi}'_3)$ en considérant l'ensemble de séquences d'entraînement $O = \{O^{(1)}, \ldots, O^{(6)}\}$.*
4. *Utilisez le programme écrit dans le devoir 2.1 pour calculer $Pr[O^{(k)}|\bar{\lambda}_3]$ et $Pr[O^{(k)}|\bar{\lambda}'_3]$ pour chaque $O^{(k)}$ où $\bar{\lambda}_3$ est le MMC obtenu à la question 5 du devoir 3.5. Que remarquez-vous ?*

$$\delta^{(1)}: \begin{array}{ccccccccccccc} x & & x & & x & & y & & x & & x & & y \\ \uparrow & & \uparrow & & \uparrow & & \uparrow & & \uparrow & & \uparrow & & \uparrow \\ b & \rightarrow & b & \rightarrow & a & \rightarrow & d & \rightarrow & b & \rightarrow & b & \rightarrow & c \end{array} \qquad \delta^{(2)}: \begin{array}{ccccccc} x & & x & & y & & y \\ \uparrow & & \uparrow & & \uparrow & & \uparrow \\ b & \rightarrow & a & \rightarrow & c & \rightarrow & c \end{array}$$

$$\delta^{(3)}: \begin{array}{ccccccccccc} x & & x & & x & & x & & y & & y \\ \uparrow & & \uparrow & & \uparrow & & \uparrow & & \uparrow & & \uparrow \\ b & \rightarrow & b & \rightarrow & b & \rightarrow & a & \rightarrow & c & \rightarrow & c \end{array} \qquad \delta^{(4)}: \begin{array}{ccccc} y & & x & & x \\ \uparrow & & \uparrow & & \uparrow \\ d & \rightarrow & b & \rightarrow & a \end{array}$$

$$\delta^{(5)}: \begin{array}{ccccccccccccc} x & & x & & y & & y & & x & & x & & x \\ \uparrow & & \uparrow & & \uparrow & & \uparrow & & \uparrow & & \uparrow & & \uparrow \\ b & \rightarrow & a & \rightarrow & c & \rightarrow & d & \rightarrow & b & \rightarrow & b & \rightarrow & a \end{array} \qquad \delta^{(6)}: \begin{array}{ccccccc} x & & x & & x & & y \\ \uparrow & & \uparrow & & \uparrow & & \uparrow \\ b & \rightarrow & b & \rightarrow & a & \rightarrow & c \end{array}$$

Chapitre 4

Comparaison de deux MMCs

4.1 Intérêt

Le but de ce chapitre est de montrer comment procéder pour comparer deux MMCs en exhibant une mesure de similarité entre deux MMC. Mais avant de le faire, commençons par répondre à la question : *Quel intérêt y a-t-il à comparer deux MMCs* ?

Lorsqu'on fait de l'apprentissage par la modélisation, le modèle (Gaussien, réseau neuronal, MMC, etc.) capture certaines propriétés caractéristiques des données $\Rightarrow$ le modèle '*apprend*' ces propriétés. Ainsi, si l'on désire comparer deux jeux de données, une astuce simple consiste à comparer leurs modèles respectifs. Cette comparaison se fait par le canal de mesures de distances ou de similarités entre deux modèles. Cependant, connaissez-vous des mesures existantes de distances ou de similarités entre deux modèles Gaussiens ou deux réseaux neuronaux ?

Le deuxième atouts majeur des MMCs (le premier étant l'apprentissage d'un ensemble de séquences) est qu'il existe de nombreuses mesures de distances et de similarités qui ont à ce jour été proposées pour comparer deux MMCs. De nombreuses mesures comparent simplement les paramètres A, B et π par des métriques entre matrices ou entre vecteurs. Mais de telles mesures ne prennent pas l'aspect temporel d'un MMC. En effet, elles comparent les modèles tels qu'ils sont observés, et ne tiennent pas compte de leur comportement sur le long terme. Heureusement, une mesure de similarité qui prend en compte cet aspect temporel a été proposée en 2008 [3]. C'est cette mesure qui sera présentée dans ce cours. La présentation de cette mesure implique que la distribution stationnaire d'un MMC soit au préalable définie.

4.2 Distribution stationnaire d'un MMC

Afin d'analyser le comportement d'un MMC sur le long terme, on s'interroge ici sur le temps global passé par un MMC $\lambda = (A, B, \pi)$ dans chaque état. Si notre MMC est perçu comme un étudiant pouvant transiter entre diverses places (états) dans la salle de classe pour suivre (observer) les cours (symboles), alors nous voulons savoir le pourcentage de temps passé par cet étudiant dans sur chaque place (a-t-il des places préférées) ? A titre d'exemple :

Quel est selon-vous le pourcentage de temps global passé dans l'état B par le MMC de l'exemple 1.6 si vous exécutez manuellement et sur le long terme plusieurs transitions d'états en respectant les probabilités de transitions ? Pourquoi ?

Même question pour le pourcentage de temps global passé dans l'état A ? Pourquoi ?

Les réponses aux questions précédentes ne sont pas exactes. Pour évaluer de manière exacte ces proportions de temps, l'analyse suivante doit être préalablement menée :

Rappelez ce que représente l'élément d'indices (i, j) de la matrice A :

Si on note $A^k = \overbrace{A \times A \times \ldots \times A}^{k \text{ fois}}$, alors en vous basant sur le principe mathématique du calcul du produit matriciel, quelle interprétation faites-vous de l'élément d'indices (i, j) de la matrice $A^2 = A \times A$?

Même question pour l'élément d'indices (i, j) de la matrice $A^3 = A^2 \times A$?

Déduisez des deux questions précédentes le sens de l'élément d'indices (i, j) de la matrice $A^k = A^{k-1} \times A$?

Quel élément permet de distinguer les composantes de la colonne j de la matrice A^k ?

Cet élément distinctif a-t-il encore de l'importance lorsque $k \to +\infty$? Pourquoi ?

Que pouvez-vous dire des composantes de la colonne j de la matrice A^k lorsque $k \to +\infty$? Quelle(s) interprétation(s) en faites-vous ?

Que pouvez-vous en déduire sur les lignes de la matrice A^k lorsque $k \to +\infty$?

Dans ces conditions, soit φ une ligne quelconque de la matrice A^k lorsque $k \to +\infty$. Que vaut le produit $\varphi.A$?

Quelle propriété fondamentale du vecteur φ en déduisez-vous ?

Définition 4.1. *Un vecteur φ est une distribution stationnaire d'un MMC $\lambda = (A, B, \pi)$ si et seulement si $\left(\sum_{i=1}^{N} \varphi_i = 1\right)$ et $(\varphi.A =$ $)$. La i^e composante de φ est généralement considérée comme le pourcentage de temps passé par λ dans l'état e_i. φ s'obtient en prenant n'importe quelle ligne de la matrice A^k lorsque $k \to +\infty$.*

Remarque 4.1. *Afin de comparer deux MMCs, il est fortement conseillé d'utiliser les distributions stationnaires des deux MMCs au lieu d'utiliser leur matrices de transitions d'états car les distributions stationnaires traduisent le transitions d'états sur le long terme.*

Devoir 4.1. *Il vous est demandé :*

1. *D'écrire un programme en langage C qui charge les paramètres d'un MMC $\lambda = (A, B, \pi)$ depuis un fichier texte suivant le formatage de votre choix puis range dans un autre fichier texte la distribution stationnaire φ de λ. Vous prendrez la valeur $k = 100$.*
2. *D'utiliser votre programme pour calculer les distributions stationnaires des MMCs $\bar{\lambda}_1$, $\bar{\lambda}_2$ et $\bar{\lambda}_3$ obtenus au devoir 3.4.*

4.3 Taux de similarité entre deux MMCs

Considérez deux MMCs λ et λ' ayant respectivement N et N' états, ayant les distributions stationnaires respectives φ et φ', et observant le même ensemble $S = \{s_1, \ldots, s_M\}$ de symboles. L'objectif est de mesurez à quel taux (en %) λ et λ' peuvent observer la même séquence, peu importent les chemins qu'ils empruntent respectivement pour réaliser cette observation identique. Il ne s'agit donc pas de mesurer la similarité visuelle entre λ et λ'. Pour y parvenir, il faut au préalable définir le taux de correspondance $q(i, i')$ entre l'état i de λ et i' de λ'. Plus formellement, $q(i, i')$ est la probabilité que λ et λ' observent le même symbole sachant que λ est dans son état i et λ' est dans son état i'. Pour que cela soit possible, il faut s'assurer :

1. Que λ est dans son état i (sur le long terme) $\Rightarrow \varphi_i$
2. Que λ' est dans son état i' (sur le long terme) $\Rightarrow \varphi'_{i'}$
3. Enfin que λ et λ' observent le même symbole sachant que λ est dans son état i et λ' est dans son état $i' \Rightarrow S(i, i')$ qu'il faut maintenant évaluer.

Il y a plusieurs manières d'évaluer $S(i, i')$, mais l'idée de base est de parcourir tous les symboles en évaluant pour chaque symbole s_k la probabilité conjointe que λ observe s_k étant dans son état i pendant qu'au même instant, λ' observe s_j étant dans son état i'. On obtient la valeur $S_1(i, i')$ suivante :

$$S_1(i, i') = \sum_{k=1}^{M} b_i(k) \times b'_{i'}(k) \tag{4.1}$$

Selon-vous, la valeur $S_1(i, i')$ est-elle une mesure de probabilité ? Justifiez votre réponse.

Dans le cas où $S_1(i, i')$ n'est pas une mesure de probabilité, que représente t-elle alors ?

Dans l'article où cette mesure de similarité est proposée, l'auteur propose (entre autre possibilités) de passer cette somme par un logarithme puis par un exponentiel, en la pondérant éventuellement par n'importe quel entier négatif comme indiqué dans la formule suivante :

$$S_2(i, i') = e^{-w.\log(S_1(i,i'))} \tag{4.2}$$

Selon-vous, y a-t-il une différence réelle entre S_1 et S_2 ? Justifiez votre réponse.

Dans le cas où il n'y a pas de réelle différence, dans quel but les auteurs de [3] préfèrent-t-ils le calcul de S_2 à celui de S_1 ?

Toute cette analyse nous permet donc de calculer $q(i, i')$ suivant le principe du nombre de cas favorables sur le nombres de cas possibles ainsi qu'il suit :

$$q(i,i') = \frac{\varphi_i \varphi'_{i'} S_2(i,i')}{\sum_{\forall j} \sum_{\forall j'} \left(\varphi_j \varphi'_{j'} S_2(j,j')\right)} \tag{4.3}$$

Il est question dans un premier temps de calculer toutes les composantes de la matrice $Q = \{q(i,i'), \forall i, \forall i'\}$. Cette matrice est appelée ***matrice de correspondances*** entre λ et λ'. Ensuite, il faut utiliser le contenu de la matrice Q pour mesurer le taux de similarité entre λ et λ'.

Comment interprétez-vous le fait d'avoir $q(i,i') = 0$? Justifiez votre réponse.

De même, comment interprétez-vous le fait d'avoir $q(i,i') = 1$? Justifiez votre réponse.

Déduisez de ce qui précède que pour que le taux de similarité entre λ et λ' soit élevé, il faut que les deux conditions C_1 et C_2 suivantes soient vérifiées simultanément :

- C_1 : Avoir uniquement des valeurs très proches de 0 ou de 1 dans Q
- C_2 : Avoir plus de valeurs très proches de 1 que de 0 dans Q

On peut donc conclure que pour évaluer le taux de similarité entre λ et λ', il faut trouver une fonction F qui analyse le contenu de Q et retourne une valeur $F(Q) = Sim(\lambda, \lambda')$ dont la proximité avec 1 est proportionnelle au degré de vérification des conditions C_1 et C_2 précédentes.

Dans ces conditions, à quel moment a-t-on $F(Q) = 1$? Justifiez votre réponse.

Si l'on décide par exemple de fixer $F_1(Q) = \frac{1}{|Q|}\left(\sum_{\forall i} \sum_{\forall i'} q(i,i')\right)$, estimez-vous que la fonction F_1 est adéquate pour évaluer la similarité entre λ et λ' ? Justifiez votre réponse.

Même question pour la fonction $F_2(Q) = \left(1 - \frac{|Q|-|Z|}{|Q|}\right) \times \left(\frac{1}{|Z|} \sum_{\forall (i,i') \in Z} q(i,i')\right)$ en considérant que $Z = \{(i,i') | q(i,i') \neq 0, \forall i, \forall i'\}$

En expliquant votre choix, proposez une fonction F_3 qui serait encore meilleure que F_1 et F_2 pour mesurer la similarité entre λ et λ'.

Dans la littérature, il existe une fonction qui '*essaye*' de retourner une valeur dont la proximité avec 1 est proportionnelle au degré de vérification des conditions C_1 et C_2, mais elle ne s'applique pas à une matrice Q mais plutôt à un vecteur x. Cette fonction notée G est connue sous le nom d'***Index Gini normalisé*** dont l'expression est donnée dans l'équation 4.4. Elle vérifie que si x est le vecteur nul, alors $G(x) = 0$ et que $G(x) = 1$ si toutes les composantes de x sont égales à 1. Dans l'équation 4.4 :

1. m est le nombre de composantes de x
2. $||x||_1$ est la somme des composantes de x
3. $x_{(k)}$ est le k^e plus petit élément de x tel que $x_{(1)} \leq x_{(2)} \leq \ldots \leq x_{(m)}$

$$G(x) = \frac{m}{m-1} - 2\sum_{k=1}^{m} \frac{x_{(k)}}{||x||_1}\left(\frac{m-k+\frac{1}{2}}{m-1}\right) \tag{4.4}$$

Pour la fonction $G(x)$ s'appliquant à un vecteur x à notre contexte dans lequel nous avons une matrice Q, il suffit de calculer la moyenne des index Gini normalisés des vecteurs colonnes puis des vecteurs lignes de Q comme cela est finalement exprimé dans l'équation 4.5.

$$Sim(\lambda, \lambda') = \frac{1}{2}\left[\frac{1}{N}\sum_{j=1}^{N} G(ligne_j) + \frac{1}{N'}\sum_{k=1}^{N'} G(colonne_k)\right] \tag{4.5}$$

Remarque 4.2. *Dans la pratique, on constate que l'index Gini normalisé n'est pas la fonction idéale pour le calcul du taux de similarité entre deux MMCs. En effet, l'équation 4.5 ne marche pas très bien pour des MMCs qui sont visuellement similaires. Dans cas là, une simple mesure du coefficient de corrélation (notamment celui de Pearson [4]) peut être utilisée.*

Devoir 4.2. *Il vous est demandé :*

1. *D'écrire un programme en langage C qui charge les paramètres de deux MMCs $\lambda = (A, B, \pi)$ et $\lambda' = (A', B', \pi')$ depuis un fichier texte suivant le formatage de votre choix puis range dans un autre fichier texte la matrice de correspondances entre λ et λ', ainsi que leur taux de similarité $Sim(\lambda, \lambda')$. Vous prendrez la valeur $k = 100$ pour le calcul des distributions stationnaires.*
2. *D'utiliser votre programme pour calculer $Sim(\lambda_1, \bar{\lambda}_1)$, $Sim(\lambda_2, \bar{\lambda}_2)$ et $Sim(\lambda_3, \bar{\lambda}_3)$ en considérant les MMCs du devoir 3.5.*

Bibliographie

[1] Lawrence R. Rabiner, *A tutorial on hidden Markov models and selected applications in speech recognition.* Proceedings of the IEEE, vol. **77**(2), pp. 257-286 (IEEE,1989).

[2] M. Anikeev and O. Makarevich, *Parallel Implementation of Baum-Welch Algorithm.* Proc. Workshop on Computer Science and Information Technologies (CSIT'06), vol. **1**, 2006, Karlsruhe, Germany, pp. 197-200.

[3] Sayed Mohammad Ebrahim Sahraeian and Byung-Jun Yoon, *A novel low-complexity HMM similarity measure.* Signal Processing Letters, vol. **18**(2), pp. 87-90 (IEEE,2011).

[4] G. Hall, *Pearson's correlation coefficient.* Url, *http ://www.hep.ph.ic.ac.uk/∼hallg/UG_ 2015/Pearsons.p* (2015)