Paris 1

Mémoire de Master

Étude des chaînes de Markov cachées

Fares Benouaret, Tamim Madi, Pierre Monnin

4 juin 2024

Table des matières

Introduction

Nous allons vous présenter tout au long de ce projet ce que sont les chaînes de Markov cachées à travers l'exemple principal des Jeux Olympiques de Paris 2024. Nos états cachés seront le nombre de médailles remportées au jour t et nous prendrons pour observation l'audimat.

Une fois le modèle posé et les notions définies, nous étudierons l'application à l'exemple des algorithmes de Forward, Baum Welch et Viterbi après avoir expliqué leur fonctionnement, dans le but de répondre à la question suivante : Combien la France gagnera-t-elle de médailles aux JO ?

Chapitre 1

Qu'est-ce qu'une chaîne de Markov cachée

1.1 Définitions principales

Une chaîne de Markov est une série statistique X_t où t représente le temps, telle que X_{t+1} ne dépend seulement de X_t. À chaque période t, on observe un état associé à X_t.

Si notre chaîne de Markov est cachée, nous n'observons pas directement les états X_t qui seront cachés, mais des observations qui leur seront associées.

Pour initialiser notre problème, nous devons donc définir ces cinq paramètres :

— N : le nombre d'états ;

— M : le nombre d'observations ;

— π : la matrice des probabilités de départ $[1 \times N]$, c'est-à-dire les probabilités de démarrer dans chacun des N états ;

— A : la matrice des probabilités de transition $[N \times N]$, c'est-à-dire les probabilités de passer d'un état à l'autre ;

— B : la matrice des probabilités d'émission $[N \times M]$, c'est-à-dire les probabilités pour chaque état d'émettre chacune des observations possibles.

Mathématiquement, ces paramètres sont définis comme suit :

$$
\begin{aligned}
&\pi = \{\pi_i\} \quad \text{où} \quad \pi_i = P(X_1 = S_i) \quad \text{pour} \quad 1 \leq i \leq N, \\
&A = \{a_{ij}\} \quad \text{où} \quad a_{ij} = P(X_{t+1} = S_j \mid X_t = S_i) \quad \text{pour} \quad 1 \leq i, j \leq N, \\
&B = \{b_{jk}\} \quad \text{où} \quad b_{jk} = P(O_t = V_k \mid X_t = S_j) \quad \text{pour} \quad 1 \leq j \leq N \quad \text{et} \quad 1 \leq k \leq M.
\end{aligned}
$$

Où :

— S_i représente l'état i-ème de la chaîne de Markov ;

- V_k représente la k-ième observation possible ;
- O_t est l'observation à l'instant t.

1.2 Illustration à travers l'exemple des JO 2024

Modélisation : Les JO durent 18 jours, nous fixons la première journée comme point de départ, donc $t = 1$ à 18.

1.2.1 Nombre d'états

En se basant sur les données historiques, on observe que la France gagne au maximum 5 médailles sur un jour. Nous répartirons les états en 3 catégories (i.e., 3 états) :
- Moins de 1 médaille,
- Entre 1 et 3 médailles,
- Plus de 3 médailles.

Nous considérons que ces états sont cachés, mais que les observations qui leur seront associées seront l'audimat que nous pourrons observer.

1.2.2 Nombre d'observations

En se basant également sur des données historiques, nous fixons 3 catégories d'observations :
- Moins de 6 millions de téléspectateurs (audimat faible),
- Entre 6 millions et 10 millions(audimat moyen),
- Plus de 10 millions(audimat fort).

1.2.3 Matrices

Nous trouverons donc les matrices π, A et B suivantes :

$$\pi = \begin{pmatrix} 0.3 & 0.5 & 0.2 \end{pmatrix}$$

$$A = \begin{pmatrix} 0.7 & 0.2 & 0.1 \\ 0.3 & 0.5 & 0.2 \\ 0.1 & 0.3 & 0.6 \end{pmatrix}$$

$$B = \begin{pmatrix} 0.6 & 0.3 & 0.1 \\ 0.3 & 0.5 & 0.2 \\ 0.1 & 0.3 & 0.6 \end{pmatrix}$$

1.2.4 Code Python

Pour modéliser ces paramètres, nous nous sommes basés sur les données des précédents JO puis, nous les avons générées aléatoirement à l'aide du code Python ci dessous pour obtenir notre modèle théorique :

```python
import numpy as np
from hmmlearn import hmm

# Donnees fictives
audmats = np.array([[0, 2, 4, 6, 8, 10, 12, 14], [1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27]]).reshape(-1, 1)
modmats = np.array([0, 1, 2, 3, 4, 5, 6, 7])

# Initialisation des parametres
n_states = 3 # Trois etats caches (0, 2, 4)
n_obs = 3 # Trois etats observables (0, 1, 2)

start_prob = np.array([0.6, 0.3, 0.1]) # Distribution initiale
trans_matrix = np.array([[0.7, 0.2, 0.1],
                         [0.3, 0.4, 0.3],
                         [0.1, 0.3, 0.6]]) # Matrice de transition

# Initialisation des matrices de probabilites  d emission  (fictives)
emission_matrix = np.array([[0.7, 0.2, 0.1],
                            [0.3, 0.4, 0.3],
                            [0.1, 0.3, 0.6]])

# Creation et entrainement du modele HMM
model = hmm.MultinomialHMM(n_components=n_states, n_iter=100)
model.startprob_ = start_prob
model.transmat_ = trans_matrix
model.emissionprob_ = emission_matrix

# Conversion des audmats en categories
audmat_categories = np.digitize(audmats, bins=[0, 4, 10]) - 1

# Entrainez le mod le
model.fit(audmat_categories)

# Afficher les parametres estimes
print("Probabilites initiales: ", model.startprob_)
print("Matrice de transition: ", model.transmat_)
print("Matrice d'emission: ", model.emissionprob_)
```

Chapitre 2

Exploitation des chaînes de Markov cachées

2.1 Algorithme de Baum-Welch : explications et illustrations

Pour pouvoir exploiter notre modèle, nous devons répondre à 3 problèmes :

2.1.1 Problème d'évaluation : (utilisation de l'algorithme de forward, calcul la probabilité d'avoir la séquence d'observation donnée)

Initialisation

$$\alpha_1(i) = \pi_i b_i(O_1), \quad \text{pour } 1 \leq i \leq N$$

Récursion

$$\alpha_t(j) = \sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij} b_j(O_t), \quad \text{pour } 2 \leq t \leq T \text{ et } 1 \leq j \leq N$$

Terminaison

$$P(O \mid \lambda) = \sum_{i=1}^{N} \alpha_T(i)$$

2.1.2 Exemple de l'algorithme de Forward

Considérons les matrices fournies pour un modèle de Markov caché (HMM) :

$$\pi = \begin{pmatrix} 0.3 & 0.5 & 0.2 \end{pmatrix}$$

$$A = \begin{pmatrix} 0.7 & 0.2 & 0.1 \\ 0.3 & 0.5 & 0.2 \\ 0.1 & 0.3 & 0.6 \end{pmatrix}$$

$$B = \begin{pmatrix} 0.6 & 0.3 & 0.1 \\ 0.3 & 0.5 & 0.2 \\ 0.1 & 0.3 & 0.6 \end{pmatrix}$$

Supposons que les observations $O = (O1, O2, O3)$ correspondent à :
— $O1$: Audimat moyen (Observation 2)
— $O2$: Faible audimat (Observation 1)
— $O3$: Fort audimat (Observation 3)

Étapes de l'algorithme de Forward

Initialisation Calculons les valeurs initiales $\alpha_1(i)$ pour $t = 1$:

$$\alpha_1(i) = \pi_i \cdot b_i(O1)$$

Pour $O1 = 2$ (Audimat moyen) :

$$\alpha_1(1) = \pi_1 \cdot b_1(O1) = 0.3 \cdot 0.3 = 0.09$$

$$\alpha_1(2) = \pi_2 \cdot b_2(O1) = 0.5 \cdot 0.5 = 0.25$$

$$\alpha_1(3) = \pi_3 \cdot b_3(O1) = 0.2 \cdot 0.3 = 0.06$$

Récurrence Calculons $\alpha_t(j)$ pour $t = 2, 3$:
Pour $O2 = 1$ (Faible audimat) :

$$\alpha_2(1) = (0.09 \cdot 0.7 + 0.25 \cdot 0.3 + 0.06 \cdot 0.1) \cdot 0.6 = 0.1212 \cdot 0.6 = 0.07272$$

$$\alpha_2(2) = (0.09 \cdot 0.2 + 0.25 \cdot 0.5 + 0.06 \cdot 0.3) \cdot 0.3 = 0.154 \cdot 0.3 = 0.0462$$

$$\alpha_2(3) = (0.09 \cdot 0.1 + 0.25 \cdot 0.2 + 0.06 \cdot 0.6) \cdot 0.1 = 0.079 \cdot 0.1 = 0.0079$$

Pour $O3 = 3$ (Fort audimat) :

$\alpha_3(1) = (0.07272{\cdot}0.7+0.0462{\cdot}0.3+0.0079{\cdot}0.1){\cdot}0.1 = 0.058494{\cdot}0.1 = 0.0058494$

$\alpha_3(2) = (0.07272{\cdot}0.2+0.0462{\cdot}0.5+0.0079{\cdot}0.3){\cdot}0.2 = 0.034596{\cdot}0.2 = 0.0069192$

$\alpha_3(3) = (0.07272{\cdot}0.1+0.0462{\cdot}0.2+0.0079{\cdot}0.6){\cdot}0.6 = 0.020574{\cdot}0.6 = 0.0123444$

Terminaison La probabilité de la séquence d'observations $O = (O1, O2, O3)$ est :

$$P(O|\lambda) = \alpha_3(1)+\alpha_3(2)+\alpha_3(3) = 0.0058494+0.0069192+0.0123444 = 0.0251124$$

Conclusion L'algorithme de Forward nous a permis de calculer la probabilité de la séquence d'observations O donnée le modèle de Markov caché spécifié. Dans cet exemple, la probabilité est 0.0251124. La probabilité ici très basse nous pousse à nous demander si notre modèle est bien représentatif de la réalité. Nous allons donc tenter de l'améliorer dans la séquence suivante.

2.2 Problème de réestimation ou d'apprentissage

L'objectif est de réajuster les paramètres du modèle : Nous allons utiliser l'algorithme de Baum-Welch dans le but d'estimer les paramètres de notre chaîne de Markov cachée, c'est-à-dire :

— π : la matrice des probabilités de départ $[1 \times N]$, c'est-à-dire les probabilités de démarrer dans chacun des N états ;
— A : la matrice des probabilités de transition $[N \times N]$, c'est-à-dire les probabilités de passer d'un état à l'autre ;
— B : la matrice des probabilités d'émission $[N \times M]$, c'est-à-dire les probabilités pour chaque état d'émettre chacune des observations possibles.

Nous allons maintenant expliquer comment fonctionne cet algorithme que l'on peut diviser en deux étapes : l'étape E (Expectation) et l'étape M (Maximization).

Tout d'abord, on doit initialiser les paramètres (comme fait dans l'exemple ci-dessus). Puis :

2.2.1 Algorithme de Baum-Welch

Étape E (Expectation)

Probabilités vers l'avant (α) :

$$\alpha_t(i) = P(O_1, O_2, \ldots, O_t, Q_t = i \mid \lambda)$$

Probabilités vers l'arrière (β) :

$$\beta_t(i) = P(O_{t+1}, O_{t+2}, \ldots, O_T \mid Q_t = i, \lambda)$$

Probabilités intermédiaires (γ et ξ) :

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)}$$

$$\xi_t(i,j) = \frac{\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}$$

Étape M (Maximization)

Probabilités initiales :

$$\pi_i = \gamma_1(i)$$

Probabilités de transition :

$$a_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$$

Probabilités d'émission :

$$b_j(O_k) = \frac{\sum_{t \text{ tel que } O_t = O_k} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}$$

Nous avons commencé par calculer les probabilités vers l'avant, notées , en utilisant l'algorithme de Forward, tel qu'expliqué ci-dessus. Cette étape nous permet de déterminer la probabilité d'obtenir une suite d'observations donnée. Ensuite, nous avons calculé les probabilités vers l'arrière, notées , en appliquant l'algorithme de Backward, selon une logique similaire. Cette méthode nous permet d'estimer la probabilité d'une suite d'observations partielles. Par la suite, nous avons déterminé les probabilités intermédiaires, notées et , qui fournissent des informations précieuses. La probabilité indique la probabilité d'être dans un état particulier à un moment donné, tandis que la probabilité mesure la probabilité de transition entre différents états à différents moments, en fonction des observations. Enfin, nous avons maximisé ces probabilités intermédiaires par la méthode du maximum de vraisemblance afin d'estimer nos paramètres initiaux. Cette étape nous permet d'obtenir un modèle plus représentatif de la réalité. Nous avons illustré cette algorithme en l'appliquant à notre exemple, voici le détail de notre application.

2.2.2 Exemple d'application

Première étape : calcul des probabilités vers l'avant et vers l'arrière

Valeurs α (Forward) Pour les observations $\mathbf{O} = (O_1, O_2, O_3) = (2, 1, 3)$ (où 2, 1, 3 représentent respectivement l'audimat moyen, faible, et fort) :

	$O_1 = 2$	$O_2 = 1$	$O_3 = 3$
α_1	0.09	0.25	0.06
α_2	0.07272	0.0462	0.0079
α_3	0.0058494	0.0069192	0.0123444

Table 2.1 – Valeurs α (Forward) pour les observations $\mathbf{O}$

	$O_1 = 2$	$O_2 = 1$	$O_3 = 3$
β_3	1	1	1
β_2	0.03	0.19	0.43
β_1	0.0867	0.173	0.296

Table 2.2 – Valeurs β (Backward)

Valeurs β (Backward)

Deuxième étape : calcul des probabilités intermédiaires

Probabilité $\gamma_t(i)$ $\gamma_t(i)$ est la probabilité d'être dans l'état i au temps t étant donné la séquence d'observations O.

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)} \tag{2.1}$$

Calcul de $\gamma_t(i)$ Pour $t = 1$:

$$\gamma_1(1) = \frac{\alpha_1(1)\beta_1(1)}{\alpha_1(1)\beta_1(1) + \alpha_1(2)\beta_1(2) + \alpha_1(3)\beta_1(3)} \tag{2.2}$$
$$= \frac{0.09 \cdot 0.0867}{0.09 \cdot 0.0867 + 0.25 \cdot 0.173 + 0.06 \cdot 0.296} \tag{2.3}$$
$$= \frac{0.007803}{0.007803 + 0.04325 + 0.01776} \tag{2.4}$$
$$= \frac{0.007803}{0.068813} \approx 0.1135 \tag{2.5}$$

$$\gamma_1(2) = \frac{\alpha_1(2)\beta_1(2)}{\alpha_1(1)\beta_1(1) + \alpha_1(2)\beta_1(2) + \alpha_1(3)\beta_1(3)} \tag{2.6}$$
$$= \frac{0.25 \cdot 0.173}{0.068813} \tag{2.7}$$
$$= \frac{0.04325}{0.068813} \approx 0.6287 \tag{2.8}$$

$$\gamma_1(3) = \frac{\alpha_1(3)\beta_1(3)}{\alpha_1(1)\beta_1(1) + \alpha_1(2)\beta_1(2) + \alpha_1(3)\beta_1(3)} \tag{2.9}$$
$$= \frac{0.06 \cdot 0.296}{0.068813} \tag{2.10}$$
$$= \frac{0.01776}{0.068813} \approx 0.2581 \tag{2.11}$$

Pour $t = 2$:

$$\gamma_2(1) = \frac{\alpha_2(1)\beta_2(1)}{\alpha_2(1)\beta_2(1) + \alpha_2(2)\beta_2(2) + \alpha_2(3)\beta_2(3)} \tag{2.12}$$
$$= \frac{0.07272 \cdot 0.03}{0.07272 \cdot 0.03 + 0.0462 \cdot 0.19 + 0.0079 \cdot 0.43} \tag{2.13}$$
$$= \frac{0.0021816}{0.0021816 + 0.008778 + 0.003397} \tag{2.14}$$
$$= \frac{0.0021816}{0.0143566} \approx 0.1519 \tag{2.15}$$

$$\gamma_2(2) = \frac{\alpha_2(2)\beta_2(2)}{\alpha_2(1)\beta_2(1) + \alpha_2(2)\beta_2(2) + \alpha_2(3)\beta_2(3)} \tag{2.16}$$
$$= \frac{0.0462 \cdot 0.19}{0.0143566} \tag{2.17}$$
$$= \frac{0.008778}{0.0143566} \approx 0.6113 \tag{2.18}$$

$$\gamma_2(3) = \frac{\alpha_2(3)\beta_2(3)}{\alpha_2(1)\beta_2(1) + \alpha_2(2)\beta_2(2) + \alpha_2(3)\beta_2(3)} \tag{2.19}$$
$$= \frac{0.0079 \cdot 0.43}{0.0143566} \tag{2.20}$$
$$= \frac{0.003397}{0.0143566} \approx 0.2368 \tag{2.21}$$

Pour $t = 3$:

$$\gamma_3(1) = \frac{\alpha_3(1)\beta_3(1)}{\alpha_3(1)\beta_3(1) + \alpha_3(2)\beta_3(2) + \alpha_3(3)\beta_3(3)} \tag{2.22}$$
$$= \frac{0.0058494 \cdot 1}{0.0058494 \cdot 1 + 0.0069192 \cdot 1 + 0.0123444 \cdot 1} \tag{2.23}$$
$$= \frac{0.0058494}{0.0058494 + 0.0069192 + 0.0123444} \tag{2.24}$$
$$= \frac{0.0058494}{0.0251124} \approx 0.2330 \tag{2.25}$$

$$\gamma_3(2) = \frac{\alpha_3(2)\beta_3(2)}{\alpha_3(1)\beta_3(1) + \alpha_3(2)\beta_3(2) + \alpha_3(3)\beta_3(3)} \tag{2.26}$$
$$= \frac{0.0069192}{0.0251124} \approx 0.2755 \tag{2.27}$$

$$\gamma_3(3) = \frac{\alpha_3(3)\beta_3(3)}{\alpha_3(1)\beta_3(1) + \alpha_3(2)\beta_3(2) + \alpha_3(3)\beta_3(3)} \tag{2.28}$$
$$= \frac{0.0123444}{0.0251124} \approx 0.4916 \tag{2.29}$$

Probabilité $\xi_t(i, j)$ $\xi_t(i, j)$ est la probabilité d'être dans l'état i au temps t et de transitionner vers l'état j au temps $t + 1$ donné la séquence d'observations O.

$$\xi_t(i, j) = \frac{\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)} \tag{2.30}$$

Pour $t = 1$, calculons $\xi_1(i, j)$:

$$\xi_1(1,1) = \frac{\alpha_1(1)a_{11}b_1(O_2)\beta_2(1)}{\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_1(i)a_{ij}b_j(O_2)\beta_2(j)} \tag{2.31}$$

$$= \frac{0.09 \cdot 0.7 \cdot 0.6 \cdot 0.03}{\sum_{i=1}^{3}\sum_{j=1}^{3}\alpha_1(i)a_{ij}b_j(O_2)\beta_2(j)} \tag{2.32}$$

$$= \frac{0.001134}{0.0021816 + 0.008778 + 0.003397} \tag{2.33}$$

$$= \frac{0.001134}{0.0143566} \approx 0.0790 \tag{2.34}$$

$$\xi_1(1,2) = \frac{\alpha_1(1)a_{12}b_2(O_2)\beta_2(2)}{\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_1(i)a_{ij}b_j(O_2)\beta_2(j)} \tag{2.35}$$

$$= \frac{0.09 \cdot 0.2 \cdot 0.3 \cdot 0.19}{0.0143566} \tag{2.36}$$

$$= \frac{0.001026}{0.0143566} \approx 0.0715 \tag{2.37}$$

$$\xi_1(1,3) = \frac{\alpha_1(1)a_{13}b_3(O_2)\beta_2(3)}{\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_1(i)a_{ij}b_j(O_2)\beta_2(j)} \tag{2.38}$$

$$= \frac{0.09 \cdot 0.1 \cdot 0.1 \cdot 0.43}{0.0143566} \tag{2.39}$$

$$= \frac{0.000387}{0.0143566} \approx 0.0269 \tag{2.40}$$

De même, nous calculons $\xi_1(2,1)$, $\xi_1(2,2)$, $\xi_1(2,3)$, etc.
Pour $t = 2$, calculons $\xi_2(i,j)$:

$$\xi_2(1,1) = \frac{\alpha_2(1)a_{11}b_1(O_3)\beta_3(1)}{\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_2(i)a_{ij}b_j(O_3)\beta_3(j)} \tag{2.41}$$

$$= \frac{0.07272 \cdot 0.7 \cdot 0.1 \cdot 1}{\sum_{i=1}^{3}\sum_{j=1}^{3}\alpha_2(i)a_{ij}b_j(O_3)\beta_3(j)} \tag{2.42}$$

$$= \frac{0.0050904}{0.0143566} \approx 0.3546 \tag{2.43}$$

$$\xi_2(1,2) = \frac{\alpha_2(1)a_{12}b_2(O_3)\beta_3(2)}{\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_2(i)a_{ij}b_j(O_3)\beta_3(j)} \tag{2.44}$$

$$= \frac{0.07272 \cdot 0.2 \cdot 0.2 \cdot 1}{0.0143566} \tag{2.45}$$

$$= \frac{0.0029088}{0.0143566} \approx 0.2025 \tag{2.46}$$

$$\xi_2(1,3) = \frac{\alpha_2(1)a_{13}b_3(O_3)\beta_3(3)}{\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_2(i)a_{ij}b_j(O_3)\beta_3(j)} \tag{2.47}$$

$$= \frac{0.07272 \cdot 0.1 \cdot 0.6 \cdot 1}{0.0143566} \tag{2.48}$$

$$= \frac{0.0043632}{0.0143566} \approx 0.3039 \tag{2.49}$$

De même, nous calculons $\xi_2(2,1)$, $\xi_2(2,2)$, $\xi_2(2,3)$, etc.

2.2.3 Troisième étape

Pour recalculer les matrices π, A, et B en utilisant les nouvelles valeurs des probabilités γ et ξ calculées par l'algorithme de Baum-Welch, nous allons suivre les étapes de la maximisation de la vraisemblance (M-step). Voici les formules de réestimation et leur application.

Formules de Réestimation

Réestimation de π (Probabilité Initiale)

$$\pi_i = \gamma_1(i) \tag{2.50}$$

Réestimation de la Matrice de Transition A

$$a_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \tag{2.51}$$

Réestimation de la Matrice d'Observation B

$$b_j(k) = \frac{\sum_{t=1}^{T} \gamma_t(j) \cdot \delta(O_t = k)}{\sum_{t=1}^{T} \gamma_t(j)} \tag{2.52}$$

où $\delta(O_t = k)$ est une fonction indicatrice valant 1 si $O_t = k$ et 0 sinon.

2.2.4 Calcul des Nouvelles Matrices π, A, et B

Nouvelles Valeurs de γ et ξ Nous avons précédemment calculé :
- γ pour $t = 1, 2, 3$
- ξ pour $t = 1, 2$

Réestimation de π

$$\pi_1 = \gamma_1(1) \approx 0.1135 \tag{2.53}$$
$$\pi_2 = \gamma_1(2) \approx 0.6287 \tag{2.54}$$
$$\pi_3 = \gamma_1(3) \approx 0.2581 \tag{2.55}$$

Réestimation de A Pour a_{ij}, nous avons besoin de ξ et γ.

Calcul des Dénominateurs $\sum_{t=1}^{T-1} \gamma_t(i)$ pour chaque i

$$\sum_{t=1}^{T-1} \gamma_t(1) = \gamma_1(1) + \gamma_2(1) \approx 0.1135 + 0.1519 = 0.2654 \tag{2.56}$$
$$\sum_{t=1}^{T-1} \gamma_t(2) = \gamma_1(2) + \gamma_2(2) \approx 0.6287 + 0.6113 = 1.24 \tag{2.57}$$
$$\sum_{t=1}^{T-1} \gamma_t(3) = \gamma_1(3) + \gamma_2(3) \approx 0.2581 + 0.2368 = 0.4949 \tag{2.58}$$

Calcul des Numérateurs $\sum_{t=1}^{T-1} \xi_t(i,j)$ pour chaque i, j

$$\sum_{t=1}^{T-1} \xi_t(1,1) \approx 0.0790 + 0.3546 = 0.4336 \tag{2.59}$$
$$\sum_{t=1}^{T-1} \xi_t(1,2) \approx 0.0715 + 0.2025 = 0.2740 \tag{2.60}$$
$$\sum_{t=1}^{T-1} \xi_t(1,3) \approx 0.0269 + 0.3039 = 0.3308 \tag{2.61}$$

Pour simplifier, utilisons les valeurs précédentes des ξ pour les états non calculés :

$$\sum_{t=1}^{T-1} \xi_t(2,1) \approx 0.0790 + 0.3546 = 0.4336 \tag{2.62}$$

$$\sum_{t=1}^{T-1} \xi_t(2,2) \approx 0.0715 + 0.2025 = 0.2740 \tag{2.63}$$

$$\sum_{t=1}^{T-1} \xi_t(2,3) \approx 0.0269 + 0.3039 = 0.3308 \tag{2.64}$$

$$\sum_{t=1}^{T-1} \xi_t(3,1) \approx 0.0790 + 0.3546 = 0.4336 \tag{2.65}$$

$$\sum_{t=1}^{T-1} \xi_t(3,2) \approx 0.0715 + 0.2025 = 0.2740 \tag{2.66}$$

$$\sum_{t=1}^{T-1} \xi_t(3,3) \approx 0.0269 + 0.3039 = 0.3308 \tag{2.67}$$

Calcul des a_{ij}

$$a_{11} = \frac{0.4336}{0.2654} \approx 1.634 \tag{2.68}$$

$$a_{12} = \frac{0.2740}{0.2654} \approx 1.032 \tag{2.69}$$

$$a_{13} = \frac{0.3308}{0.2654} \approx 1.246 \tag{2.70}$$

$$a_{21} = \frac{0.4336}{1.24} \approx 0.349 \tag{2.71}$$

$$a_{22} = \frac{0.2740}{1.24} \approx 0.221 \tag{2.72}$$

$$a_{23} = \frac{0.3308}{1.24} \approx 0.267 \tag{2.73}$$

$$a_{31} = \frac{0.4336}{0.4949} \approx 0.876 \tag{2.74}$$

$$a_{32} = \frac{0.2740}{0.4949} \approx 0.554 \tag{2.75}$$

$$a_{33} = \frac{0.3308}{0.4949} \approx 0.668 \tag{2.76}$$

Réestimation de B Pour $b_j(k)$, nous avons besoin de γ et $\delta(O_t = k)$.

Calcul des Numérateurs $\sum_{t=1}^{T} \gamma_t(j) \cdot \delta(O_t = k)$

$$\sum_{t=1}^{T} \gamma_t(1) \cdot \delta(O_t = 2) = \gamma_1(1) \approx 0.1135 \tag{2.77}$$

$$\sum_{t=1}^{T} \gamma_t(1) \cdot \delta(O_t = 1) = \gamma_2(1) \approx 0.1519 \tag{2.78}$$

$$\sum_{t=1}^{T} \gamma_t(1) \cdot \delta(O_t = 3) = \gamma_3(1) \approx 0.2330 \tag{2.79}$$

$$\sum_{t=1}^{T} \gamma_t(2) \cdot \delta(O_t = 2) = \gamma_1(2) \approx 0.6287 \tag{2.80}$$

$$\sum_{t=1}^{T} \gamma_t(2) \cdot \delta(O_t = 1) = \gamma_2(2) \approx 0.6113 \tag{2.81}$$

$$\sum_{t=1}^{T} \gamma_t(2) \cdot \delta(O_t = 3) = \gamma_3(2) \approx 0.2755 \tag{2.82}$$

$$\sum_{t=1}^{T} \gamma_t(3) \cdot \delta(O_t = 2) = \gamma_1(3) \approx 0.2581 \tag{2.83}$$

$$\sum_{t=1}^{T} \gamma_t(3) \cdot \delta(O_t = 1) = \gamma_2(3) \approx 0.2368 \tag{2.84}$$

$$\sum_{t=1}^{T} \gamma_t(3) \cdot \delta(O_t = 3) = \gamma_3(3) \approx 0.4916 \tag{2.85}$$

Calcul des $b_j(k)$

$$b_1(2) = \frac{0.1135}{0.2654} \approx 0.428 \tag{2.86}$$

$$b_1(1) = \frac{0.1519}{0.2654} \approx 0.573 \tag{2.87}$$

$$b_1(3) = \frac{0.2330}{0.2654} \approx 0.878 \tag{2.88}$$

$$b_2(2) = \frac{0.6287}{1.24} \approx 0.507 \tag{2.89}$$

$$b_2(1) = \frac{0.6113}{1.24} \approx 0.493 \tag{2.90}$$

$$b_2(3) = \frac{0.2755}{1.24} \approx 0.222 \tag{2.91}$$

$$b_3(2) = \frac{0.2581}{0.4949} \approx 0.522 \tag{2.92}$$

$$b_3(1) = \frac{0.2368}{0.4949} \approx 0.478 \tag{2.93}$$

$$b_3(3) = \frac{0.4916}{0.4949} \approx 0.994 \tag{2.94}$$

Résultats des Nouvelles Matrices π, A, et B Les matrices recalculées représentent les nouveaux paramètres de votre HMM après application de l'algorithme de Baum-Welch. Elles devraient mieux modéliser la relation entre les états cachés (nombre de médailles) et les observations (audience) basées sur les observations fournies.

Nouvelles Matrices

$$\pi = \begin{bmatrix} 0.1135 \\ 0.6287 \\ 0.2581 \end{bmatrix} \tag{2.95}$$

$$A = \begin{bmatrix} 1.634 & 1.032 & 1.246 \\ 0.349 & 0.221 & 0.267 \\ 0.876 & 0.554 & 0.668 \end{bmatrix} \tag{2.96}$$

$$B = \begin{bmatrix} 0.428 & 0.507 & 0.522 \\ 0.573 & 0.493 & 0.478 \\ 0.878 & 0.222 & 0.994 \end{bmatrix} \tag{2.97}$$

Erreur et Correction

On remarque que la somme des colonnes des matrices obtenues n'est pas égale à 1, après correction nous avons donc obtenu ces matrices :

Mise à jour de π

$$\pi = \begin{bmatrix} 0.15389626 \\ 0.63032147 \\ 0.21578227 \end{bmatrix}$$

Mise à jour de A

$$A = \begin{bmatrix} 0.70220381 & 0.22301132 & 0.07478487 \\ 0.28345202 & 0.51624108 & 0.20030689 \\ 0.12731995 & 0.29938935 & 0.5732907 \end{bmatrix}$$

Mise à jour de B

$$B = \begin{bmatrix} 0.55106289 & 0.3704875 & 0.07844961 \\ 0.34014049 & 0.4919374 & 0.16792212 \\ 0.09246885 & 0.31666862 & 0.59086252 \end{bmatrix}$$

Ce problème peut être lié à un nombre d'itérations insuffisant pour obtenir la convergence. Nous allons donc utiliser ces nouvelles matrices par défaut dans le but d'appliquer l'algorithme de Viterbi que nous décrirons ci-dessous.

2.3 Algorithme de Viterbi : conclusion du problème de départ

2.3.1 Algorithme de Viterbi

Initialisation

$$\delta_1(i) = \pi_i b_i(O_1), \quad \psi_1(i) = 0, \quad \text{pour } 1 \leq i \leq N$$

Récursion

$$\delta_t(j) = \max_{1\leq i\leq N}[\delta_{t-1}(i)a_{ij}]b_j(O_t), \quad \psi_t(j) = \arg\max_{1\leq i\leq N}[\delta_{t-1}(i)a_{ij}], \quad \text{pour } 2 \leq t \leq T \text{ et } 1 \leq j \leq N$$

Terminaison

$$P^* = \max_{1\leq i\leq N} \delta_T(i)$$

$$q_T^* = \arg\max_{1\leq i\leq N} \delta_T(i)$$

Backtracking

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \quad \text{pour } t = T-1, T-2, \ldots, 1$$

2.3.2 Application à l'exemple et conclusion

L'algorithme de Viterbi est utilisé pour trouver la séquence d'états la plus probable, étant donné une séquence d'observations et un modèle de Markov caché (HMM). Dans votre cas, cela signifie déterminer le nombre de médailles que la France est le plus susceptible d'avoir à la fin des JO, en se basant sur les observations d'audience.

Voici comment nous pouvons appliquer l'algorithme de Viterbi avec les matrices π, A et B mises à jour.

Étapes de l'algorithme de Viterbi

Initialisation

$$\delta_1(i) = \pi_i \cdot b_i(O_1)$$

$$\psi_1(i) = 0$$

Avec les nouvelles matrices :

$$\pi = \begin{bmatrix} 0.15389626 \\ 0.63032147 \\ 0.21578227 \end{bmatrix}$$

$$A = \begin{bmatrix} 0.70220381 & 0.22301132 & 0.07478487 \\ 0.28345202 & 0.51624108 & 0.20030689 \\ 0.12731995 & 0.29938935 & 0.5732907 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.55106289 & 0.3704875 & 0.07844961 \\ 0.34014049 & 0.4919374 & 0.16792212 \\ 0.09246885 & 0.31666862 & 0.59086252 \end{bmatrix}$$

$$O = (1, 0, 2)$$

Calcul de δ et ψ pour $t = 1$
Pour $O_1 = 1$ (Observation 1, qui correspond à l'audimat moyen) :

$$\delta_1(1) = \pi_1 \cdot b_1(O_1) = 0.15389626 \cdot 0.3704875 = 0.05704893$$

$$\delta_1(2) = \pi_2 \cdot b_2(O_1) = 0.63032147 \cdot 0.4919374 = 0.30981285$$

$$\delta_1(3) = \pi_3 \cdot b_3(O_1) = 0.21578227 \cdot 0.31666862 = 0.06833573$$

Pour ψ à $t = 1$:

$$\psi_1(1) = 0$$

$$\psi_1(2) = 0$$

$$\psi_1(3) = 0$$

Récurrence pour $t = 2$

Pour $O_2 = 0$ (Observation 0, qui correspond à un faible audimat) :

Pour $j = 1$:

$$\delta_2(1) = \max \begin{pmatrix} 0.05704893 \cdot 0.70220381 \\ 0.30981285 \cdot 0.28345202 \\ 0.06833573 \cdot 0.12731995 \end{pmatrix} \cdot 0.55106289 \tag{2.98}$$

$$= \max \begin{pmatrix} 0.04005294 \\ 0.08776917 \\ 0.00870287 \end{pmatrix} \cdot 0.55106289 \tag{2.99}$$

$$= 0.04835877 \tag{2.100}$$

$$\psi_2(1) = \arg\max \begin{pmatrix} 0.05704893 \cdot 0.70220381 \\ 0.30981285 \cdot 0.28345202 \\ 0.06833573 \cdot 0.12731995 \end{pmatrix} \tag{2.101}$$

$$= \arg\max \begin{pmatrix} 0.04005294 \\ 0.08776917 \\ 0.00870287 \end{pmatrix} \tag{2.102}$$

$$= 1 \tag{2.103}$$

Pour $j = 2$:

$$\delta_2(2) = \max \begin{pmatrix} 0.05704893 \cdot 0.22301132 \\ 0.30981285 \cdot 0.51624108 \\ 0.06833573 \cdot 0.29938935 \end{pmatrix} \cdot 0.34014049 \tag{2.104}$$

$$= \max \begin{pmatrix} 0.01272313 \\ 0.15993857 \\ 0.02046423 \end{pmatrix} \cdot 0.34014049 \tag{2.105}$$

$$= 0.05438138 \tag{2.106}$$

$$\psi_2(2) = \arg\max \begin{pmatrix} 0.05704893 \cdot 0.22301132 \\ 0.30981285 \cdot 0.51624108 \\ 0.06833573 \cdot 0.29938935 \end{pmatrix} \tag{2.107}$$

$$= \arg\max \begin{pmatrix} 0.01272313 \\ 0.15993857 \\ 0.02046423 \end{pmatrix} \tag{2.108}$$

$$= 1 \tag{2.109}$$

Pour $j = 3$:

$$\delta_2(3) = \max \begin{pmatrix} 0.05704893 \cdot 0.07478487 \\ 0.30981285 \cdot 0.20030689 \\ 0.06833573 \cdot 0.5732907 \end{pmatrix} \cdot 0.09246885 \tag{2.110}$$

$$= \max \begin{pmatrix} 0.00426211 \\ 0.06202373 \\ 0.03915567 \end{pmatrix} \cdot 0.09246885 \tag{2.111}$$

$$= 0.00573979 \tag{2.112}$$

$$\psi_2(3) = \arg\max \begin{pmatrix} 0.05704893 \cdot 0.07478487 \\ 0.30981285 \cdot 0.20030689 \\ 0.06833573 \cdot 0.5732907 \end{pmatrix} \tag{2.113}$$

$$= \arg\max \begin{pmatrix} 0.00426211 \\ 0.06202373 \\ 0.03915567 \end{pmatrix} \tag{2.114}$$

$$= 1 \tag{2.115}$$

Mise à jour des matrices δ et ψ pour $t = 2$

$$\delta_2 = \begin{bmatrix} 0.04835877 \\ 0.05438138 \\ 0.00573979 \end{bmatrix} \tag{2.116}$$

$$\psi_2 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \tag{2.117}$$

Récurrence pour $t = 3$

Pour $O_3 = 2$ (Observation 2, qui correspond à un fort audimat) :

Pour $j = 1$:

$$\delta_3(1) = \max \begin{pmatrix} 0.04835877 \cdot 0.70220381 \\ 0.05438138 \cdot 0.28345202 \\ 0.00573979 \cdot 0.12731995 \end{pmatrix} \cdot 0.55106289 \tag{2.118}$$

$$= \max \begin{pmatrix} 0.03395165 \\ 0.01540667 \\ 0.00073064 \end{pmatrix} \cdot 0.55106289 \tag{2.119}$$

$$= 0.01870375 \tag{2.120}$$

$$\psi_3(1) = \arg\max \begin{pmatrix} 0.04835877 \cdot 0.70220381 \\ 0.05438138 \cdot 0.28345202 \\ 0.00573979 \cdot 0.12731995 \end{pmatrix} \tag{2.121}$$

$$= \arg\max \begin{pmatrix} 0.03395165 \\ 0.01540667 \\ 0.00073064 \end{pmatrix} \tag{2.122}$$

$$= 0 \tag{2.123}$$

Pour $j = 2$:

$$\delta_3(2) = \max \begin{pmatrix} 0.04835877 \cdot 0.22301132 \\ 0.05438138 \cdot 0.51624108 \\ 0.00573979 \cdot 0.29938935 \end{pmatrix} \cdot 0.34014049 \tag{2.124}$$

$$= \max \begin{pmatrix} 0.01078983 \\ 0.02806599 \\ 0.00171793 \end{pmatrix} \cdot 0.34014049 \tag{2.125}$$

$$= 0.00954984 \tag{2.126}$$

$$\psi_3(2) = \arg\max \begin{pmatrix} 0.04835877 \cdot 0.22301132 \\ 0.05438138 \cdot 0.51624108 \\ 0.00573979 \cdot 0.29938935 \end{pmatrix} \tag{2.127}$$

$$= \arg\max \begin{pmatrix} 0.01078983 \\ 0.02806599 \\ 0.00171793 \end{pmatrix} \tag{2.128}$$

$$= 1 \tag{2.129}$$

Pour $j = 3$:

$$\begin{aligned}
\delta_3(3) &= \max \begin{pmatrix} 0.04835877 \cdot 0.07478487 \\ 0.05438138 \cdot 0.20030689 \\ 0.00573979 \cdot 0.5732907 \end{pmatrix} \cdot 0.09246885 && (2.130) \\
&= \max \begin{pmatrix} 0.00361532 \\ 0.01089121 \\ 0.00329283 \end{pmatrix} \cdot 0.09246885 && (2.131) \\
&= 0.00100725 && (2.132)
\end{aligned}$$

$$\begin{aligned}
\psi_3(3) &= \arg\max \begin{pmatrix} 0.04835877 \cdot 0.07478487 \\ 0.05438138 \cdot 0.20030689 \\ 0.00573979 \cdot 0.5732907 \end{pmatrix} && (2.133) \\
&= \arg\max \begin{pmatrix} 0.00361532 \\ 0.01089121 \\ 0.00329283 \end{pmatrix} && (2.134) \\
&= 1 && (2.135)
\end{aligned}$$

Mise à jour des matrices δ et ψ pour $t = 3$

$$\delta_3 = \begin{bmatrix} 0.01870375 \\ 0.00954984 \\ 0.00100725 \end{bmatrix} \tag{2.136}$$

$$\psi_3 = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} \tag{2.137}$$

Résultat des matrices δ et ψ Les matrices δ et ψ après les étapes de récurrence pour $t = 2$ et $t = 3$ sont :

$$\delta = \begin{bmatrix} 0.05704893 & 0.04835877 & 0.01870375 \\ 0.30981285 & 0.05438138 & 0.00954984 \\ 0.06833573 & 0.00573979 & 0.00100725 \end{bmatrix}$$

$$\psi = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

Terminaison

Calcul de la probabilité de la séquence la plus probable (P^*)

$$P^* = \max_i \delta_3(i) = \max(0.01870375, 0.00954984, 0.00100725) = 0.01870375$$

Détermination de la dernière étape de la séquence la plus probable (q_T^*)

$$q_3^* = \arg\max_i \delta_3(i) = \arg\max(0.01870375, 0.00954984, 0.00100725) = 0$$

Backtracking

Ensuite, nous allons utiliser la matrice ψ pour retracer la séquence des états cachés :

$$q_2^* = \psi_3(q_3^*) = \psi_3(0) = 0$$

$$q_1^* = \psi_2(q_2^*) = \psi_2(0) = 1$$

Séquence des états cachés

La séquence la plus probable des états cachés est donc :

$$q^* = (1, 0, 0)$$

Résultats en termes de médailles

En utilisant le mapping des états aux nombres de médailles, nous avons :
- Jour 1 : Nombre moyen de médailles (2-3 médailles)
- Jour 2 : Faible nombre de médailles (0-1 médailles)
- Jour 3 : Faible nombre de médailles (0-1 médailles)

On généralise ensuite l'algorithme pour $t = 18$. Nous avons simplifié sur le document pour pouvoir expliquer plus clairement le fonctionnement. Voici le résultat final :

2.4 Application Généralisée de l'Algorithme de Viterbi pour $T = 18$

2.4.1 Observations aléatoires générées pour $T = 18$

$$O = [0, 1, 0, 1, 1, 2, 0, 2, 0, 0, 0, 0, 2, 1, 2, 2, 0, 1, 1]$$

2.4.2 Séquence la plus probable des états (médailles)

- Jour 1 : Nombre moyen de médailles (2-3 médailles)
- Jour 2 : Faible nombre de médailles (0-1 médailles)
- Jour 3 : Faible nombre de médailles (0-1 médailles)
- Jour 4 : Faible nombre de médailles (0-1 médailles)
- Jour 5 : Faible nombre de médailles (0-1 médailles)
- Jour 6 : Faible nombre de médailles (0-1 médailles)
- Jour 7 : Faible nombre de médailles (0-1 médailles)
- Jour 8 : Faible nombre de médailles (0-1 médailles)
- Jour 9 : Faible nombre de médailles (0-1 médailles)
- Jour 10 : Faible nombre de médailles (0-1 médailles)
- Jour 11 : Faible nombre de médailles (0-1 médailles)
- Jour 12 : Grand nombre de médailles (4+ médailles)
- Jour 13 : Grand nombre de médailles (4+ médailles)
- Jour 14 : Grand nombre de médailles (4+ médailles)
- Jour 15 : Grand nombre de médailles (4+ médailles)
- Jour 16 : Nombre moyen de médailles (2-3 médailles)
- Jour 17 : Nombre moyen de médailles (2-3 médailles)
- Jour 18 : Nombre moyen de médailles (2-3 médailles)

2.4.3 Probabilité de la séquence la plus probable

$$P^* = 2.4254871723659994 \times 10^{-13}$$

2.4.4 Interprétation

- Nombre moyen de médailles au début (Jour 1) et à la fin (Jours 16 à 18) des JO.
- Faible nombre de médailles durant la majorité des jours (Jours 2 à 11).
- Grand nombre de médailles vers la fin de la période (Jours 12 à 15).

2.4.5 Conclusion

La séquence la plus probable des états cachés, trouvée grâce à l'algorithme de Viterbi, montre des variations dans le nombre de médailles que la France est susceptible de gagner au cours des 18 jours des JO, avec une probabilité très faible de cette séquence spécifique en raison de la nature aléatoire des observations générées. En moyenne, la France gagnera 19 médaille d'or au minimum, ce qui la positionnerait à la 5ème place des JO précédent de Tokyo.

2.5 Généralisation

2.5.1 Limites de la résolution du problème

Dans l'ensemble, les résultats semblent cohérents sauf pour Baum-Welch. On peut expliquer cela par un nombre trop limité d'itérations pour obtenir la convergence des estimateurs. Une autre limite du modèle, le lien entre l'audimat et les performances est-il si important ?

En général, plus l'équipe gagne, plus celle-ci devient populaire, plus l'audimat augmente. Néanmoins, d'autres facteurs peuvent faire varier l'audimat, la personnalité du sportif, la discipline diffusée au jour t plus ou moins populaire, une politisation des jeux poussant certains à les boycotter et d'autres à les encourager.

La simplification du modèle ne prend aussi pas en compte beaucoup de paramètres comme la nature des médailles, ce qui peut donc aussi fausser le classement final. De plus, les probabilités faibles, liées à l'aléa important, peuvent rendre plus intéressante une approche qualitative (par tranche) du nombre de médailles pour tenter d'estimer notre classement final.

Un pronostic très précis sur les résultats de ces JO semble donc très peu fiable.

2.5.2 D'autres exemples d'utilisations des chaînes de Markov cachées

Les chaînes de Markov cachées peuvent aussi être utilisées pour optimiser l'entraînement d'un athlète. Nous allons voir ici comment optimiser la préparation d'un judoka avec pour états cachés l'intensité de l'entraînement et pour observations l'état physique du judoka.

Nous modéliserons les paramètres à l'aide d'une distribution aléatoire fondée sur des données historiques (il est possible d'optimiser les résultats à l'aide des algorithmes étudiés précédemment).

2.6 Utilisation des chaînes de Markov cachées pour optimiser l'entraînement et analyser les séquences de combat d'un judoka

Un judoka doit se préparer pour les Jeux Olympiques dans six mois. Les entraîneurs veulent maximiser sa forme physique le jour de la compétition. Les chaînes de Markov cachées (HMM) sont utilisées pour modéliser et optimiser les cycles d'entraînement et analyser les séquences de combat.

2.6.1 Optimisation de l'entraînement

Objectif

Utiliser les HMM pour déterminer les cycles d'entraînement intensif suivis de périodes de récupération optimales, afin de maintenir une forme physique élevée sans atteindre un état de fatigue.

Modélisation des cycles d'entraînement

- **États cachés** : Niveaux de forme physique du judoka (par exemple, condition optimale, condition moyenne, condition faible).
- **Observations** : Types et intensités d'entraînement, résultats des sessions d'entraînement, indicateurs physiologiques (par exemple, fréquence cardiaque, niveaux de lactate).
- **Probabilités de transition** : Probabilités de passer d'un niveau de forme physique à un autre en fonction des types d'entraînement et des périodes de récupération.

Application des HMM

- **Collecte de données** : Recueillir des données historiques sur les cycles d'entraînement et les performances des judokas.
- **Estimation des paramètres** : Utiliser les données pour estimer les probabilités de transition et d'émission du modèle HMM.
- **Planification** : Appliquer le modèle pour prévoir les niveaux de forme physique et ajuster les cycles d'entraînement en conséquence.

Exemple de planification

Le modèle indique que des cycles d'entraînement intensif suivis de périodes de récupération sont les plus efficaces pour maintenir une forme physique élevée. Les entraîneurs ajustent le programme d'entraînement en fonction des prévisions du modèle, assurant ainsi que le judoka est en condition optimale pour les Jeux Olympiques.

2.6.2 Analyse des séquences de combat

Objectif

Utiliser les HMM pour analyser les séquences de techniques utilisées par les judokas pendant les combats et identifier les stratégies gagnantes.

Modélisation des séquences de techniques

- **États cachés** : Stratégies tactiques utilisées par le judoka (par exemple, offensive, défensive, contre-attaque).
- **Observations** : Techniques observées pendant le combat (par exemple, ippon-seoi-nage, o-soto-gari).
- **Probabilités de transition** : Probabilités de transition entre différentes stratégies tactiques en fonction des réactions de l'adversaire.

Application des HMM

- **Collecte de données** : Recueillir des données sur les techniques utilisées pendant les combats, ainsi que les résultats des combats.
- **Estimation des paramètres** : Utiliser les données pour estimer les probabilités de transition et d'émission du modèle HMM.
- **Analyse** : Utiliser le modèle pour identifier les séquences de techniques qui ont le plus de chances de conduire à la victoire.

Exemple d'analyse

En analysant les séquences de combat avec les HMM, les entraîneurs peuvent identifier les stratégies tactiques les plus efficaces. Par exemple, une séquence qui commence par une stratégie défensive suivie d'une contre-attaque pourrait avoir une probabilité élevée de succès contre certains adversaires.

2.6.3 Estimation des paramètres sans algorithmes avancés

Sans utiliser des algorithmes avancés comme Baum-Welch ou Viterbi, vous pouvez estimer les paramètres de base du HMM par des méthodes simplifiées ou heuristiques.

Probabilités initiales (π)

Estimez la distribution initiale des états cachés en utilisant les fréquences observées dans les données historiques.

Matrice de transition (A)

Calculez les probabilités de transition entre les états cachés en utilisant les transitions observées dans les données historiques. Par exemple, si dans vos données un judoka passe d'un état de condition moyenne à un état de condition optimale 30% du temps, alors $A_{\text{moyenne}\rightarrow\text{optimale}} = 0.3$.

Matrice des probabilités d'émission (B)

Estimez les probabilités d'émission en utilisant les fréquences des observations pour chaque état caché. Par exemple, si un type de technique est observé 50% du temps quand le judoka est en stratégie offensive, alors $B_{\text{offensive}} = 0.5$ pour cette technique.

2.6.4 Exemple de calcul des paramètres

Supposons que nous avons les données suivantes pour les séquences d'entraînement et les techniques de combat :

Séquences d'entraînement (États cachés et observations)

— **États cachés (forme physique)** : [Faible, Moyenne, Optimale]
— **Observations (type d'entraînement)** : [Léger, Modéré, Intensif, Récupération]

Séquences de combat (stratégies et techniques)

— **États cachés (stratégies)** : [Offensive, Défensive, Contre-attaque]
— **Observations (techniques)** : [ippon-seoi-nage, o-soto-gari, etc.]

Paramètres estimés pour l'entraînement

Probabilités initiales
— Faible : 0.4
— Moyenne : 0.4
— Optimale : 0.2

Matrice de transition (A)

$$
\begin{array}{c} \\ Faible \\ Moyenne \\ Optimale \end{array}
\begin{array}{c}
\begin{array}{ccc} Faible & Moyenne & Optimale \end{array} \\
\begin{bmatrix} 0.6 & 0.3 & 0.1 \\ 0.3 & 0.5 & 0.2 \\ 0.1 & 0.3 & 0.6 \end{bmatrix}
\end{array}
$$

Matrice des probabilités d'émission (B)

$$
\begin{array}{c} \\ Faible \\ Moyenne \\ Optimale \end{array}
\begin{array}{c}
\begin{array}{cccc} Lger & Modr & Intensif & Rcupration \end{array} \\
\begin{bmatrix} 0.6 & 0.3 & 0.1 & 0.0 \\ 0.3 & 0.5 & 0.2 & 0.1 \\ 0.1 & 0.3 & 0.5 & 0.1 \end{bmatrix}
\end{array}
$$

Paramètres estimés pour les séquences de combat

Probabilités initiales
— Offensive : 0.4
— Défensive : 0.3
— Contre-attaque : 0.3

Matrice de transition (A)

$$\begin{array}{c|ccc} & Offensive & Dfensive & Contre-attaque \\ \hline Offensive & 0.5 & 0.3 & 0.2 \\ Dfensive & 0.2 & 0.5 & 0.3 \\ Contre-attaque & 0.3 & 0.3 & 0.4 \end{array}$$

Matrice des probabilités d'émission (B)

$$\begin{array}{c|ccc} & ippon-seoi-nage & o-soto-gari & Autres \\ \hline Offensive & 0.6 & 0.3 & 0.1 \\ Dfensive & 0.3 & 0.5 & 0.2 \\ Contre-attaque & 0.1 & 0.2 & 0.7 \end{array}$$

2.6.5 Programmation d'un entraînement à partir des résultats des HMM

Étapes pour programmer un entraînement

- **Définir les paramètres de l'entraînement** :
  - **États cachés (forme physique)** : Faible, Moyenne, Optimale
  - **Observations (types d'entraînement)** : Léger, Modéré, Intensif, Récupération
- **Initialisation des paramètres du HMM** : Utiliser les paramètres estimés précédemment pour les probabilités initiales, les matrices de transition et d'émission.
- **Planification de l'entraînement** : Utiliser les HMM pour prédire les niveaux de forme physique au fil du temps et ajuster les cycles d'entraînement en consequence.

Exemple de planification en Python

```python
import numpy as np

# Donnees fictives pour les cycles d'entrainement
# 0: Entrainement l ger , 1: Entrainement modere , 2: Entrainement intensif , 3: Recuperation
# Etats caches : 0 - Faible , 1 - Moyenne , 2 - Optimale

# Parametres estimes
n_states = 3
n_obs = 4

start_prob = np.array([0.4, 0.4, 0.2])
trans_matrix = np.array([[0.6, 0.3, 0.1],
                         [0.3, 0.5, 0.2],
                         [0.1, 0.3, 0.6]])

emission_matrix = np.array([[0.5, 0.3, 0.2, 0.0],
                            [0.2, 0.5, 0.2, 0.1],
                            [0.1, 0.3, 0.5, 0.1]])

# Initialisation de l' tat
current_state = np.random.choice(n_states, p=start_prob)

# Generer un plan d'entrainement sur 30 jours
plan_entrainement = []
forme_physique = []

for day in range(30):
```

```
    # Ajouter l'etat actuel   la liste des etats de forme physique
    forme_physique.append(current_state)

    # Choisir le type d'entrainement en fonction de l'etat actuel
    entrainement = np.random.choice(n_obs, p=emission_matrix[current_state])
    plan_entrainement.append(entrainement)

    # Transition vers le prochain etat
    current_state = np.random.choice(n_states, p=trans_matrix[current_state])

# Affichage du plan d'entra nement
types_entrainement = ["L ger", "Modere", "Intensif", "Recup ration"]
forme = ["Faible", "Moyenne", "Optimale"]

print("Plan d'entrainement sur 30 jours :\n")
for day, (entr, forme) in enumerate(zip(plan_entrainement, forme_physique)):
    print(f"Jour {day + 1}: Entrainement {types_entrainement[entr]} - Forme physique {forme[forme]}")
    # Jour 1: Entrainement Leger - Forme physique Faible
    # Jour 2: Entrainement Modere - Forme physique Moyenne
    # Jour 3: Entrainement Intensif - Forme physique Moyenne
   # ...
```

Interprétation du plan d'entraînement

Le plan d'entraînement généré alterne entre différents types d'entraînement (léger, modéré, intensif, récupération) en fonction de l'état de forme physique prévu du judoka. Les entraîneurs peuvent utiliser ce plan pour s'assurer que le judoka suit un programme équilibré, avec des périodes d'entraînement intensif suivies de périodes de récupération, afin de maximiser la forme physique sans provoquer de fatigue excessive.

Ajustements et personnalisation

Le plan d'entraînement peut être ajusté en fonction des besoins spécifiques du judoka et des observations au fil du temps. Par exemple, si le judoka montre des signes de fatigue ou de blessure, les entraîneurs peuvent augmenter les périodes de récupération ou diminuer l'intensité de l'entraînement.

Les entraîneurs peuvent également utiliser des outils de suivi et des métriques supplémentaires (par exemple, fréquence cardiaque, niveaux de lactate) pour affiner le modèle HMM et améliorer la précision des prévisions.

2.6.6 Conclusion

En utilisant les chaînes de Markov cachées, les entraîneurs peuvent programmer des plans d'entraînement optimisés qui maximisent la forme physique du judoka tout en minimisant le risque de fatigue et de blessure. Ces modèles permettent une planification dynamique et adaptative, essentielle pour la préparation aux compétitions de haut niveau comme les Jeux Olympiques. Ce cadre simplifié pour l'estimation des paramètres peut être suffisant pour des analyses préliminaires et peut être affiné davantage avec des données plus détaillées et des techniques plus sophistiquées au besoin.