

Reconnaissance de séquences

DTW + HMM

Catherine ACHARD
Institut des Systèmes Intelligents et de Robotique

catherine.achard@sorbonne-universite.fr

Séquences de longueur variable. Comment faire ?

- Mesurer une distance entre séquence (Dynamic Time Warping, DTW) puis classification par KPPV (méthode discriminative)
- Modéliser les séquences (Hidden Markov Model, HMM) et classifieur par une approche générative

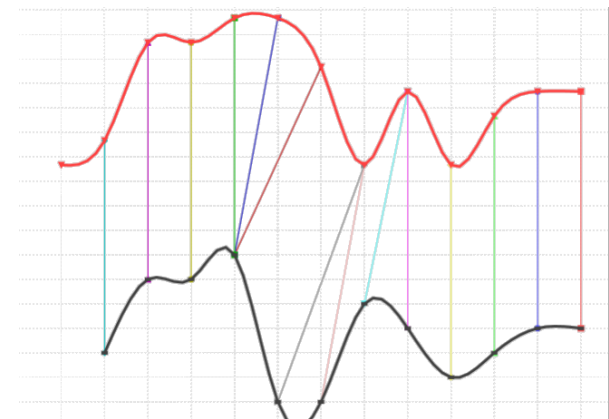
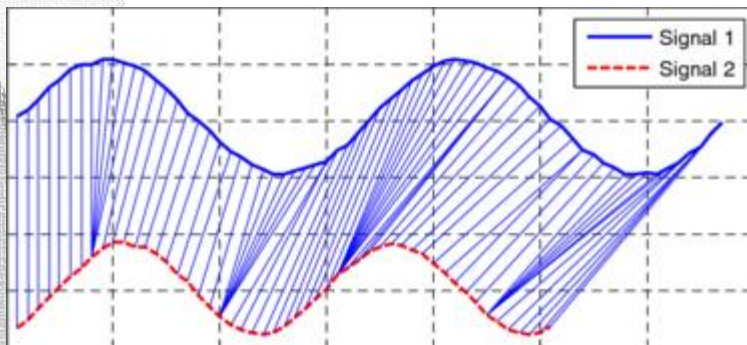
DTW

Dynamic Time Warping

Il faut établir une distance entre deux séquences

Problèmes:

- Les deux séquences à comparer n'ont pas forcément la même longueur
 - Les séquences ne sont pas toujours exactement alignées dans le temps
- ➔ Il n'est pas possible d'utiliser une distance euclidienne



Il faut donc aligner temporellement les signaux

Soit

- $G = \{g_1, g_2, \dots, g_k\}$ et $H = \{h_1, h_2, \dots, h_l\}$ deux séquences de longueur k et l et de dimension n (chaque g_i ou h_j est de dimension n)
- $d(g_i, h_j)$: distance entre g_i et h_j

Le DTW réalise à la fois l'alignement des séquences et calcule leur distance. Il est fondé sur un principe récursif :

Si $D(i, j)$ est la distance entre les séquences $\{g_1, g_2, \dots, g_i\}$ et $\{h_1, h_2, \dots, h_j\}$ alors,

$$D(i, j) = d(g_i, h_j) + \min \begin{cases} D(i-1, j) \\ D(i, j-1) \\ D(i-1, j-1) \end{cases}$$

La distance entre les deux séquences est donnée par $D(k, l)$

$$D(i, j) = d(g_i, h_j) + \min \begin{cases} D(i-1, j) \\ D(i, j-1) \\ D(i-1, j-1) \end{cases}$$



	h_1	h_2	h_3	h_4	h_5
g_1					
g_2			D(2,3)	D(2,4)	
g_3			D(3,3)	D(3,4)?	
g_4					

$$D(3, 4) = d(g_3, h_4) + \min \begin{cases} D(2, 3) \\ D(2, 4) \\ D(3, 3) \end{cases}$$

Algorithme DTW
$$D(1, 1) = d(g_1, h_1)$$
for $i = 2$ à k **do**
$$D(i, 1) = D(i-1, 1) + d(g_i, h_1)$$
end for**for** $j = 2$ à l **do**
$$D(1, j) = D(1, j-1) + d(g_1, h_j)$$
end for**for** $i = 2$ à k **do****for** $j = 2$ à l **do**

$$D(i, j) = d(g_i, h_j) + \min \begin{cases} D(i-1, j) \\ D(i, j-1) \\ D(i-1, j-1) \end{cases}$$

end for**end for**

La distance entre les deux séquences est donnée par $D(k, l)$

Exemple entre deux signaux 1D très simples:

$G = \{2.0, 3.0, 7.0, 7.0, 8.0, 8.0, 6.0, 2.0, 5.0, 2.0, 4.0, 5.0, 5.0\}$

$H = \{4.0, 7.0, 7.0, 8.0, 2.0, 2.0, 6.0, 5.0, 3.0, 4.0, 5.0, 5.0\}$

	2	3	7	7	8	8	6	2	5	2	4	5	5
4													
7													
7													
8													
2													
2													
6													
5													
3													
4													
5													
5													

Exemple entre deux signaux 1D très simples:

$G = \{2.0, 3.0, 7.0, 7.0, 8.0, 8.0, 6.0, 2.0, 5.0, 2.0, 4.0, 5.0, 5.0\}$

$H = \{4.0, 7.0, 7.0, 8.0, 2.0, 2.0, 6.0, 5.0, 3.0, 4.0, 5.0, 5.0\}$

	2	3	7	7	8	8	6	2	5	2	4	5	5
4	2	?											
7													
7													
8													
2													
2													
6													
5													
3													
4													
5													
5													

Exemple entre deux signaux 1D très simples:

$G = \{2.0, 3.0, 7.0, 7.0, 8.0, 8.0, 6.0, 2.0, 5.0, 2.0, 4.0, 5.0, 5.0\}$

$H = \{4.0, 7.0, 7.0, 8.0, 2.0, 2.0, 6.0, 5.0, 3.0, 4.0, 5.0, 5.0\}$

	2	3	7	7	8	8	6	2	5	2	4	5	5
4	2	3											
7	7												
7													
8													
2													
2													
6													
5													
3													
4													
5													
5													

Exemple entre deux signaux 1D très simples:

$G = \{2.0, 3.0, 7.0, 7.0, 8.0, 8.0, 6.0, 2.0, 5.0, 2.0, 4.0, 5.0, 5.0\}$

$H = \{4.0, 7.0, 7.0, 8.0, 2.0, 2.0, 6.0, 5.0, 3.0, 4.0, 5.0, 5.0\}$

	2	3	7	7	8	8	6	2	5	2	4	5	5
4	2	3	6	9	13	17	19	21	22	24	24	25	26
7	7	6	3	3	4	5	6	11	13	18	21	23	25
7	12	10	3	3	4	5	6	11	13	18	21	23	25
8	18	15	4	4	3	3	5	11	14	19	22	24	26
2	18	16	9	9	9	9	7	5	8	8	10	13	16
2	18	17	14	14	15	15	11	5	8	8	10	13	16
6	22	20	15	15	16	17	11	9	6	10	10	11	12
5	25	22	17	17	18	19	12	12	6	9	10	10	10
3	26	22	21	21	22	23	15	13	8	7	8	10	12
4	28	23	24	24	25	26	17	15	9	9	7	8	9
5	31	25	25	26	27	28	18	18	9	12	8	7	7
5	34	27	27	27	29	30	19	21	9	12	9	7	

Exemple entre deux signaux 1D très simples:

$G = \{2.0, 3.0, 7.0, 7.0, 8.0, 8.0, 6.0, 2.0, 5.0, 2.0, 4.0, 5.0, 5.0\}$

$H = \{4.0, 7.0, 7.0, 8.0, 2.0, 2.0, 6.0, 5.0, 3.0, 4.0, 5.0, 5.0\}$

	2	3	7	7	8	8	6	2	5	2	4	5	5
4	2	3	6	9	13	17	19	21	22	24	24	25	26
7	7	6	3	3	4	5	6	11	13	18	21	23	25
7	12	10	3	3	4	5	6	11	13	18	21	23	25
8	18	15	4	4	3	3	5	11	14	19	22	24	26
2	18	16	9	9	9	9	7	5	8	8	10	13	16
2	18	17	14	14	15	15	11	5	8	8	10	13	16
6	22	20	15	15	16	17	11	9	6	10	10	11	12
5	25	22	17	17	18	19	12	12	6	9	10	10	10
3	26	22	21	21	22	23	15	13	8	7	8	10	12
4	28	23	24	24	25	26	17	15	9	9	7	8	9
5	31	25	25	26	27	28	18	18	9	12	8	7	7
5	34	27	27	27	29	30	19	21	9	12	9	7	7

Exemple entre deux signaux 1D très simples:

$G = \{2.0, 3.0, 7.0, 7.0, 8.0, 8.0, 6.0, 2.0, 5.0, 2.0, 4.0, 5.0, 5.0\}$

$H = \{4.0, 7.0, 7.0, 8.0, 2.0, 2.0, 6.0, 5.0, 3.0, 4.0, 5.0, 5.0\}$

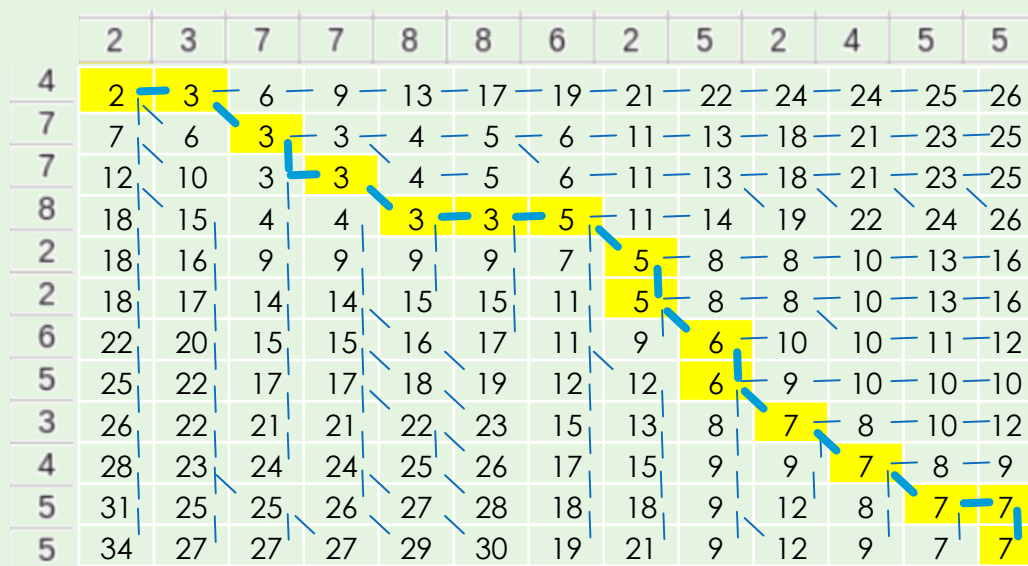
	2	3	7	7	8	8	6	2	5	2	4	5	5
4	2	3	6	9	13	17	19	21	22	24	24	25	26
7	7	6	3	3	4	5	6	11	13	18	21	23	25
7	12	10	3	3	4	5	6	11	13	18	21	23	25
8	18	15	4	4	3	3	5	11	14	19	22	24	26
2	18	16	9	9	9	9	7	5	8	8	10	13	16
2	18	17	14	14	15	15	11	5	8	8	10	13	16
6	22	20	15	15	16	17	11	9	6	10	10	11	12
5	25	22	17	17	18	19	12	12	6	9	10	10	10
3	26	22	21	21	22	23	15	13	8	7	8	10	12
4	28	23	24	24	25	26	17	15	9	9	7	8	9
5	31	25	25	26	27	28	18	18	9	12	8	7	7
5	34	27	27	27	29	30	19	21	9	12	9	7	7

La distance entre les signaux vaut 7

Exemple entre deux signaux 1D très simples:

$G = \{2.0, 3.0, 7.0, 7.0, 8.0, 8.0, 6.0, 2.0, 5.0, 2.0, 4.0, 5.0, 5.0\}$

$H = \{4.0, 7.0, 7.0, 8.0, 2.0, 2.0, 6.0, 5.0, 3.0, 4.0, 5.0, 5.0\}$



En jaune, le chemin minimal qui nous donne l'alignement des signaux

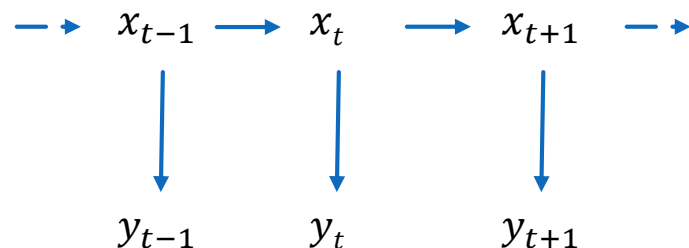
HMM

Hidden Markov Model

Un HMM décrit la loi jointe de variables cachées (états) et de variables observées

Deux hypothèses :

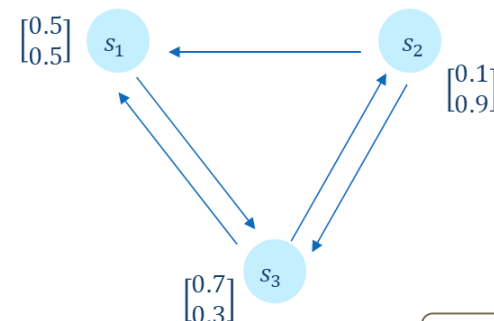
- ✓ L'état caché courant ne dépend que de l'état caché précédent
- ✓ L'observation courante ne dépend que de l'état caché courant.



A l'instant t , l'état caché est représenté par x_t , $t = 1, \dots, T$ et prend valeur dans un ensemble d'états cachés $\{s_1, s_2, \dots, s_N\}$

A l'instant t , l'observation est représentée par y_t , $t = 1, \dots, T$ et prend valeur dans un ensemble de symboles distincts observables $\{o_1, o_2, \dots, o_M\}$

Exemple : 3 états, 2 observations et leur probabilité associée



Un HMM est entièrement défini par $\theta = (A, B, \Pi)$:

✓ La **matrice de transition A**:

$$A = \{a_{ij}\} = \{P(x_{t+1} = s_j | x_t = s_i)\} = \{P(s_j | s_i)\}$$

A ne dépend pas du temps -> l'état suivant ne dépend que de l'état précédant

✓ Le **vecteur des probabilités initiales Π** des états cachés :

$$\Pi = \{\pi_i\} = \{P(x_1 = s_i)\} = \{P(s_i)\}$$

✓ La **matrice d'observation B** de taille :

$$B = \{b_{ki}\} = \{P(y_t = o_k | x_t = s_i)\} = \{P(o_k | s_i)\}$$

Quelles sont les dimensions de ces variables ?

Un HMM est entièrement défini par $\theta = (A, B, \Pi)$:

- ✓ La matrice de transition de taille $N \times N$ (N états) :

$$A = \{a_{ij}\} = \{P(x_{t+1} = s_j | x_t = s_i)\} = \{P(s_j | s_i)\}$$

A ne dépend pas du temps -> l'état suivant ne dépend que de l'état précédant

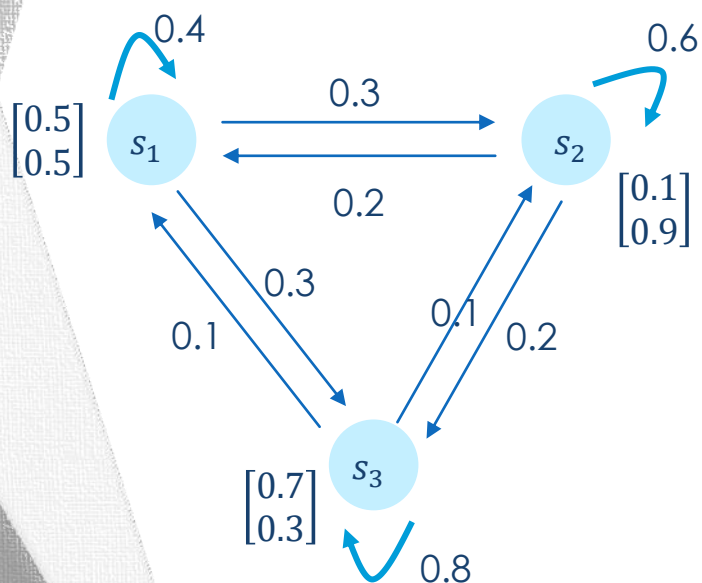
- ✓ Les probabilités initiales des états cachés Π de taille $1 \times N$:

$$\Pi = \{\pi_i\} = \{P(x_1 = s_i)\} = \{P(s_i)\}$$

- ✓ La matrice d'observation de taille $M \times N$:

$$B = \{b_{ki}\} = \{P(y_t = o_k | x_t = s_i)\} = \{P(o_k | s_i)\}$$

Un HMM est représenté par une machine à états finis



$$A = \begin{bmatrix} 0.4 & 0.3 & 0.3 \\ 0.2 & 0.6 & 0.2 \\ 0.1 & 0.1 & 0.8 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.5 & 0.1 & 0.7 \\ 0.5 & 0.9 & 0.3 \end{bmatrix}$$

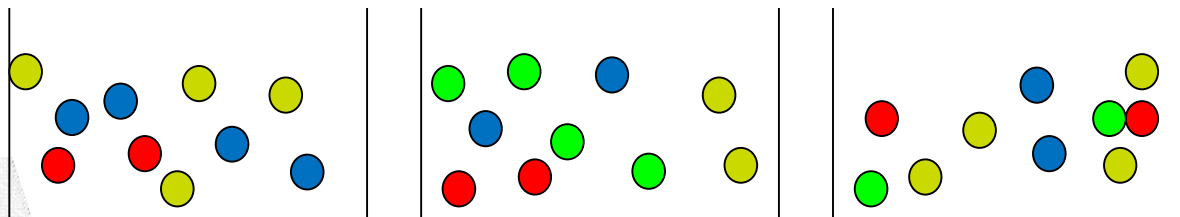
Exemple:

Considérons un ensemble de N boîtes.

Dans chaque boîte se trouvent des balles de M couleurs différentes.

Les observations sont générées ainsi : une personne choisit une boîte initiale en fonction d'une loi uniforme. Puis elle choisit une balle. La couleur de la balle constitue l'observation. La personne choisit alors une autre boîte avec la règle suivante : elle a deux fois plus de chance de rester dans la même boîte que de changer de boîte. Si elle change de boîte, les deux autres boîtes sont équiprobables.

Donner les paramètres $\theta = (\mathbf{A}, \mathbf{B}, \Pi)$ de la chaîne de Markov qui modélise le processus représenté par les boîtes ci-dessous



Exemple:

On aura $N=3$ états, un pour chaque boîte

Il y aura $M=4$ observations (une pour chaque couleur de balle)

une personne choisit une boîte initiale en fonction d'une loi uniforme

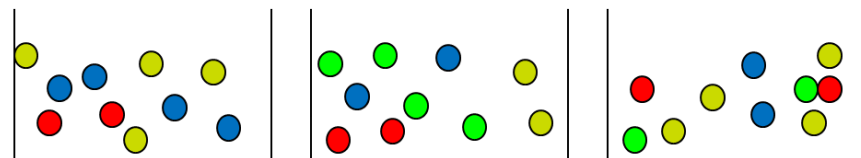
$$\Pi = \{\pi_i\} = \{P(x_1 = s_i)\} = \{1/3; 1/3; 1/3\}$$

La couleur de la balle constitue l'observation

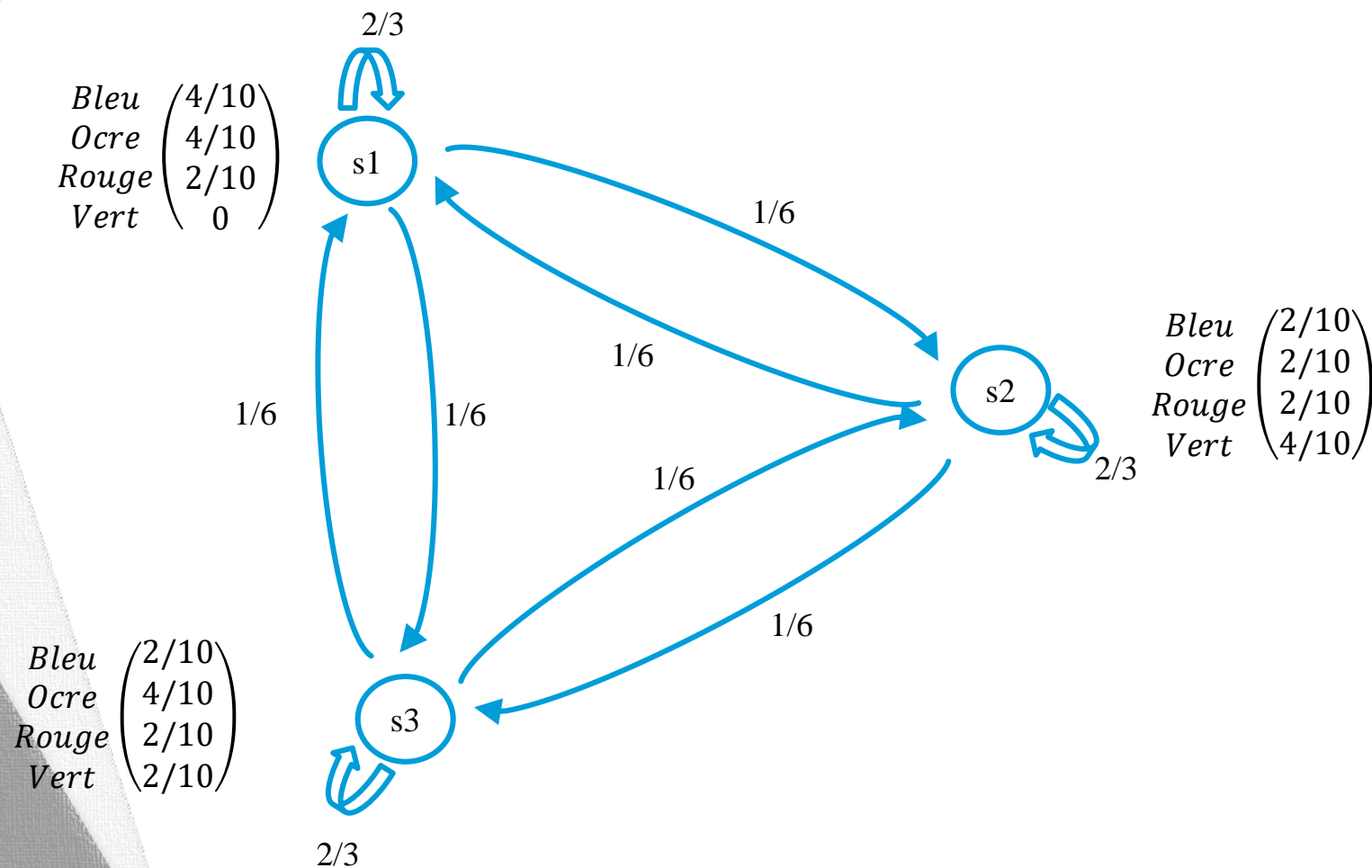
$$B = \{b_{ki}\} = \{P(y_t = o_k | x_t = s_i)\} = \begin{matrix} \text{Bleu} \\ \text{Ocre} \\ \text{Rouge} \\ \text{Vert} \end{matrix} \begin{pmatrix} 4/10 & 2/10 & 2/10 \\ 4/10 & 2/10 & 4/10 \\ 2/10 & 2/10 & 2/10 \\ 0 & 4/10 & 2/10 \end{pmatrix}$$

La personne choisit alors une autre boîte avec la règle suivante : elle a deux fois plus de chance de rester dans la même boîte que de changer de boîte. Si elle change de boîte, les deux autres boîtes sont équiprobables.

$$A = \{a_{ij}\} = P(s_j | s_i) = \begin{pmatrix} 2/3 & 1/6 & 1/6 \\ 1/6 & 2/3 & 1/6 \\ 1/6 & 1/6 & 2/3 \end{pmatrix}$$



Machine à états finis correspondante



Remarque:

Le HMM est un modèle génératif, il peut générer des données (une séquence d'observations):

1. $t = 1$: choix de l'état initial en fonction du vecteur de probabilité initial $\Pi = \{P(s_i)\}$

$$x_t = s_i$$

2. Choix de l'observation y_t avec la probabilité $b_{ki} = P(o_k | s_i)$

$$y_t = o_k$$

3. Transition vers le nouvel état $x_{t+1} = s_j$ avec la probabilité $a_{ij} = P(s_j | s_i)$

4. $t = t + 1$; si $t < T$, retourner en 2, sinon *fin*

Trois problèmes de base :

1. Reconnaissance

Etant donnée une séquence $y_{1:T}$ et θ , quelle est la probabilité que cette séquence ait été générée par le HMM θ ? (**algorithme Forward ou Backward**)

2. Segmentation

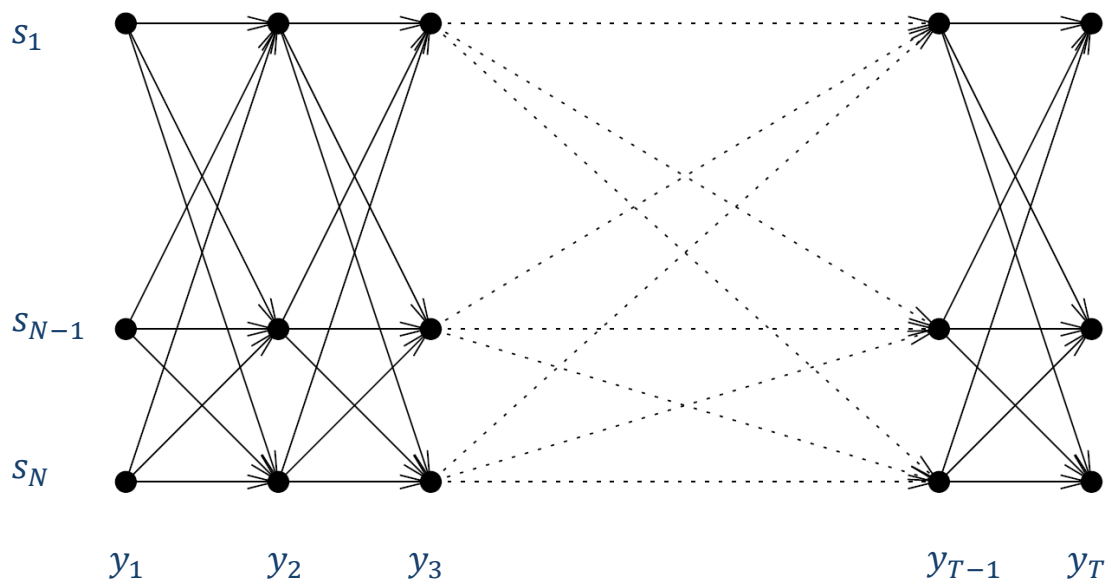
Etant donné $y_{1:T}$ et θ , quelle est la séquence optimale $x_{1:T}$ qui a produit $y_{1:T}$? (**algorithme de Viterbi**)

3. Apprentissage

Etant donnée une séquence (ou un ensemble de séquences) $y_{1:T}$, estimer les paramètres θ du HMM qui maximisent la vraisemblance d'apparition de $y_{1:T}$ (**Algorithme de Baum-Welch**)

Problème 1 - Reconnaissance

Représentation sous forme de treillis



Chaque noeud du treillis est l'événement où une observation y_t est générée alors que le modèle occupait l'état s_i

Problème 1 - Reconnaissance

La vraisemblance \mathfrak{L} qu'un HMM ait généré la séquence d'observation y_1, \dots, y_T est donnée par :

$$\mathfrak{L} = P(y_1, \dots, y_T)$$

→ Estimation de la vraisemblance avec l'**algorithme forward ou backward**

Algorithme forward

Définissons $\alpha_i(t)$, la probabilités d'avoir la séquence d'observation $\{y_1, \dots, y_t\}$ et l'état $x_t = s_i$:

$$\alpha_i(t) = P(y_1, \dots, y_t, x_t = s_i)$$

Comment estimer la vraisemblance \mathfrak{L} à partir des variables forward $\alpha_i(t)$?

Par définition,

$$\mathcal{L} = P(y_1, \dots, y_T) \quad \text{et} \quad \alpha_i(t) = P(y_1, \dots, y_t, x_t = s_i)$$

$$\mathcal{L} = P(y_1, \dots, y_T, x_T = s_1) + P(y_1, \dots, y_T, x_T = s_2) + \dots + P(y_1, \dots, y_T, x_T = s_N)$$

$$\mathcal{L} = \sum_{i=1}^N \alpha_i(T)$$

Reconnaissance, algorithme forward

Calcul des $\alpha_i(t) = P(y_1, \dots, y_t, x_t = s_i)$

Initialisation :

$$\alpha_i(1) = ???$$

Reconnaissance, algorithme forward

Calcul des $\alpha_i(t) = P(y_1, \dots, y_t, x_t = s_i)$

Initialisation :

$$\alpha_i(1) = \pi_i * b_{y_1, i}$$

Induction :

Exprimer $\alpha_i(t)$ en fonction des $\alpha_j(t-1)$

Reconnaissance, algorithme forward

Calcul des $\alpha_i(t) = P(y_1, \dots, y_t, x_t = s_i)$

Initialisation :

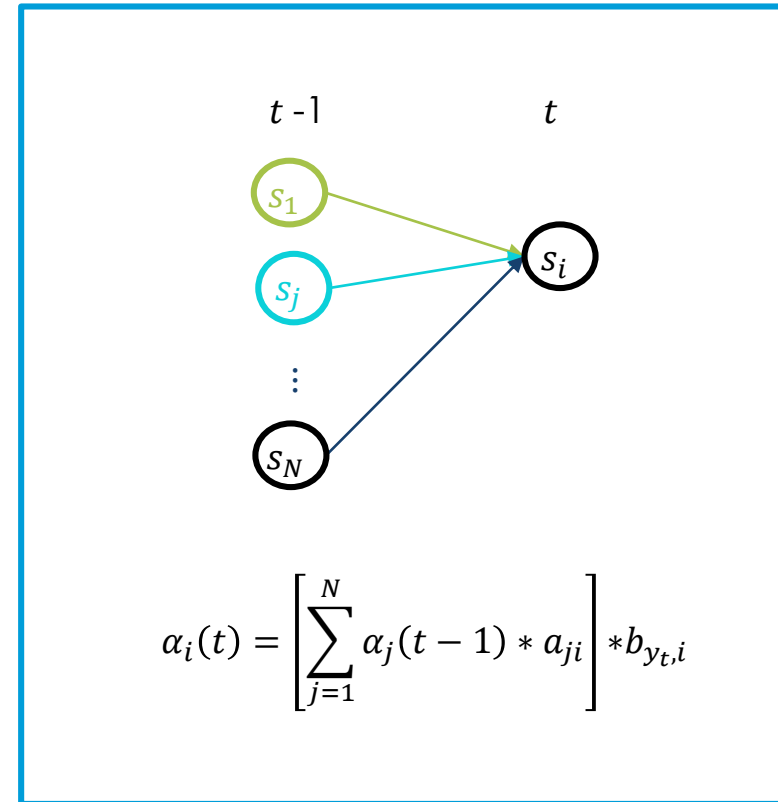
$$\alpha_i(1) = \pi_i * b_{y_1,i}$$

Induction :

$$\alpha_i(t) = \left[\sum_{j=1}^N \alpha_j(t-1) * a_{ji} \right] * b_{y_t,i}$$

Estimation de la vraisemblance :

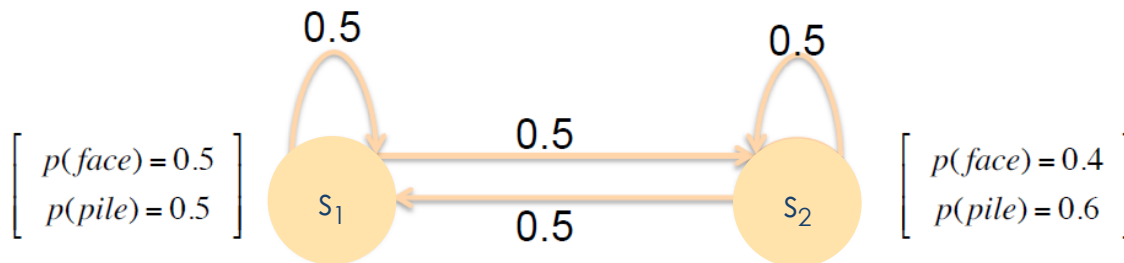
$$\mathfrak{L} = \sum_{i=1}^N \alpha_i(T)$$



Exercice :

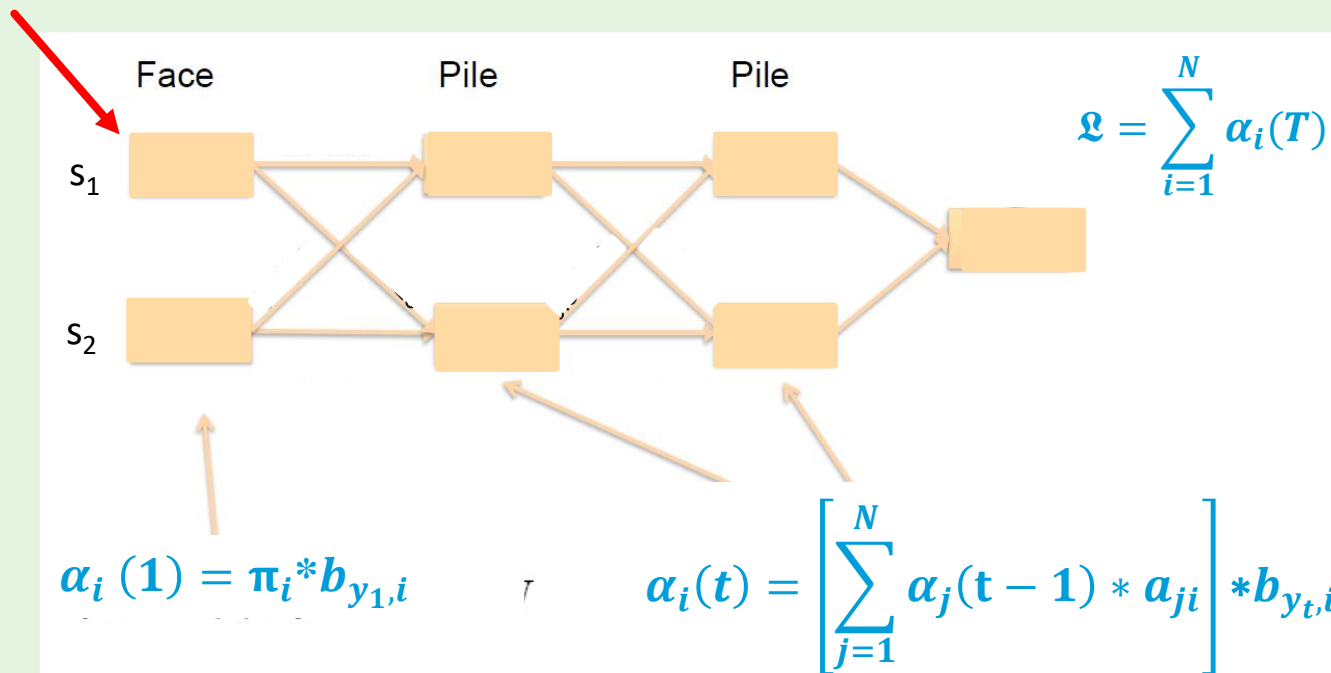
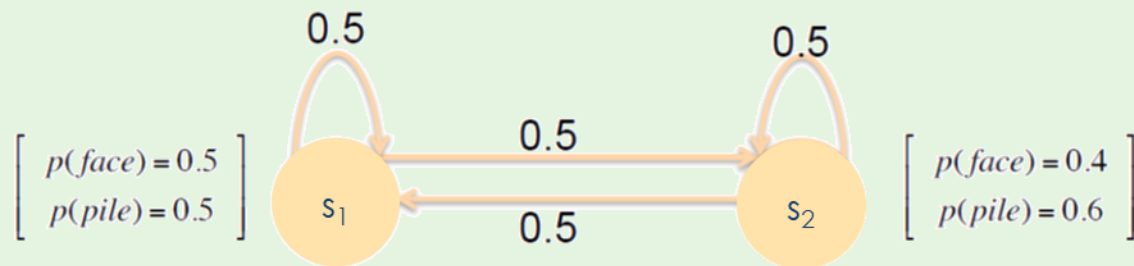
Jeu de pile ou face avec 2 pièces de monnaie (même proba initiale).

Calculer la vraisemblance de la séquence : $\mathbf{y} = \{\text{Face, Pile, Pile}\}$



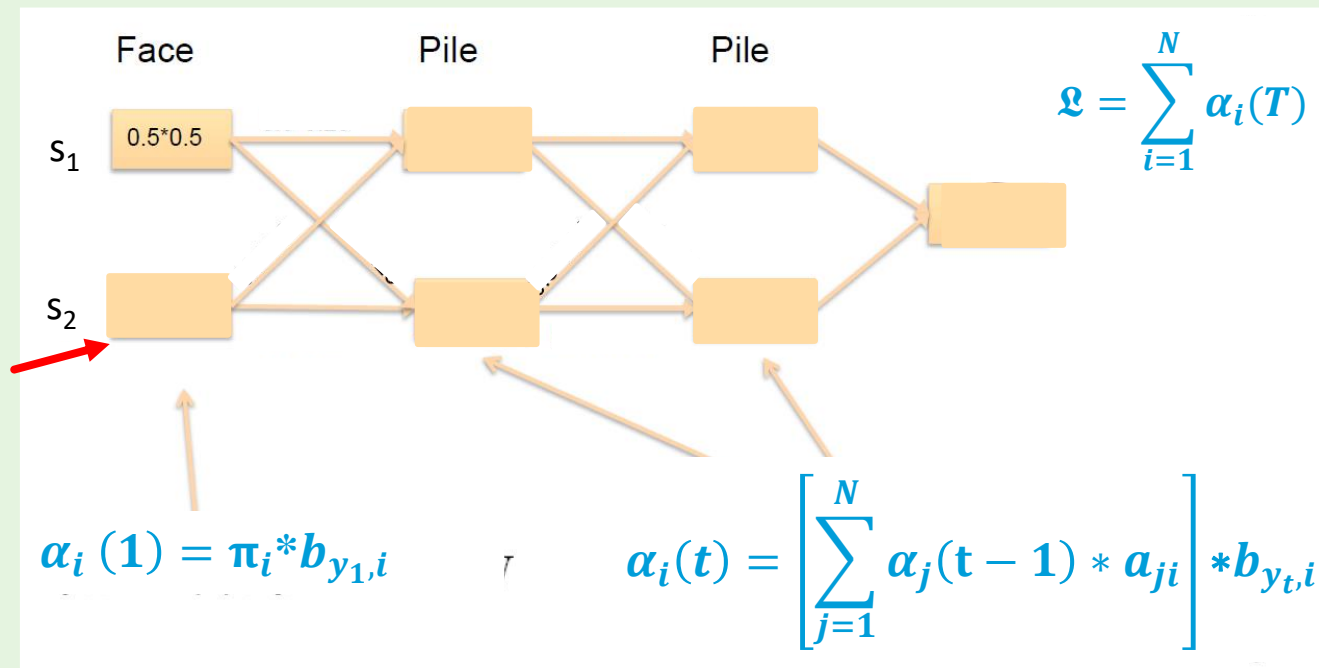
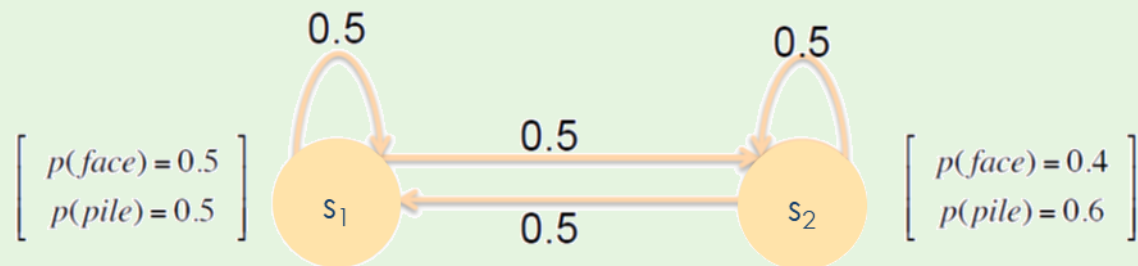
Exercice :

Jeu de pile ou face avec 2 pièces de monnaie. $\mathbf{y} = \{\text{Face, Pile, Pile}\}$



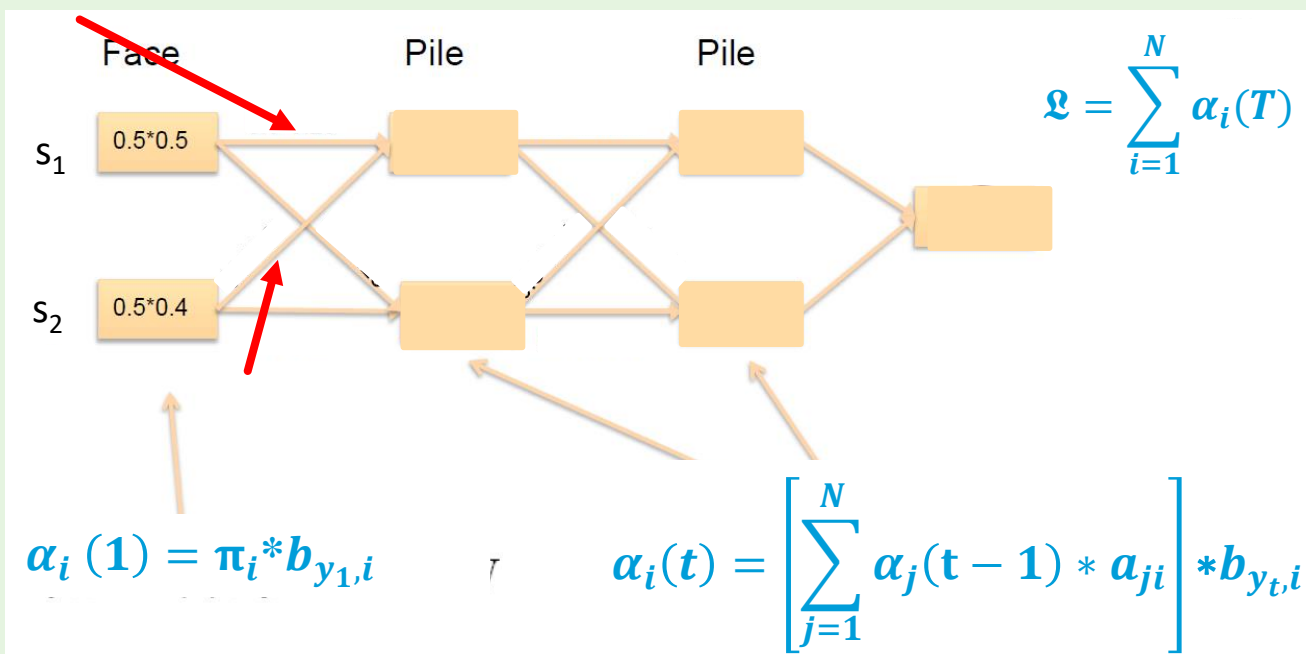
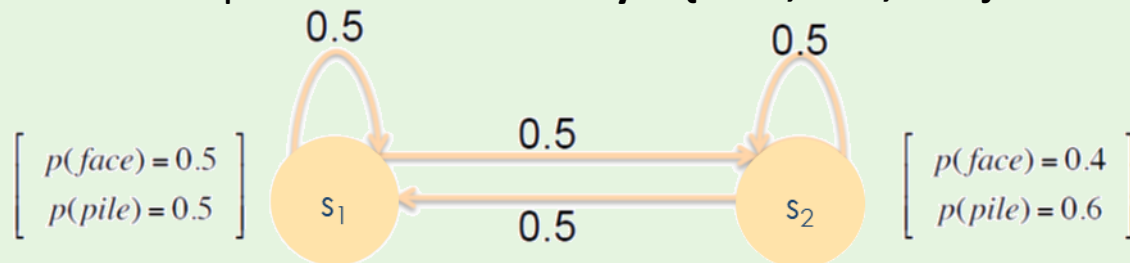
Exercice :

Jeu de pile ou face avec 2 pièces de monnaie. $\mathbf{y} = \{\text{Face, Pile, Pile}\}$



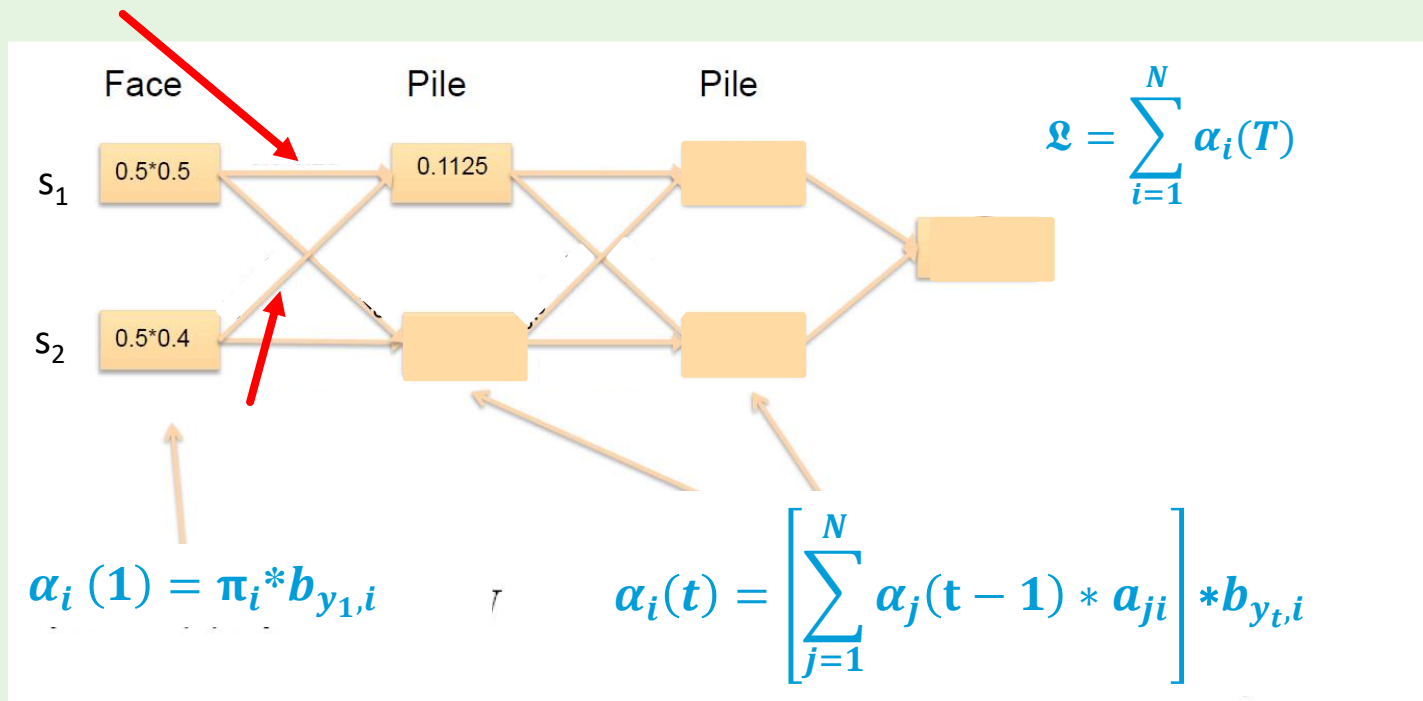
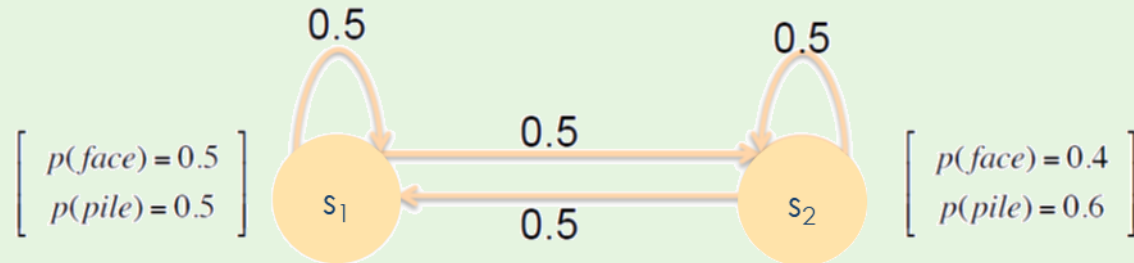
Exercice :

Jeu de pile ou face avec 2 pièces de monnaie. $\mathbf{y} = \{\text{Face}, \text{Pile}, \text{Pile}\}$



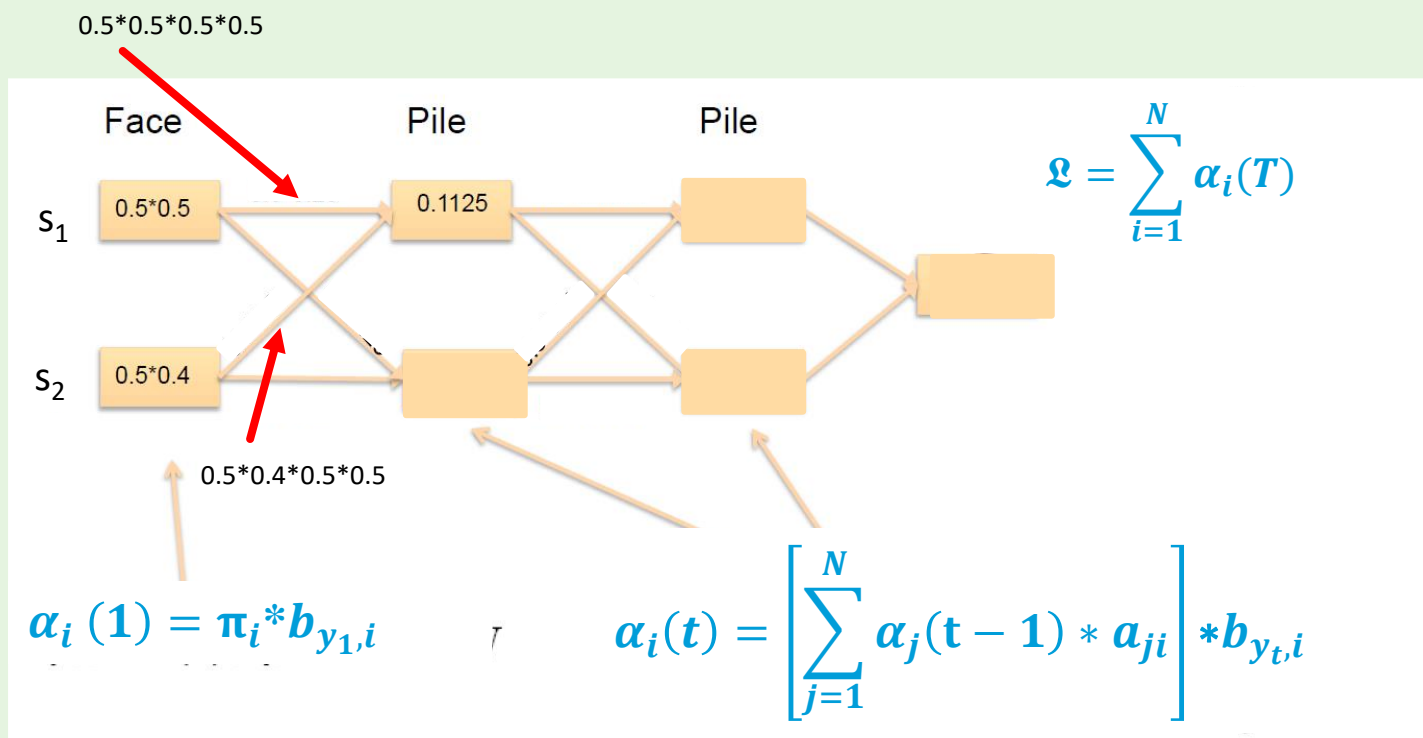
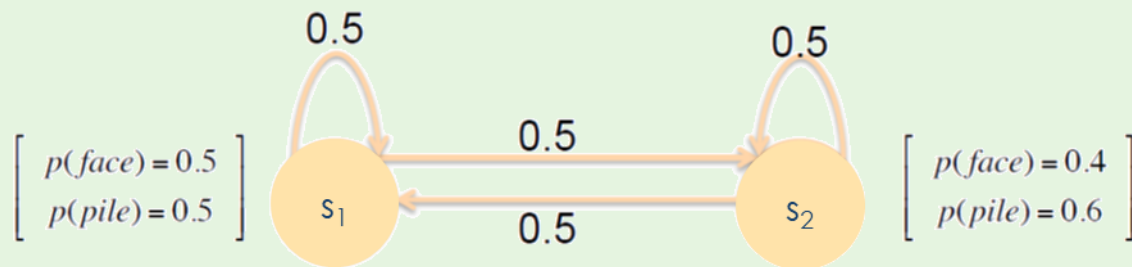
Exercice :

Jeu de pile ou face avec 2 pièces de monnaie. $\mathbf{y} = \{\text{Face, Pile, Pile}\}$



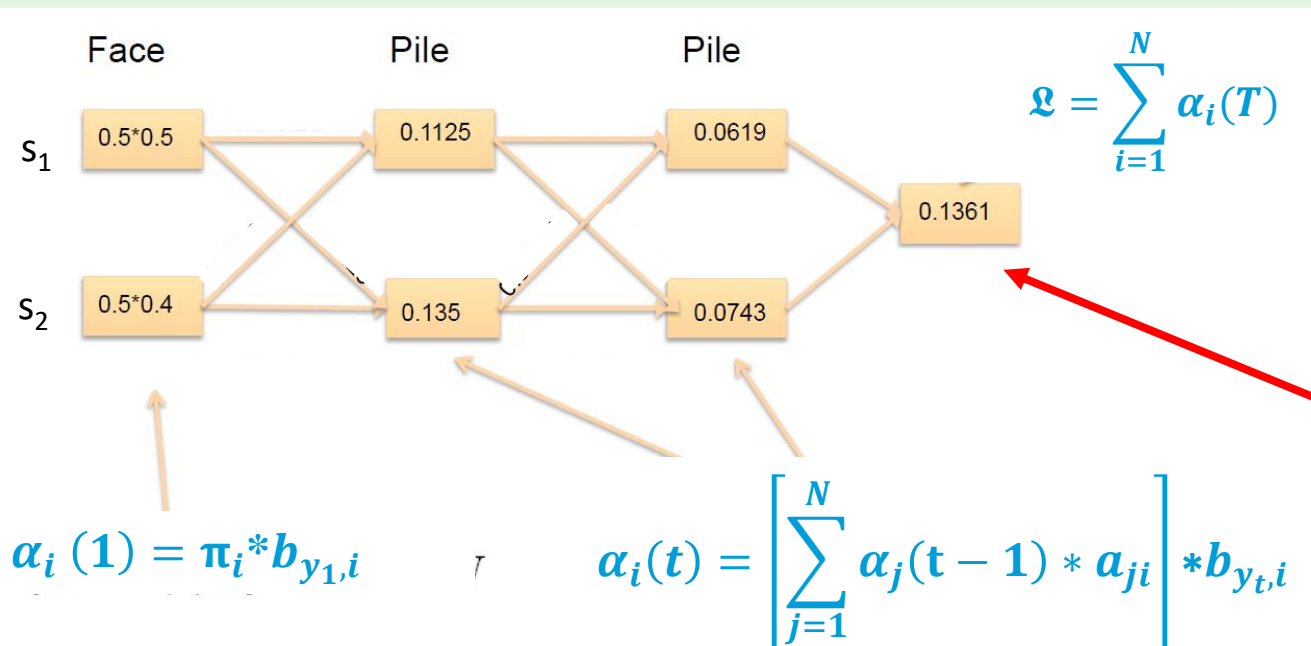
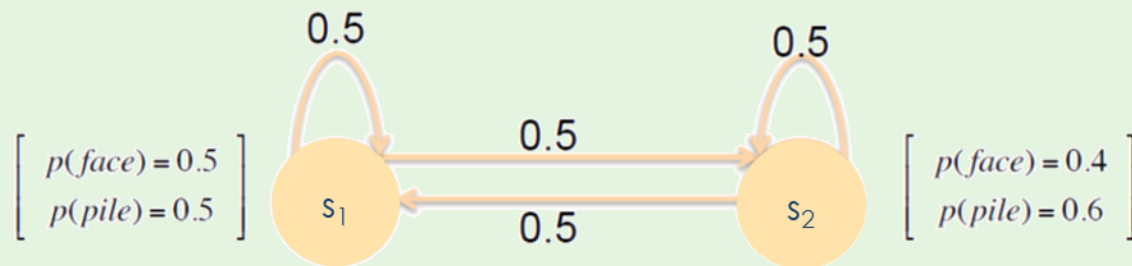
Exercice :

Jeu de pile ou face avec 2 pièces de monnaie. $\mathbf{y} = \{\text{Face}, \text{Pile}, \text{Pile}\}$



Exercice :

Jeu de pile ou face avec 2 pièces de monnaie. $\mathbf{y} = \{\text{Face}, \text{Pile}, \text{Pile}\}$



Reconnaissance, algorithme backward

On peut utiliser la même démarche mais en partant de la fin.

On définit alors $\beta_i(t)$, la probabilité d'avoir la séquence d'observation $\{y_{t+1}, \dots, y_T\}$ et l'état $x_t = s_i$:

$$\beta_i(t) = P(y_{t+1}, \dots, y_T, x_t = s_i)$$

La vraisemblance de la séquence y_1, \dots, y_T est donnée par :

$$\mathcal{Q} = P(y_1, \dots, y_T)$$

$$\mathcal{Q} = \sum_{i=1}^N \beta_i(1) b_{y_1, i} \pi_i$$

Reconnaissance, algorithme backward

Calcul des $\beta_i(t) = P(y_{t+1}, \dots, y_T, x_t = s_i)$

Initialisation :

$$\beta_i(T) = 1$$

Induction :

Déterminer $\beta_i(t-1)$ en fonction des $\beta_j(t)$

Reconnaissance, algorithme backward

Calcul des $\beta_i(t) = P(y_{t+1}, \dots, y_T, x_t = s_i)$

Initialisation :

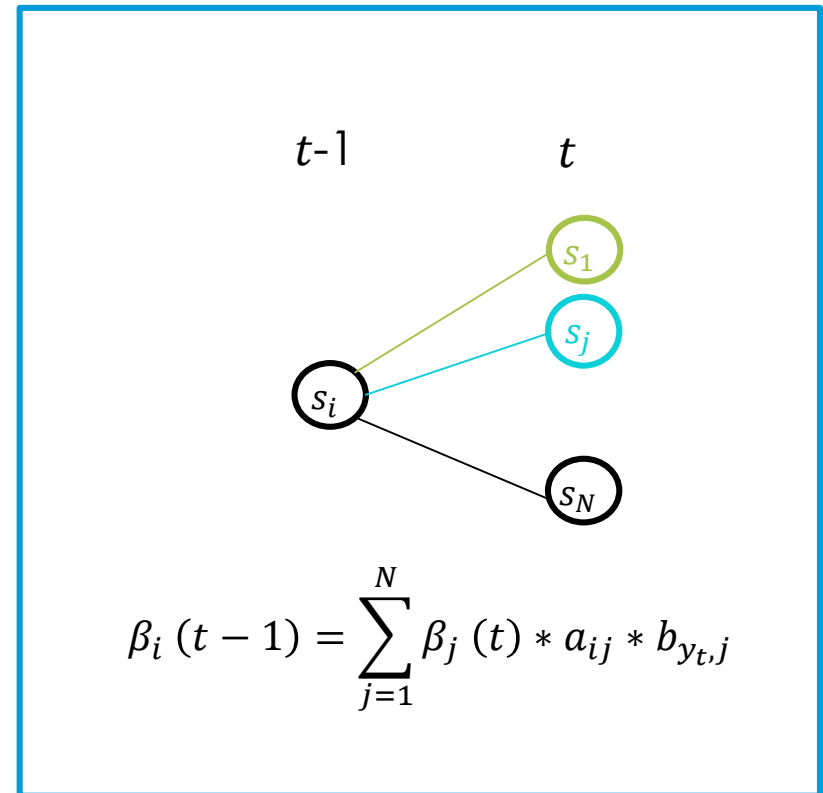
$$\beta_i(T) = 1$$

Induction :

$$\beta_i(t-1) = \sum_{j=1}^N \beta_j(t) * a_{ij} * b_{y_{t,j}}$$

Estimation de la vraisemblance :

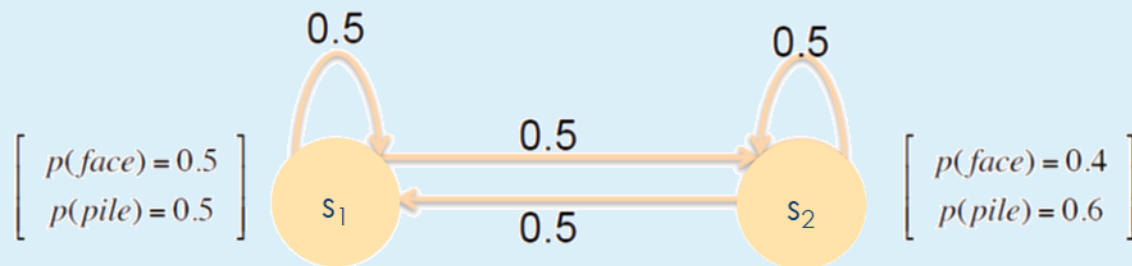
$$\mathfrak{L} = \sum_{i=1}^N \beta_i(1) b_{y_1,i} \pi_i$$



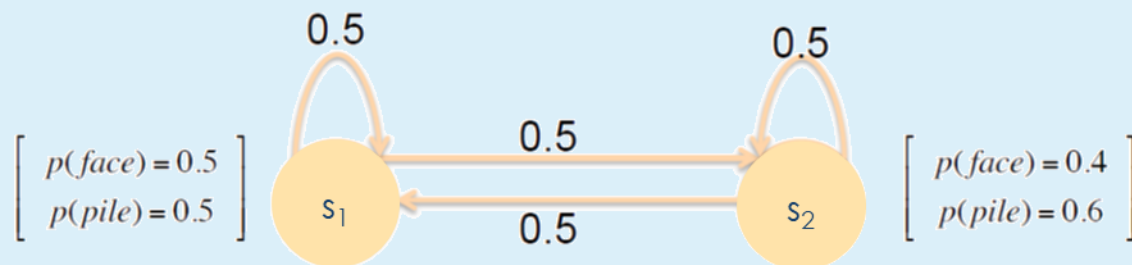
Exercice :

Jeu de pile ou face avec 2 pièces de monnaie.

Calculer la vraisemblance de la séquence $\mathbf{y} = \{\text{Face, Pile, Pile}\}$ avec l'algorithme backward

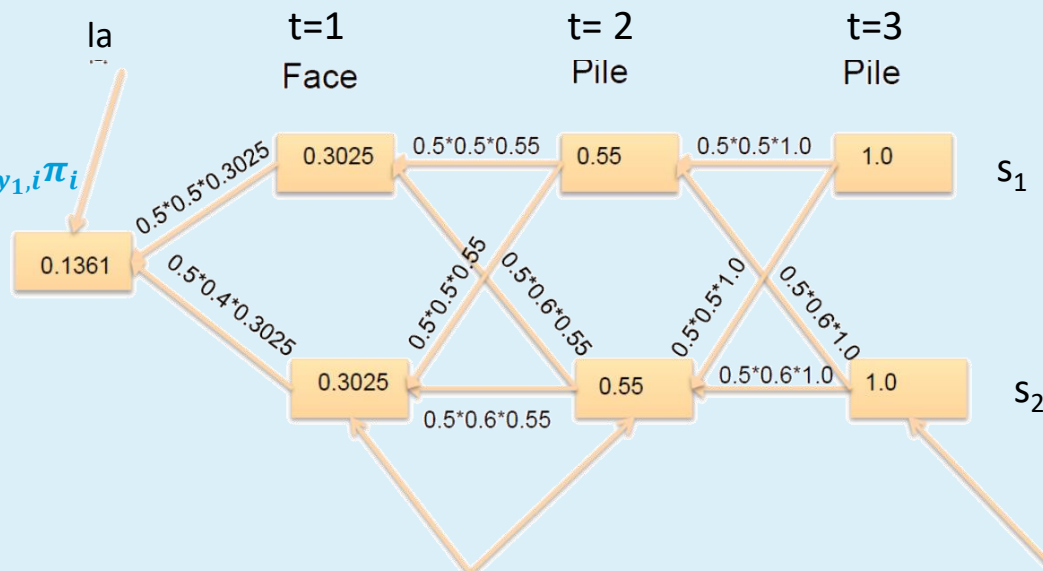


Exercice :



Estimation de la vraisemblance :

$$\mathcal{L} = \sum_{i=1}^N \beta_i(1) b_{y_{1,i}} \pi_i$$



Induction :

$$\beta_i(t-1) = \sum_{j=1}^N \beta_j(t) * a_{ij} * b_{y_{t,j}}$$

Initialisation :

$$\beta_i(T) = 1$$

Reconnaissance, algorithme forward/backward

$$\alpha_i(t) = P(y_1, \dots, y_t, x_t = s_i)$$

$$\beta_i(t) = P(y_{t+1}, \dots, y_T, x_t = s_i)$$

$$\alpha_i(t)\beta_i(t) = P(y_1, \dots, y_T, x_t = s_i)$$

Trois facons d'estimer la vraisemblance $\mathfrak{L} = P(y_1, \dots, y_T | \theta)$:

$$\mathfrak{L} = \sum_{i=1}^N \alpha_i(T)$$

$$\mathfrak{L} = \sum_{i=1}^N \beta_i(1) b_{y_1, i} \pi_i$$

$$\forall t, \mathfrak{L} = \sum_{i=1}^N \alpha_i(t) \beta_i(t)$$

Les résultats sont rigoureusement identiques

Problème 2 – Segmentation

Algorithme de Viterbi

Etant donnée une séquence y_1, \dots, y_T et un HMM θ , quelle est la séquence d'état optimale x_1, \dots, x_T qui a produit y_1, \dots, y_T ?

On introduit la variable $\delta_i(t)$: probabilité du meilleur chemin qui arrive à l'état s_i à l'instant t

$$\delta_i(t) = \max_{s_1, \dots, s_{t-1}} P(s_1, s_2, \dots, s_t = s_i, y_1, \dots, y_t)$$

On raisonne par récurrence :

$$\delta_i(1) = ??$$

Problème 2 – Segmentation

Algorithme de Viterbi

$$\delta_i(1) = \pi_i * b_{y_1,i}$$

$$\delta_i(t) = \max_{s_1, \dots, s_{t-1}} P(s_1, s_2, \dots, s_t = s_i, y_1, \dots, y_t)$$

On raisonne par récurrence :

$$\delta_i(t+1) = \left[\max_j \delta_j(t) a_{ji} \right] * b_{y_{t+1},i}$$

En gardant trace de meilleure séquence:

$$\Psi_i(t) = \left[\operatorname{argmax}_j \delta_j(t) a_{ji} \right]$$
$$\Psi_1(t) = 0$$

A la fin des itérations, le chemin associé à la valeur maximale de $\delta_i(t)$ donne la séquence d'états optimale

Exercice

On considère le HMM :

$$A = \begin{bmatrix} 0,3 & 0,5 & 0,2 \\ 0 & 0,3 & 0,7 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\pi = \begin{bmatrix} 0,6 \\ 0,4 \\ 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 & 0,5 & 0 \\ 0 & 0,5 & 1 \end{bmatrix}$$

Quelle est la séquence d'états optimale qui a produit la séquence [1, 1, 2, 2] ?

$$A = \begin{bmatrix} 0,3 & 0,5 & 0,2 \\ 0 & 0,3 & 0,7 \\ 0 & 0 & 1 \end{bmatrix} \quad \pi = \begin{bmatrix} 0,6 \\ 0,4 \\ 0 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 0,5 & 0 \\ 0 & 0,5 & 1 \end{bmatrix} \quad y = [1, 1, 2, 2] ?$$

$$\delta_i(1) = \pi_i * b_{y_1,i} \quad \delta_i(t+1) = \left[\max_j \delta_j(t) a_{ji} \right] * b_{y_{t+1},i}$$

$\delta_i(t)$	t=1	t=2	t=3	t=4
i=1	$\delta_1(1)$	$\delta_1(2)$	$\delta_1(3)$	$\delta_1(4)$
i=2	$\delta_2(1)$	$\delta_2(2)$	$\delta_2(3)$	$\delta_2(4)$
i=3	$\delta_3(1)$	$\delta_3(2)$	$\delta_3(3)$	$\delta_3(4)$

$$A = \begin{bmatrix} 0,3 & 0,5 & 0,2 \\ 0 & 0,3 & 0,7 \\ 0 & 0 & 1 \end{bmatrix} \quad \pi = \begin{bmatrix} 0,6 \\ 0,4 \\ 0 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 0,5 & 0 \\ 0 & 0,5 & 1 \end{bmatrix} \quad y = [1, 1, 2, 2] ?$$

$$\delta_i(1) = \pi_i * b_{y_1,i} \quad \delta_i(t+1) = \left[\max_j \delta_j(t) a_{ji} \right] * b_{y_{t+1},i}$$

$\delta_i(t)$	t=1	t=2	t=3	t=4
i=1	$\delta_1(1) =$ $\pi_1 * b_{1,1} =$ $0,6 * 1 =$ 0,6	$\delta_1(2)$	$\delta_1(3)$	$\delta_1(4)$
i=2	$\delta_2(1)$	$\delta_2(2)$	$\delta_2(3)$	$\delta_2(4)$
i=3	$\delta_3(1)$	$\delta_3(2)$	$\delta_3(3)$	$\delta_3(4)$

$$A = \begin{bmatrix} 0,3 & 0,5 & 0,2 \\ 0 & 0,3 & 0,7 \\ 0 & 0 & 1 \end{bmatrix} \quad \pi = \begin{bmatrix} 0,6 \\ 0,4 \\ 0 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 0,5 & 0 \\ 0 & 0,5 & 1 \end{bmatrix} \quad y = [1, 1, 2, 2] ?$$

$$\delta_i(1) = \pi_i * b_{y_1,i} \quad \delta_i(t+1) = \left[\max_j \delta_j(t) a_{ji} \right] * b_{y_{t+1},i}$$

$\delta_i(t)$	t=1	t=2	t=3	t=4
i=1	$\delta_1(1) = \pi_1 * b_{1,1} =$ 0,6	$\delta_1(2)$	$\delta_1(3)$	$\delta_1(4)$
i=2	$\delta_2(1) = \pi_2 * b_{1,2} =$ $0,4 * 0,5 =$ 0,2	$\delta_2(2)$	$\delta_2(3)$	$\delta_2(4)$
i=3	$\delta_3(1) = \pi_3 * b_{1,3} =$ $0 * 0 =$ 0	$\delta_3(2)$	$\delta_3(3)$	$\delta_3(4)$

$$A = \begin{bmatrix} 0,3 & 0,5 & 0,2 \\ 0 & 0,3 & 0,7 \\ 0 & 0 & 1 \end{bmatrix} \quad \pi = \begin{bmatrix} 0,6 \\ 0,4 \\ 0 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 0,5 & 0 \\ 0 & 0,5 & 1 \end{bmatrix} \quad y = [1, 1, 2, 2] ?$$

$$\delta_i(1) = \pi_i * b_{y_1,i}$$

$$\delta_i(t+1) = \left[\max_j \delta_j(t) a_{ji} \right] * b_{y_{t+1},i}$$

$$\max \begin{pmatrix} 0.6 * 0.3 \\ 0.2 * 0 \\ 0 * 0 \end{pmatrix} * 1 = 0.18$$

$\delta_i(t)$	t=1	t=2	t=3	t=4
i=1	$\delta_1(1) = \pi_1 * b_{1,1} = 0,6$	$\delta_1(2) = \left[\max_j \delta_j(1) a_{j1} \right] * b_{1,1} = 0,18$	$\delta_1(3)$	$\delta_1(4)$
i=2	$\delta_2(1) = \pi_2 * b_{1,2} = 0,2$	$\delta_2(2)$	$\delta_2(3)$	$\delta_2(4)$
i=3	$\delta_3(1) = \pi_3 * b_{1,3} = 0$	$\delta_3(2)$	$\delta_3(3)$	$\delta_3(4)$

?

$$A = \begin{bmatrix} 0,3 & 0,5 & 0,2 \\ 0 & 0,3 & 0,7 \\ 0 & 0 & 1 \end{bmatrix} \quad \pi = \begin{bmatrix} 0,6 \\ 0,4 \\ 0 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 0,5 & 0 \\ 0 & 0,5 & 1 \end{bmatrix} \quad y = [1, 1, 2, 2] ?$$

$$\delta_i(1) = \pi_i * b_{y_1,i}$$

$$\delta_i(t+1) = \left[\max_j \delta_j(t) a_{ji} \right] * b_{y_{t+1},i}$$

$$\max \begin{pmatrix} 0,6 * 0,5 \\ 0,2 * 0,3 \\ 0 * 0 \end{pmatrix} * 0,5 = 0,15$$

$\delta_i(t)$	t=1	t=2	t=3	t=4
i=1	$\delta_1(1) = \pi_1 * b_{1,1} =$ 0,6	$\delta_1(2) = \left[\max_j \delta_j(1) a_{j1} \right] * b_{1,1}$ =0,18	$\delta_1(3)$	$\delta_1(4)$
i=2	$\delta_2(1) = \pi_2 * b_{1,2} =$ 0,2	$\delta_2(2) =$ =0,15	$\delta_2(3)$	$\delta_2(4)$
i=3	$\delta_3(1) = \pi_3 * b_{1,3} =$ 0	$\delta_3(2)$	$\delta_3(3)$	$\delta_3(4)$

$\delta_i(t)$	$t=1$	$t=2$	$t=3$	$t=4$
$i=1$	$\delta_1(1) = \pi_1 * b_{1,1} = 0,6$	$\delta_1(2) = \left[\max_j \delta_j(1) a_{j1} \right] * b_{1,1} = 0,18$	$\delta_1(3) = \left[\max_j \delta_j(2) a_{j1} \right] * b_{21} = 0$	$\delta_1(4) = \left[\max_j \delta_j(3) a_{j1} \right] * b_{2,1} = 0$
$i=2$	$\delta_2(1) = \pi_2 * b_{1,2} = 0,2$	$\delta_2(2) = \left[\max_j \delta_j(1) a_{j2} \right] * b_{1,2} = 0,15$	$\delta_2(3) = \left[\max_j \delta_j(2) a_{j2} \right] * b_{22} = 0,045$	$\delta_2(4) = \left[\max_j \delta_j(3) a_{j2} \right] * b_{2,2} = 0,0067$
$i=3$	$\delta_3(1) = \pi_3 * b_{1,3} = 0$	$\delta_3(2) = \left[\max_j \delta_j(1) a_{j3} \right] * b_{1,3} = 0$	$\delta_3(3) = \left[\max_j \delta_j(2) a_{j3} \right] * b_{23} = 0,105$	$\delta_3(4) = \left[\max_j \delta_j(3) a_{j3} \right] * b_{2,3} = 0,105$

Valeur maximale de $\delta_i(4)$
Suite d'états optimaux : **1 2 3 3**

Problème 3 - Apprentissage: algorithme de Baum-Welch

Etant donnée une séquence $y_{1:T}$, estimer les paramètres θ du HMM qui maximisent la vraisemblance d'apparition de $y_{1:T}$

Quatre variables temporaires :

- ✓ Variable forward : $\alpha_i(t) = P(y_1, \dots, y_t, x_t = s_i)$
- ✓ Variable backward : $\beta_i(t) = P(y_{t+1}, \dots, y_T, x_t = s_i)$
- ✓ $\gamma_i(t)$: probabilité d'être dans l'état s_i à t sachant l'ensemble des observations y_1, \dots, y_T :

$$\gamma_i(t) = P(x_t = s_i | y_1, \dots, y_T) = \frac{\alpha_i(t)\beta_i(t)}{\Omega}$$

- ✓ $\xi_{ij}(t)$: probabilité d'être dans l'état s_i à t et s_j à $t + 1$ sachant l'ensemble des observations y_1, \dots, y_T

$$\xi_{ij}(t) = P(x_t = s_i, x_{t+1} = s_j | y_1, \dots, y_T) = \frac{a_{i,j} \alpha_i(t) \beta_j(t+1) b_{y_{t+1},j}}{\Omega}$$

La mise à jour est ensuite faite avec :

$$\pi_i = \gamma_i(1)$$

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

$$b_i(o_k) = \frac{\sum_{t=1}^T \alpha_i(t) P(y_t / s_i) \mathbf{1}_{y_t=o_k}}{\sum_{t=1}^T \alpha_i(t) \beta_i(t)} \text{ où } \mathbf{1}_{y_t=o_k} = \begin{cases} 1 & \text{si } y_t = o_k \\ 0 & \text{sinon} \end{cases}$$

D'où l'algorithme:

Algorithme de Baum-Welch

```

1:  repeat
2:      Appliquer l'algorithme forward et estimer les  $\alpha_i(t)$ 
3:      Appliquer l'algorithme backward estimer les  $\beta_i(t)$ 
3:      Calcul de la vraisemblance :  $\mathcal{L} = \sum_{i=1}^N \alpha_i(T) \beta_i(T)$ 
5:      for  $t = 1$  to  $T-1$  do
6:          for  $i = 1$  to  $N$  do
7:              for  $j = 1$  to  $N$  do
8:                   $\xi_{ij}(t) = \frac{P(s_j/s_i)\alpha_{i(t)}\beta_{j(t+1)}P(y_{t+1}/s_j)}{\mathcal{L}}$ 
9:              end for
10:              $\gamma_i(t) = \frac{\alpha_i(t)\beta_i(t)}{\mathcal{L}}$ 
11:         end for
12:     end for
13:     Mise à jour de  $\pi_i = \gamma_i(1)$ 
14:     Mise à jour de  $a_{ij} = \frac{\sum_{t=1}^{T-1} \xi_{ij}(t)}{\sum_{t=1}^T \gamma_i(t)}$ 
15:     Mise à jour de  $b_i(o_k) = \frac{\sum_{t=1}^T \alpha_i(t)\beta_i(t)\mathbf{1}_{y_t=o_k}}{\sum_{t=1}^T \alpha_i(t)\beta_i(t)}$  où  $\mathbf{1}_{y_t=o_k} = \begin{cases} 1 & \text{si } y_t = o_k \\ 0 & \text{sinon} \end{cases}$ 
16:
17: until :  $\mathcal{L}$  n'augmente plus ou que le nombre maximum d'itérations est atteint
  
```

Classification par HMM

Connaissant les paramètres θ d'un HMM, la vraisemblance \mathfrak{L} mesure la probabilité que la séquence d'observation y_1, \dots, y_T ait été générée par ce HMM

$$\mathfrak{L} = P(y_1, \dots, y_T / \theta)$$

Apprentissage

- On apprend un HMM de paramètre θ_c pour chaque classe c

Reconnaissance

Pour reconnaître la classe d'une nouvelle séquence y_1, \dots, y_T

- Calcule pour chaque classe la vraisemblance $\mathfrak{L}_c = P(y_1, \dots, y_T / \theta_c)$ que le HMM c ait généré la séquence y_1, \dots, y_T
- La séquence reconnue par classification bayésienne :

$$P(\theta_c / y_1, \dots, y_T) = \frac{P(y_1, \dots, y_T / \theta_c) \cdot P(\theta_c)}{P(y)}$$

FIN