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

- Modéliser les séquences (Hidden Markov Model, HMM)

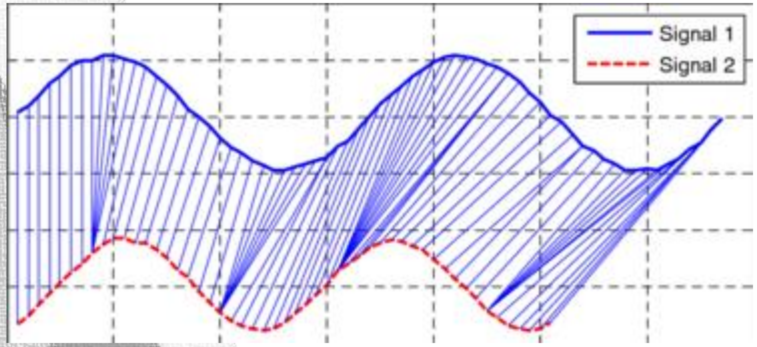
Dynamic Time Warping, DTW

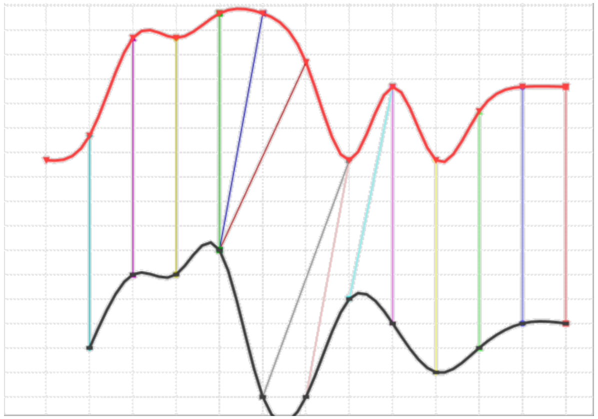
Il faut établir une distance entre deux séquences

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





Dynamic Time Warping, DTW

Il faut donc aligner temporellement les signaux

Soit

$\mathbf{G} = \{\mathbf{g_1}, \mathbf{g_2}, \ldots, \mathbf{g_k}\}$ et $\mathbf{H} = \{\mathbf{h_1}, \mathbf{h_2}, \ldots, \mathbf{h_l}\}$ deux séquences de longueur *k* et *l* et de dimension *n* (chaque $\mathbf{g_i}$ ou $\mathbf{h_i}$ est de dimension *n*)

$\mathbf{d(g_i, h_j)}$: distance entre $\mathbf{g_i}$ ou $\mathbf{h_j}$

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

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

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

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

Dynamic Time Warping, DTW

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

j

i

	h_1	h_2	h_3	h_4	h_5
g_1					
g_2			D(i-1,j-1)	D(i-1,j)	
g_3			D(i,j-1)	$D(i,j)$?	
g_4					

Dynamic Time Warping, DTW

Algorithme DTW

$D(1, 1) = (g_1, h_1)$

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)

Dynamic Time Warping, DTW

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													

Dynamic Time Warping, DTW

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

En jaune, le chemin minimal qui nous donne l’alignement des signaux
La distance entre les signaux vaut 7

Dynamic Time Warping, DTW

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

En jaune, le chemin minimal qui nous donne l'alignement des signaux
La distance entre les signaux vaut 7

Dynamic Time Warping, DTW

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

En jaune, le chemin minimal qui nous donne l'alignement des signaux
La distance entre les signaux vaut 7

Hidden Markov Model, HMM

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

$$\begin{tikzcd}
{} \arrow[r, dashed] & x_{t-1} \arrow[r] \arrow[d] & x_t \arrow[r] \arrow[d] & x_{t+1} \arrow[r, dashed] \arrow[d] & {} \\
& y_{t-1} & y_t & y_{t+1} &
\end{tikzcd}$$

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

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

Hidden Markov Model, HMM

Un HMM est entièrement défini par θ = (A, B, Π):

- ✓ La matrice de transition de taille NxN (N états) :

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

- ✓ Les probabilités initiales des états cachés Pi de taille 1xN:

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

- ✓ La matrice d'observation de taille MxN :

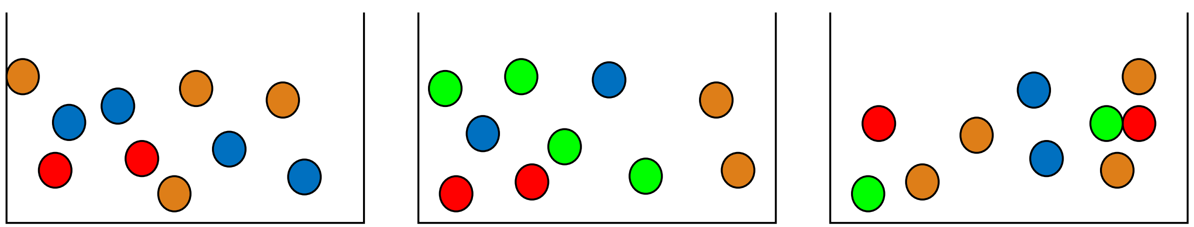
$$B = \{b_{ki}\} = \{P(y_t = o_k | x_t = s_i)\} = \{P(o_k | s_i)\}$$

Exemple:

Considérons un ensemble de ***N*** boites.
Dans chaque boite se trouvent des balles de ***M*** couleurs différentes.

Les observations sont générées ainsi : une personne choisit une boite 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 boite avec la règle suivante : elle a deux fois plus de chance de rester dans la même boite que de changer de boite. Si elle change de boite, les deux autres boites sont équiprobables.

Donner les paramètres $\boldsymbol{\theta = (A, B, \Pi)}$ de la chaine de Markov qui modélise ce processus



Hidden Markov Model, HMM

Exemple:

On aura N=3 états, un pour chaque boite
Il y aura M= 4 observations (une pour chaque couleur de balle)

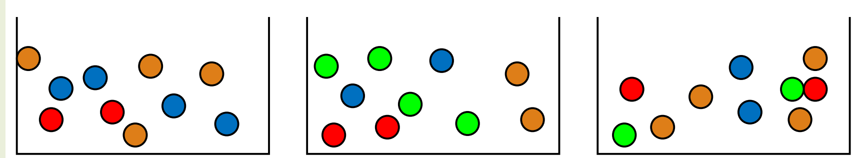
une personne choisit une boite initiale en fonction d'une loi uniforme
$\Pi = \{\pi_i\} = \{P(x_1 = s_i)\} = \{1/3;\ 1/3;\ 1/3\}$

La couleur de la balle constitue l'observation

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

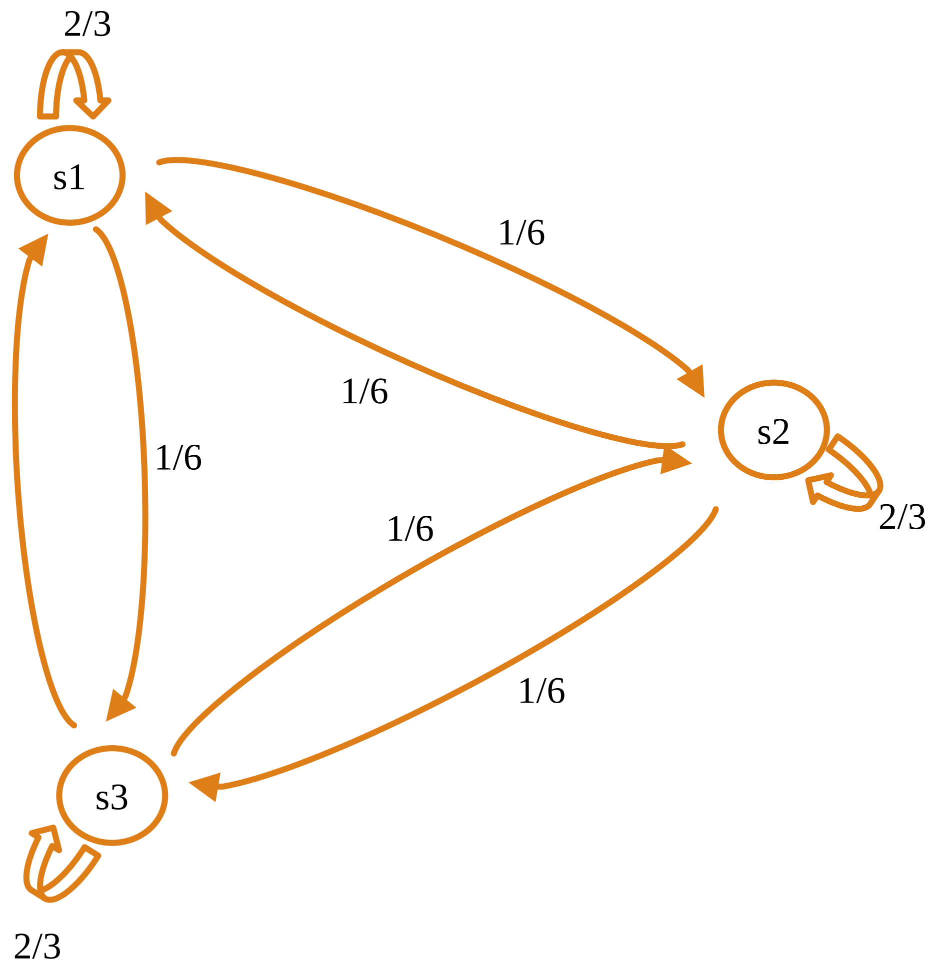
La personne choisit alors une autre boite avec la règle suivante : elle a deux fois plus de chance de rester dans la même boite que de changer de boite. Si elle change de boite, les deux autres boites sont équiprobables.

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



Hidden Markov Model, HMM

Exemple:



$$\begin{array}{l} Bleu \\ Ocre \\ Rouge \\ Vert \end{array}\begin{pmatrix} 4/10 \\ 4/10 \\ 2/10 \\ 0 \end{pmatrix}$$

$$\begin{array}{l} Bleu \\ Ocre \\ Rouge \\ Vert \end{array}\begin{pmatrix} 2/10 \\ 2/10 \\ 2/10 \\ 4/10 \end{pmatrix}$$

$$\begin{array}{l} Bleu \\ Ocre \\ Rouge \\ Vert \end{array}\begin{pmatrix} 2/10 \\ 4/10 \\ 2/10 \\ 2/10 \end{pmatrix}$$

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 $y_{1:T}$? (**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 $y_{1:T}$ (**Baum-Welch**)

Classification par HMM

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

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

Apprentissage

- On apprend un HMM de paramètre $\boldsymbol{\theta c}$ pour chaque classe $\boldsymbol{c}$

Reconnaissance

Pour reconnaitre la classe d'une nouvelle séquence $\boldsymbol{y_1, \dots, y_T}$

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

- Le séquence reconnue par classification bayésienne :

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

Problème :
Comment estimer la vraisemblance $\mathfrak{L}$ qu'un HMM ait généré la séquence d'observation $y_1, \ldots, y_T$?
➔ **algorithme forward ou backward**

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

$$\boldsymbol{\alpha_i\,(t)\ =\ P(y_1, \ldots, y_t, x_t\ = s_i)}$$

La vraisemblance de la séquence $y_1, \ldots, y_T$ connaissant les paramètres θ du HMM est donnée par :

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

Exprimer $\mathfrak{L}$ en fonction des $\alpha_i\,(t)$

Reconnaissance, algorithme forward

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

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

La vraisemblance de la séquence $y_1, \ldots, y_T$ du HMM est donnée par :

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

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

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

Hidden Markov Model, HMM

Reconnaissance, algorithme forward

Mais comment estimer les α_i (T) $= P(y_1, \ldots, y_T, x_T = s_i)$
Commençons par

α_1 (1) $=?$

α_{i} (1) $=$

Reconnaissance, algorithme forward

Mais comment estimer les α_i (T) $= P(y_1, \ldots, y_T, x_T = s_i)$

Commençons par

$$\alpha_1\ (1) = P(y_1, x_1 = s_1)$$
$$\alpha_1\ (1) = \mathrm{P}(y_1 | x_1 = s_1) * \mathrm{P}(x_1 = s_1)$$
$$\alpha_1\ (1) = b_{y_1,1}\ \pi_1$$

Et d'une manière générale

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

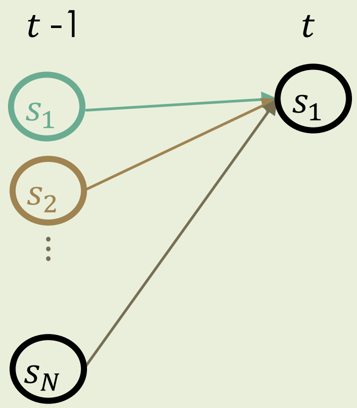
Reconnaissance, algorithme forward

Estimer

$\alpha_{\mathrm{i}}(t)$

SORBONNE UNIVERSITÉ

Reconnaissance, algorithme forward



On était en s_1 à t=1

On était en s_2 à t=1

Au temps t, on aura

α_1 (t) $= P(y_1, .., y_t, x_t \; = s_1)$

$= P(y_1, .., y_{t-1}, x_{t-1} = \boldsymbol{s_1}) * P(s_1/\boldsymbol{s_1})$*P$(y_t/s_1)$ + $P(y_1, .., y_{t-1}, x_{t-1} = \boldsymbol{s_2}) * P(s_1/\boldsymbol{s_2})$*P$(y_t/s_1)$+…

$= \alpha_1$ (t − 1) $* a_{11} * b_{y_t,1}$ + α_2 (t − 1) $* a_{21} * b_{y_t,1}$+…

$= \left[\sum_{J=1}^{N} \alpha_j \text{ (t − 1)} * a_{j1}\right] * b_{y_t,1}$

Et d'une manière générale

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

!! $a_{ij} = P(x_t = s_j/x_{t-1} = \boldsymbol{s_i})$

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 :

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

Reconnaissance, algorithme backward

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

$$\mathfrak{L} = P(y_1, \dots, y_T, x_1 = s_1) + P(y_1, \dots, y_T, x_1 = s_2) + \dots + P(y_1, \dots, y_T, x_1 = s_N)$$

$$\mathfrak{L} = P(y_2, \dots, y_T, x_1 = s_1) P(y_1, x_1 = s_1) + P(y_2, \dots, y_T, x_1 = s_2) P(y_1, x_1 = s_2) + \dots$$

$$\mathfrak{L} = P(y_2, \dots, y_T, x_1 = s_1) P(y_1 / x_1 = s_1) \mathrm{P}(s_1) + P(y_2, \dots, y_T, x_1 = s_2) P(y_1 / x_1 = s_2) \mathrm{P}(s_2) + \dots$$

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

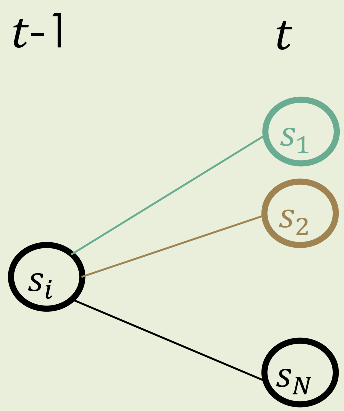
Reconnaissance, algorithme backward

Il va falloir estimer les $\beta_i\,(t) \;=\; P(y_{t+1},\ldots,y_T, x_t \;= s_i)$ en commençant de la fin :

Initialisation $\boldsymbol{\beta_i\,(T)\;=1}$

$\beta_i\,(t-1) \;= P(y_t,\ldots,y_T, x_{t-1} \;= s_i) = ?$

Reconnaissance, algorithme backward



on est en s_1 à t

on est en s_2 à t

$$\beta_i\,(t-1) = P(y_t, \ldots, y_T, x_{t-1} = s_i)$$

$$\beta_i\,(t-1) = P(y_{t+1}, \ldots, y_T, x_t = s_1) * P(s_1/s_i) * P(y_t/\,s_1) + P(y_{t+1}, \ldots, y_T, x_t = s_2) * P(s_2/\,s_i) * P(y_t/s_2) + \cdots$$

$$\beta_i\,(t-1) = \beta_1\,(t) * a_{i1} * b_{y_t,1} + \beta_2\,(t) * a_{i2} * b_{y_t,2}$$

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

Hidden Markov Model, HMM

Exercice

On considère le HMM :

$$A = \begin{bmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{bmatrix}$$

$$\pi = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.3 & 0.7 \\ 0.6 & 0.2 \\ 0.1 & 0.1 \end{bmatrix}$$

Combien y a-t-il d'états ? Combien y a-t-il d'observations ?
Quelle est la vraisemblance de la séquence y=[1 1 3 2] avec l'algorithme forward ?

Hidden Markov Model, HMM

Exercice

On considère le HMM :

$$A = \begin{bmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{bmatrix} \quad \pi = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad B = \begin{bmatrix} 0.3 & 0.7 \\ 0.6 & 0.2 \\ 0.1 & 0.1 \end{bmatrix}$$

y=[1 1 3 2] ?

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

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

$\alpha_i(t)$	t=1	t=2	t=3	t=4
i=1				
i=2				

Exercice

On considère le HMM :

$$A = \begin{bmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{bmatrix} \quad \pi = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad B = \begin{bmatrix} 0.3 & 0.7 \\ 0.6 & 0.2 \\ 0.1 & 0.1 \end{bmatrix}$$

y=[1 1 3 2] ?

$\beta_i\ (T)\ = 1$

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

$\alpha_{\mathrm{i}}(t)$	t=1	t=2	t=3	t=4
i=1	$\boldsymbol{\alpha_1(1) =}$ $\pi(1)$*B(1,1) 0.3	$\boldsymbol{\alpha_1(2) =}$ $[\alpha_1(1) * A(1,1)$	$\boldsymbol{\alpha_1(3) =}$ $[\alpha_1(2) * A(1,1)$ $+ \alpha_2(2)$	$\boldsymbol{\alpha_1(4) =}$ 0.043
i=2	$\boldsymbol{\alpha_2(1) =}$ $\pi(2)$*B(1,2) 0	$\boldsymbol{\alpha_2(2) =}$ $[\alpha_1(1) * A(1,2)$	$\boldsymbol{\alpha_2(3) =}$ 0.057	$\boldsymbol{\alpha_2(4) =}$ 0.011

$$\mathfrak{L}(y_1, \ldots, y_T/\theta) = \sum_{i=1}^{N} \alpha_i(T) = 0.054$$

Reconnaissance, algorithme forward/backward

$$\alpha_i\,(\mathrm{t}) = P(y_1, \dots, y_t, x_t \; = s_i)$$
$$\beta_i\,(t) \;=\; P(y_{t+1}, \dots, y_T, x_t \; = s_i)$$

$$\alpha_i\,(\mathrm{t})\beta_i\,(t) = P(y_1, \dots, y_T, x_t \; = s_i)$$

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

Algorithme de Viterbi

Etant donné $y_{1:T}$ et θ, quelle est *la* séquence d'état optimale $x_{1:T}$ qui a produit $y_{1:T}$?

$\delta_i(t)$: 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 :

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

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

En gardant trace de meilleure séquence:

$$\Psi_i(t) = \left[arg\max_j \delta_j(t) a_{ji}\right]$$

$$\Psi_1(t) = 0$$

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] ?

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

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

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

SORBONNE UNIVERSITÉ

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 le modèle θ et l'ensemble des observations $y_1, \dots, y_T$:

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

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

$$\xi_{ij}(t) = P\left(x_t = s_i, x_{t+1} = s_j \middle| y_1, \dots, y_T\right) = \frac{P\left(s_j/s_i\right)\alpha_i(t)\beta_j(t+1)P(y_{t+1}/s_j)}{\mathfrak{L}}$$

Hidden Markov Model, HMM

La mise à jour est ensuite faite avec :

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

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

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

D'où l'algorithme:

Algorithme de Baum-Welch

```
repeat
    Appliquer l'algorithme forward et estimer les α_i(t)
    Appliquer l'algorithme backward estimer les β_i(t)
    Calcul de la vraisemblance : 𝔏 = ∑_{i=1}^{N} α_i(T) β_i(T)
    for t = 1 to T-1 do
        for i = 1 to N do
            for j = 1 to N do
                ξ_ij(t) = P(s_j/s_i) α_i(t) β_j(t+1) P(y_{t+1}/s_j) / 𝔏
            end for
            γ_i(t) = α_i(t) β_i(t) / 𝔏
        end for
    end for
    Mise à jour de π_i = γ_i(1)
    Mise à jour de a_ij = P(s_j, s_i) = ∑_{t=1}^{T-1} ξ_ij(t) / ∑_{t=1}^{T} γ_i(t)
    Mise à jour de b_i(o_k) = ∑_{t=1}^{T} α_i(t)β_i(t)1_{y_t=o_k} / ∑_{t=1}^{T} α_i(t)β_i(t)   où 1_{y_t=o_k} = {1 si y_t = o_k ; 0 sinon}

until : 𝔏 n'augmente plus ou que le nombre maximum d'itérations est atteint
```

Formules :

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

$$\xi_{ij}(t) = \frac{P(s_j/s_i)\alpha_{i(t)}\beta_j(t+1)P(y_{t+1}/s_j)}{\mathfrak{L}}$$

$$\gamma_i(t) = \frac{\alpha_i(t)\beta_i(t)}{\mathfrak{L}}$$

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

$$a_{ij} = P(s_j, s_i) = \frac{\sum_{t=1}^{T-1} \xi_{ij}(t)}{\sum_{t=1}^{T} \gamma_i(t)}$$

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

HMMs continus

Dans le cas où les séquences d’observation sont continues, la probabilité d’observation de chaque état est modalisée par un mélange de gaussiennes.