SI340
Module Parole

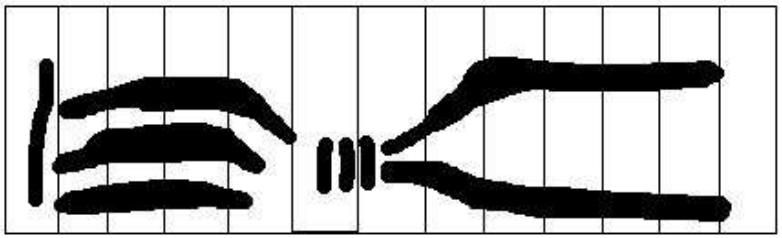
Alignement temporel et Programmation dynamique (DTW)

Gaël RICHARD

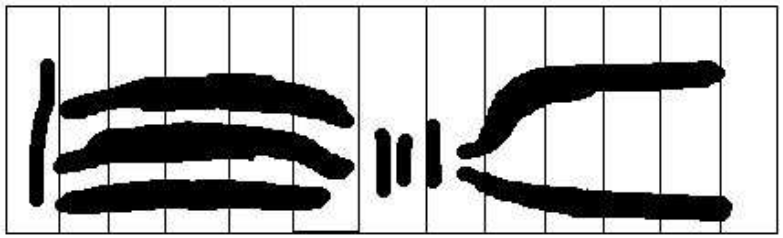
Mai 2010

Distorsions temporelles

- **Un même locuteur ne peut pas prononcer plusieurs fois une même séquence vocale avec exactement le même rythme et la même durée totale**



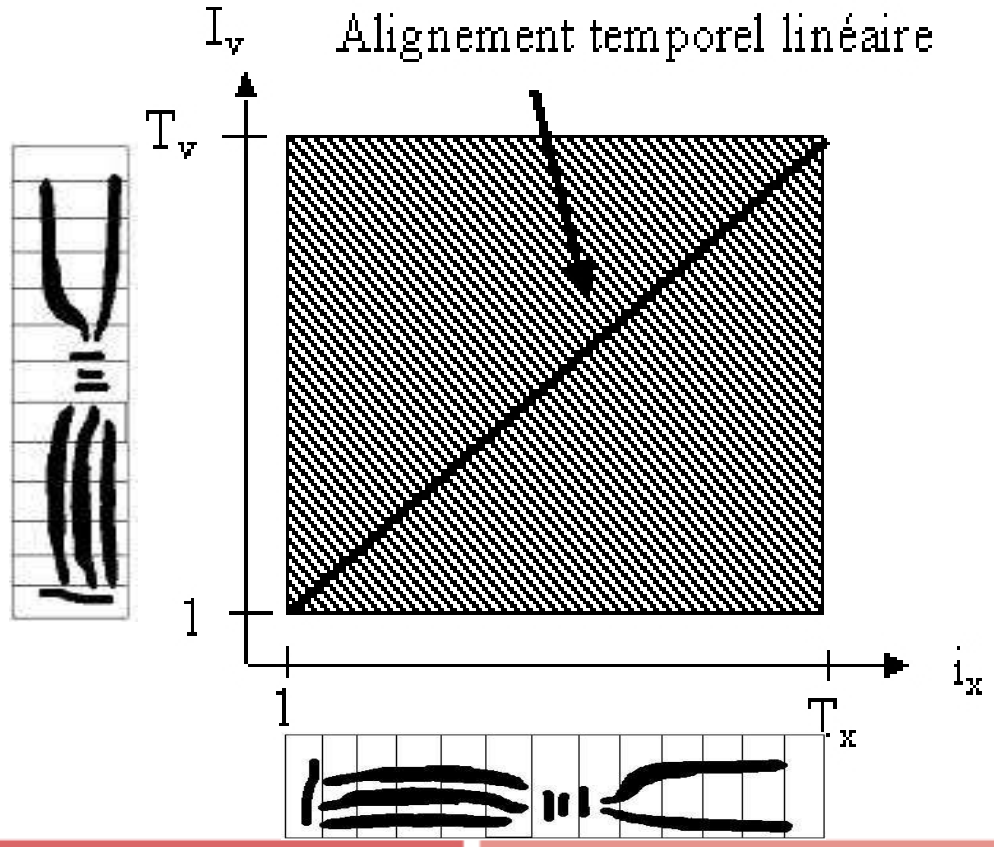
- **Les échelles temporelles de deux occurences d'un même mot ne coïncident pas**



- **Les suites de vecteurs issus de la paramétrisation ne peuvent pas être comparées entre elles**

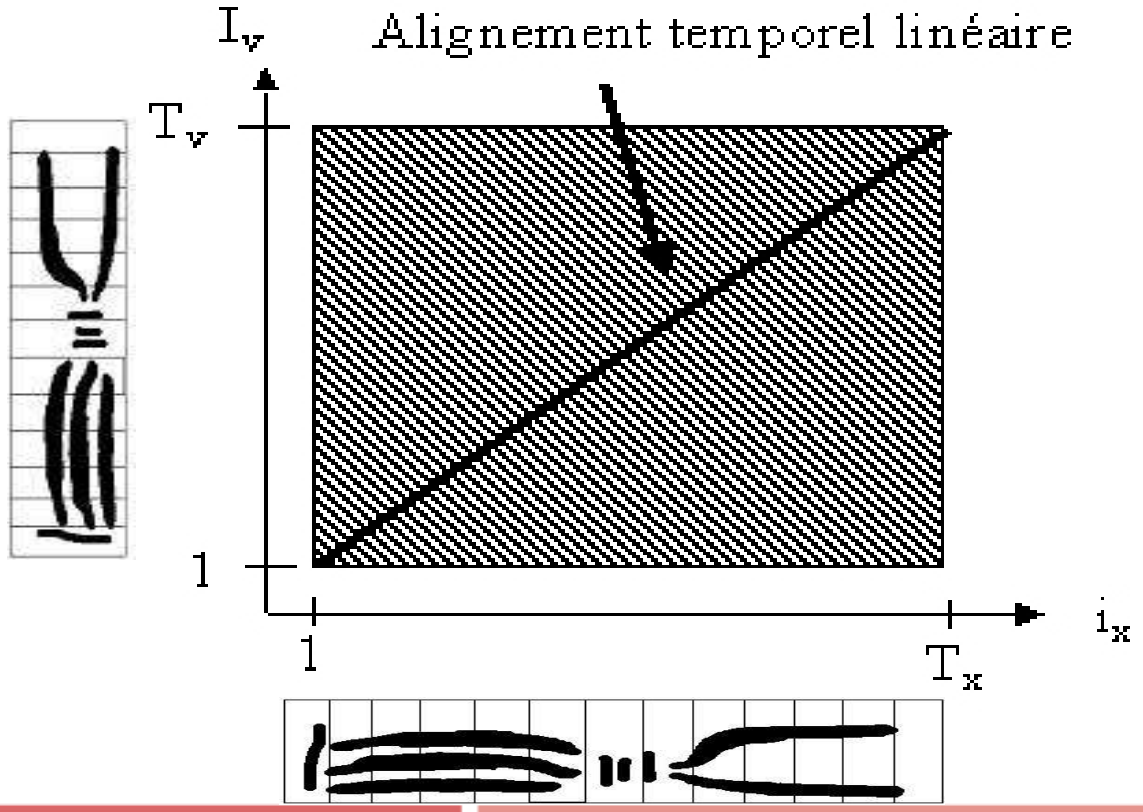
Alignement

- Cas (*irréaliste*) où les deux séquences sont prononcées avec exactement le même rythme et la même durée (T_y=T_x)



Alignement temporel linéaire

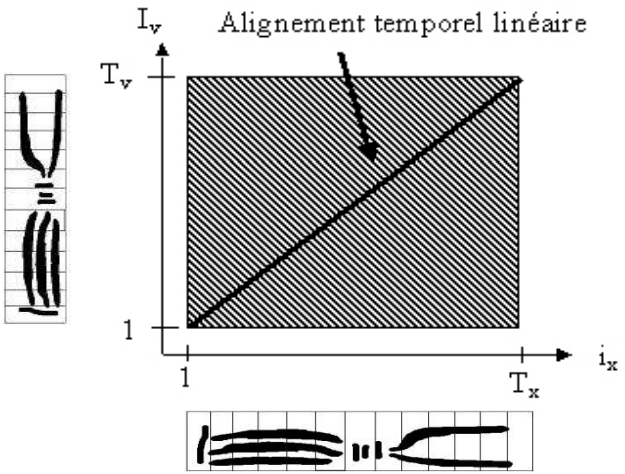
- Cas (*un peu plus réaliste*) où la déformation est linéaire: un mot et chacun de ses segments est prononcé plus rapidement et dans la même proportion



Alignement temporel linéaire

- **Distance entre les séquences**

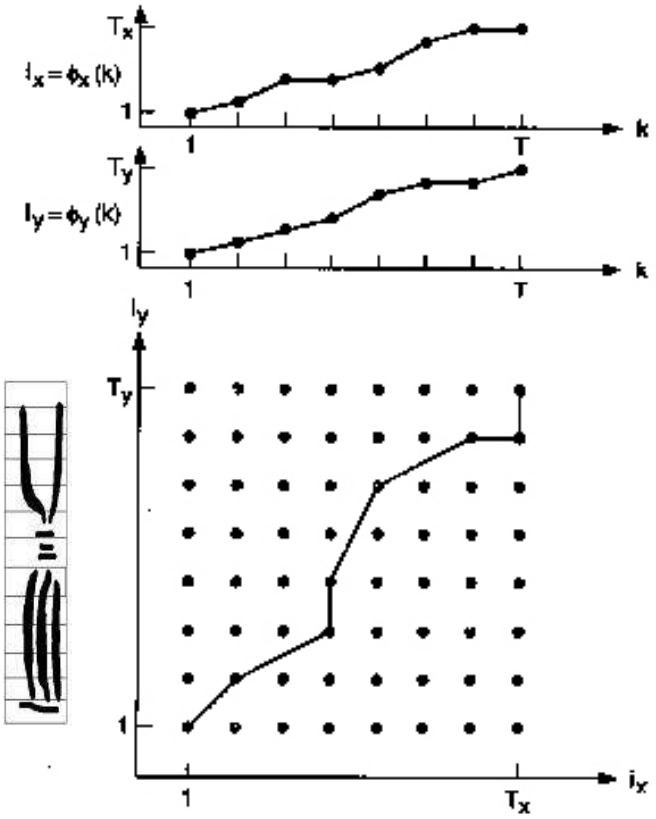
$$d(\chi,\xi) = \sum_{i_x=1}^{T_x} d(i_x, i_y)$$



$$i_y = \frac{T_y}{T_x} i_x$$

Alignement temporel dynamique

- Cas (***beaucoup plus réaliste***) où la **déformation** entre les séquences est **dynamique**



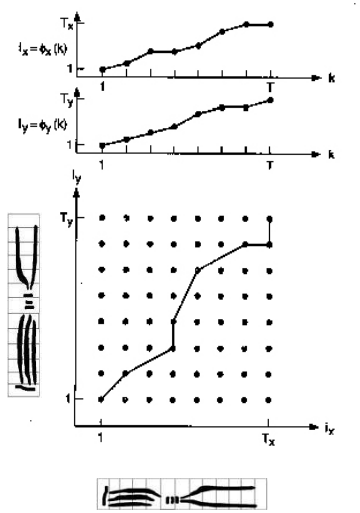
Alignement temporel dynamique

- **Fonctions de déformation**

$$i_x = \phi_x(k) \text{ pour } k = 1,2,\ldots,T$$
$$i_y = \phi_y(k) \text{ pour } k = 1,2,\ldots,T$$

- **Mesure de similarité entre les séquences**

$$d_\phi(\chi,\xi) = \sum_{k=1}^{T} d(\phi_x(k),\phi_y(k)))m(k)/M_\phi$$



- **Choix du meilleur chemin**

$$d(\chi,\xi)) = \min_{\phi} d_\phi(\chi,\xi)$$

Programmation dynamique (DTW)

- Permet (sous certaines conditions) d'obtenir la solution optimale sans devoir considérer toutes les solutions possibles
- Principe de base: la solution optimale peut être obtenue à partir de solutions intermédiaires optimales
- La distance optimale est obtenue en calculant, pour chaque point (i_x, i_y) la distance cumulée $D(i_x, i_y)$ correspondant à la distance optimale que l'on obtient en comparant les deux sous-séquences (sous-politiques)

Programmation dynamique (DTW)

- **Distance accumulée minimale entre (1,1) et (i_x,i_y)**

$$D(i_x,i_y) = \min_{\phi_x,\phi_y,T'} \sum_{k=1}^{T'} d(\phi_x(k),\phi_y(k))m(k)$$

où

$$\phi_x(T') = i_x \ ; \ \phi_y(T') = i_y$$

- Le facteur de normalisation sera utilisé une fois que le point final aura été atteint

$$M_\phi = \sum_{k=1}^{T} m(k)$$

Programmation dynamique (DTW)

- Rajout de **contraintes**

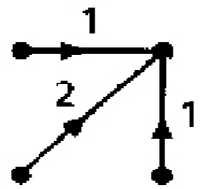
$$D(i_x,i_y) = \min_{(i'_x,i'_y)} [D(i'_x,i'_y) + \zeta((i'_x,i'_y),(i_x,i_y))]$$

- Avec la distance pondérée définie par:

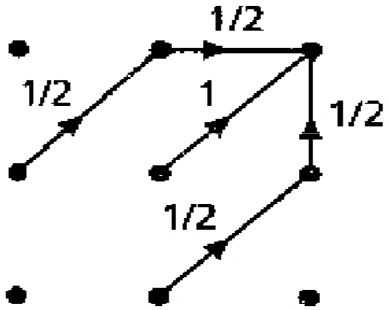
$$\zeta((i'_x,i'_y),(i_x,i_y)) = \sum_{l=0}^{L_s} d(\phi_x(T'-l),\phi_y(T'-l)m(T'-l)$$

- Ls est le nombre de déplacements dans le chemin pour aller de (i'_x,i'_y) à (i_x,i_y)

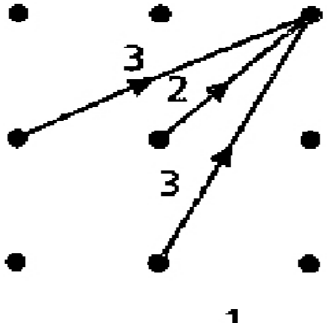
Exemples de contraintes locales



$$\min \left\{ \begin{array}{l} D(i_x - 1, i_y) + d(i_x, i_y), \\ D(i_x - 1, i_y - 1) + 2d(i_x, i_y), \\ D(i_x, i_y - 1) + d(i_x, i_y) \end{array} \right\}$$



$$\min \left\{ \begin{array}{l} D(i_x - 2, i_y - 1) + \frac{1}{2}[d(i_x - 1, i_y) + d(i_x, i_y)], \\ D(i_x - 1, i_y - 1) + d(i_x, i_y), \\ D(i_x - 1, i_y - 2) + \frac{1}{2}[d(i_x, i_y - 1) + d(i_x, i_y)] \end{array} \right\}$$



$$\min \left\{ \begin{array}{l} D(i_x - 2, i_y - 1) + 3d(i_x, i_y), \\ D(i_x - 1, i_y - 1) + 2d(i_x, i_y), \\ D(i_x - 1, i_y - 2) + 3d(i_x, i_y), \end{array} \right\}$$

Utilisation en reconnaissance vocale

- **Repose sur l'utilisation de contraintes supplémentaires:**

  - **Des contraintes de monocité**
    - Point de départ (début des deux mots): (1,1)
    - Point d'arrivée (fin des deux mots): (T_x,T_y,)

  - **Des contraintes globales**
    - Réduction de l'espace de recherche

  - **Des contraintes locales**
    - Prédécesseurs limités à quelques éléments proches
    - Chemins uniquement Gauche-droite
    - Utilisation de poids (pénalités) suivant les chemins

Implémentation (DTW)

- Initialisation de la matrice D des distances cumulées

$$D_A(1,1) = d(1,1)m(1)$$

- Calculer les distance locales pour tous les autres éléments de la première colonne de D: $d(1,i)$

- Si la transition verticale est autorisée, calculer les distances accumulées de la première colonne:

$$D(i_x,i_y) = \min_{(i'_x,i'_y)} [D(i'_x,i'_y) + \zeta((i'_x,i'_y),(i_x,i_y))]$$

  Sinon les distances sont égales à l'infini.

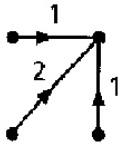
- Passer à la colonne suivante et ainsi de suite….
- Lorsque le dernier point est atteint, réinjecter le coefficient de normalisation

$$d(\chi,\xi) = \frac{D_A(T_x,T_y)}{M_\phi}$$

Exemple (DTW)

- **Soit les séquences**
  - A=(1 2 2 4 6) (séquence test))
  - B =(2 3 4 5 6 7) (référence en mémoire)
  - C= (1 2 4 4 6 6) (référence en mémoire)

- **Calculer la matrice des distances locales avec**
  - $d(i,j) = (s(i) - s(j))^2$
- **Calculer la matrice des distances cumulées avec la contrainte locale suivante:**



- **Retrouver le chemin optimal en suivant les valeurs les plus faibles (et au besoin en favorisant la diagonale)**

Exemple: corrigé (DTW)

Distances locales

B					
7	36	25	25	9	1
6	25	16	16	4	0
5	16	9	9	1	1
4	9	4	4	0	4
3	4	1	1	1	9
2	1	0	0	4	16
	1	2	2	4	6

A

Distances cumulées

∞	91	56 ~~66~~	56 ~~66~~	16	**4**
∞	55	31	31	7	**3**
∞	30	15	15	**3**	4
∞	14	6	6	**2**	6
∞	5	2	**2**	3	12
∞	**1**	**1**	1	5	21
	∞	∞	∞	∞	∞

C= 4 / 9

Exemple: corrigé (DTW)

Distances locales

C					
6	25	16	16	4	0
6	25	16	16	4	0
4	9	4	4	0	4
4	9	4	4	0	4
2	1	0	0	4	16
1	0	1	1	9	25
	1	2	2	4	6

A

Distances cumulées

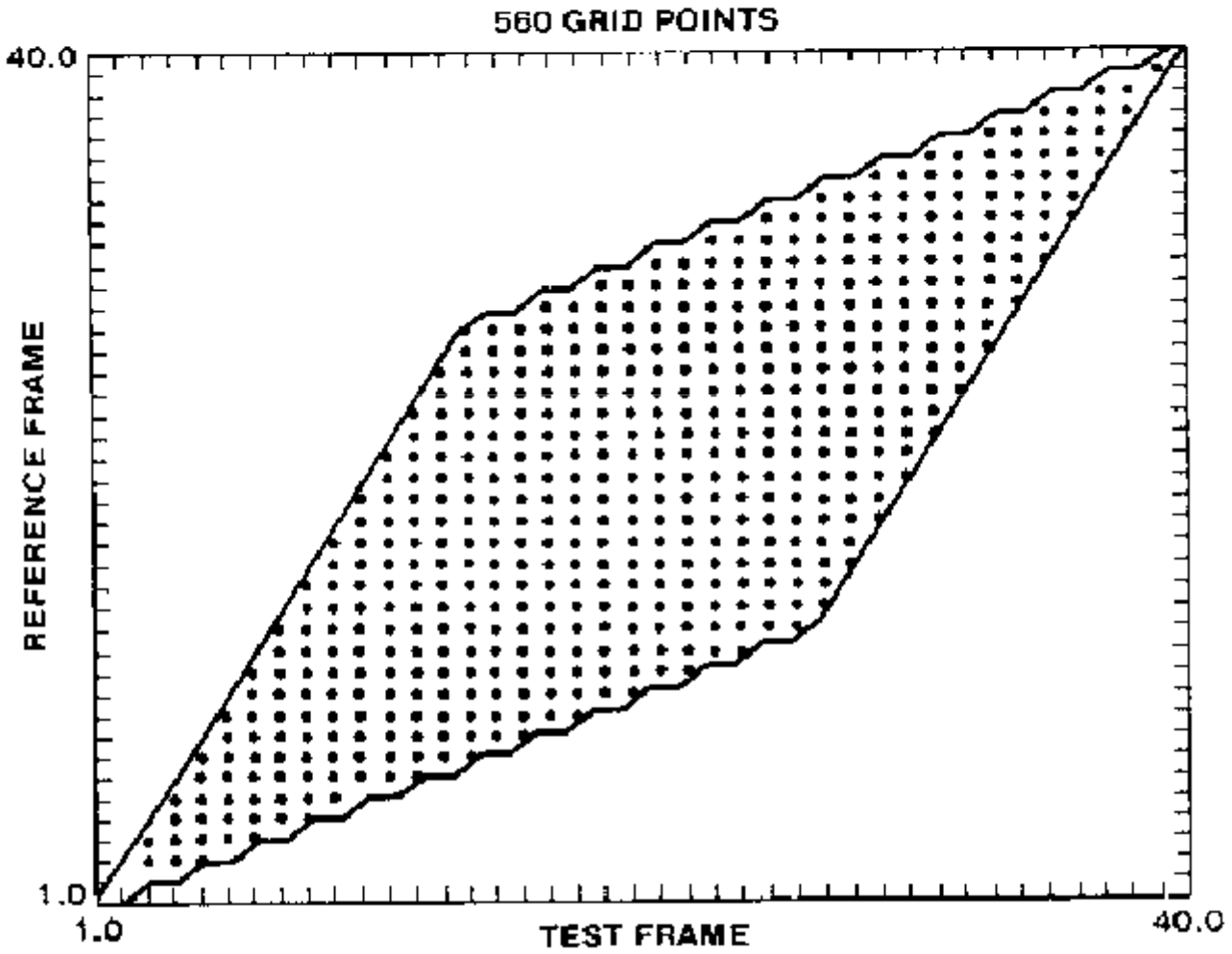
∞	69	40	40	8	**0**
∞	44	24	24	4	**0**
∞	19	8	8	**0**	4
∞	10	4	4	**0**	4
∞	1	**0**	**0**	4	20
∞	**0**	1	2	11	36
	∞	∞	∞	∞	∞

C= 0 /9 =0

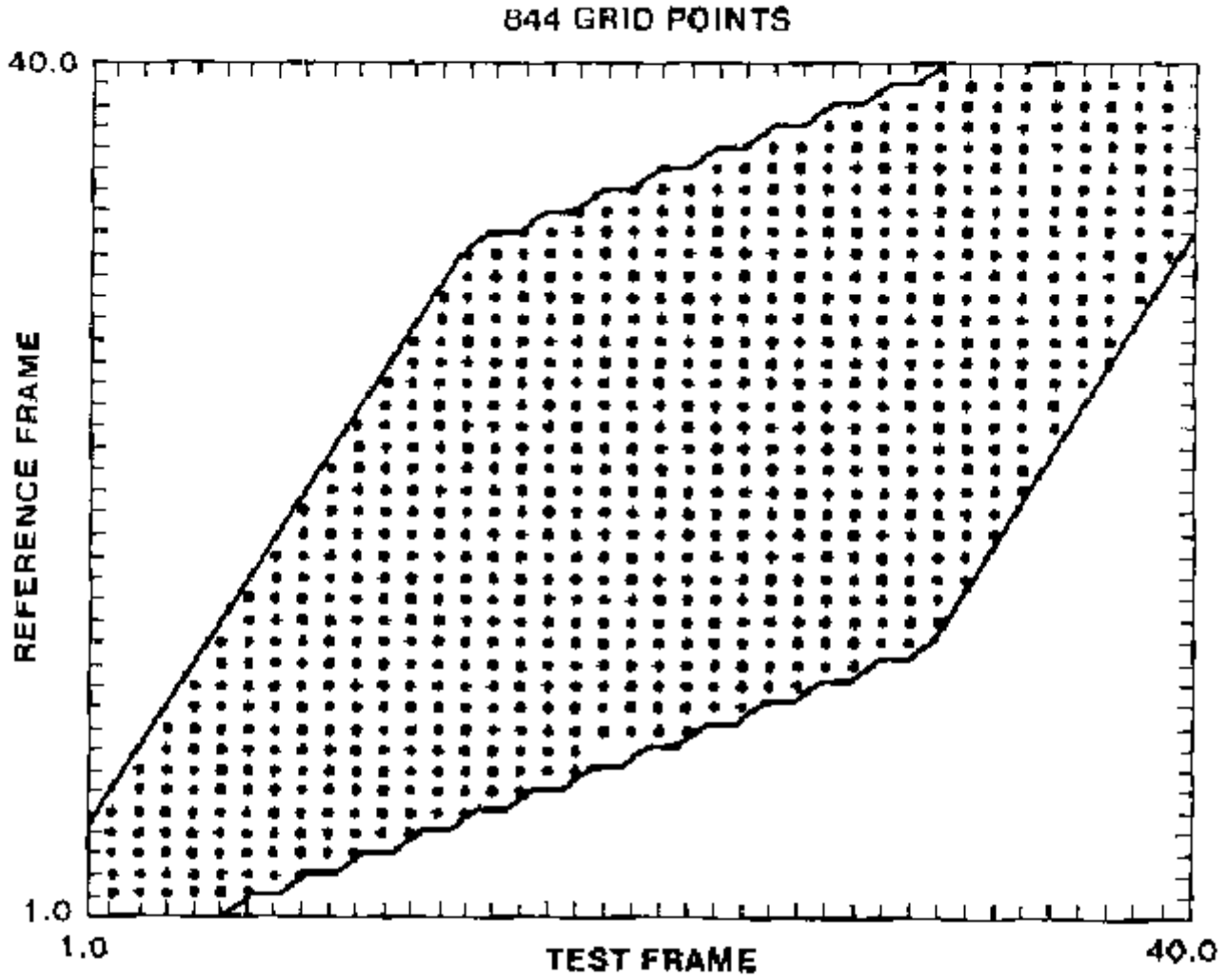
Possibilité d'initialiser D(0,0) à 1

Région de recherche du chemin optimal

(d'après Rabiner93)

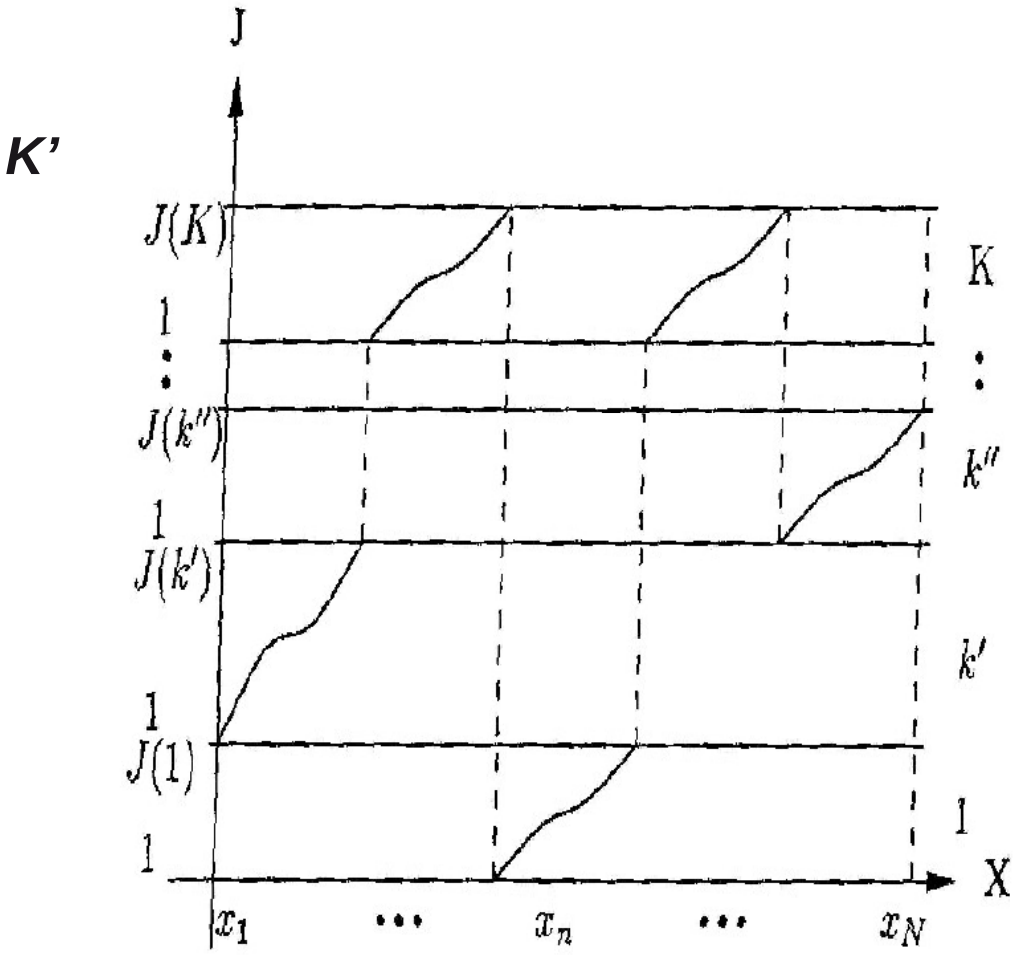


Relâchement de contraintes aux extrémités



Reconnaissance de mots connectés

- **Séquence reconnue:**



DTW: discussion

- **Très utilisée dès les années 70.**
- **Actuellement moins utilisée mais est à la base des systèmes utilisant les HMM**

- **Nombreuses améliorations développées:**
  - Adaptation à la variabilité
    - Multi-références
    - Références obtenues par Quantification Vectorielle (k-means)
  - Adaptation multi-locuteur

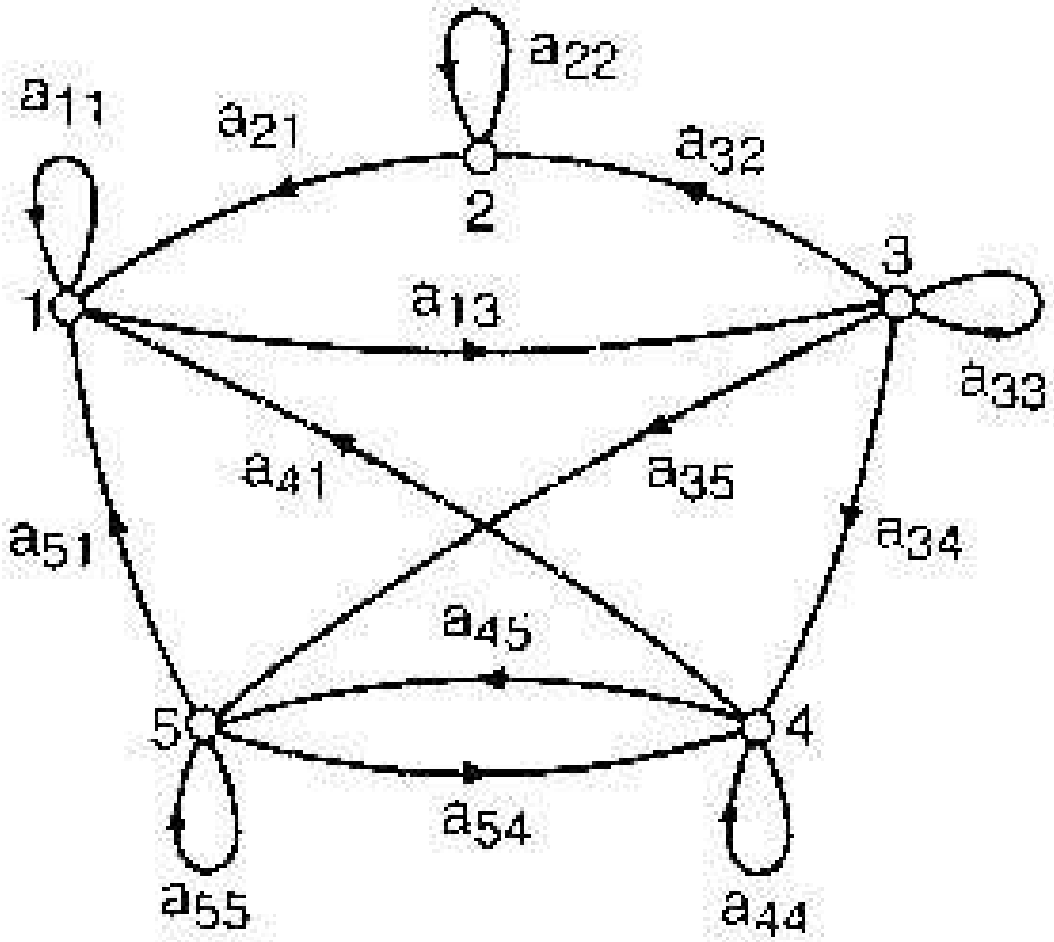
SI340
Module Parole

Introduction aux Modèles de Markov
utilisation en parole

Gaël RICHARD

Mai 2010

Chaînes de Markov discrètes



Chaînes de Markov discrètes

Notations

- ***t=1,2 …*** sont les instants de changement d'états
- q_t est l'état à l'instant ***t***
- La probabilité d'être dans l'état ***j*** sachant que l'on a été dans l état ***i*** au temps ***t-1*** et dans l'état ***k*** à l'état ***t-2***, etc… est:

$$P[q_t = j | q_{t-1} = i, q_{t-2} = k, \ldots]$$

- Chaînes de Markov du premier ordre:

$$P[q_t = j | q_{t-1} = i, q_{t-2} = k, \ldots] = P[q_t = j | q_{t-1} = i]$$

Chaînes de Markov discrètes (notations)

- **Système dont les changements d'états sont indépendants du temps:**

  Avec les propriétés suivantes:

$$a_{ij} = P[q_t = j | q_{t-1} = i] \qquad \text{pour} \quad 1 \leq i,j \leq N$$

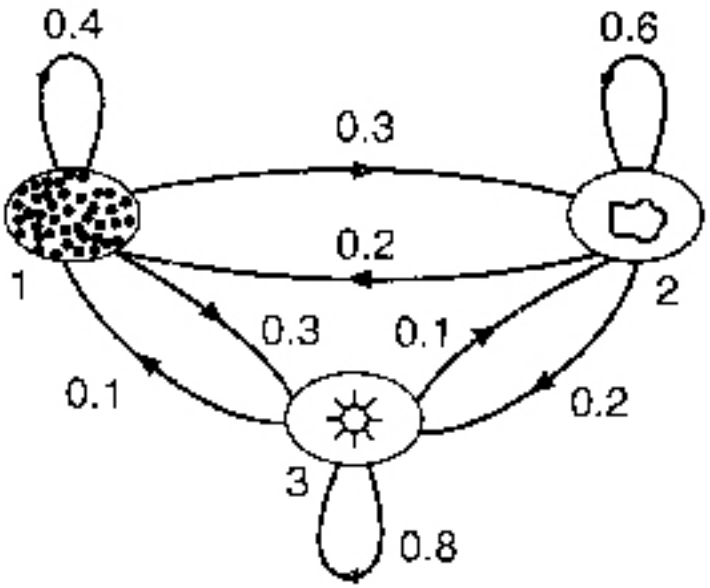
$$a_{ij} \geq 0 \qquad \forall j,i$$

$$\sum_{j=1}^{N} a_{ij} = 1 \qquad \forall i$$

Probabilité de l'état initial

$$\pi_i = P[q_1 = i], \qquad 1 \leq i \leq N$$

Exercice: Modèle Météo à 3 états



- **Ecrire la matrice de transition A**
- **Sachant qu'à t=1 le temps est ensoleillé, quelle est la probabilité (connaissant le modèle) que le temps pour les 7 prochains jours soit *(Soleil, Soleil, pluie, pluie, Soleil, Nuageux, Soleil)?***
- **Sachant que le système est dans un état connu, quelles est la probabilité qu'il reste dans cet état pendant exactement *d* jours ?**

Exercice: Modèle Météo à 3 états (correction)

- **La matrice A vaut:**

$$A = \begin{pmatrix} 0.4_{11} & 0.3_{12} & 0.3_{13} \\ 0.2_{21} & 0.6_{22} & 0.2_{23} \\ 0.1_{31} & 0.1_{32} & 0.8_{33} \end{pmatrix}$$

- **Soit l'observation:**

$$\mathbf{O} = (\text{ Soleil, Soleil, Soleil, Pluie, Pluie, Soleil, Nuageux, Soleil, })$$

$$\begin{aligned} P(\mathbf{O}|Model) &= P[3,3,3,1,1,3,2,3,|Model] \\ &= P[3]P[3|3]^2P[1|3]P[1|1]P[3|1]P[2|3]P[3|2] \\ &= \pi_3 \cdot (a_{33})^2 a_{31}a_{11}a_{13}a_{32}a_{23} \\ &= (1.0)(0.8)^2(0.1)(0.4)(0.3)(0.1)(0.2) \\ &= 1.536 \times 10^{-4} \end{aligned}$$

Exercice: Modèle Météo à 3 états (correction)

- **Soit la séquence d'observation:**
  - O = (i, i, i, i, …. i, j≠i)

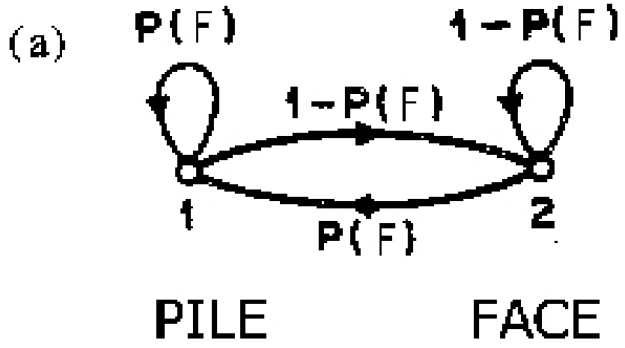
$$
\begin{aligned}
P(\mathbf{O}|Model, q_1 = i) &= P(\mathbf{O}, q_1 = i|Model)/P(q_1 = i) \\
&= P[i]P[i|i]^{d-1}P[i|j]/P[i] \\
&= \pi_i(a_{ii})^{d-1}(1 - a_{ii})/\pi_i \\
&= (a_{ii})^{d-1}(1 - a_{ii})
\end{aligned}
$$

Extension aux modèles de Markov cachés

- **Modèle Pile (P) ou Face (F)**
- **Fournit une séquence d'observations**

$$\begin{aligned} \mathbf{O} &= (\ \mathbf{O_1}\ \ \mathbf{O_2}\ \ \mathbf{O_3}\ \ \ldots\ \ \mathbf{O_T}\ ) \\ &= (\ \mathrm{F}\ \ \mathrm{P}\ \ \mathrm{F}\ \ \ldots\ \ \mathrm{F}\ ) \end{aligned}$$

- Le premier modèle qui vient à l'esprit est le modèle suivant où une seule pièce de monnaie est utilisée

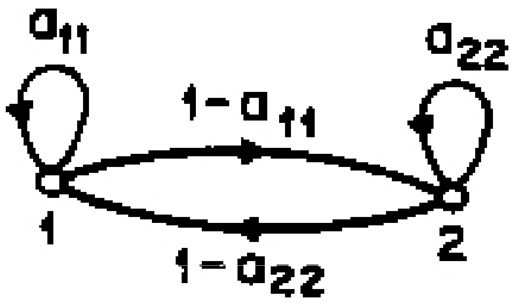


Extension aux modèles de Markov cachés

- **Construire d'autres modèles en supposant que l'on a:**
  - **2 pièces de monnaie** qui possèdent un biais différent (par exemple la pièce A a une probabilité 0.6 de sortir Face alors que la pièce B a une probabilité 0.4 de sortir Face)
  - **Extrapoler à un modèle de 3 pièces de monnaie**

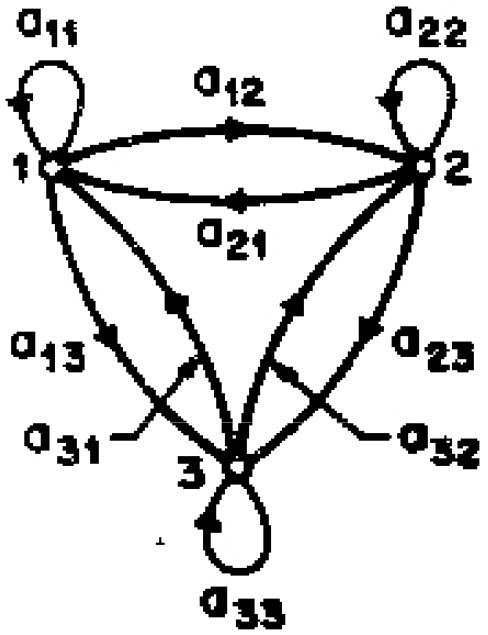
Extension aux modèles de Markov cachés (solution)

(b)



$$P(F) = P_1 \qquad P(F) = P_2$$
$$P(P) = 1-P_1 \qquad P(P) = 1-P_2$$

(c)

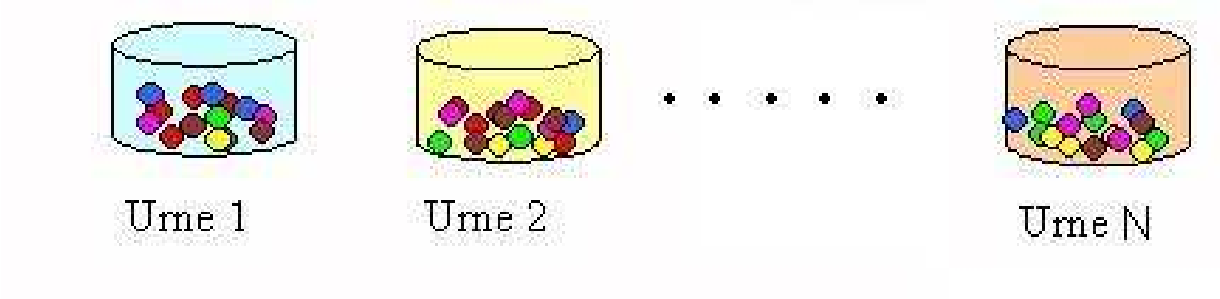


ETAT

	1	2	3
$P(F)$	P_1	P_2	P_3
$P(P)$	$1-P_1$	$1-P_2$	$1-P_3$

Extension aux modèles de Markov cachés

- **Modèle des boules et des urnes**



- **M couleurs**
- **Le tirage s'effectue de la façon suivante:**
  - Une urne est sélectionnée (selon une procédure aléatoire)
  - Une boule est ensuite tirée dans cette urne. La couleur de la boule constitue l'observation
  - La boule est replacée dans l'urne et une nouvelle urne peut être sélectionnée pour le tirage suivant

Extension aux modèles de Markov cachés

- **Le tirage est effectué dans une autre pièce**

  ne permet pas de savoir dans quelle urne a été tirée chaque boule

- **Un modèle approprié: modèle de markov à N états**

Caractérisation des HMM

- **Un modèle HMM pour des données d'observations discrètes sera caractérisé par:**

  - **Le nombre *N* d'états du modèle**
    - Modèle HMM ergodique
    - Modèle HMM Gauche-droite
    - q_t est l'état à l'instant ***t***

  - **Le nombre *M* de symboles distincts d'observation** par état (soit la taille de l'alphabet)
    - Ex P et F pour le modèle Pile ou Face
    - Ex. Les couleurs pour le modèle Boules et urnes
    - On notera ces symboles sous la forme:

$$V = \mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_M$$

Caractérisation des HMM

- **La matrice $A=a_{ij}$ de transition** entre états où

$$a_{ij} = P[q_t = j | q_{t-1} = i] \qquad \text{pour} \quad 1 \leq i,j \leq N$$

Notons que a_{ij} =0 pour les transitions impossibles

- **La distribution de probabilité $B=b_j(k)$ d'observation** des symboles où

$$b_j(k) = P[\mathbf{o}_t = \mathbf{v}_k | q_t = j], \qquad \text{pour} \quad 1 \leq k \leq M$$

définit la distribution de probabilité des symboles dans l'état *j*

- **La distribution de l'état initial pour laquelle**

$$\pi_j = P[q_1 = j], \qquad \text{pour} \quad 1 \leq j \leq N$$

Caractérisation des HMM

- **En résumé, la spécification complète d'un HMM est donnée par:**

  - Paramètres *N* et *M* du modèle
  - La specification des symboles d'observation
  - La spécification des probabilités *A*, *B* et Π

  - On notera par convention $\lambda = (A,B,\pi)$ pour désigner le modèle complet.
  - Ce modèle inclut une mesure de probabilité soit: $P(\mathbf{O}|\lambda)$

HMM: Générateur d ’observations (Exercice)

- **Supposons les modèles à 3 états suivants dans le cadre d ’une expérience Pile ou Face:**

	Etat 1	Etat 2	Etat3
P(F)	0.5	0.75	0.25
P(P)	0.5	0.25	0.75

avec chaque probabilité de transition étant gale à 1/3 (la probabilité d ’état initiale étant également égale à 1/3)

Questions:

1. Connaissant la séquence O=(F F F F P F P P P P), quelle est la séquence d ’états la plus probable ? Quelle est la probabilité d ’observer cette séquence et celle de cette séquence d ’états
2. Quelle est la probabilité que la séquence d ’observation provienne entièrement de l ’état 1?
3. En est-il de même avec la séquence O=(F P P F P F F P P F) ?
4. Quelles seraient vos réponses avec la matrice de transition suivante:

a11=0.9	A21=0.45	a33=0.45
a12=0.05	a22=0.1	a32=0.45
a13=0.05	a23=0.45	a33=0.1

Les trois problèmes des HMM

- **Problème 1: Évaluer la probabilité d'une séquence d'observations.**

  connaissant la séquence d'observation $\mathbf{O} = (\mathbf{o_1}, \mathbf{o_2}, \mathbf{o_3}, \ldots, \mathbf{o_T})$ *et le modèle,* $\lambda = (A, B, \pi)$ *comment peut-on calculer* $P(\mathbf{O}|\lambda)$ *qui est la probabilité de la séquence d'observations, connaissant le modèle.*

- **Problème 2: Retrouver la séquence d'états optimale.**

  connaissant la séquence d'observation $\mathbf{O} = (\mathbf{o_1}, \mathbf{o_2}, \mathbf{o_3}, \ldots, \mathbf{o_T})$ *et le modèle* $\lambda = (A, B, \pi)$ *comment choisit-on la séquence d'état* $q = (q_1, q_2, q_3, \ldots, q_T)$ *qui est optimale au sens d'un certain critère (i.e. la séquence d'états qui "explique" au mieux les observations)*

Les trois problèmes des HMM

- **Problème 3: Ré-estimer les paramètres du modèle.**

  Comment ajuste-t-on les paramètres du modèle $\lambda = (A,B,\pi)$ *pour maximiser* $P(\mathbf{O}|\lambda)$ *qui est la probabilité de la séquence d'observation connaissant le modèle.*

Problème 1: Évaluer la probabilité d'une séquence d'observations.

connaissant la séquence d'observation et le modèle, comment peut-on calculer la probabilité de la séquence d'observations, connaissant le modèle.

- ✓ **Solution intuitive:** énumérer toutes les séquences d'états de taille T (N^T séquences possibles)

Soit **q** l'une de ces séquences: $\mathbf{q} = (q_1, q_2, \ldots q_T)$

En supposant que les observations sont statistiquement indépendantes:

$$P(\mathbf{O}|\mathbf{q},\lambda) = \prod_{t=1}^{T} P(\mathbf{o}_t|q_t,\lambda)$$

On en déduit: $$P(\mathbf{O}|\mathbf{q},\lambda) = b_{q_1}(\mathbf{o}_1) \cdot b_{q_2}(\mathbf{o}_2) \ldots b_{q_T}(\mathbf{o}_T)$$

Problème 1: Évaluer la probabilité d'une séquence d'observations.

- la probabilité d'une telle séquence d'états est aussi donnée par:

$$P(\mathbf{q}|\lambda) = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} \ldots a_{q_{T-1} q_T}$$

- La probabilité jointe de O et q est le produit des deux termes

$$P(\mathbf{O},\mathbf{q}|\lambda) = P(\mathbf{O}|\mathbf{q},\lambda) \cdot P(\mathbf{q}|\lambda)$$

- La probabilité de la séquence **O** est donnée en sommant cette probabilité jointe sur l'ensemble des séquences d'états possibles **q**

$$\begin{aligned} P(\mathbf{O}|\lambda) &= \sum_{all\ \mathbf{q}} P(\mathbf{O}|\mathbf{q},\lambda) \cdot P(\mathbf{q}|\lambda) \\ &= \sum_{q_1,q_2,\ldots q_T} \pi_{q_1} b_{q_1}(\mathbf{o}_1) a_{q_1 q_2} b_{q_2}(\mathbf{o}_2) a_{q_2 q_3} \ldots b_{q_T}(\mathbf{o}_T) a_{q_{T-1} q_T} \end{aligned}$$

Problème 1: Évaluer la probabilité d'une séquence d'observations.

- Interprétation
- à t=1: nous sommes dans l ’état q_1 avec la probabilité π_{q_1} et nous générons un symbole o_1 avec la probabilité $b_{q_1}(\mathbf{o}_1)$
- à t=2, nous faisons une transition à l ’état q_2 à partir de l ’état q1 avec la probabilité $a_{q_1 q_2}$ et générons le symbole o_2 avec la probabilité $b_{q_2}(\mathbf{o}_2)$
- et ainsi de suite jusqu ’à o_T

Problème 1: Évaluer la probabilité d'une séquence d'observations.

- **Complexité de l ’approche pour évaluer** $P(\mathbf{O}|\lambda)$
  - $(2T-1)N^T$ multiplications
  - $N^T - 1$ additions

- **Calcul impossible même pour des petites valeurs de N (nombre d ’états) et T (nombre d ’observations).**

- **Une solution: l ’algorithme forward (ou récurrence avant)**

Algorithme Forward

- **Soit la variable définie par :**

$$\alpha_t(i) = P(\mathbf{o_1}, \mathbf{o_2}, \mathbf{o_3} \ldots \mathbf{o_t}, q_t = i|\lambda)$$

qui est la probabilité de la séquence d 'observations partielle $\mathbf{o_1}, \mathbf{o_2}, \mathbf{o_3} \ldots \mathbf{o_t}$ et de l 'état *i* à l 'instant *t* connaissant le modèle

- **Algorithme:**
  - Initialisation $\alpha_1(i) = \pi_i b_i(\mathbf{o}_1), \qquad 1 \leq i \leq N$
  - Récursion $\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_i(\mathbf{o}_{t+1}), \qquad 1 \leq t \leq T-1, \quad 1 \leq j \leq N$
  - Arrêt $P(\mathbf{O}|\lambda) = \sum_{i=1}^{N} \alpha_T(i)$

Algorithme Forward

- **Approche largement moins complexe:**
  - $N(N+1)T-1)+N$ multiplications
  - $N(N-1)(T-1)$ additions

- **De façon similaire, on peut définir un algorithme backward à partir de la probabilité de la séquence d 'observation partielle de *t+1* à la fin étant donné l 'état *i* au temps t et le modèle:**

$$\beta_t(i) = P(\mathbf{o_{t+1}}, \mathbf{o_{t+2}}, \mathbf{o_{t+3}} \ldots \mathbf{o_T} | q_t = i, \lambda)$$

Problème 2: Retrouver la séquence d'états optimale.

- On cherche ici à trouver une séquence optimale d ’états connaissant la séquence d ’observations
- L ’approche couramment retenue est de trouver l ’unique meilleure séquence d ’états (ou encore chemin)
- Méthode basée sur la programmation dynamique : l ’algorithme de Viterbi

Problème 2: Retrouver la séquence d'états optimale.

- On introduit la notion de meilleur chemin partiel jusqu ’au temps t et finissant à l ’état i:

$$\delta_t(i) = \max_{q_1, q_2, \cdots q_{t-1}} P[q_1 q_2 \ldots q_{t-1}, q_t = i, \mathbf{o}_1 \mathbf{o}_2, \ldots \mathbf{o}_t | \lambda]$$

- Par récurrence, on peut alors déterminer:

$$\delta_{t+1}(j) = [\max_i \delta_t(i) a_{ij}] \cdot b_j(\mathbf{o}_{t+1})$$

- En pratique, il sera aussi nécessaire de garder la séquence d ’états pour chaque temps *t*. Ce sera réalisé à l ’aide du tableau $\psi_t(j)$

Problème 2: Retrouver la séquence d'états optimale.

- **Algorithme:**
  - Initialisation

$$\begin{aligned} \delta_1(i) &= \pi_i b_i(\mathbf{o}_1), \ 1 \leq i \leq N \\ \psi_1(i) &= 0 \end{aligned}$$

$$\begin{aligned} \delta_t(j) &= \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}] \cdot b_j(\mathbf{o}_t), \qquad 2 \leq t \leq T-1; \quad 1 \leq j \leq N \\ \psi_t(j) &= \arg \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}], \qquad 2 \leq t \leq T-1; \quad 1 \leq j \leq N \end{aligned}$$

  - Arrêt

$$\begin{aligned} P^* &= \max_{1 \leq i \leq N} [\delta_T(i)] \\ q_T^* &= \arg \max_{1 \leq i \leq N} [\delta_T(i)] \end{aligned}$$

  - Rétropropagation (chemin optimal)

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

Exercice: Algorithme de Viterbi

- **Soit le modèle choisi pour l'expérience avec 3 pièces avec les probabilités suivantes:**
  - Etat1: P(P) = P(F) = 0.5
  - Etat2: P(P)=0.75; P(F) = 0.25
  - Etat 3: P(P) = 0.25; P(F)=0.75
  - Tous les aij =1/3; probabilités initiales = 1/3

  - Quel est la séquence d'état optimale obtenue par l'algorithme de Viterbi pour l'observation (F F F F P F P P P P) ?
  - Pouvez vous trouver la séquence d'état optimale obtenue pour la matrice de transition suivante ?

a11=0.9	A21=0.45	a33=0.45
a12=0.05	a22=0.1	a32=0.45
a13=0.05	a23=0.45	a33=0.1

Problème 3: Ré-estimer les paramètres du modèle.

- **Problème plus complexe**
- **Utilisation de l ’algorithme de Baum-Welch (aussi connu sous le nom d ’algorithme Expectation-Maximisation)**
- **<u>Objectif:</u> Ré-estimer les paramètres du modèle** $\lambda = (A, B, \pi)$ **pour maximiser localement** $P(\mathbf{O}|\lambda)$

Ré-estimer les paramètres du modèle (2)

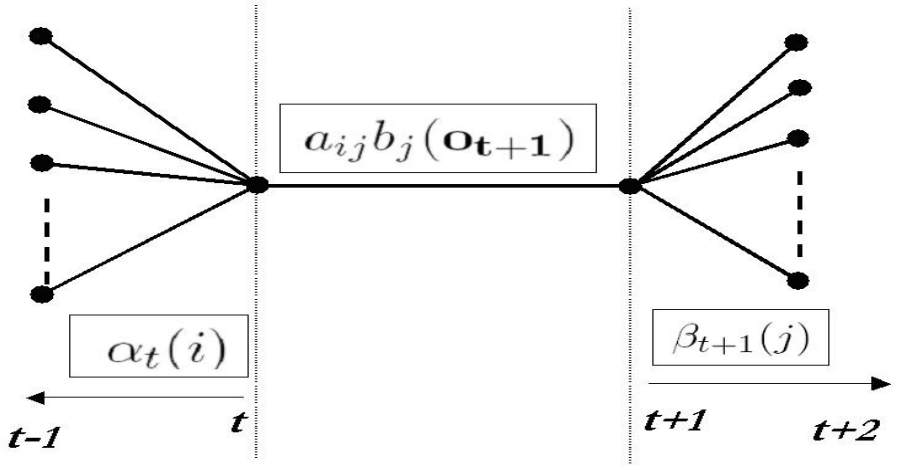
- **Définissons la probabilité d'être dans l'état i à l'instant t et dans l'état j à l'instant t + 1 connaissant le modèle, et la séquence d'observation O:** $\xi_t(i,j)$

- **On a:**

$$\begin{aligned}\xi_t(i,j) &= P(q_t = i, q_{t+1} = j | \mathbf{O}, \lambda) \\ &= \frac{P(q_t = i, q_{t+1} = j, \mathbf{O} | \lambda)}{P(\mathbf{O}|\lambda)}\end{aligned}$$

Ré-estimer les paramètres du modèle (2)

- **Interprétation:**



- $\xi_t(i,j)$ **est l'ensemble des chemins vérifiant les conditions requises par l'équation précédente:**

$$\xi_t(i,j) = \frac{\alpha_t(i)a_{ij}b_j(\mathbf{o_{t+1}})\beta_{t+1}(j)}{P(\mathbf{O}|\lambda)}$$

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

Ré-estimer les paramètres du modèle (3)

- La probabilité d'être dans l'état *i* à l'instant *t* connaissant la séquence d'observation O et le modèle est $\gamma_t(i)$, et s'écrit:

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j)$$

- Par ailleurs, si l'on somme $\gamma_t(i)$ sur le temps *t*, on obtient une quantité qui peut être interprétée comme une estimation du nombre de fois que l'état *i* est visité (Si l'on ne somme que sur les T-1 premiers indices, on a une estimation du nombre de transitions à partir de l´état *i)*.

- De même, si l'on somme la variable $\xi_t(i,j)$ sur le temps (sur les T premiers indices), on obtient une estimation du nombre de transitions de l'état *i* vers l'état *j.*

Ré-estimer les paramètres du modèle (4)

- **On obtient finalement:**

$$\begin{aligned}\overline{\pi}_j &= \text{estimation du nombre de fois dans l'état } i \text{ à l'instant } t=1 \\ &= \gamma_1(i)\end{aligned}$$

$$\begin{aligned}\overline{a}_{ij} &= \frac{\text{estimation du nombre de transitions de l'état } i \text{ à l'état } j}{\text{estimation du nombre de transitions à partir de l'état } i} \\ &= \frac{\sum_{t=1}^{T-1}\xi_t(i,j)}{\sum_{t=1}^{T-1}\gamma_t(i)}\end{aligned}$$

$$\begin{aligned}\overline{b}_j(k) &= \frac{\text{estimation du nombre de fois dans l'état } j \text{ en y observant le symbole } \mathbf{v_k}}{\text{estimation du nombre de fois dans l'état } j} \\ &= \frac{\sum_{t=1,\mathbf{o_t}=\mathbf{v_k}}^{T-1}\gamma_t(i)}{\sum_{t=1}^{T-1}\gamma_t(i)}\end{aligned}$$

Ré-estimer les paramètres du modèle (5)

- Ainsi, à partir du modèle courant $\lambda = (A,B,\pi)$ on obtient un modèle réestimé $\overline{\lambda} = (\overline{A},\overline{B},\overline{\pi})$ pour lequel, on a:

$$P(\mathbf{O}|\overline{\lambda}) > P(\mathbf{O}|\lambda)$$

- On peut montrer que l'approche décrite est équivalente à l'utilisation de l'algorithme EM qui maximisera par rapport à λ la fonction auxilliaire de Baum :

$$Q(\lambda',\lambda) = \sum_{\mathbf{q}} P(\mathbf{O},\mathbf{q}|\lambda') \log P(\mathbf{O},\mathbf{q}|\lambda)$$

Densités d'observations continues

- **Les paramètres prennent en général des valeurs continues, d'où l'intérêt d'utiliser des densités d'observation continues**

- **Pour des modèles simples (par exemple sommes de Gaussiennes), on peut trouver des expressions analytiques pour les formules de réestimations.**

$$b_j(\mathbf{o}) = \sum_{k=1}^{M} c_{jk} \mathcal{N}(\mathbf{o}, \mu_{jk} \Gamma_{jk}), \qquad 1 \leq j \leq N$$

Densités d'observations continues

- **Modèle par somme de Gaussiennes**

$$b_j(\mathbf{o}) = \sum_{k=1}^{M} c_{jk} \aleph(\mathbf{o}, \mu_{jk} \Gamma_{jk}), \quad 1 \leq j \leq N$$

- **Avec:**

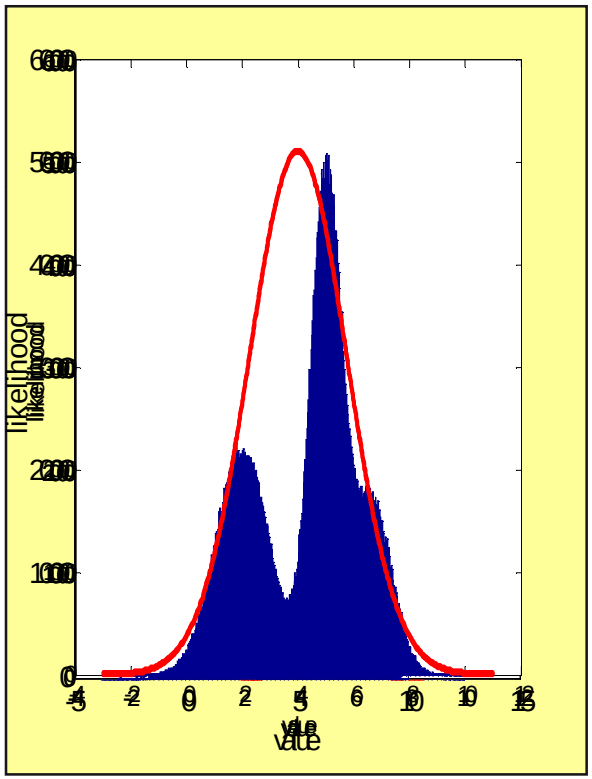
$$\sum_{k=1}^{M} c_{jk} = 1, \quad 1 \leq i \leq N$$

$$c_{jk} \geq 0, \quad 1 \leq j \leq N, \quad 1 \leq k \leq M$$

Approche par Mélanges de Gaussiennes

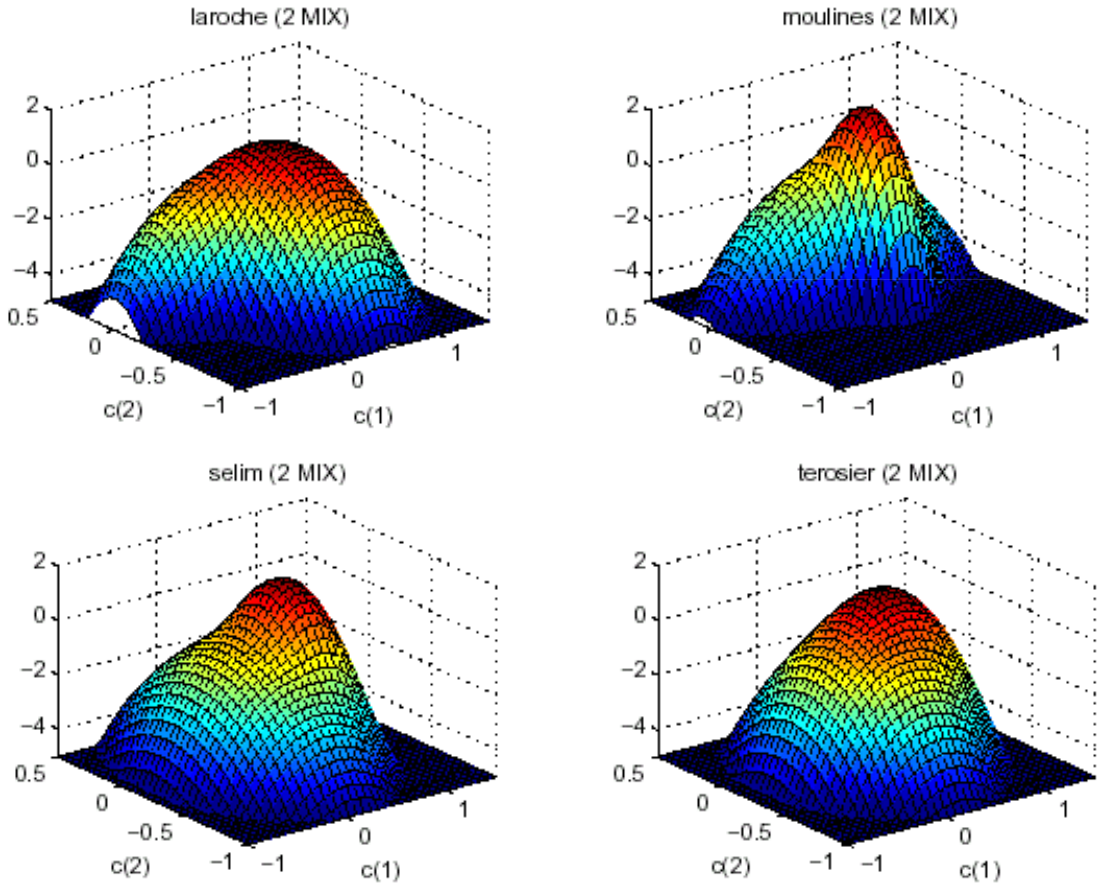
(Voir Cours O. Cappé http://tsi.enst.fr/~ocappe/em_tap.pdf)

- **Modèle de mélange**
  - Exemple à 2 dimensions avec 1, 2 puis 3 gaussiennes



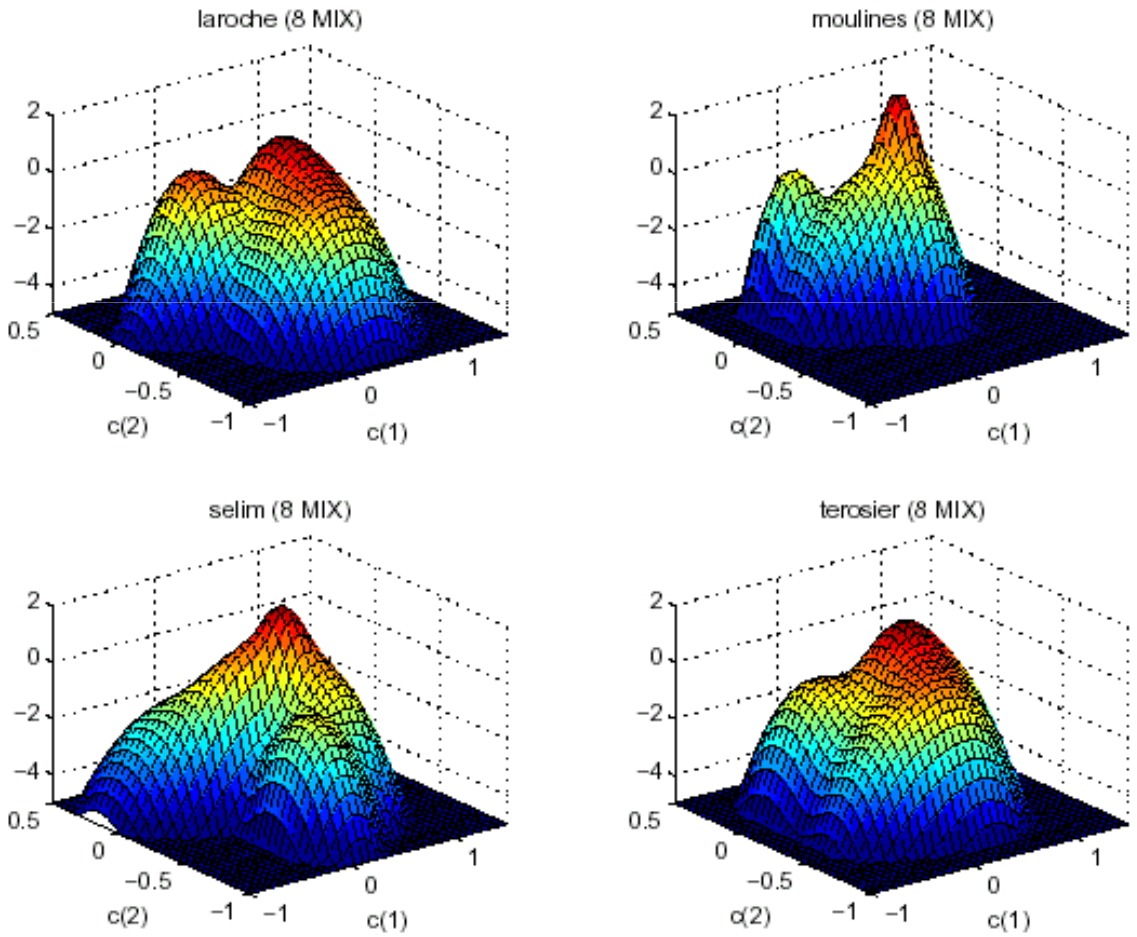
Approche par Mélanges de Gaussiennes (GMM)

- **Exemple de modèles à 2 composantes**



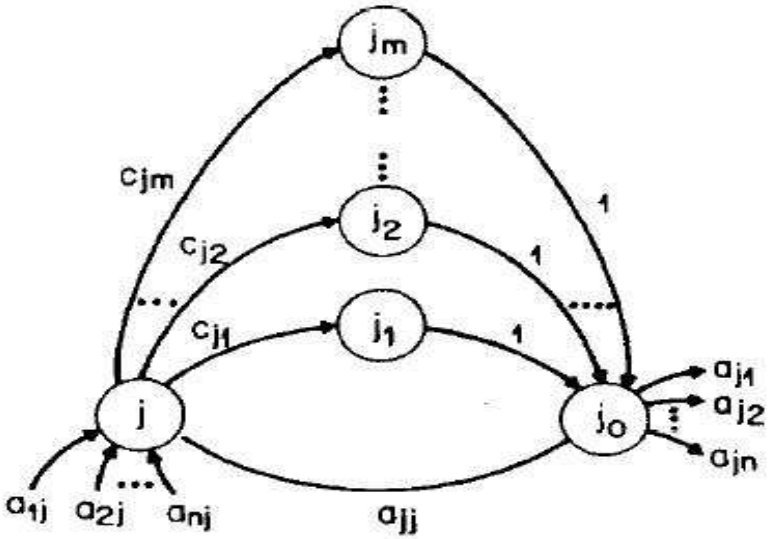
Approche par Mélanges de Gaussiennes (GMM)

- **Exemple de modèles à 2 composantes**



Equivalence de modèles

- **Equivalence entre**
  - un état d'une chaîne de Markov cachée avec une densité d'observation sous la forme d'un mélange de Gausiennes
  - un modèle multi-états en parallèle contenant chacun une des Gaussiennes du mélange:



Formules de ré-estimation

- Où la probabilité d'être dans l'état j au temps t avec la k^{ième} Gaussienne du mélange représentant o_t est donnée par:

$$\overline{c_{jk}} = \frac{\sum_{t=1}^{T} \gamma_t(j,k)}{\sum_{t=1}^{T} \sum_{k=1}^{M} \gamma_t(j,k)}$$

$$\overline{\mu_{jk}} = \frac{\sum_{t=1}^{T} \gamma_t(j,k).\mathbf{o}_t}{\sum_{t=1}^{T} \gamma_t(j,k)}$$

$$\overline{\mathbf{\Gamma}_{jk}} = \frac{\sum_{t=1}^{T} \gamma_t(j,k).(\mathbf{o}_t - \mu_{jk})(\mathbf{o}_t - \mu_{jk})'}{\sum_{t=1}^{T} \gamma_t(j,k)}$$

$$\gamma_t(j,k) = \left[\frac{\alpha_t(j)\beta_t(j)}{\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)}\right] \left[\frac{c_{jk}\aleph(\mathbf{o_t}, \mu_{jk}\Gamma_{jk})}{\sum_{m=1}^{M} c_{jm}\aleph(\mathbf{o_t}, \mu_{jm}\Gamma_{jm})}\right]$$