

# Plan

.

- Processus stochastiques
- Chaînes de Markov
- Chaînes de Markov Cachées
- Applications
  - Etiquetage morpho-syntaxique
  - Reconnaissance automatique de la parole

# Processus stochastique

- Un **processus stochastique** (ou processus aléatoire) est une séquence  $X_1, X_2 \dots X_N$  de variables aléatoires fondées sur le même ensemble fondamental  $S$ .
- Les valeurs possibles des variables aléatoires sont appelées les états possibles du processus.
- La variable  $X_t$  représente l'état du processus au temps  $t$  (on dit aussi l'observation au temps  $t$ ).
- Les différentes variables aléatoires ne sont en général pas indépendantes les unes des autres. Ce qui fait réellement l'intérêt des processus stochastiques est la dépendance entre les variables aléatoires.

# Processus stochastique

.

- Pour spécifier entièrement un processus stochastique, il suffit de spécifier :
  1. la loi de probabilité de la première variable aléatoire  $X_1$ , qui spécifie donc l'état du processus lors de la première observation.
  2. pour toute valeur de  $t > 1$  la probabilité conditionnelle :

$$P(X_t = j | X_1 = i_1, \dots, X_{t-1} = i_{t-1})$$

# Propriété de Markov

.

Une chaîne de Markov est un type particulier de processus stochastique qui vérifie deux conditions :

- L'état au temps  $t$  du processus ne dépend que de son état au temps  $t - 1$  :

$$P(X_t = j | X_1 = i_1, \dots, X_{t-1} = i_{t-1}) = P(X_t = j | X_{t-1} = i_{t-1})$$

- La probabilité de passage d'un état  $i$  à un état  $j$  ne varie pas avec le temps :

$$\forall t, 1 < t \leq N, \quad P(X_t = j | X_{t-1} = i) = C$$

# Processus de Markov

.

Un processus de Markov peut être décrit par

- une matrice de transition  $T$  telle que :

$$T(i, j) = P(X_t = j | X_{t-1} = i), 1 < t \leq N$$

$$\text{avec } T(i, j) \geq 0, \forall i, j$$

$$\text{et } \sum_{j=1}^N T(i, j) = 1 \forall i$$

- L'état du processus à l'instant 1 donc la loi de probabilité, notée  $\pi$ , de la variable  $X_1$  :

$$\pi(i) = P(X_1 = i)$$

# Processus de Markov

▪

On peut éviter le recours à la loi  $\pi$  en imposant que le processus débute toujours dans le même état 0, par exemple et en utilisant les transitions depuis cet état pour représenter les probabilités  $\pi$  :

$$T(0, i) = \pi(i), \text{ pour tout état } i \text{ du processus}$$

# Processus de Markov

.

Un processus de Markov peut aussi être représenté par un automate fini :

- Chaque état du processus est représenté par un état de l'automate
- Une transition de l'état  $i$  à l'état  $j$  est étiqueté par la probabilité  $T(i, j)$ .

# Exemple

- On admet que le fait qu'il ait plu ou non un jour donné est la seule considération à prendre en compte pour prévoir s'il pleuvra le lendemain. Plus précisément, s'il pleut aujourd'hui, il pleuvra demain aussi avec une probabilité de  $\alpha$  et s'il ne pleut pas aujourd'hui la probabilité qu'il pleuve demain est  $\beta$ .
- On convient de dire que le système est dans l'état 1 s'il pleut et 2 s'il ne pleut pas. La situation peut être représentée par une chaîne de Markov à deux états dont la matrice de transition est :

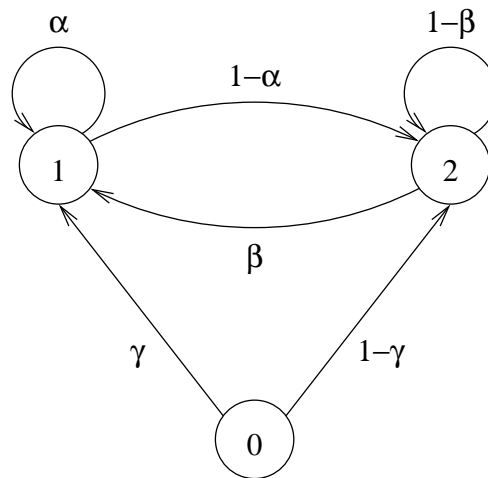
$$\begin{vmatrix} \alpha & 1 - \alpha \\ \beta & 1 - \beta \end{vmatrix}$$



# Exemple (suite)

.

- De plus, la probabilité que le processus soit dans l'état 1 à l'instant 1 est égale à  $\gamma$ .
- Le même processus peut être représenté par l'automate :



# Probabilité d'une suite d'observations

- Les propriétés de Markov permettent de calculer simplement la probabilité qu'une suite d'états particulière de longueur  $T$  soit observée (la loi de probabilité conjointe de  $(X_1, X_2, \dots, X_T)$ ) :

$$p(X_1, X_2, \dots, X_T) = \text{(règle de multiplication)}$$

$$p(X_1)p(X_2|X_1)p(X_3|X_1, X_2) \dots p(X_T|X_1, \dots, X_{T-1}) =$$

$$p(X_1)p(X_2|X_1)p(X_3|X_2) \dots p(X_T|X_{T-1}) \text{(hypothèse de Markov)}$$

# Exemple

.

- Etant donné le processus de Markov de l'exemple précédent, la probabilité d'avoir trois jours consécutifs de pluie est égale à :

$$\begin{aligned} p(X_1 = 1, X_2 = 1, X_3 = 1) &= p(X_1 = 1)p(X_2 = 1|X_1 = 1)p(X_3 = 1|X_2 = 1) \\ &= \gamma \times \alpha \times \alpha \\ &= \gamma\alpha^2 \end{aligned}$$

- Afin d'alléger les notations, on omettra de spécifier les variables aléatoires dans l'écriture de la probabilité d'une suite d'états. On notera, par exemple,  $p(1, 1, 1)$  la probabilité  $p(X_1 = 1, X_2 = 1, X_3 = 1)$ .

# Modèles de Markov Cachés

- Dans les chaînes de Markov, les observations correspondent aux états du processus.
- Dans un modèle de Markov caché, on ne peut observer directement les états du processus, mais des symboles (appelés aussi *observables*) émis par les états selon une certaine loi de probabilité.
- Au vu d'une séquence d'observation on ne peut savoir par quelle séquence d'états (ou *chemin*) le processus est passé, d'où le nom de modèles de Markov cachés (MMC).
- On distingue le processus  $X = X_1, X_2, \dots, X_T$  qui représente l'évolution des états du MMC et le processus  $O = O_1, O_2, \dots, O_T$  qui représente la suite des symboles émis par le MMC.

# Exemple

▪

# Éléments d'un MMC

.

Un MMC est défini par un quintuplet  $\langle S, A, \pi, T, E \rangle$  où :

- $S$  est l'ensemble des états :  $\{1, \dots, N\}$
- $A$  est l'alphabet des symboles émis par les états :  $\{a_1, \dots, a_M\}$
- $\pi$  est la loi de probabilité de l'état initial  $\pi(i) = P(X_1 = i)$ .  
 $\pi$  étant une loi de probabilité, on a :

$$\sum_{i=1}^N \pi(i) = 1$$

# Éléments d'un MMC - 2

.

Un MMC est défini par un quintuplet  $\langle S, A, \pi, T, E \rangle$  où :

- $T$  est la matrice des probabilités de transition d'un état vers un autre. La probabilité de transition d'un état  $i$  vers un état  $j$  ( $P(X_t = j | X_{t-1} = i)$ ) est notée  $T(i, j)$ . Comme dans le cas des chaînes de Markov, la somme des probabilités des transitions émanant d'un état vaut 1 :

$$\sum_{j=1}^N T(i, j) = 1, \forall i \in S$$

# Éléments d'un MMC - 3

.

Un MMC est défini par un quintuplet  $\langle S, A, \pi, T, E \rangle$  où :

- $E$  est la matrice des probabilités d'émission des symboles de  $A$  pour chaque état. La probabilité que l'état  $i$  émette le symbole  $j$  ( $P(O_t = j | X_t = i)$ ) est notée  $E(i, j)$ . Les probabilités d'émission de symboles de  $A$  pour chaque état du MMC constituent une loi de probabilité, on a par conséquent :

$$\sum_{j=1}^M E(i, o_j) = 1, \quad \forall i \in S$$

L'ensemble constitué des probabilités initiales, des probabilités de transition et d'émission d'un MMC  $\lambda$  est souvent appelé les *paramètres*  $\lambda$ .



# Exemple

.

$\lambda_1 = \langle \{1, 2, 3\}, \{a, b, c\}, \pi, T, E \rangle$  avec :

$$E(1, a) = 0,6 \quad E(2, a) = 0 \quad E(3, a) = 0,3$$

$$E(1, b) = 0,2 \quad E(2, b) = 0,5 \quad E(3, b) = 0 \quad \text{et}$$

$$E(1, c) = 0,2 \quad E(2, c) = 0,5 \quad E(3, c) = 0,7$$

$$T(1, 1) = 0,3 \quad T(2, 1) = 0,6 \quad T(3, 1) = 0,2$$

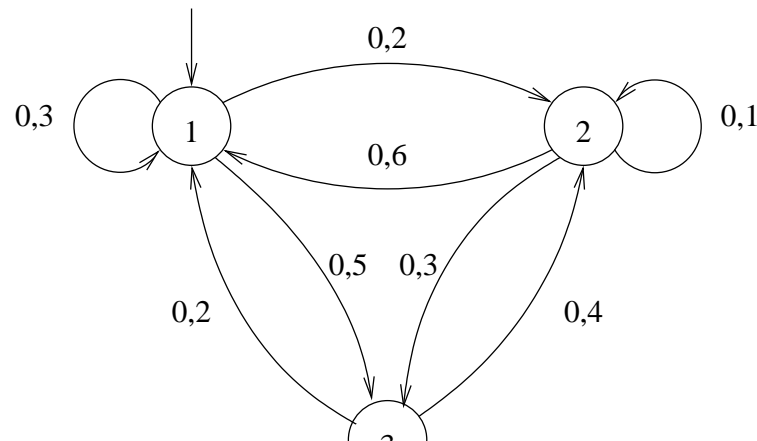
$$T(1, 2) = 0,2 \quad T(2, 2) = 0,1 \quad T(3, 2) = 0,4$$

$$T(1, 3) = 0,5 \quad T(2, 3) = 0,3 \quad T(3, 3) = 0,4$$

et

$$\pi(1) = 1 \quad \pi(2) = 0 \quad \pi(3) = 0$$

## représentation graphique



# Calcul de la probabilité d'une séquence d'observations

- Etant donné un MMC  $\lambda = \langle S, A, \pi, T, E \rangle$ , la suite d'observation  $o = o_1 o_2, \dots, o_T$  peut généralement être générée en suivant différents chemins dans le MMC
- La probabilité que  $\lambda$  émette la séquence  $o$  est égale à la somme des probabilités que la séquence  $o$  soit émise en empruntant les différents chemins pouvant émettre  $o$ .
- Ce raisonnement correspond en fait à l'application de la formule des probabilités totales à la probabilité  $P(o)$

$$P(o) = \sum_{x \in \mathcal{C}_T} P(o|x)P(x)$$

où  $\mathcal{C}_T$  est l'ensemble des séquences de  $T$  états de  $\lambda$  et  $x = x_1, \dots, x_T$  ( $x_i \in S$ ,  $1 \leq i \leq T$ ) une de ces séquences

# Calcul de la probabilité d'une séquence d'observations - 2

- la probabilité conditionnelle que  $o$  soit générée lorsque  $\lambda$  passe successivement par la séquence d'états  $x = x_1, \dots, x_T$  est le produit des probabilités que l'état atteint à l'instant  $t$  ( $x_t$ ) émette le symbole observé à cet instant ( $o_t$ ) :

$$P(o|x) = \prod_{t=1}^T E(x_t, o_t)$$

# Calcul de la probabilité d'une séquence d'observations - 3

- et la probabilité que le MMC suive une séquence particulière d'états  $x$  est le produit des probabilités que  $\lambda$  passe de l'état  $x_t$  à l'état  $x_{t+1}$  entre les instants  $t$  et  $t + 1$ , comme dans un modèle de Markov *visible* :

$$P(x) = \pi(x_1) \prod_{t=1}^{T-1} T(x_t, x_{t+1})$$

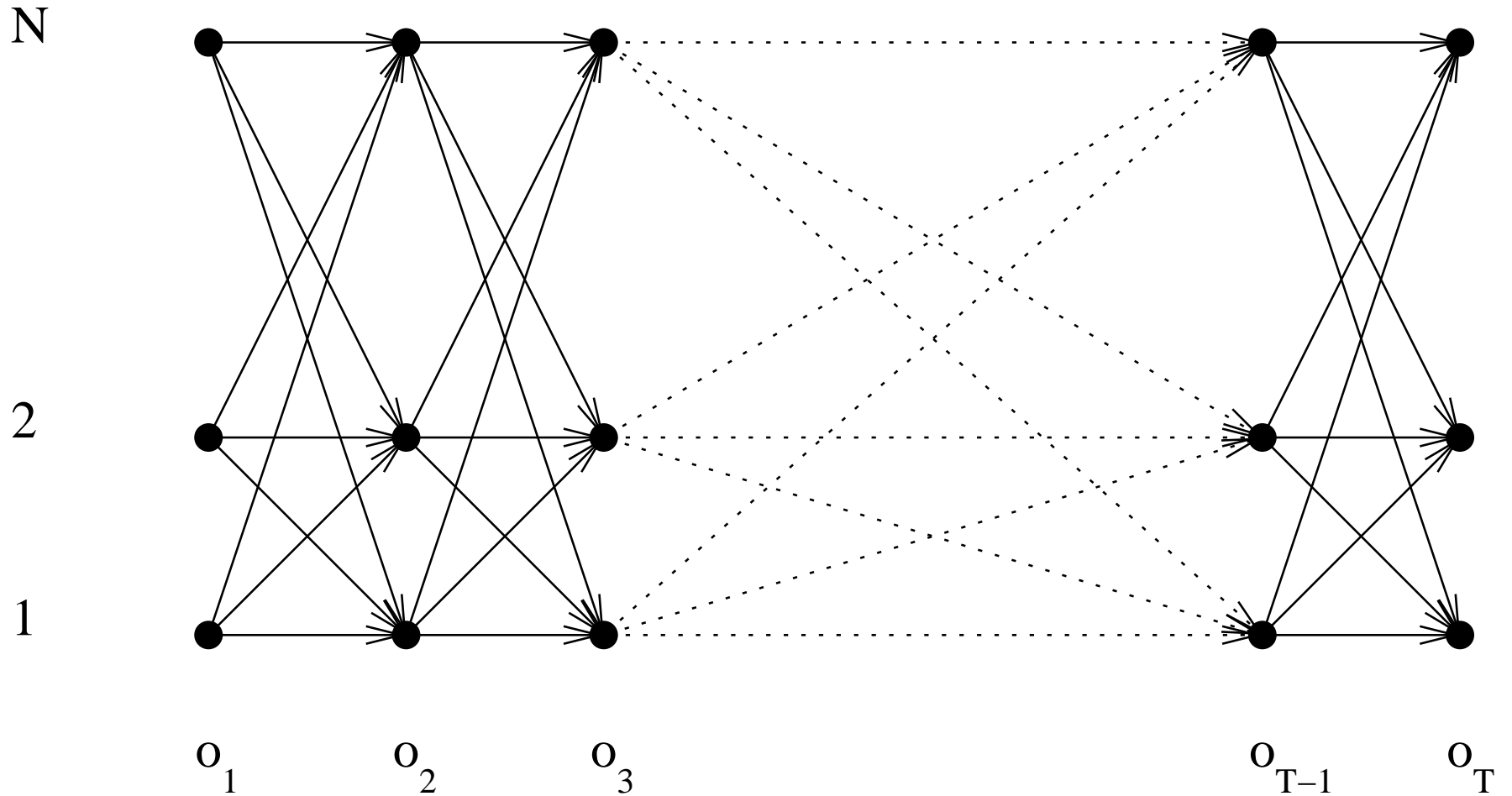
- En remplaçant  $P(o|x)$  et  $P(x)$  dans l'équation initiale, il vient :

$$P(o) = \sum_{x \in \mathcal{C}_T} \pi(x_1) \times \prod_{t=1}^{T-1} E(o_t, x_t) T(x_t, x_{t+1}) \times E(o_T, x_T)$$

# Treillis - 1

- Le calcul précédent est particulièrement inefficace, il nécessite dans le cas général (où tous les états sont reliés entre eux par une transition et chaque état peut émettre chacun des  $N$  symboles)  $2 \times T \times N^T$  multiplications ( $N^T$  chemins et  $2T$  multiplications à effectuer par chemin.).
- On a recours à une méthode de programmation dynamique pour effectuer ce calcul. Cette méthode repose sur la représentation, sous forme d'un *treillis*, de l'évolution du MMC ayant donné lieu à une suite d'observables  $o_1 \dots o_k$ .

# Treillis - 2



# Treillis - 3

.

- On associe à chaque sommet  $(i, t)$  du treillis la variable  $\alpha(i, t)$  qui correspond à la probabilité de se trouver dans l'état  $i$  du MMC  $\lambda$  à un instant  $t$ , ayant observé la suite  $o_1 \dots o_{t-1}$  :

$$\alpha(i, t) = P(o_1 \dots o_{t-1}, X_t = i)$$

# Treillis - 4

- L'avantage du treillis réside dans le fait qu'il est possible de *résumer* au niveau d'un sommet  $(i, t)$  de ce dernier des informations portant sur l'ensemble des chemins menant à l'état  $i$  à l'instant  $t$  tout en ayant observé la séquence  $o_1 \dots o_{t-1}$ . Dans notre cas, cette information est la somme des probabilités de ces chemins.
- Cette particularité permet de calculer la probabilité de se trouver dans un état quelconque à un instant  $t$  en fonction de la probabilité de se trouver dans les différents états à l'instant  $t - 1$ , c'est l'étape récursive de l'algorithme suivant.



# Algorithme de calcul de $P(o)$

.

1. Initialisation :

$$\alpha(i, 1) = \pi(i), \quad 1 \leq i \leq N$$

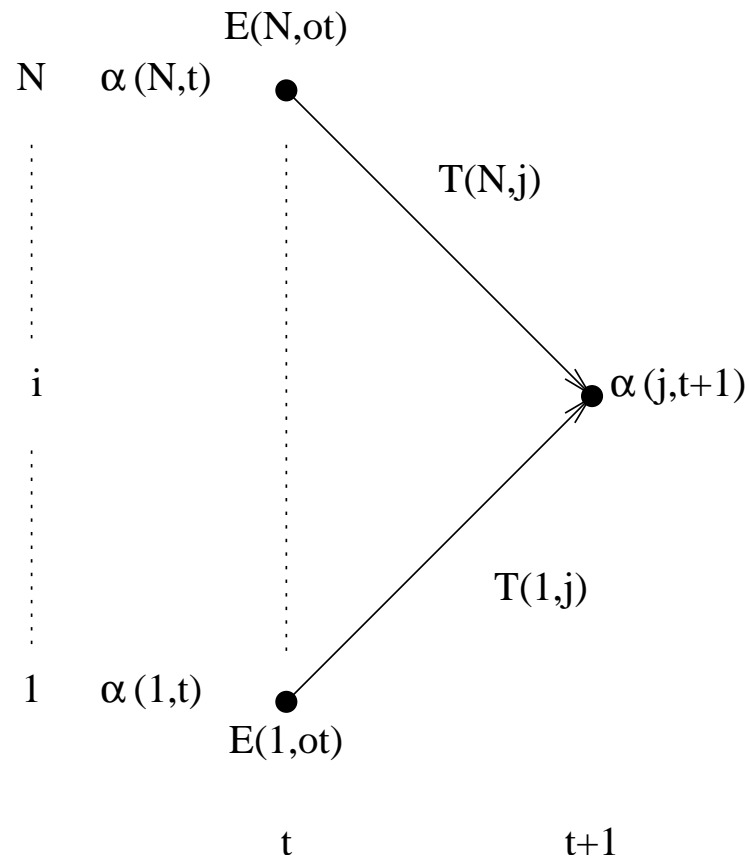
2. Etape récursive :

$$\alpha(j, t+1) = \sum_{i=1}^N \alpha(i, t) E(i, o_t) T(i, j), \quad 1 \leq t < T-1, \quad 1 \leq j \leq N$$

3. Calcul de la probabilité totale :

$$P(o) = \sum_{i=1}^N \alpha(i, T) E(i, o_T)$$

# Calcul de $\alpha(j, t + 1)$



Cette façon de calculer  $P(o)$  est bien plus économique puisqu'elle n'exige (dans le cas général) que  $2N^2T$  multiplications :  $N \times T$  sommets et  $2N$  multiplications par sommet.

# Calcul backward

.

- La procédure de calcul de  $P(o)$  présentée ci-dessus est appelée quelquefois procédure *forward* (en avant) car le calcul de la probabilité à un instant  $t$  est effectué à partir de la probabilité à un instant  $t - 1$ , en parcourant le treillis de la gauche vers la droite.
- Il est aussi possible d'effectuer le calcul dans l'ordre inverse, où la probabilité à un instant  $t$  est calculée à partir de la probabilité à l'instant  $t + 1$ .
- On définit la variable  $\beta(i, t)$  de la façon suivante :

$$\beta(i, t) = P(o_t \dots o_T | X_t = i)$$

# Calcul backward

- La procédure de calcul de  $P(o)$  présentée ci-dessus est appelée quelquefois procédure *forward* (en avant) car le calcul de la probabilité à un instant  $t$  est effectué à partir de la probabilité à un instant  $t - 1$ , en parcourant le treillis de la gauche vers la droite.
- Il est aussi possible d'effectuer le calcul dans l'ordre inverse, où la probabilité à un instant  $t$  est calculée à partir de la probabilité à l'instant  $t + 1$ .
- On définit la variable  $\beta(i, t)$  de la façon suivante :

$$\beta(i, t) = P(o_t \dots o_T | X_t = i)$$

**Attention :**  $\alpha(i, t) = P(o_1 \dots o_{t-1}, X_t = i)$

# Algorithme de calcul de $P(o)$ grâce aux probabilités *backward*

1. Initialisation :

$$\beta(i, T) = E(i, o_T), \quad 1 \leq i \leq N$$

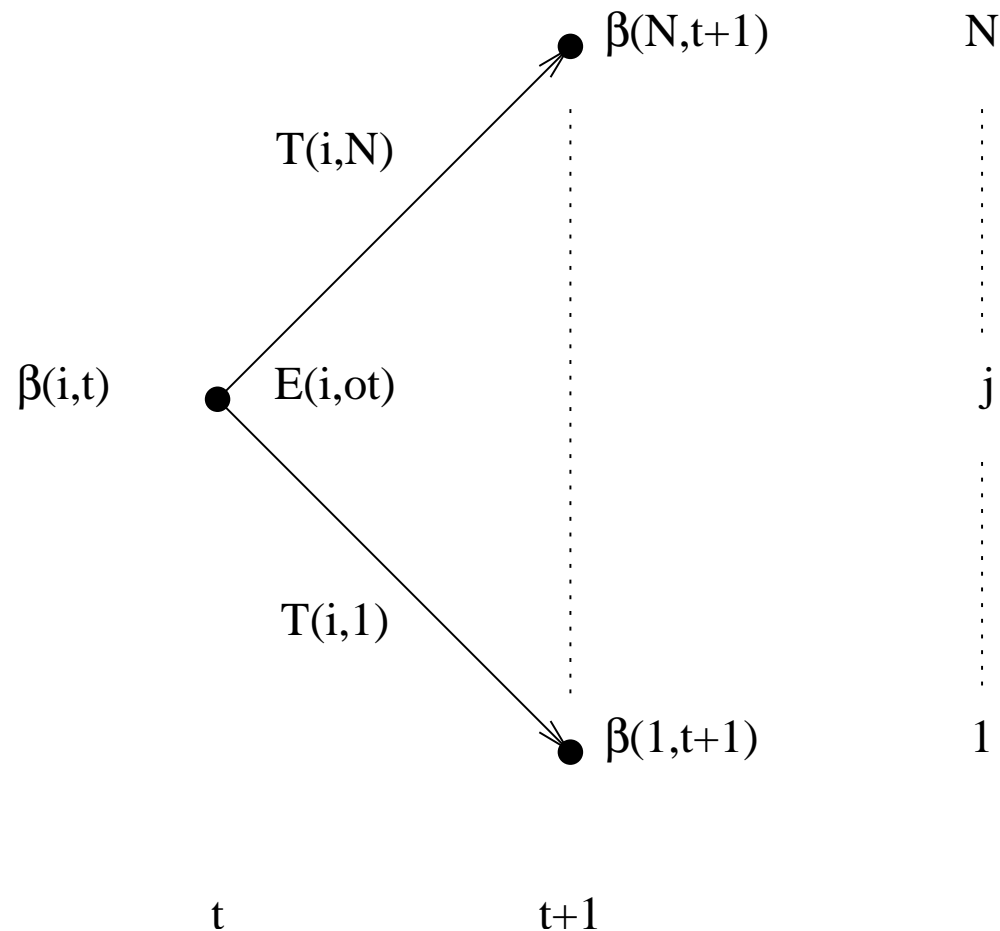
2. Etape récursive :

$$\beta(i, t) = \sum_{j=1}^N \beta(j, t+1) T(i, j) E(i, o_t), \quad 1 \leq t \leq T-1, \quad 1 \leq i \leq N$$

3. Calcul de la probabilité totale :

$$P(o) = \sum_{i=1}^N \pi(i) \beta(i, 1)$$

# Calcul de $\beta(i, t)$



$$\beta(i, t) = \sum_{j=1}^N \beta(j, t+1) T(i, j) E(i, o_t), \quad 1 \leq t \leq T-1, \quad 1 \leq i \leq N$$

# Combinaison des probabilités *backward* et *forward*

.

- Les probabilités forward et backward peuvent être combinées pour calculer  $P(o)$  de la façon suivante :

$$P(o) = \sum_{i=1}^N \alpha(i, t) \beta(i, t) \quad \forall t \quad 1 \leq t \leq T$$

- Ce résultat est établi en utilisant d'une part la formule des probabilités totales :

$$P(o) = \sum_{i=1}^N P(o, X_t = i)$$

# Combinaison des probabilités *backward* et *forward*

- puis en remarquant que chacun des termes de la somme peut être exprimée en fonction des probabilités forward et backward de la façon suivante :

$$\begin{aligned} P(o, X_t = i) &= P(o_1, \dots, o_T, X_t = i) \\ &= P(o_1, \dots, o_{t-1}, X_t = i, o_t, \dots, o_T) \\ &= P(o_1, \dots, o_{t-1}, X_t = i) \times P(o_t, \dots, o_T | o_1, \dots, o_{t-1}, X_t = i) \\ &= P(o_1, \dots, o_{t-1}, X_t = i) \times P(o_t, \dots, o_T | X_t = i) \\ &= \alpha(i, t) \beta(i, t) \end{aligned}$$



# Recherche du chemin le plus probable

- Il est souvent intéressant, étant donné un MMC  $\lambda$  et une séquence d'observations  $o = o_1 \dots o_T$  de déterminer la séquence d'états  $\hat{x} = \hat{x}_1, \hat{x}_2, \dots, \hat{x}_T$  la plus probable ayant pu générer  $o$ .
- Première solution : déterminer toutes les séquences d'états ayant pu générer  $o$ , puis de calculer leur probabilités afin de déterminer la plus probable.
- Méthode particulièrement coûteuse car, dans le cas général, il existe  $N^T$  chemins possibles.
- Solution : utiliser le treillis (algorithme de Viterbi)

# Algorithme de Viterbi

- Idée générale : on détermine, pour chaque sommet du treillis, le meilleur chemin (le chemin de probabilité maximale) menant à ce sommet, tout en ayant généré la suite  $o_1 \dots o_t$ .
- On définit pour chaque sommet  $(j, t)$  du treillis la variable  $\delta(j, t)$  :

$$\delta(j, t) = \max_{x \in \mathcal{C}_{t-1}} P(x, o_1 \dots o_t, X_t = j)$$

où  $\mathcal{C}_{t-1}$  est l'ensemble des séquences de  $t - 1$  états de  $\lambda$  et  $x$  une de ces séquences.

# Algorithme de Viterbi

.

- On définit de plus, pour chaque sommet  $(j, t)$  la variable  $\psi(j, t)$  dans laquelle est stocké l'état du MMC au temps  $t - 1$  qui a permis de réaliser le meilleur score, qui n'est donc autre que l'état précédent dans le meilleur chemin menant à  $(j, t)$ .

# Algorithme de Viterbi

1. Initialisation du treillis :

$$\delta(j, 1) = \pi(j)E(j, o_1), \quad 1 \leq j \leq N$$

2. Etape récursive :

$$\delta(j, t+1) = \max_{1 \leq i \leq N} \delta(i, t)T(i, j)E(j, o_{t+1}), \quad 1 \leq t < T, \quad 1 \leq j \leq N$$

stockage du meilleur état précédent :

$$\psi(j, t+1) = \arg \max_{1 \leq i \leq N} \delta(i, t)T(i, j)E(j, o_{t+1}), \quad 1 \leq t < T, \quad 1 \leq j \leq N$$

# Algorithme de Viterbi

.

1. Détermination du meilleur chemin :

$$\hat{x}_T = \arg \max_{1 \leq i \leq N} \delta(i, T)$$

$$\hat{x}_t = \psi(\hat{x}_{t+1}, t + 1)$$

$$P(\hat{x}) = \max_{1 \leq i \leq M} \delta(i, T)$$

# Estimation des paramètres d'un MMC

- Les paramètres d'un MMC ne sont généralement pas donnés par avance, ils doivent être estimés à partir de données.
- On suppose que l'on dispose d'une longue suite d'observations  $o = o_1 \dots o_T$ , appelée *données d'apprentissage* qui est sensée être représentative du type de données que le MMC peut produire.
- On suppose de plus que la structure du MMC (le nombre d'états et les transitions possibles entre états) est fixée.

# Estimation des paramètres d'un MMC

- L'objectif est de déterminer les paramètres qui rendent le mieux compte de  $o$ , ou, en d'autres termes, de déterminer les paramètres qui, parmi l'ensemble des paramètres possibles, attribuent à  $o$  la meilleure probabilité.
- Si l'on note  $P_\lambda(o)$  la probabilité qu'attribue le MMC  $\lambda$  à la suite  $o$ , le but de l'estimation est de déterminer le MMC  $\hat{\lambda}$  qui maximise  $P_\lambda(o)$  :

$$\hat{\lambda} = \arg \max_{\lambda} P_\lambda(o)$$

# Estimation des paramètres d'un MMC

- Nous allons supposer que la séquence  $o$  a été générée par un MMC. Ceci n'est qu'une vision de l'esprit et l'on ne connaît pas le processus qui est à l'origine de  $o$ .

- Deux cas peuvent alors se présenter :

**données complètes** : on dispose des données d'apprentissage  $o$  et de la séquence d'états  $x = x_1 \dots x_T$  ayant permis la génération de  $o$ .

**données incomplètes** : on ne dispose que de la suite d'observation  $o$ .



# Données complètes

.

états	$x =$	$x_1$	$x_2$	$x_3$	$\dots$	$x_N$
observations	$o =$	$o_1$	$o_2$	$o_3$	$\dots$	$o_N$

On définit les variables :

- $\mathcal{C}_e(i) = \sum_{t=1}^N \delta_{x_t, i}$
- $\mathcal{C}_{o,e}(a, i) = \sum_{t=1}^N \delta_{o_t, a} \times \delta_{x_t, i}$
- $\mathcal{C}_{e,e}(i, j) = \sum_{t=2}^N \delta_{x_{t-1}, i} \times \delta_{x_t, j}$

Une façon naturelle d'estimer les probabilités d'émission et de transition est :

$$E_{\hat{\lambda}}(i, a) = \frac{\mathcal{C}_{o,e}(a, i)}{\mathcal{C}_e(i)} \quad T_{\hat{\lambda}}(i, j) = \frac{\mathcal{C}_{e,e}(i, j)}{\mathcal{C}_e(i)}$$

Cette méthode d'estimation des probabilités est appelée estimation par maximum de vraisemblance.

# Données incomplètes

états  $x = ? \quad ? \quad ? \quad \dots \quad ?$   
observations  $o = o_1 \quad o_2 \quad o_2 \quad \dots \quad o_N$

- On ne dispose que des données d'apprentissage  $o$  et de la structure du MMC  $\hat{\lambda}$ .
- On ne connaît pas de méthode permettant de calculer directement  $\hat{\lambda}$ .
- Il existe une procédure, appelée algorithme de Baum-Welsh ou algorithme forward-backward qui permet de s'en approcher.
- Procédure itérative : on calcule une suite de MMC  $\lambda_0, \lambda_1, \dots, \lambda_n$  où  $\lambda_{i+1}$  est construit à partir de  $\lambda_i$  et tel que :

$$P_{\lambda_{i+1}}(o) \geq P_{\lambda_i}(o)$$

# Algorithme de Baum-Welsh

- On donne aux paramètres de  $\lambda_0$  des valeurs arbitraires, qui peuvent être aléatoires, comme elles peuvent être guidées par la connaissance a priori que nous avons du problème.
- On considère que  $o$  a été généré par  $\lambda_0$ . Cette hypothèse permet de calculer la probabilité, notée  $\gamma(i, t)$ , que  $\lambda_0$  soit dans l'état  $i$  à l'instant  $t$  :

$$\begin{aligned}\gamma(i, t) &= P(X_t = i | o) \\ &= \frac{P(X_t = i, o)}{p(o)} \\ &= \frac{\alpha(i, t)\beta(i, t)}{\sum_{j=1}^N \alpha(j, t)\beta(j, t)}\end{aligned}$$

# Algorithme de Baum-Welsh - 2

- On effectue la somme  $\sum_{t=1}^T \gamma(i, t)$
- Somme des probabilités que  $\lambda_0$  soit passé par l'état  $i$  aux différents instants  $t$  de la génération de  $o$ .
- Il ne s'agit pas d'une probabilité :
  - elle peut être supérieure à 1
  - on ne voit à quel événement elle correspond.
- On l'interprète comme une approximation du nombre de fois que  $\lambda_0$  est passé par l'état  $i$  lors de la génération de  $o$ .
- On se retrouve dans une situation proche de l'estimation avec des données complètes.

# Réestimation des probabilités d'émission

.

On peut calculer (on dit aussi réestimer) de nouvelles probabilités d'émission, notées  $E_1$ , par maximum de vraisemblance :

$$\begin{aligned} E_1(i, a_j) &= \frac{\text{nombre de fois que } \lambda_0 \text{ s'est trouvé dans l'état } i \text{ et que } a \text{ a été émis}}{\text{nombre de fois que } \lambda_0 \text{ s'est trouvé dans l'état } i} \\ &= \frac{\sum_{t:o_t=a} \gamma(i, t)}{\sum_{t=1}^T \gamma(i, t)} \end{aligned}$$

# Réestimation des probabilités initiales

▪

les probabilités initiales peuvent, elles, être réestimées de la façon suivante :

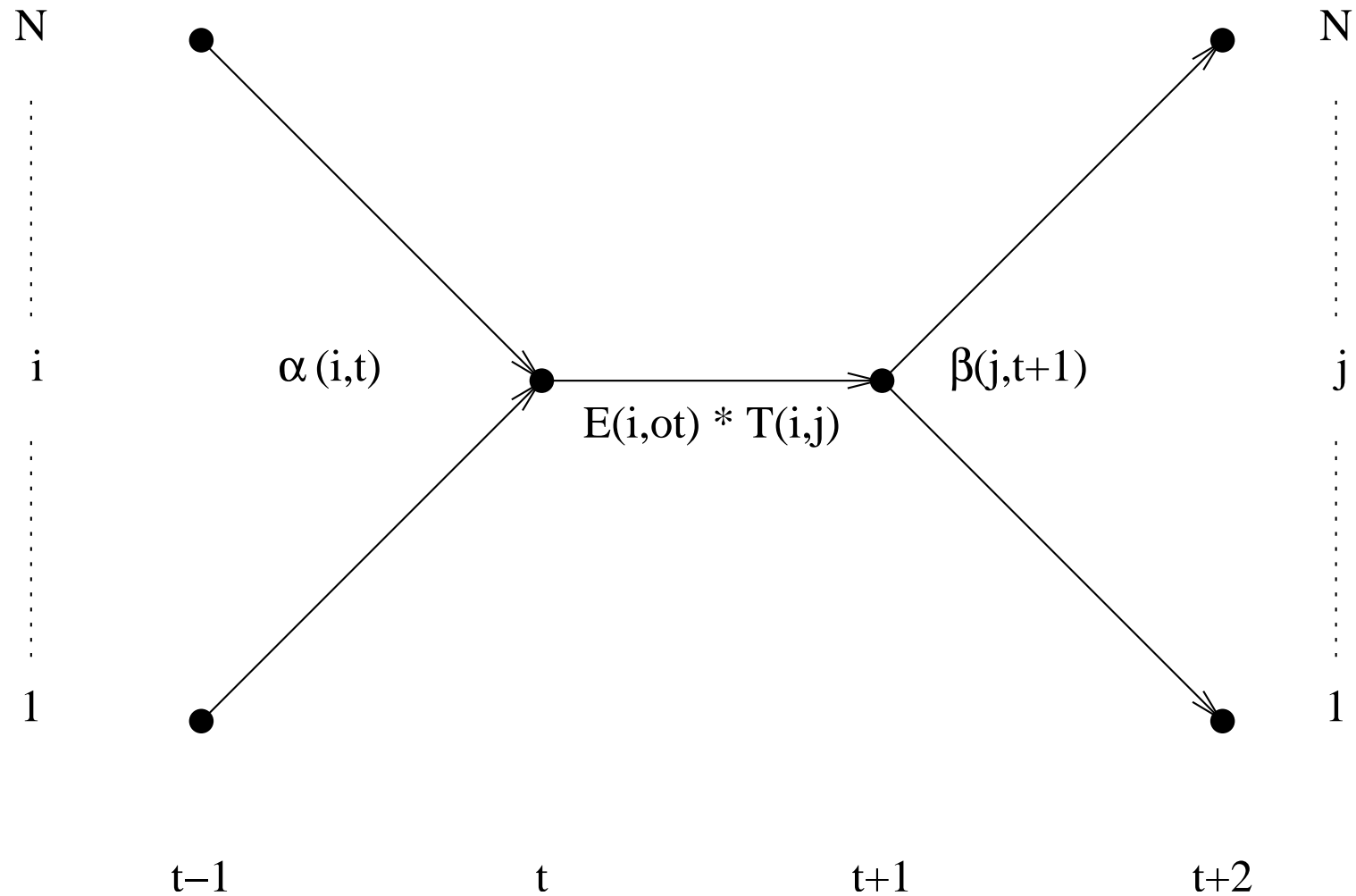
$$\begin{aligned}\pi_1(i) &= \text{probabilité d'être en } i \text{ à l'instant } t = 1 \\ &= \gamma(i, 1)\end{aligned}$$

# Réestimation des probabilités de transition

- On note  $p_t(i, j)$  la probabilité que  $\lambda_0$  soit passé de l'état  $i$  à l'état  $j$  entre les instants  $t$  et  $t + 1$  :

$$\begin{aligned} p_t(i, j) &= P(X_t = i, X_{t+1} = j | o) \\ &= \frac{P(X_t = i, X_{t+1} = j, o)}{P(o)} \\ &= \frac{\alpha(i, t) \times E(i, o_t) \times T(i, j) \times \beta(j, t + 1)}{\sum_{k=1}^N \alpha(k, t) \beta(k, t)} \end{aligned}$$

# Réestimation des probabilités de transition - 2





# Réestimation des probabilités de transition - 2

.

On peut maintenant effectuer la somme  $\sum_{t=1}^T p_t(i, j)$  que nous allons interpréter comme le nombre de fois qu'une transition de  $i$  vers  $j$  a été empruntée lors de la génération de  $o$  et on peut recalculer à partir de cette quantité des nouvelles probabilités de transition  $T_1$  par maximum de vraisemblance :

$$\begin{aligned} T_1(i, j) &= \frac{\text{nombre de fois qu'une transition de } i \text{ vers } j \text{ a été empruntée}}{\text{nombre de fois qu'un transition émanant de } i \text{ a été empruntée}} \\ &= \frac{\sum_{t=1}^T p_t(i, j)}{\sum_{t=1}^T \gamma(i, t)} \end{aligned}$$

# Réestimation des probabilités de transition - 2

.

- $\lambda_1$  possède la propriété remarquable d'attribuer à la séquence  $o$  une probabilité meilleure ou égale à celle que lui attribuait  $\lambda_0$  :

$$P_{\lambda_1}(o) \geq P_{\lambda_0}(o)$$

- Cette propriété s'explique par le fait que lors du calcul des paramètres de  $\lambda_1$ , nous avons augmenté la probabilité des transitions et des émissions qui étaient à l'origine de la génération de  $o$ , et ce faisant, diminué les autres probabilités.

# Réestimation des probabilités de transition - 2

- En réitérant le processus de réestimation des probabilités, nous obtiendrons des paramètres attribuant une probabilité de plus en plus élevée à la séquence  $o$ , jusqu'à ce qu'une valeur limite soit atteinte, pour un MMC  $\lambda_n$ .
- $\lambda_n$  n'est cependant pas le meilleur possible, il peut s'agir d'un maximum local, qui dépend de  $\lambda_0$  :

