Plan

Deux modèles génératifs

- Permettent de générer des séquences d'observables $o = o_1 \dots o_k$
- Chaînes de Markov Cachées (HMM)
  - modèle sous jacent : automate fini
  - pas de mémoire : la probabilité de génération d'un symbole ne dépend que de l'état dans lequel se trouve l'automate
- Grammaires hors-contexte probabilistes (PCFG)
  - modèle sous jacent : automate à pile
  - la pile permet de modéliser des dépendances de longueur arbitraire

Deux modèles génératifs

Trois questions :

1. Comment calculer $P(o)$
2. Comment calculer la structure sous-jacente la plus probable pour o
   - HMM : séquence d'états de l'automate
   - PCFG : arbre syntaxique
3. Comment optimiser les paramètres du modèle étant donné une (longue) séquence d'observables $o_1 \ldots o_N$

Processus stochastique

- Un **processus stochastique** (ou processus aléatoire) est une séquence $X_1, X_2 \ldots X_N$ de variables aléatoires fondées sur le même ensemble fondamental.
- 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'interêt 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 alétaoire X_1, qui spécifie donc l'état du processus lors de la première observation.
  2. pour toute valeur de $t > 1$ la probabilité conditionnelle :

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

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 $t-1$:

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

- La probabilité de passage d'un état i à un état j est **constante**, elle ne varie pas avec le temps :

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

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 π, de la variable X_1 :

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

Processus de Markov

On peut éviter le recours à la loi π 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 π :

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

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 $T(i, j)$.

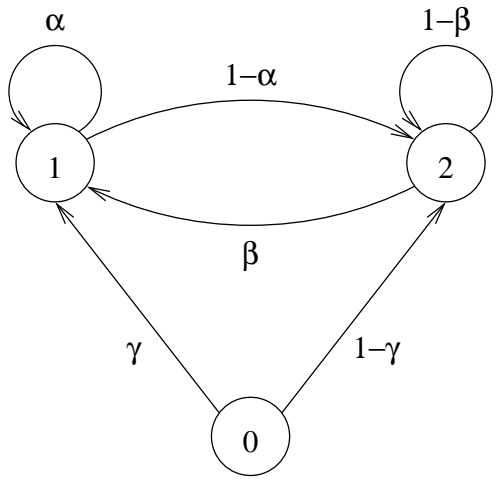
Exemple

- On admet que le fait que le temps qu'il fera demain ne dépend que du temps qu'il fait aujourd'hui.
- Plus précisément, s'il pleut aujourd'hui, il pleuvra demain aussi avec une probabilité de α et s'il ne pleut pas aujourd'hui la probabilité qu'il pleuve demain est β.
- 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 γ.
- 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 $(X_1, X_2, \ldots X_T)$) :

  $P(X_1, X_2, \ldots, X_T) =$ (règle de multiplication)
  $P(X_1)P(X_2|X_1)P(X_3|X_1, X_2)\ldots P(X_T|X_1, \ldots, X_{T-1}) =$
  $P(X_1)P(X_2|X_1)P(X_3|X_2)\ldots P(X_T|X_{T-1})$(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 :

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

Modèles de Markov Cachés

- Dans les chaînes de Markov, les observations correspondent aux états du processus.
- Dans un modèle de Markov caché, on ne peux 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'observations on ne peux savoir par quelle séquence d'états (ou *chemin*) le processus est passé, d'où le nom de modèles de Markov cachés (HMM).
- On distingue le processus $X = X_1, X_2, \ldots, X_T$ qui représente l'évolution des états du HMM et le processus $O = O_1, O_2, \ldots, O_T$ qui représente la suite des symboles émis par le HMM.

Eléments d'un HMM

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

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

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

Eléments d'un HMM - 2

Un HMM 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 $T(i, j)$.
- La somme des probabilités des transitions émanant d'un état vaut 1 :

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

Eléments d'un HMM - 3

Un HMM 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 HMM constituent une loi de probabilité :

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

L'ensemble constitué des probabilités initiales, des probabilités de transition et d'émission d'un HMM λ est souvent appelé les *paramètres* λ.

Exemple

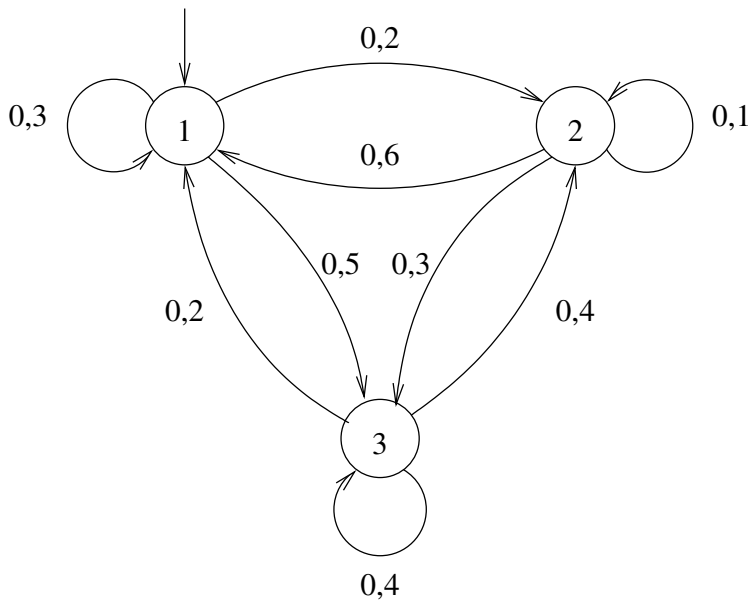
$\lambda_1 = \langle \{1,2,3\}, \{a,b,c\}, \pi, T, E \rangle$ avec :

$$\begin{array}{lll} E(1,a) = 0,6 & E(2,a) = 0 & E(3,a) = 0,3 \\ E(1,b) = 0,2 & E(2,b) = 0,5 & E(3,b) = 0 \\ E(1,c) = 0,2 & E(2,c) = 0,5 & E(3,c) = 0,7 \end{array} \quad \text{et} \quad \begin{array}{lll} T(1,1) = 0,3 & T(2,1) = 0,6 & T(3,1) = 0,2 \\ T(1,2) = 0,2 & T(2,2) = 0,1 & T(3,2) = 0,4 \\ T(1,3) = 0,5 & T(2,3) = 0,3 & T(3,3) = 0,4 \end{array}$$

et

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

représentation graphique



Trois questions

- Calcul de la probabilité d'une séquence d'observations o :

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

où $\mathcal{C}_T$ est l'ensemble des séquences de T états.

- Calcul du chemin le plus probable :

$$\hat{x} = \arg\max_{x \in \mathcal{C}_T} P(x|o)$$

- Estimation des paramètres du HMM :

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

Calcul de $P(o)$

- Etant donné un HMM $\lambda = \langle S, A, \pi, T, E \rangle$, la suite d'observation $o = o_1 o_2, \ldots, o_T$ peut généralement être générée en suivant différents chemins dans le HMM
- La probabilité que λ 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 λ et $x = x_1, \ldots, x_T$ ($x_i \in S$, $1 \leq i \leq T$) une de ces séquences

Calcul de $P(o)$

- la probabilité conditionnelle que o soit générée lorsque λ passe successivement par la séquence d'états $x = x_1, \ldots, x_T$ 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 $P(o)$

- et la probabilité que le HMM suive une séquence particulière d'états x est le produit des probabilités que λ 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* :

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

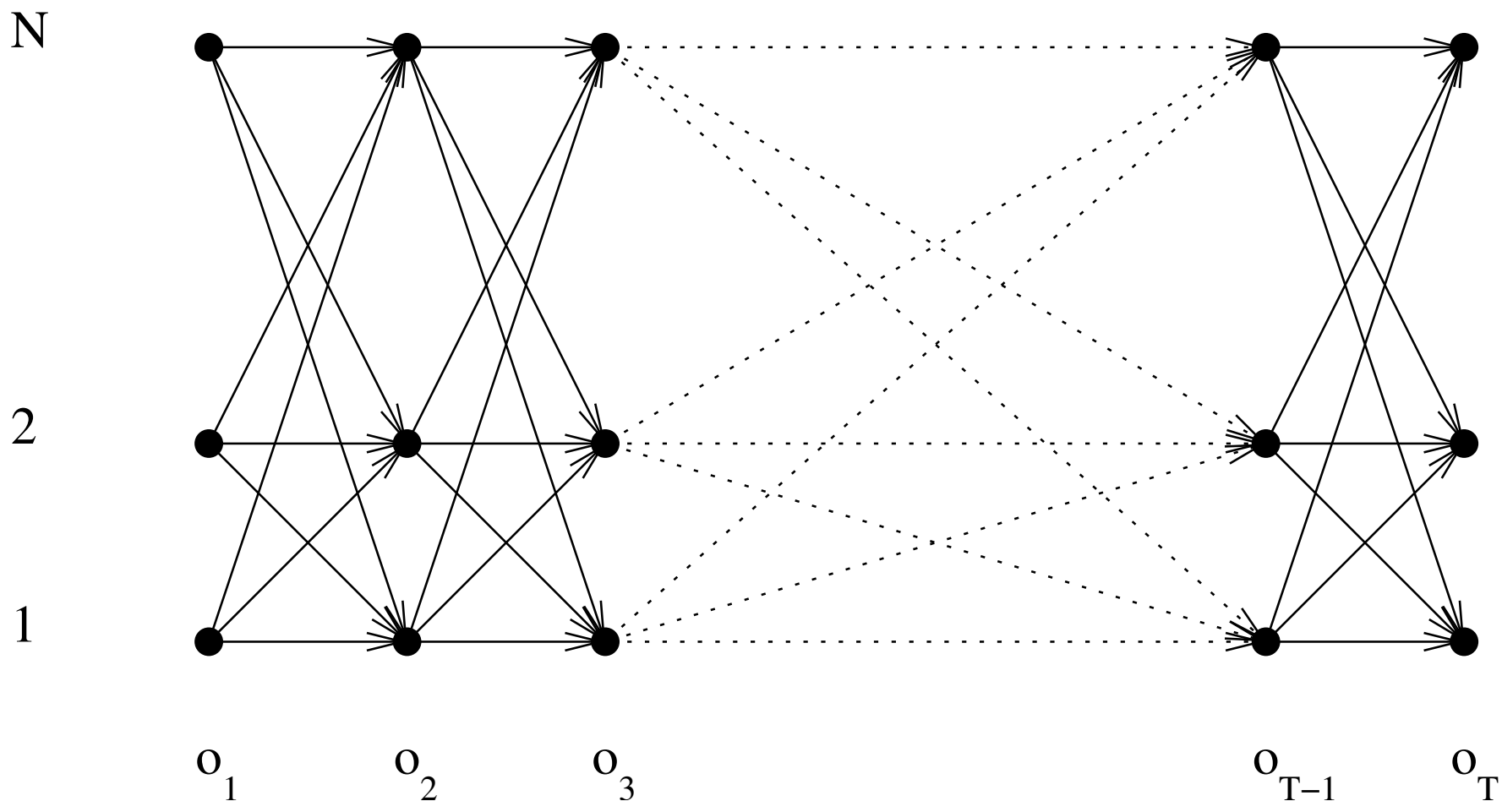
- En remplaçant $P(o|x)$ et $P(x)$ dans l'équation initiale, on obtient :

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

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 HMM ayant donné lieu à une suite d'observables $o_1 \ldots o_k$.

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 HMM λ 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

- Le treillis permet de *résumer* au niveau d'un sommet (i, t) 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 :

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

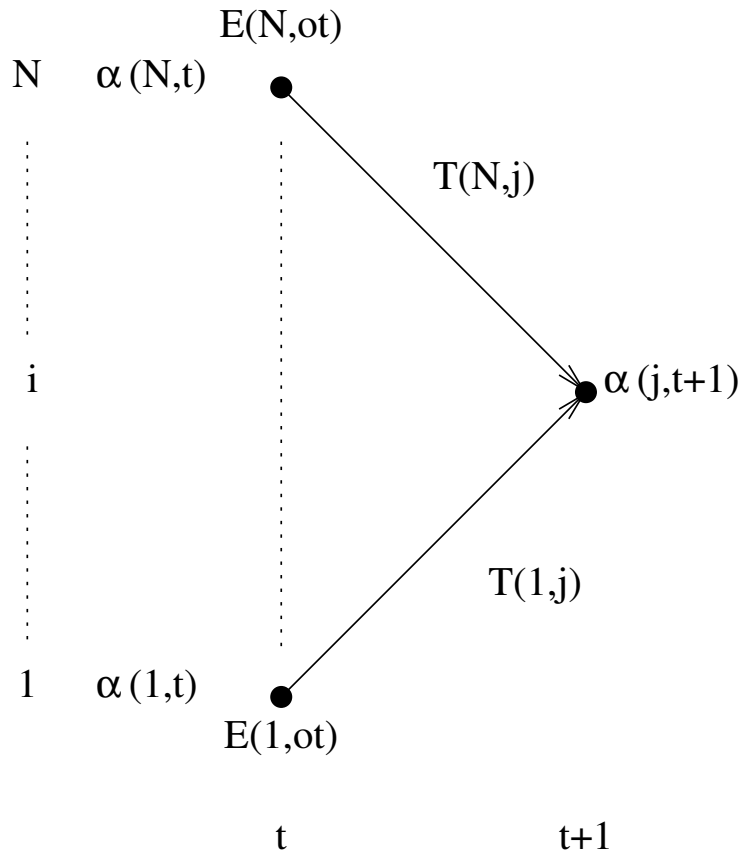
2. Etape récursive :

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

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 :

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

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 :

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

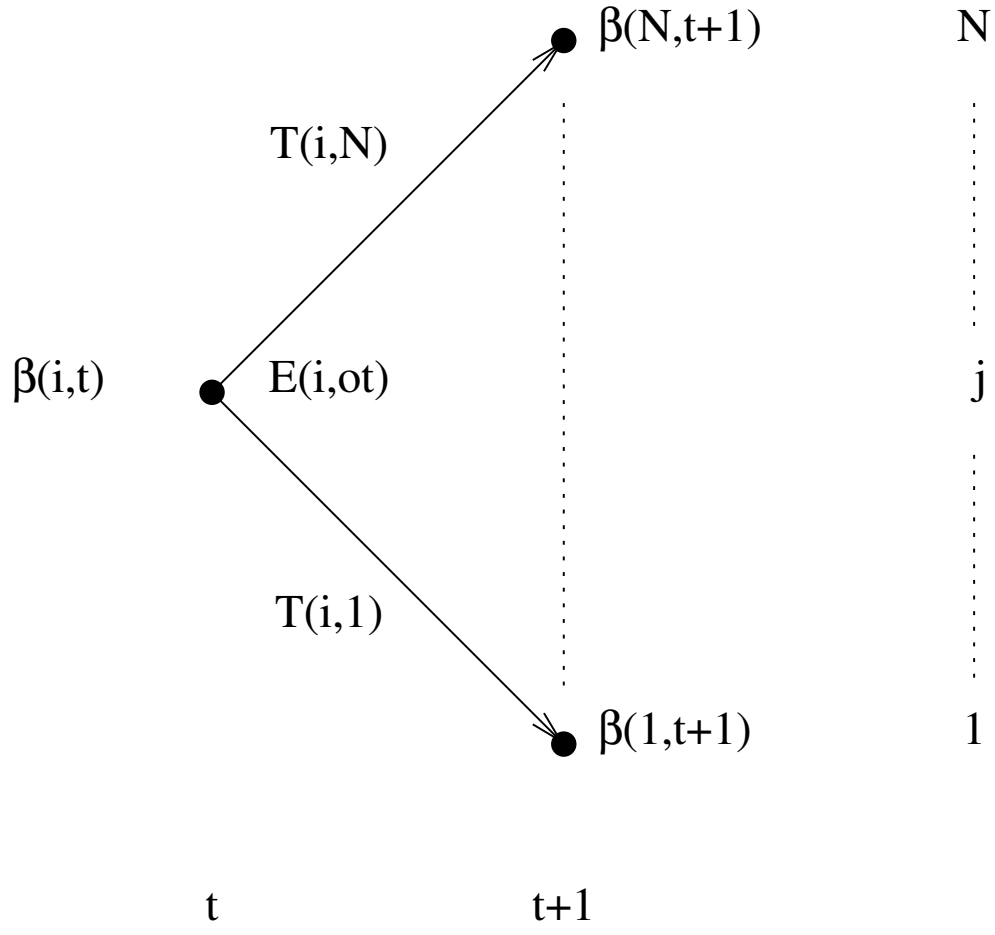
2. Etape récursive :

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

3. Calcul de la probabilité totale :

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

Calcul de $\beta(i,t)$



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

Combinaison des probabilités *backward* et *forward*

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

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

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

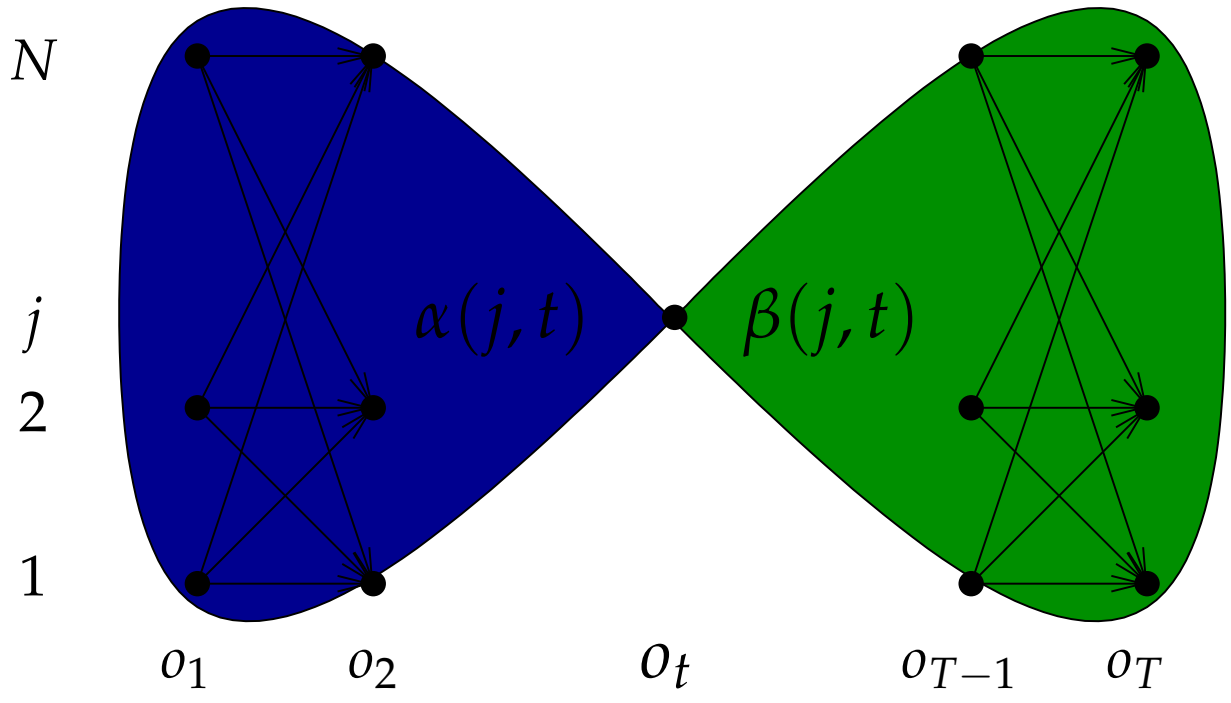
$$P(o) = \sum_{i=1}^{N} P(o, X_t = i)$$

Combinaison des probabilités *backward* et *forward*

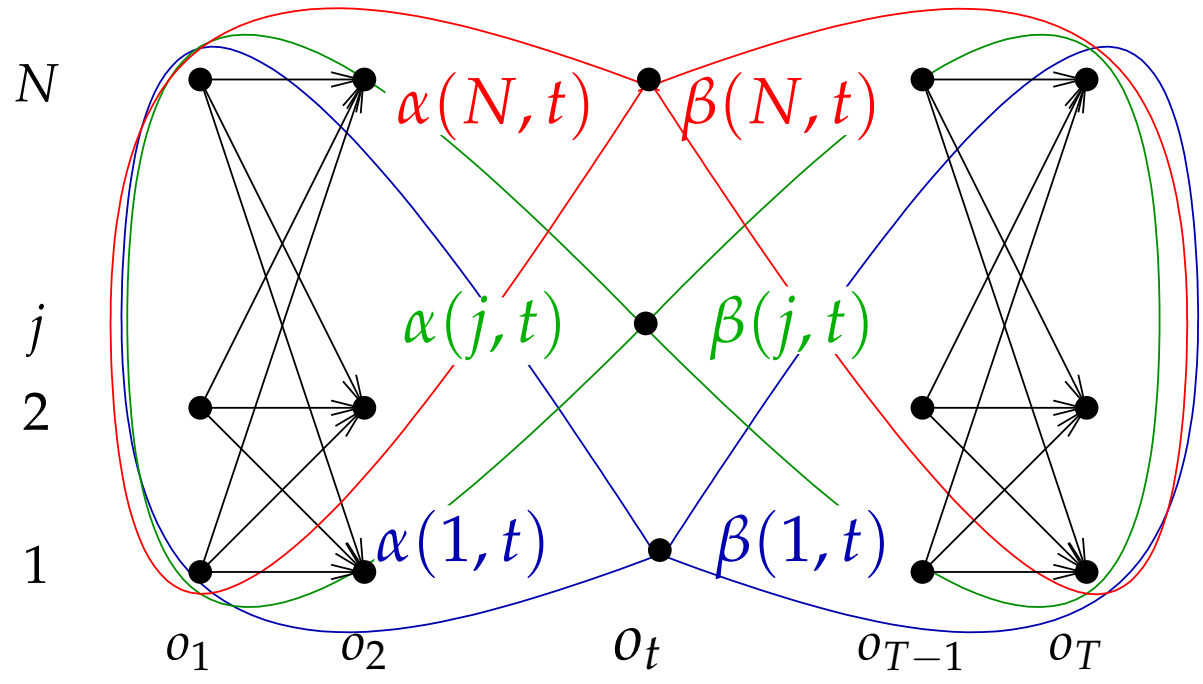
- puis en remarquant que chacun des termes de la somme peut être exprimée en fonction des probabilités forward et backward de la façon suivante :

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

Combinaison des probabilités *backward* et *forward*



Probabilité de o



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

Recherche du chemin le plus probable

- Il est souvent intéressant, étant donné un HMM λ 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 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 λ 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 HMM 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 :

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

2. Etape récursive :

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

stockage du meilleur état précédent :

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

Algorithme de Viterbi

1. Détermination du meilleur chemin :

$$\begin{aligned} \hat{x}_T &= \arg \max_{1 \leq i \leq N} \delta(i, T) \\ \hat{x}_t &= \psi(\hat{x}_{t+1}, t+1) \\ P(\hat{x}) &= \max_{1 \leq i \leq M} \delta(i, T) \end{aligned}$$

Estimation des paramètres d'un HMM

- Les paramètres d'un HMM ne sont généralement pas données 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 HMM peut produire.
- On suppose de plus que la structure du HMM (le nombre d'états et les transitions possibles entre états) est fixée.

Estimation des paramètres d'un HMM

- 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 HMM λ à la suite o, le but de l'estimation est de déterminer le HMM $\hat{\lambda}$ qui maximise $P_\lambda(o)$:

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

Estimation des paramètres d'un HMM

- Nous allons supposer que la séquence o a été générée par un HMM. 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	$\ldots$	x_T
observations	$o =$	o_1	o_2	o_3	$\ldots$	o_T

On définit les variables :

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

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

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

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

Données incomplètes

états	$x =$	?	?	?	...	?
observations	$o =$	o_1	o_2	o_2	...	o_T

- On ne dispose que des données d'apprentissage o et de la structure du HMM $\hat{\lambda}$.
- 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 HMM $\lambda_0, \lambda_1, \ldots, \lambda_n$ où λ_{i+1} est construit à partir de λ_i et tel que :

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

Algorithme de Baum-Welsh

- On donne aux paramètres de λ_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 λ_0. Cette hypothèse permet de calculer la probabilité, notée $\gamma(i,t)$, que λ_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 λ_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 évenement elle correspond.
- On l'interprète comme une approximation du nombre de fois que λ_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 :

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

Réestimation des probabilités initiales

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

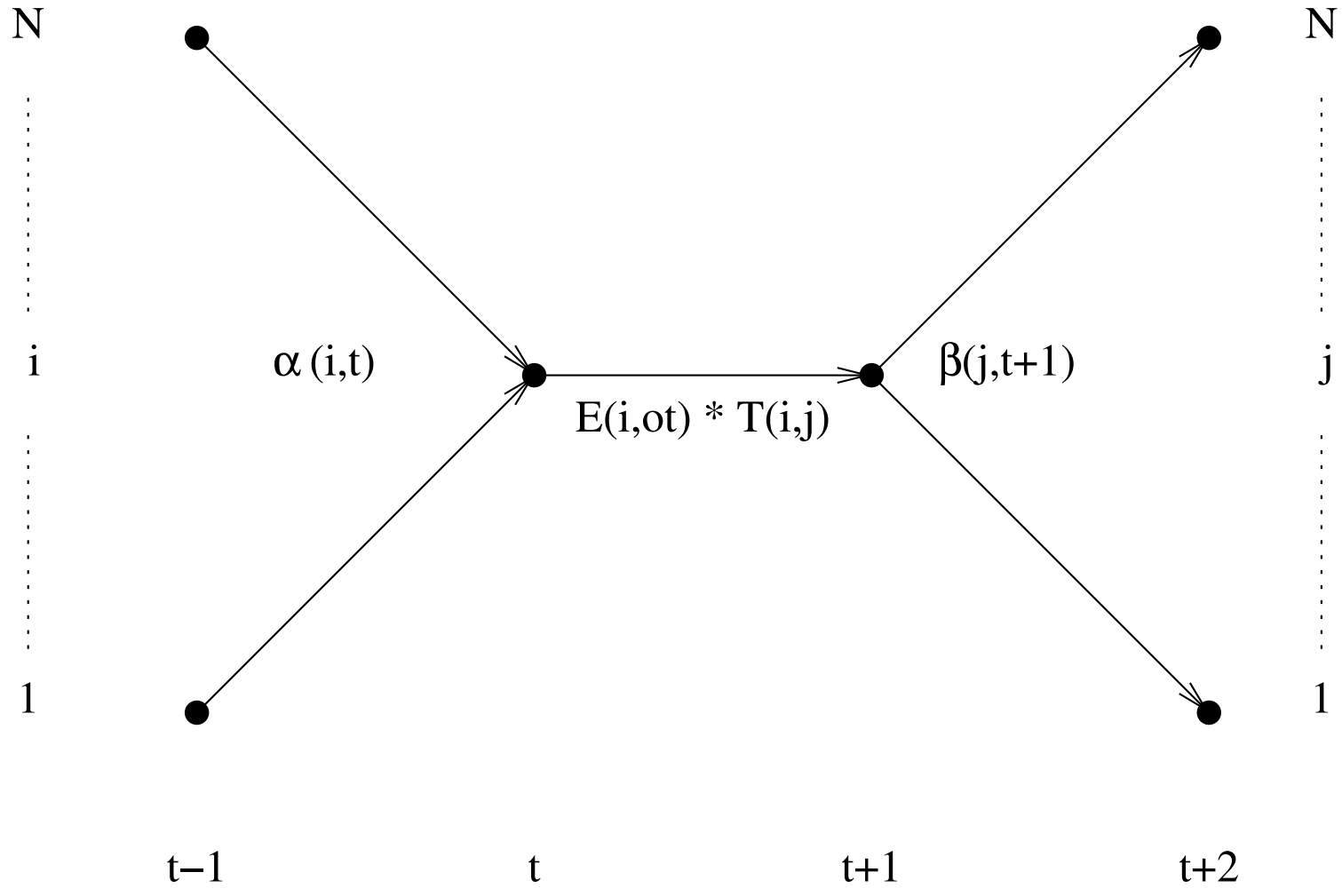
$$\begin{aligned} \pi_1(i) &= \text{probabilité d'être en } i \text{ à l'instant } t = 1 \\ &= \gamma(i, 1) \end{aligned}$$

Réestimation des probabilités de transition

- On note $p_t(i,j)$ la probabilité que λ_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 effectue la somme $\sum_{t=1}^{T} p_t(i,j)$
- Estimation du nombre de fois qu'une transition de i vers j a été empruntée lors de la génération de o
- on recalcule à partir de cette quantité des nouvelles probabilités de transition T_1 par maximum de vraisemblance :

$$T_1(i,j) = \frac{\text{nombre de fois qu'une transition de i vers j a été empruntée}}{\text{nombre de fois qu'un transition émanant de i a été empruntée}}$$

$$= \frac{\sum_{t=1}^{T} p_t(i,j)}{\sum_{t=1}^{T} \gamma(i,t)}$$

Réestimation des probabilités de transition - 2

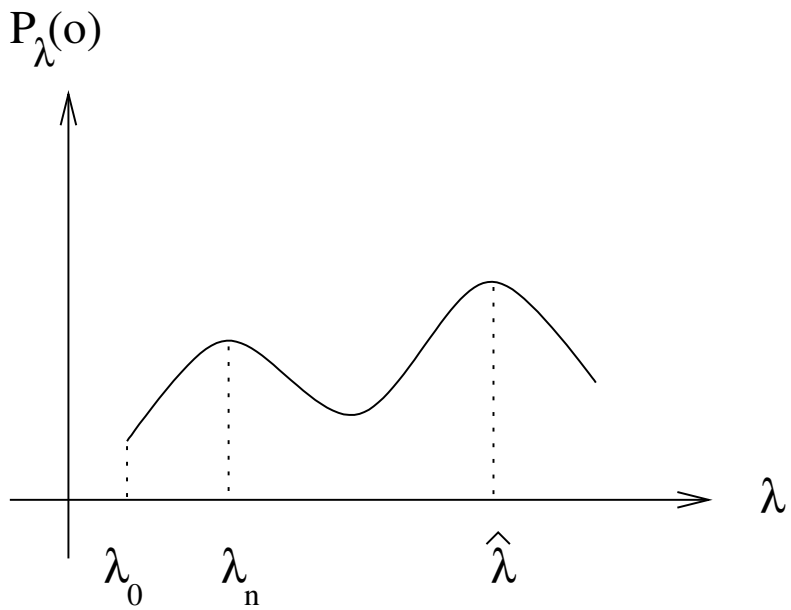
- λ_1 possède la propriété remarquable d'attribuer à la séquence o une probabilité meilleure ou égale à celle que lui attribuait λ_0 :

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

- Cette propriété s'explique par le fait que lors du calcul des paramètres de λ_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 pocessus 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 HMM λ_n.
- λ_n n'est cependant pas le meilleur possible, il peut s'agir d'un maximum local, qui dépend de λ_0 :



Grammaires hors-contexte probabiliste

Une grammaire hors-contexte probabiliste est composée de :

- Un alphabet non terminal $\mathcal{N} = \{N^1 \dots N^n\}$
- Un alphabet terminal $\mathcal{T} = \{t^1 \dots t^m\}$
- Un axiome N^1
- Un ensemble de règles $N^i \rightarrow \alpha$ avec $\alpha \in (\mathcal{N} \cup \mathcal{T})^*$
- Une distribution de probabilité associée à tout N^i :

$$\sum_j P(N^i \rightarrow \alpha^j) = 1$$

- On fera l'hypothèse que la grammaire est sous forme normale de Chomsky.

Probabilités

- La probabilité d'une règle est la probabilité de choisir cette règle pour réécrire le symbole de la partie gauche.

$$P(N^i \rightarrow \alpha^j) = P(N^i \rightarrow \alpha^j | N^i)$$

- Probabilité d'un arbre T :

$$P(T) = \prod_{n \in T} P(r(n))$$

où $n \in \mathcal{N}$ et $r(n)$ désigne la règle ayant permis de réécrire n.

- Probabilité d'une phrase S :

$$P(S) = \sum_{T \in \mathcal{T}(S)} P(T)$$

où $\mathcal{T}(S)$ est l'ensemble des analyses de S.

Trois problèmes à résoudre

- Calcul de la probabilité d'une phrase S :

$$P(S) = \sum_{T \in \mathcal{T}(S)} P(T)$$

- Construction de l'arbre d'analyse de S le plus probable :

$$\hat{T} = \arg \max_{T \in \mathcal{T}(S)} P(T)$$

- Estimation des probabilités de G à partir de données D :

$$\hat{G} = \arg \max_{G} P(D|G)$$

Parallèle avec les chaînes de Markov cachées

- Calcul de la probabilité d'une séquence d'observables o :

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

où $\mathcal{C}_T$ est l'ensemble des séquences de T états et x une de ces séquences.

- Calcul du chemin le plus probable :

$$\hat{x} = \arg\max_{x \in \mathcal{C}_T} P(x|o)$$

- Estimation des probabilités du HMM :

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

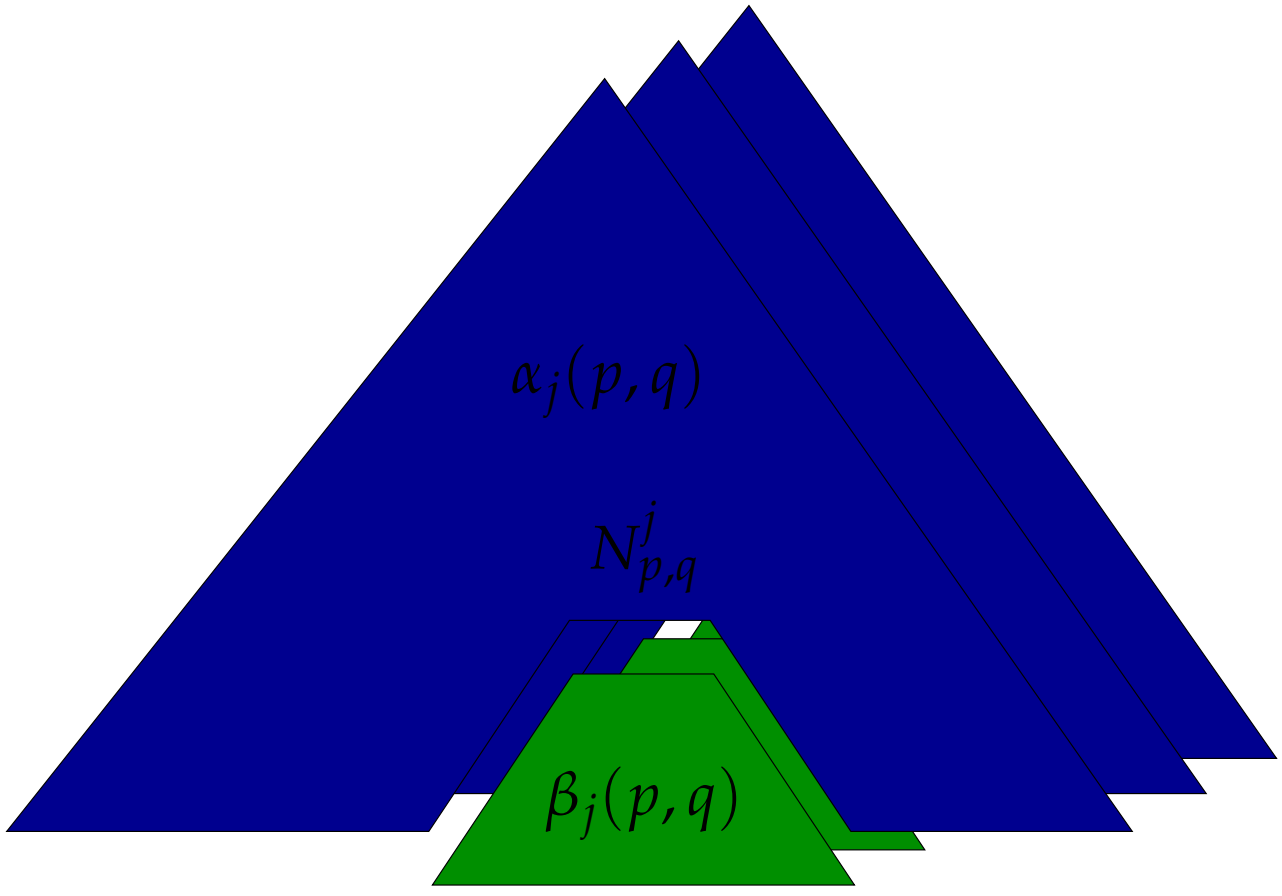
Différences

- Dans le cas d'un HMM une séquence d'états peut correspondre à plusieurs séquences d'observables alors qu'un arbre de dérivation correspond à une seule séquence d'observables (de terminaux).
- Etant donné un HMM et une séquence d'observables o, il est facile de déterminer tous les chemins ayant pu générer o. Dans le cas d'une grammaire probabiliste, il faut déterminer l'ensemble $\mathcal{T}(S)$: faire l'analyse syntaxique de S.

Notations

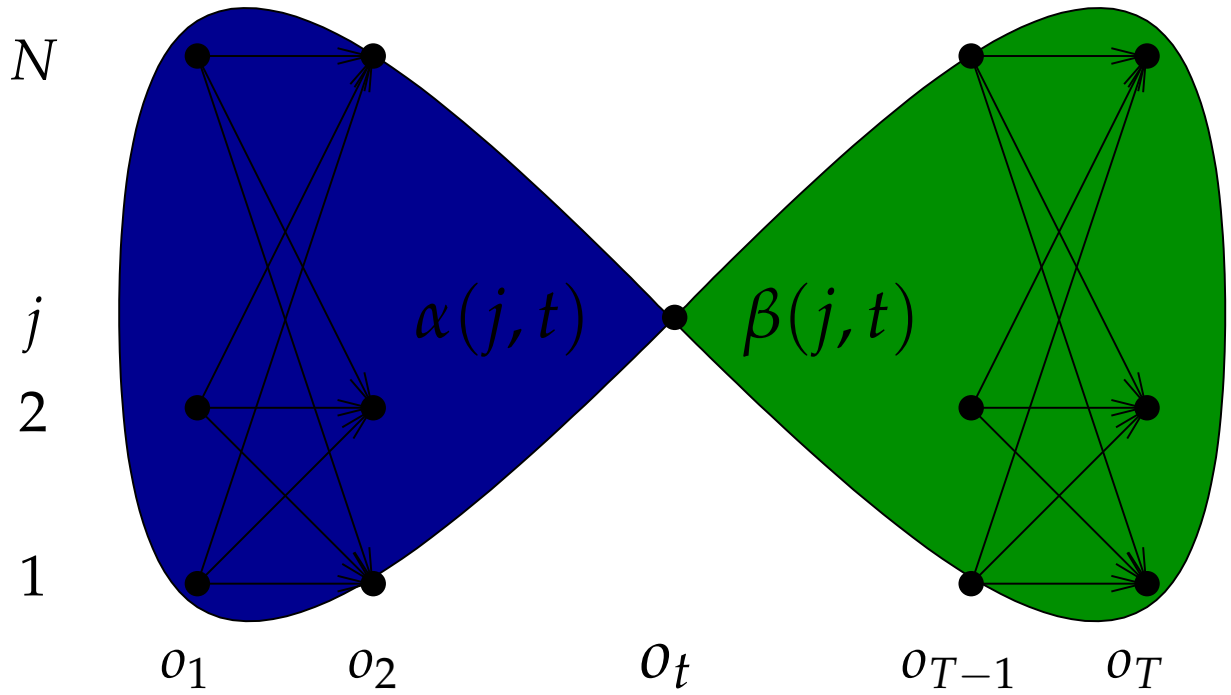
$m_1 \ldots m_n$	phrase à analyser
$m_{p,q}$	segment $m_1 \ldots m_n$ de la phrase
m^i	symbole de l'alphabet terminal
N^j	symbole de l'alphabet non terminal
$N^j_{p,q}$	le symbole N^j permet de dériver le segment $m_{p,q}$

Probabilités extérieures, probabilités intérieures



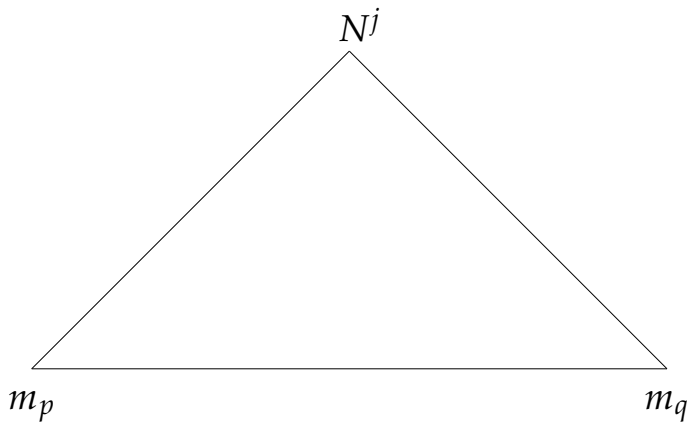
$$\beta_j(p,q) \stackrel{def}{=} P(m_{p,q}|N^j_{p,q}) \qquad \alpha_j(p,q) \stackrel{def}{=} P(m_{1,p-1}, N^j_{p,q}, m_{q+1,m})$$

Parallèle avec les HMM



Probabilité intérieure

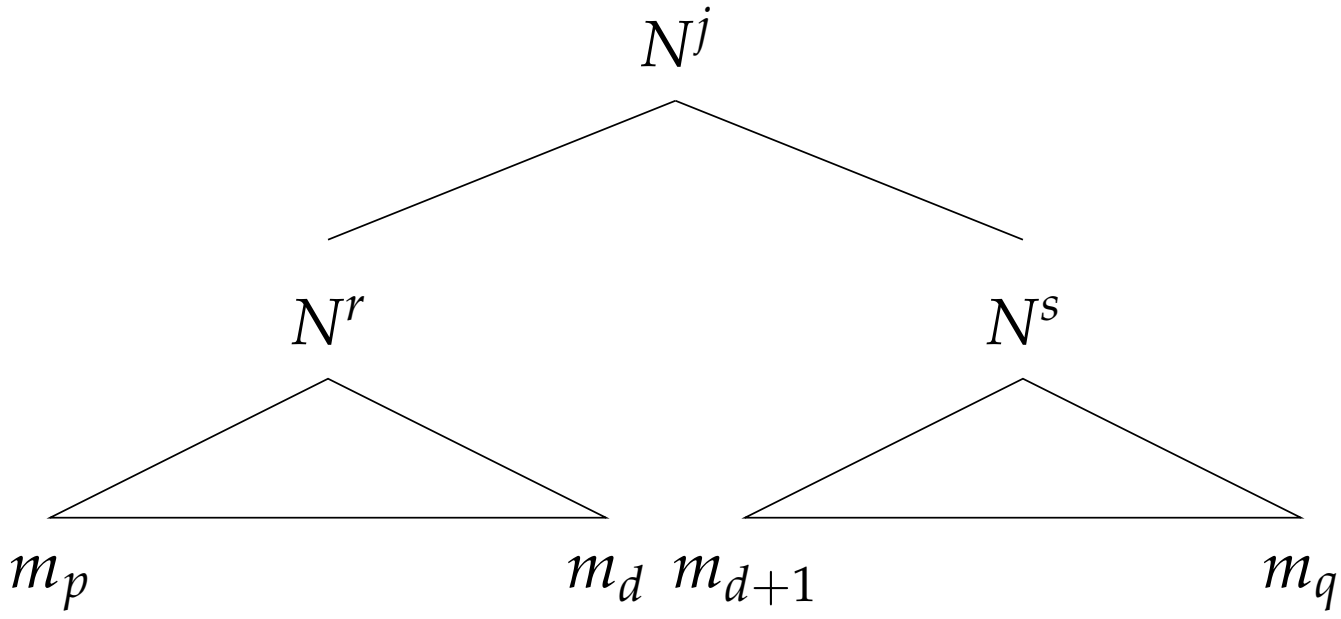
$$\beta_j(p,q) \stackrel{def}{=} P(m_{p,q}|N^j_{p,q})$$



Probabilité d'une phrase

$$
\begin{aligned}
P(m_{1,n}) &= P(N^1 \overset{*}{\Rightarrow} m_{1,n}) \\
&= P(m_{1,n}|N^1) \\
&= \beta_1(1,n)
\end{aligned}
$$

Calcul récursif des probabilités intérieures



- Calcul ascendant
- On calcule $\beta_j(p,q)$ à partir de $\beta_r(p,d)$ et $\beta_s(d+1,q)$

Calcul récursif des probabilités intérieures

Relation de récurrence

$$
\begin{aligned}
\beta_j(p,q) &= P(m_{p,q}|N^j_{p,q}) \\
&= \sum_{r,s}\sum_{d=p}^{q-1} P(m_{p,d}, N^r_{p,d}, m_{d+1,q}, N^s_{d+1,q}|N^j_{p,q}) \\
&= \sum_{r,s}\sum_{d=p}^{q-1} P(N^r_{p,d}, N^s_{d+1,q}|N^j_{p,q}) P(m_{p,d}|N^r_{p,d}, N^s_{d+1,q}, N^j_{p,q}) \\
&\qquad P(m_{d+1,q}|N^r_{p,d}, N^s_{d+1,q}, N^j_{p,q}) \\
&= \sum_{r,s}\sum_{d=p}^{q-1} P(N^j \to N^r, N^s)\beta_r(p,d)\beta_s(d+1,q)
\end{aligned}
$$

Calcul récursif des probabilités intérieures

Cas terminal

$$\begin{aligned} \beta_j(k,k) &= P(m_k | N^j_{k,k}) \\ &= P(N^j \rightarrow m_k) \end{aligned}$$

Relation avec l'algorithme CYK

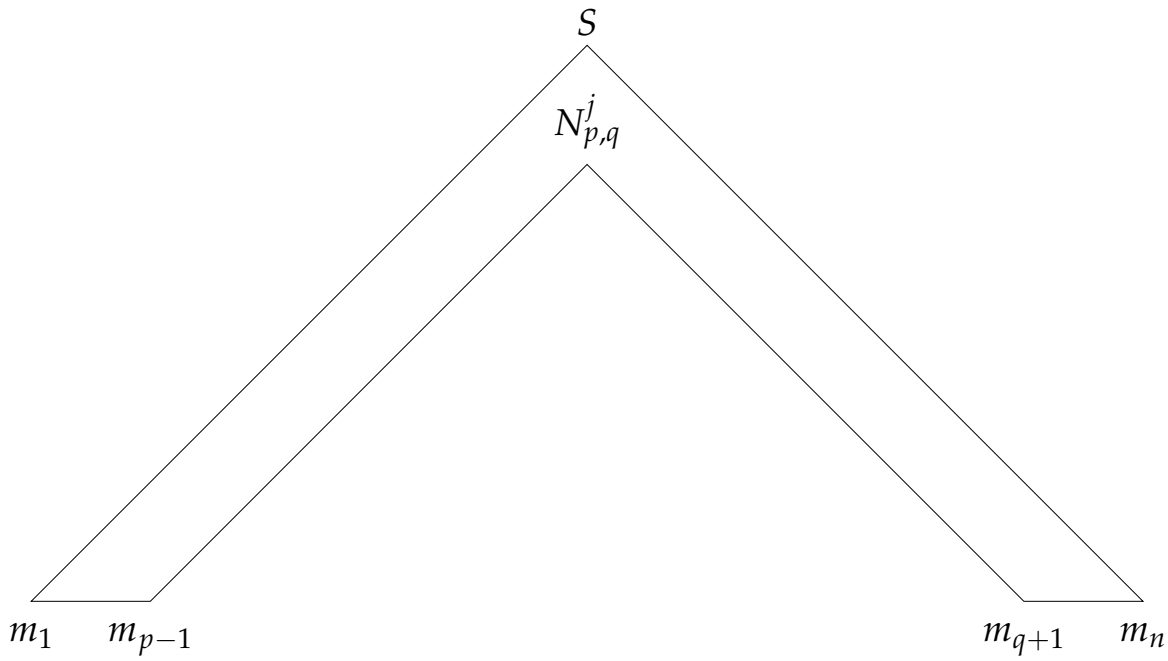
- $N^j_{p,q}$ correspond à la présence du symbole N^j dans la case $t_{p,q}$
- On peut calculer les $\beta(p,q)$ au fur et à mesure que l'on remplit la matrice d'analyse.

Calcul de $P(S)$ façon CYK

pour $q = 1$ à n faire { INITIALISATION }
  pour $p = q$ à 1 faire
    si $(p == q)$
      $\beta_j(p,p) = P(N^j \to m_p)$
    sinon
      $\beta_j(p,q) = 0$
pour $q = 1$ à n faire
  pour $p = q-1$ à 1 faire
    pour $d = p$ à $q-1$ faire
      $\beta_j(p,q) = \beta_j(p,q) + P(N^j \to N^r, N^s)\beta_r(p,d)\beta_s(d+1,q)$
      avec $N^r \in t_{p,d}$ et $N^s \in t_{d+1,q}$
$P(S) = \beta_1(1,n)$

Probabilité extérieure

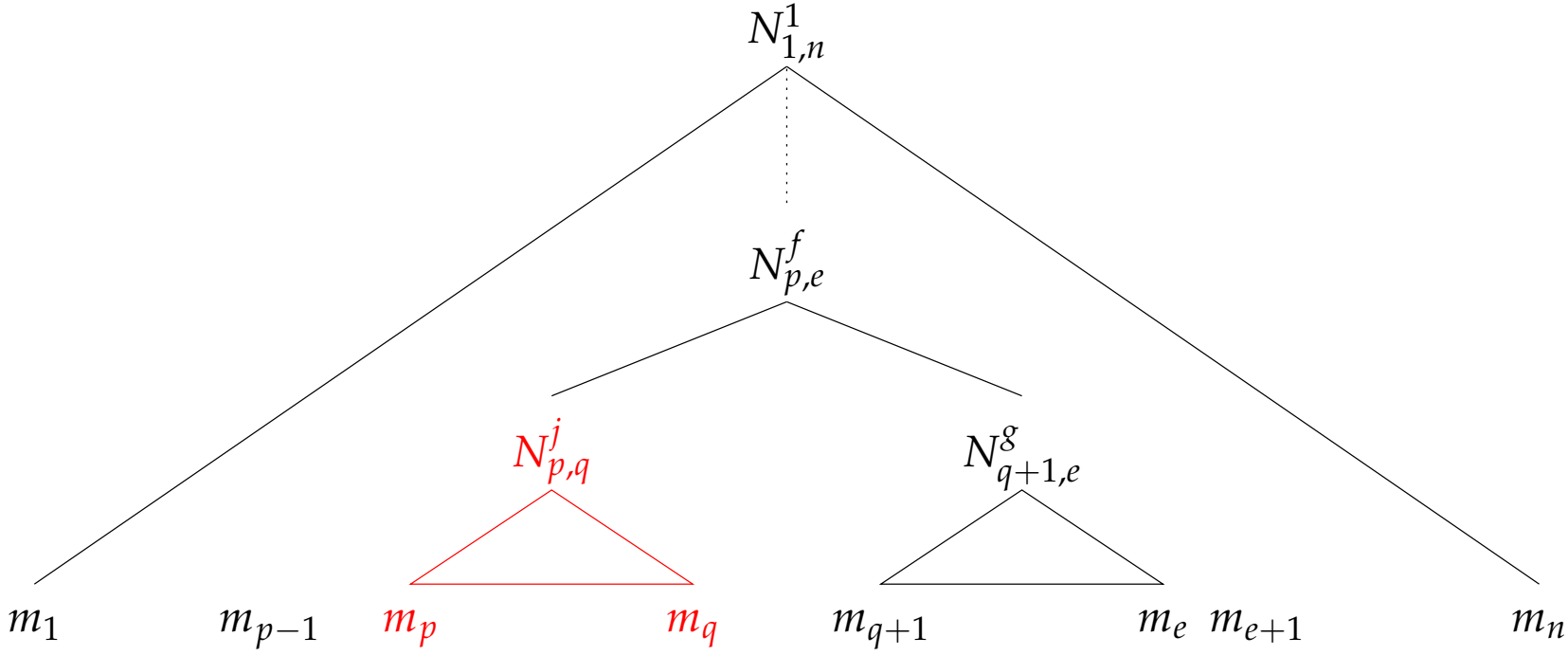
$$\alpha_j(p,q) \stackrel{def}{=} P(m_{1,p-1}, N^j_{p,q}, m_{q+1,m})$$



Calcul récursif des probabilités extérieures

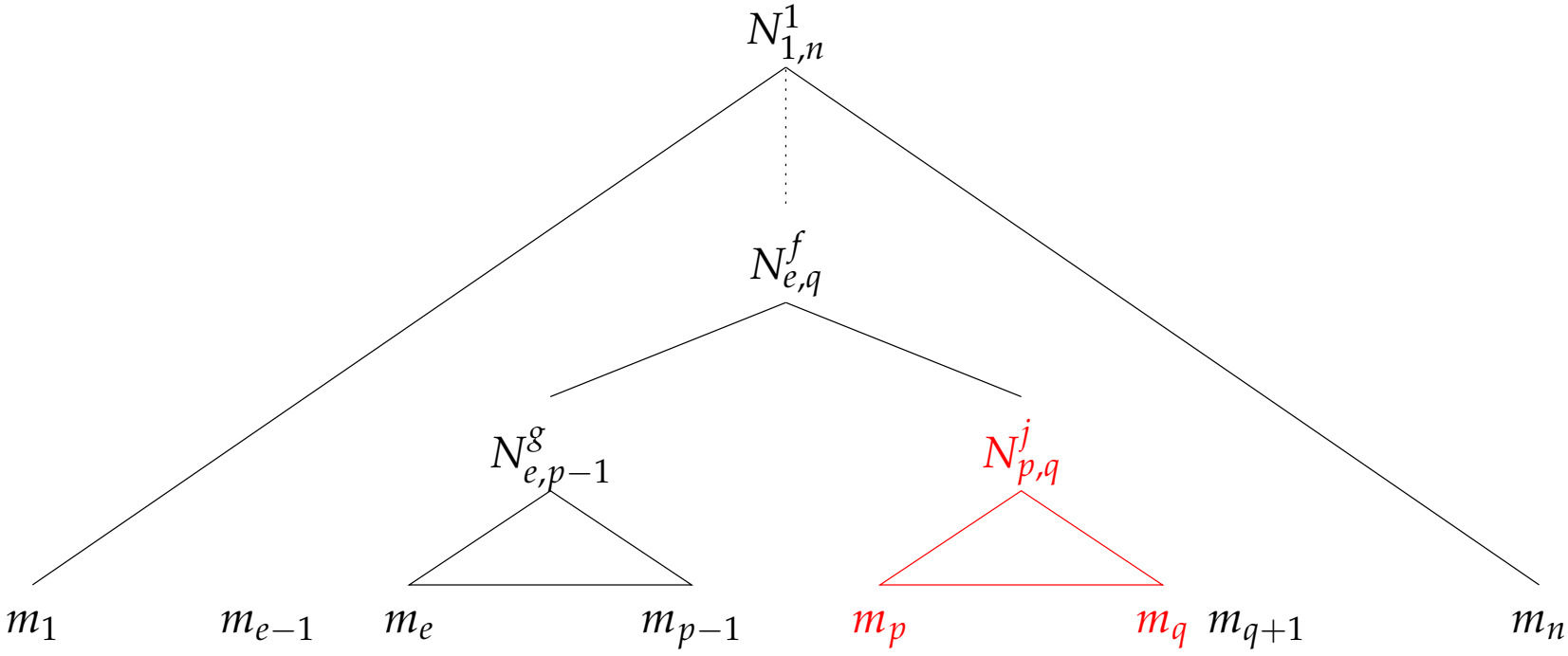
- Calcul descendant
- Le calcul des probabilités extérieures fait appel au calcul des probabilités intérieures, qui doivent avoir été préalablement calculées.

Calcul récursif des probabilités extérieures



$$\alpha_j^G(p,q) = \sum_{f,g} \sum_{e=q+1}^{n} P(m_{1,p-1}, m_{q+1,n}, N^f_{p,e}, N^j_{p,q}, N^g_{q+1,e})$$

Calcul récursif des probabilités extérieures



$$\alpha_j^D(p,q) = \sum_{f,g} \sum_{e=1}^{p-1} P(m_{1,p-1}, m_{q+1,n}, N^f_{e,q}, N^g_{e,p-1}, N^j_{p,q})$$

Calcul récursif des probabilités extérieures

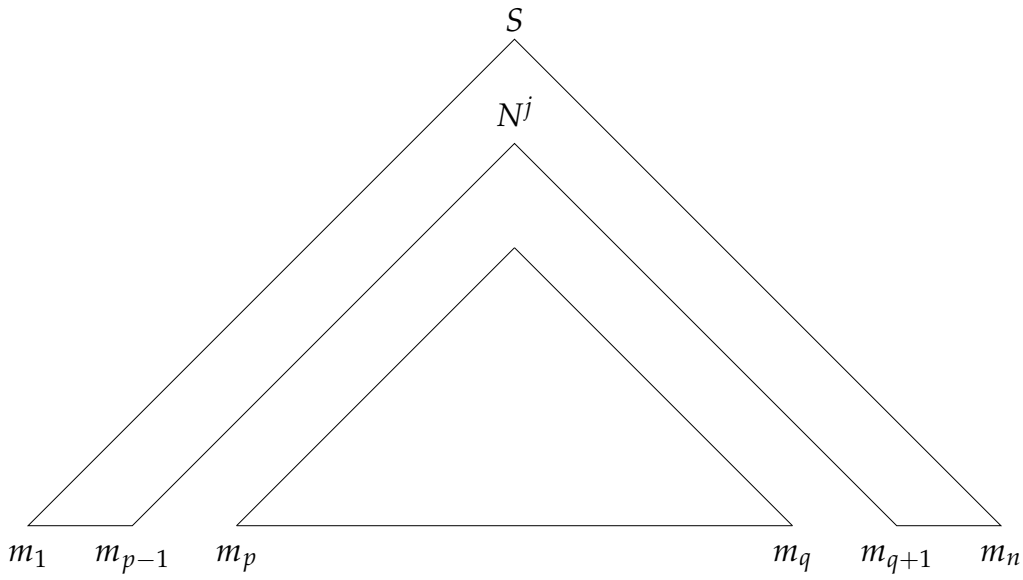
$$
\begin{aligned}
\alpha_j(p,q) &= \alpha_j^G(p,q) + \alpha_j^D(p,q) \\
&\dots \\
&= [\sum_{f,g \neq j} \sum_{e=q+1}^{n} \alpha_f(p,e) P(N^f \rightarrow N^j N^g) \beta_g(q+1,e)] \\
&\quad + [\sum_{f,g} \sum_{e=1}^{p-1} \alpha_f(e,q) P(N^f \rightarrow N^j N^g) \beta_g(e,p-1)]
\end{aligned}
$$

Calcul récursif des probabilités extérieures

Cas terminal :

$$\alpha_j(1,n) = \begin{cases} 1 & \text{si } j = 1 \\ 0 & \text{sinon} \end{cases}$$

Combinaison des probabilités extérieures et intérieures



$$\begin{aligned}
\alpha_j(p,q)\beta_j(p,q) &= P(m_{1,p-1}, N^j_{p,q}, m_{q+1,n})P(m_{p,q}|N^j_{p,q}) \\
&= P(m_{1,p-1}, N^j_{p,q}, m_{q+1,n})P(m_{p,q}|m_{1,p-1}, N^j_{p,q}, m_{q+1,n}) \\
&= P(m_{1,n}, N^j_{p,q})
\end{aligned}$$

Probabilité d'un syntagme

- Etant donné une phrase $m_{1,n}$, et une grammaire G, on peut calculer la probabilité que le segment $m_{p,q}$ constitue un syntagme de type N^j :

$$P(m_{1,n}, N^j_{p,q}) = \alpha_j(p,q)\beta_j(p,q)$$

- et la probabilité que le segment $m_{p,q}$ constitue un syntagme de type quelconque :

$$P(m_{1,n}, N_{p,q}) = \sum_j \alpha_j(p,q)\beta_j(p,q)$$

- On peut aussi calculer la probabilité d'occurrence d'un syntagme de type N^j dans la phrase :

$$P(m_{1,n}, N^j) = \sum_{1 \leq p \leq q \leq n} \alpha_j(p,q)\beta_j(p,q)$$

Calcul de la probabilité de $\hat{T}$

$\delta_i(p,q) =$ la probabilité du sous-arbre $N^j_{p,q}$ le plus probable.

1. Initialisation

$$\delta_i(p,p) = P(N^i \to m_p)$$

2. Récurrence

$$\delta_i(p,q) = \max_{1 \leq j,k \leq n,\ p \leq d < q} P(N^i \to N^j N^k)\delta_j(p,d)\delta_k(d+1,q)$$

3. Fin

$$P(\hat{T}) = \delta_1(1,n)$$

Calcul de $P(\hat{T})$

pour $q = 1$ à n faire { INITIALISATION }
  pour $p = q$ à 1 faire
    si $(p == q)$
      $\delta_j(p,p) = P(N^j \to m_p)$
    sinon
      $\delta_j(p,q) = 0$
pour $q = 1$ à n faire
  pour $p = q-1$ à 1 faire
    pour $d = p$ à $q-1$ faire
      $\delta_j(p,q) = \max(\delta_j(p,q), P(N^i \to N^j N^k)\delta_j(p,d)\delta_k(d+1,q))$
      avec $N^r \in t_{p,d}$ et $N^s \in t_{d+1,q}$
$P(\hat{T}) = \beta_1(1,n)$

Construction de $\hat{T}$

$\psi_i(p,q) = \langle j,k,r \rangle$ où j,k,r désignent l'application de la règle ayant réalisé le maximum $\delta_i(p,q)$

$$\psi_i(p,q) = \arg\max_{j,k,d} P(N^i \rightarrow N^j N^k)\delta_j(p,d)\delta_k(d+1,q)$$

- racine($\hat{T}$) = $N^1_{1,n}$
- si $\psi_i(p,q) = \langle j,k,r \rangle$ alors
  - fils gauche($N^i_{p,q}$) = $N^j_{p,r}$
  - fils droit($N^i_{p,q}$) = $N^k_{r+1,q}$

Estimation des probabilités de la grammaire

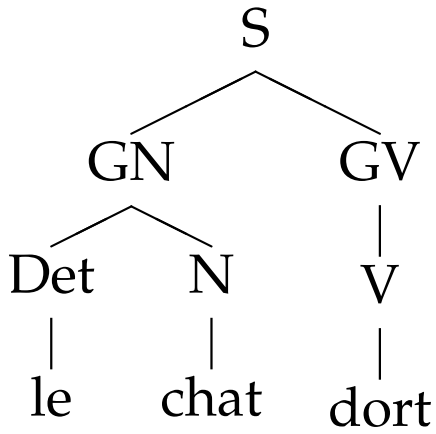
Deux cas :

- Données complètes : on dispose d'un ensemble de phrases et de leur analyse syntaxique (banque d'arbres).
- Données incomplètes : on ne dispose que d'un ensemble de phrases.

Exemples de banques d'arbres

corpus	Penn Treebank	corpus Paris 7
mots	1 000 000	400 000
phrases	45 000	15 000
cat. syntag.	26	13
parties de discours	36	14
règles	9 657	

Construction de la grammaire



$$
\begin{array}{lcl}
S & \rightarrow & GN\ GV \\
GN & \rightarrow & Det\ N \\
GV & \rightarrow & V \\
Det & \rightarrow & le \\
N & \rightarrow & chat \\
V & \rightarrow & dort
\end{array}
$$

Estimation des probabilités des règles

- On compte le nombre d'occurrences des symboles non terminaux A dans la banque d'arbres : $C(A)$
- On compte le nombre d'occurrences des règles $A \rightarrow \alpha$ dans la banque d'arbres : $C(A \rightarrow \alpha)$
- On estime $P(A \rightarrow \alpha)$ par maximum de vraisemblance :

$$P(A \rightarrow \alpha) = \frac{C(A \rightarrow \alpha)}{C(A)}$$

Estimation à partir de données incomplètes

- On fixe la partie algébrique de la grammaire (alphabets, règles)
- On recherche la distribution de probabilités de la grammaire qui maximise la probabilité des données d'apprentissage (des phrases dont on dispose)
- On ne sait calculer ces distributions directement, on fait appel à un algorithme itératif du type Expectation Maximization appelé algorithme intérieur-extérieur (inside-outside).

Principe de l'algorithme intérieur extérieur

- On fixe des distributions de probabilités initiales
- On calcule la probabilité des différents symboles et règles étant donné une phrase S
- On estime le nombre d'occurrences des différents symboles et règles lors des dérivations de S
  - $C(N^j)$
  - $C(N^j \rightarrow N^r N^s)$
  - $C(N^j \rightarrow m^k)$
- On calcule de nouvelles probabilités pour les règles
  - $\hat{P}(N^j \rightarrow N^r N^s)$
  - $\hat{P}(N^j \rightarrow m^k)$
- On calcule $P(S)$
- On itère tant que $P(S)$ augmente

Occurrences estimées

- Symbole

$$C(N^j) = \sum_{p=1}^{n} \sum_{q=p}^{n} P(N^j_{p,q}) = \sum_{p=1}^{n} \sum_{q=p}^{n} \alpha_j(p,q)\beta_j(p,q)$$

- Règles binaires

$$C(N^j \rightarrow N^r N^s) =$$

$$\sum_{p=1}^{n-1} \sum_{q=p+1}^{n} \sum_{d=p}^{q-1} \alpha_j(p,q) P(N^j \rightarrow N^r N^s) \beta_r(p,d) \beta_s(d+1,q)$$

- Règles unaires

$$C(N^j \rightarrow m^k) = \sum_{h=1}^{n} \alpha_j(h,h) P(N^j \rightarrow m_h) P(m_h = m^k) \beta_j(h,h)$$

Nouvelles probabilités

- $\hat{P}(N^j \rightarrow N^r N^s) = \frac{C(N^j \rightarrow N^r N^s)}{C(N^j)}$
- $\hat{P}(N^j \rightarrow m^k) = \frac{C(N^j \rightarrow m^k)}{C(N^j)}$

Nouvelles probabilités (2)

$$\hat{P}(N^j \rightarrow m^k) = \frac{\sum_{h=1}^{n} \alpha_j(h,h) P(N^j \rightarrow m_h), P(m_h = m^k) \beta_j(h,h)}{\sum_{p=1}^{n} \sum_{q=p}^{n} \alpha_j(p,q) \beta_j(p,q)}$$

$$\hat{P}(N^j \rightarrow N^r N^s) =$$

$$\frac{\sum_{p=1}^{n-1} \sum_{q=p+1}^{n} \sum_{d=p}^{q-1} \alpha_j(p,q) P(N^j \rightarrow N^r, N^s) \beta_r(p,d) \beta_s(d+1,q)}{\sum_{p=1}^{n} \sum_{q=p}^{n} \alpha_j(p,q) \beta_j(p,q)}$$

Problèmes de l'algorithme extérieur intérieur

- Efficacité : pour chaque phrase du corpus d'apprentissage et chaque itération, la complexité est $O(n^3V^3)$ où n est la longueur de la phrase et V le nombre de non terminaux de la grammaire.
- Maximum local : l'algorithme est très sensible aux distributions de probabilités initiales. Des distributions différentes aboutissent à des maxima différents.
- Choix du nombre de non terminaux : on ne sait pas combien de non terminaux choisir, les expériences ont montré qu'un nombre important de non terminaux améliore les résultats, ce qui augmente le problème d'efficacité.

Limites des PCFG (1)

- Indépendance lexicale
  - La réecriture d'un symbole pré-terminal X ne dépend pas du contexte d'occurrence de X
  - mise en défaut : la préposition introduisant un complément d'un verbe dépend de la nature lexicale du verbe
    Exemple : *Jean* **pense à** *Marie*
  - cette dépendance n'est pas modélisée :
    $GV \rightarrow V\ GP$
    $V \rightarrow pense$
    $GP \rightarrow P\ GN$
    $P \rightarrow à$

Limites des PCFG (2)

- Indépendance structurale
  - le choix d'une règle pour la réécriture d'un symbole X est indépendant du contexte d'occurrence de X
  - mise en défaut : Dans le corpus switchboard, 91% des sujets sont des pronoms alors que seule 34% des objets le sont (Francis et al 99)
  - Or il n'y a qu'une règle de la forme $GN \rightarrow Pro$
  - et une seule probabilité lui correspondant