

MDI 341

Introduction aux modèles graphiques - Partie II: Modèles de Markov Cachés

Mars 2017

Laurence Likforman-Sulem
Telecom ParisTech/IDS
likforman@telecom-paristech.fr



Plan

■ Modèles de Markov Cachés

- discrets/continus
- modèle génératif
- décodage Viterbi, Baum-Welch
- apprentissage

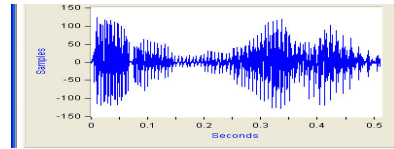
HMMs (Hidden Markov Models)

- analyse de séquences

- parole, écriture, texte

- applications

- reconnaissance de la parole
- reconnaissance de l'écriture
- reconnaissance d'objets, de visages dans les videos,...
- Natural Language Processing (NLP): étiquetage grammatical
 - tagging
 - corrections orthographiques



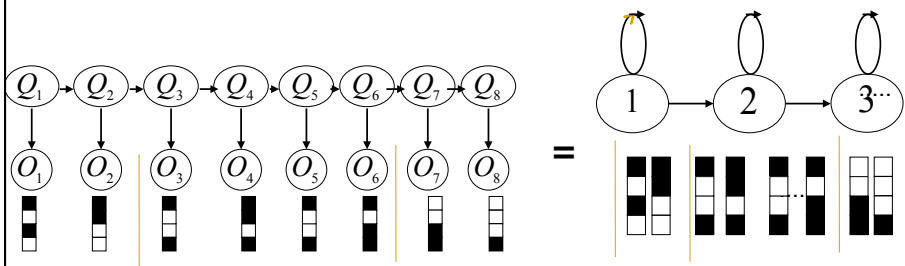
THE → TGE



Laurence Likforman-Telecom ParisTech

3

modèle de Markov caché : cas particulier de DBN



- HMM: Hidden Markov Model
- structure d'arbre
- 1 variable d'état + 1 variable d'observation à chaque instant t

$(Q_t)_{1 \leq t \leq T}$: variables d'état (cachées)

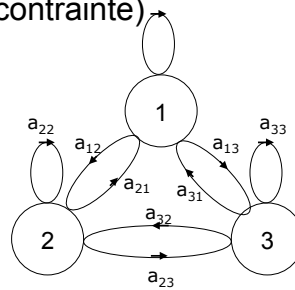
$(O_t)_{1 \leq t \leq T}$: variables d'observation (observées) générées par les états

4

topologie du modèle: ergodique / gauche droite

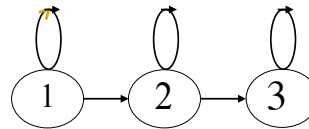
- modèle ergodique (sans contrainte)

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



- modèle gauche droite (contrainte: transitions $i \rightarrow j \geq i$)

$$A = \begin{bmatrix} 0.4 & 0.6 & 0 \\ 0 & 0.8 & 0.2 \\ 0 & 0 & 1 \end{bmatrix}$$



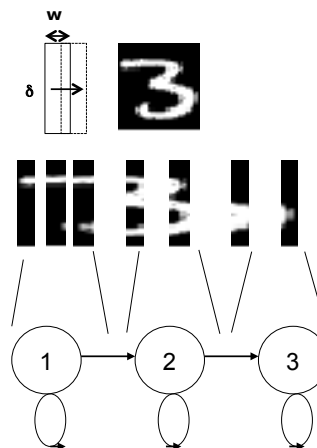
5

Modèles de Markov Cachés

- une classe de forme
 - modèle λ
- combinaison de 2 processus stochastiques
 - un observé
 - un caché
- on n'observe pas la séquence d'états

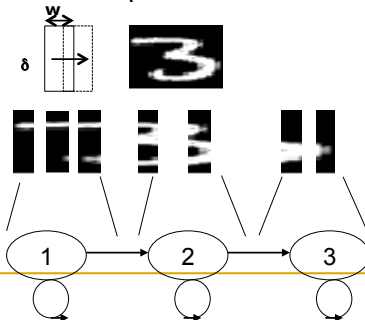
$q = q_1 q_2 \dots q_T$
- on observe la séquence d'observations

$o = o_1 o_2 \dots o_T$
- les observations sont générées (émises) par les états



Processus stochastiques

- variables d'états indexées par temps discret
 - q_t prend ses valeurs dans $\{1, 2, \dots, Q\}$ (nombre fini d'états)
 - évolution: probabilités de transition (chaîne Markov stationnaire ordre 1)
- variables d'observations
 - discrètes ou continues
 - évolution: émission par les états



Laurence Likforman
Telecom ParisTech

7

HMMs discrets

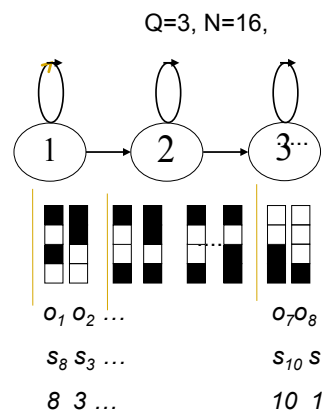
- ensemble de Q états discrets $\{1, 2, \dots, Q\}$
- ensemble de N symboles discrets
 - $\{s_1, s_2, s_3, \dots, s_N\} \rightarrow \{1, 2, 3, \dots, N\}$

- on observe $o = o_1 o_2 o_t \dots o_T$

- $o = s_8 s_3 s_{13} s_6 s_8 s_5 s_{10} s_1$
- $o = 8 \ 3 \ 13 \ 6 \ 8 \ 5 \ 10 \ 1$

- o correspond à séquence d'états (cachés)

- $q = q_1 q_2 q_t \dots q_T$
- $q = 1 \ 1 \ 2 \ 2 \ 2 \ 2 \ 3 \ 3$



8

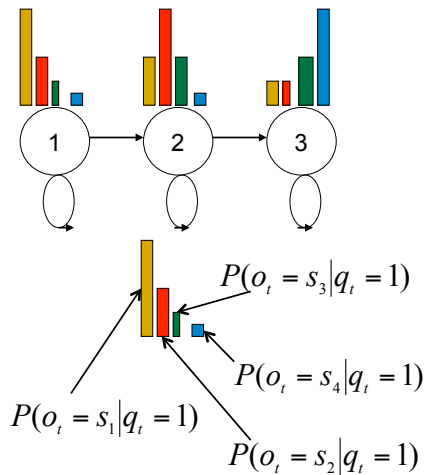
HMMs discrets

- HMM λ discret est défini par
 - π vecteur probabilités initiales
 - A : matrice transition
 - B : matrice des probabilités d'observation des symboles (dans les états)

$$\pi = (\pi_1, \pi_2, \dots, \pi_Q) \quad \pi_i = P(q_1 = i)$$

$$A = \{a_{ij}\} = P(q_t = j | q_{t-1} = i)$$

$$B = \{b_{ki}\} = P(o_t = s_k | q_t = i)$$

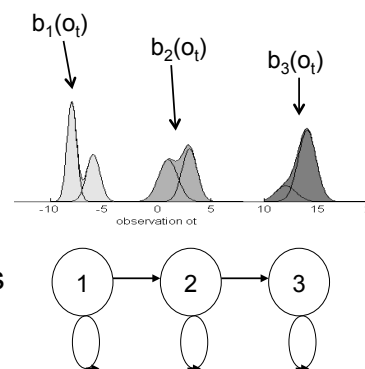


Laurence Likforman-Telecom
ParisTech

9

modèles de Markov cachés continus

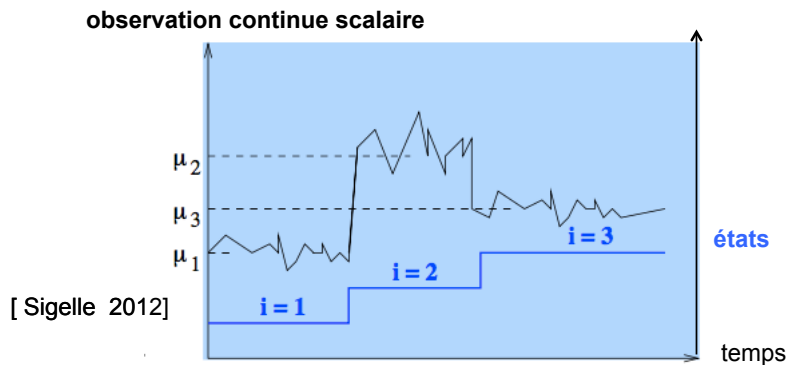
- HMM λ continu défini par :
- π vecteur de probabilités initiales
- A : matrice de transition entre états
- $b_i(o_t)$: densité de probabilité des observations dans état i , $i=1, \dots, Q$
→ gaussienne ou mélange gaussiennes



L. Likforman - Telecom ParisTech

10

modèle d'observations Gaussien



$$P(o_t / q_t = i, \lambda) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp -\frac{(o_t - \mu_i)^2}{2\sigma_i^2}$$

modèle: inclut μ_i et σ_i , $i=1,2,3$

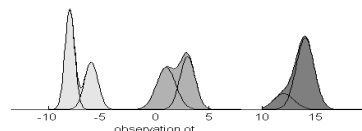
Laurence Likforman-Telecom ParisTech

11

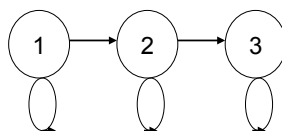
mélange de gaussiennes

$$b_i(o_t) = \sum_{k=1}^M c_{ik} \mathcal{N}(o_t; \Sigma_{ik}, \mu_{ik}) \quad \forall i = 1, \dots, Q.$$

observations continues (scalaires ou vectorielles)



c_{ik} : poids de la k ème loi gaussienne du mélange de M gaussiennes, associée à l'état i



modèle λ : inclut c_{ik} , μ_{ik} et Σ_{ik} , $i=1,2,3$ et $k=1,\dots,M$

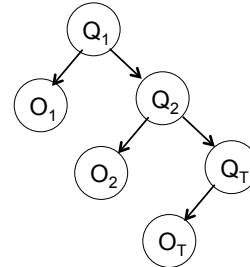
L. Likforman - Telecom ParisTech

12

hypothèses fondamentales

- indépendance des observations conditionnellement aux états

$$P(o_1, \dots, o_T | q_1, \dots, q_T, \lambda) = \prod_{t=1}^T P(o_t | q_t, \lambda)$$



- chaîne de Markov stationnaire (transitions entre états)

$$P(q_1, q_2, \dots, q_T) = P(q_1)P(q_2 | q_1)P(q_3 | q_2) \dots P(q_T | q_{T-1})$$

hypothèses fondamentales

- probabilité jointe pour une séquence d'observations et un chemin d'états

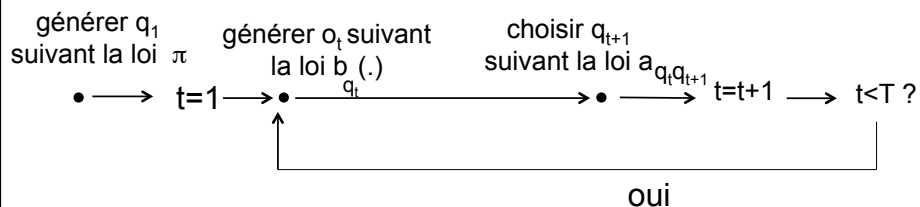
$$\begin{aligned} P(o_1, \dots, o_T, q_1, \dots, q_T | \lambda) &= \pi_{q_1} b_{q_1}(o_1) \prod_{t=2}^T a_{q_{t-1}, q_t} P(o_t | q_t, \lambda) \\ &= \pi_{q_1} b_{q_1}(o_1) \prod_{t=2}^T a_{q_{t-1}, q_t} b_{q_t}(o_t) \\ &= P(o_1, \dots, o_T | q_1, \dots, q_T, \lambda) P(q_1, \dots, q_T) \end{aligned}$$

MODELE GENERATIF

15

générer une séquence d'observations

- générer la séquence d'états q_1, \dots, q_T , puis générer la séquence observations à partir de chaque état
- ou générer q_1 puis o_1 ($q_1 \rightarrow o_1$); générer q_2 à partir de q_1 ($q_1 \rightarrow q_2$), puis o_2 ($q_2 \rightarrow o_2$), etc...

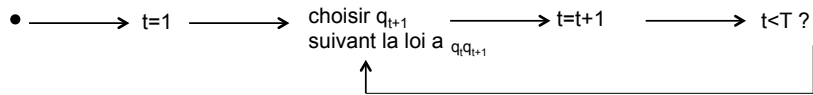


L. Likforman - Telecom ParisTech

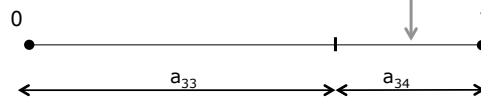
16

étape 1 : générer une séquence d'états

générer q_1
suivant la loi π



ex: $q_6=3$



$a_{31}=0; a_{32}=0;$

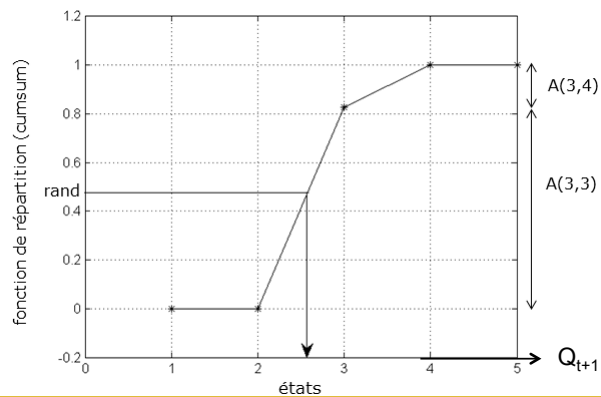
$\rightarrow q_7=4$

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

$$A = \begin{bmatrix} 0.1 & 0.9 & 0 & 0 \\ 0 & 0.15 & 0.85 & 0 \\ 0 & 0 & 0.8 & 0.2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

générer une séquence d'états: simulation Monte Carlo

$$A = \begin{bmatrix} 0.1 & 0.9 & 0 & 0 \\ 0 & 0.15 & 0.85 & 0 \\ 0 & 0 & 0.8 & 0.2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



$$F_{Q_{t+1}|q_t=i}(j) = P(Q_{t+1} \leq j | q_t = i, \lambda)$$

18

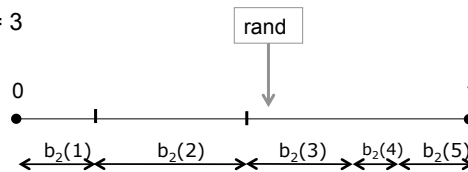
étape 2 : générer les observations (discrètes)

■ séquence états

- $q_1 = 1; q_2 = 1; q_3 = 1; q_4 = 2; q_5 = 2; q_6 = 3; \dots$

■ générer l'observation à $t=4$

- $q_4 = 2;$
- $\rightarrow o_4 = 3$



générer une séquence d'états: mini-TD

- générer séquence d'états de longueur T
- on donne
 - $\pi = [0.35 \ 0.65]$ $A = \begin{bmatrix} 0.35 & 0.65 \\ 0.2 & 0.8 \end{bmatrix}$

□ *on tire les nombres aléatoires suivants:*

- $u_1 = 0.92$ (q_1) $u_2 = 0.31$
- $u_3 = 0.1$ $u_4 = 0.4$
- $u_5 = 0.01$

DÉCODAGE

21

HMM pour la reconnaissance des formes

- chaque classe m est modélisée par un modèle HMM λ_m
- pour une séquence d'observations $o=o_1, \dots, o_T$ extraite d'une forme, calcul de la vraisemblance:

$$P(o_1, \dots, o_T | \lambda_m)$$

- attribution de la forme à la classe \hat{m} telle que:

$$\hat{m} = \arg \max_m P(o_1, \dots, o_T | \lambda_m)$$

HMM pour étiquetage morpho-syntaxique

- observations: suite de mots
 - « Bob mange la pomme »
- états cachés (à trouver par Viterbi)
 - 'nom propre' 'verbe' déterminant' 'nom'
- modèle
 - probabilités de transitions entre éléments grammaticaux, bi-grams (tags)
 - probabilités d'observer les mots pour un élément grammatical donné (tag)
 - $P(\text{« le »} | \text{verbe})$, $P(\text{« le »} | \text{pronom})$ etc....

Laurence Likforman-Telecom
ParisTech

23

algorithme de décodage de Viterbi

- calcul de la vraisemblance
- séquence observation $o=o_1, \dots, o_T$

$$P(o | \lambda) = \sum_q P(o, q | \lambda)$$

- au lieu de sommer sur toutes les séquences d'états, recherche de la séquence optimale :

$$\hat{q} = \arg \max_q P(q, o | \lambda)$$

- puis estimer la vraisemblance par :

$$P(o | \lambda) \approx P(o, \hat{q} | \lambda)$$

Laurence Likforman-Telecom
ParisTech

24

algorithme de Viterbi (décodage)

- $\delta_t(i)$: proba. (jointe) meilleure séquence partielle d'états aboutissant à l'état i au temps t et correspondant à la séquence partielle d'observations $o_1 \dots o_t$.

$$\delta_t(i) = \max_{q_1 q_2 \dots q_{t-1}} P(q_1 q_2 \dots q_t = i, o_1 o_2 \dots o_t | \lambda)$$

- récurrence

$$\begin{aligned} & P(q_1 q_2 \dots q_t = i, q_{t+1} = j, o_1 o_2 \dots o_t o_{t+1} | \lambda) \\ &= P(o_{t+1}, q_{t+1} = j | o_1 \dots o_t, q_1 \dots q_{t-1} q_t = i, \lambda) P(o_1 \dots o_t, q_1 \dots q_{t-1} q_t = i | \lambda) \\ &= P(o_{t+1} | q_{t+1} = j, \lambda) P(q_{t+1} = j | q_t = i, \lambda) P(o_1 \dots o_t, q_1 \dots q_{t-1} q_t = i | \lambda) \\ & \max_{q_1 q_2 \dots q_t} P(q_1 q_2 \dots q_t = i, q_{t+1} = j, o_1 o_2 \dots o_t o_{t+1} | \lambda) = \max_i b_j(o_{t+1}) a_{ij} \delta_t(i) \end{aligned}$$

$$\delta_{t+1}(j) = \max_i b_j(o_{t+1}) a_{ij} \delta_t(i) = b_j(o_{t+1}) \max_i a_{ij} \delta_t(i)$$

$$P(o, \hat{q}) = \max_j \delta_T(j)$$

Laurence Likforman-Telecom ParisTech

25

algorithme de décodage de Viterbi

- 1ere colonne: Initialisation

$$\delta_1(i) = P(q_1 = i, o_1) = b_i(o_1) \pi_i \quad i = 1, \dots, Q$$

- colonnes 2 à T : récursion

$$\delta_{t+1}(j) = b_j(o_{t+1}) \max_i a_{ij} \delta_t(i) \quad t = 1, \dots, T-1, j = 1, \dots, Q$$

$$\varphi_{t+1}(j) = \arg \max_i a_{ij} \delta_t(i) \quad \text{sauvegarde meilleur chemin (état précédent)}$$

- terminaison

$$P(o, \hat{q}) = \max_j \delta_T(j)$$

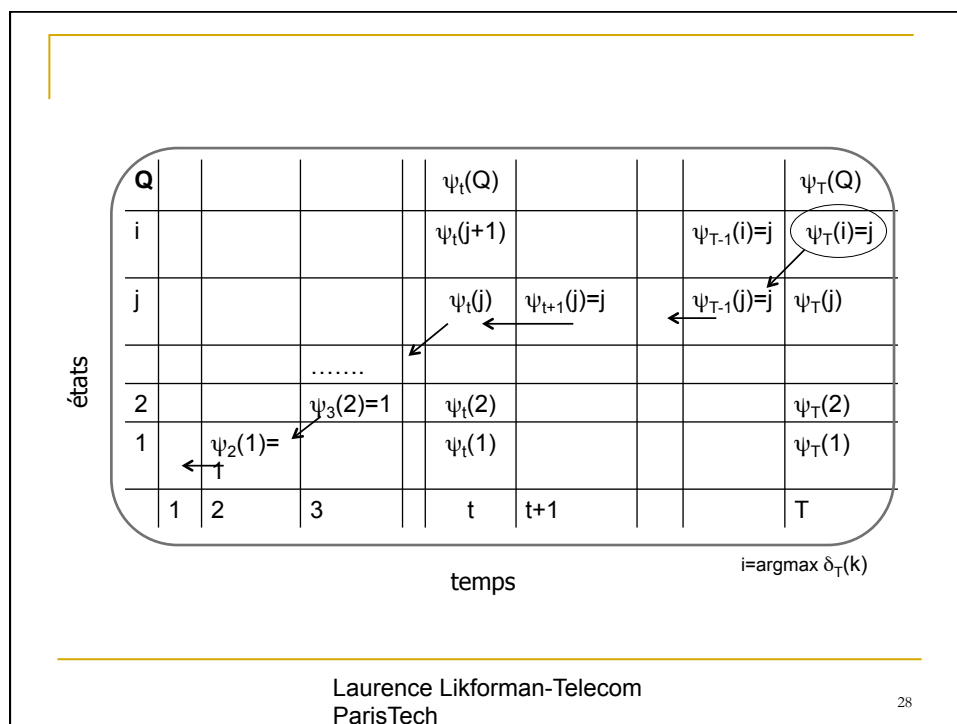
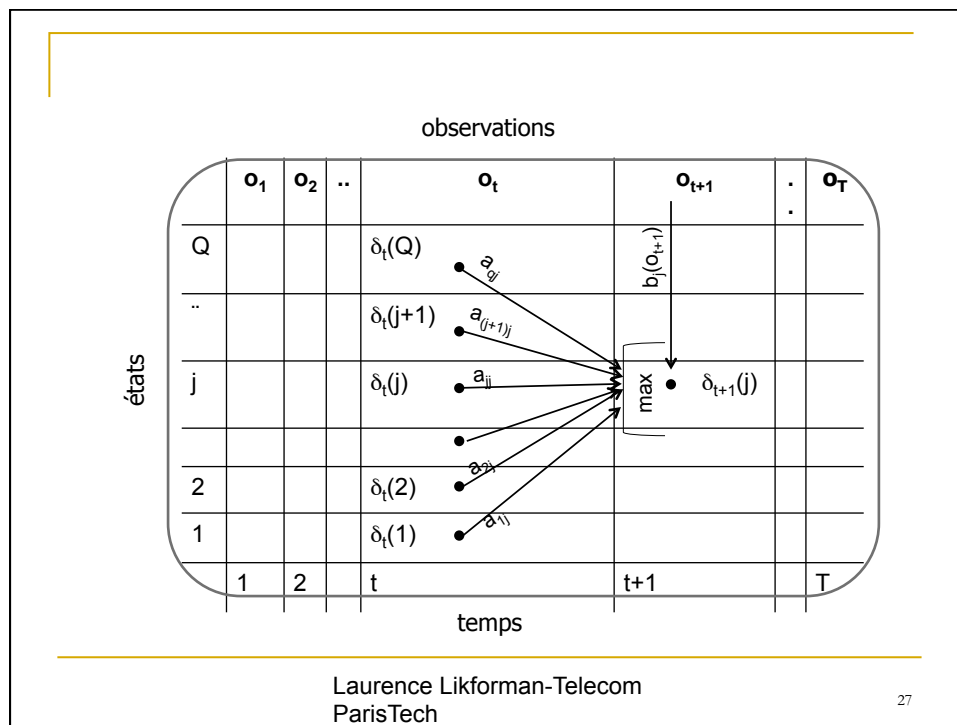
$$\hat{q}_T = \arg \max_j \delta_T(j)$$

- backtrack

$$\hat{q}_t = \varphi(\hat{q}_{t+1}) \quad t = T-1, T-2, \dots, 1$$

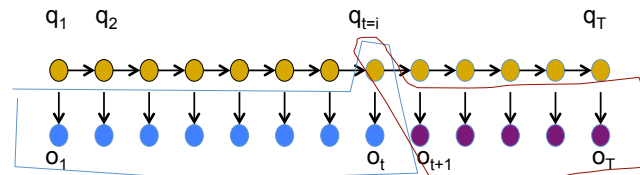
Laurence Likforman-Telecom
ParisTech

26



variables forward-backward

$$\begin{aligned}
 P(o|\lambda) &= \sum_i P(o, q_t = i|\lambda) \\
 P(o, q_t = i|\lambda) &= P(o_1 \dots o_t, q_t = i, o_{t+1} \dots o_T|\lambda) \\
 &= P(o_{t+1} \dots o_T | o_1 \dots o_t, q_t = i, \lambda) P(o_1 \dots o_t, q_t = i|\lambda) \\
 &= \underbrace{P(o_{t+1} \dots o_T | q_t = i, \lambda)}_{\beta_t(i)} \underbrace{P(o_1 \dots o_t, q_t = i|\lambda)}_{\alpha_t(i)} \\
 &= \beta_t(i) \alpha_t(i) \quad \beta_t(i) : \text{variable backward (analogue à } \lambda) \\
 &\quad \alpha_t(i) : \text{variable forward (analogue à } \pi)
 \end{aligned}$$



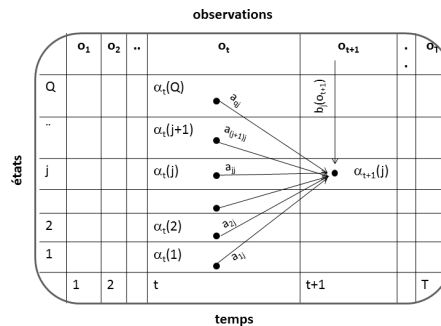
Laurence Likforman-Telecom ParisTech

29

algorithme de décodage forward-backward

- calcul exact de la vraisemblance $P(o|\text{modele})$: Baum-Welch
- basé sur les variables forward et/ou backward

$$\begin{aligned}
 \alpha_1(j) &= b_j(o_1) \pi_j \\
 \alpha_{t+1}(j) &= b_j(o_{t+1}) \sum_{i=1}^Q \alpha_t(i) a_{ij} \\
 P(o|\lambda) &= \sum_{j=1}^Q \alpha_T(j)
 \end{aligned}$$



Laurence Likforman-Telecom
ParisTech

30

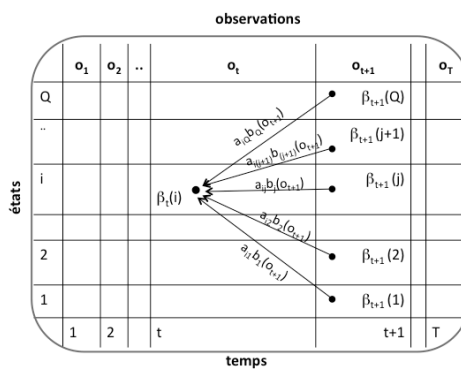
algorithme de décodage forward-backward

- calcul exact de la vraisemblance $P(o|\text{modele})$: Baum-Welch
- basé sur les variables forward et/ou backward

$$\beta_T(i) = 1$$

$$\beta_t(i) = \sum_{j=1}^Q a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)$$

$$P(O|\lambda) = \sum_{j=1}^Q \beta_1(j) \pi_j b_j(o_1)$$



Laurence Likforman-Telecom
ParisTech

31

ESTIMATION DE PARAMETRES

32

Apprentissage en données complètes

- pour chaque modèle λ , estimer les paramètres
- base d'apprentissage
 - L séquences d'observation $o^{(l)}$, $l=1 \dots L$
 - + séquences d'états associées
- séquence $o=o_1 \dots o_T$ associée à séquence d'états $q=q_1 \dots q_T$

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} 1_{\{q_t=i, q_{t+1}=j\}}}{\sum_{t=1}^{T-1} 1_{\{q_t=i\}}} \quad \hat{b}_i(s_k) = \frac{\sum_{t=1}^T 1_{\{o_t=s_k, q_t=i\}}}{\sum_{t=1}^T 1_{\{q_t=i\}}}$$

33

Apprentissage en données complètes

- sur la base d'apprentissage totale

$$\hat{a}_{ij} = \frac{\sum_{l=1}^L \sum_{t=1}^{T^{(l)}-1} 1_{\{q_t^{(l)}=i, q_{t+1}^{(l)}=j\}}}{\sum_{l=1}^L \sum_{t=1}^{T^{(l)}-1} 1_{\{q_t^{(l)}=i\}}}$$

$$\hat{b}_i(s_k) = \frac{\sum_{l=1}^L \sum_{t=1}^{T^{(l)}} 1_{\{o_t^{(l)}=s_k, q_t^{(l)}=i\}}}{\sum_{l=1}^L \sum_{t=1}^{T^{(l)}} 1_{\{q_t^{(l)}=i\}}}$$

Laurence Likforman-Telecom ParisTech

34

apprentissage données complètes

HMM continu, gaussienne mono-variee,

$$\hat{\mu}_i = \frac{\sum_{l=1}^L \sum_{t=1}^{T^{(l)}} o_t^{(l)} \mathbb{1}_{q_t^{(l)}=i}}{\sum_{l=1}^L \sum_{t=1}^{T^{(l)}} \mathbb{1}_{q_t^{(l)}=i}}$$
$$\widehat{(\sigma_i)^2} = \frac{\sum_{l=1}^L \sum_{t=1}^{T^{(l)}} (o_t^{(l)} - \hat{\mu}_i)^2 \mathbb{1}_{q_t^{(l)}=i}}{\sum_{l=1}^L \sum_{t=1}^{T^{(l)}} \mathbb{1}_{q_t^{(l)}=i}}$$

35

Apprentissage en données incomplètes

- estimer les paramètres, modèle λ
- on a une base d'apprentissage
 - L séquences d'observation $o^{(l)}$, $l=1 \dots L$
- pas connaissance des états cachés
 - plus difficile
- algorithme apprentissage
 - Baum-Welch
 - Viterbi

apprentissage Viterbi

- apprentissage en données incomplètes
- décodage par Viterbi
 - séquence états optimale
 - on se ramène au cas « données complètes »

conclusion

- modèles de Markov Cachés
 - apprentissage cas discret et données complètes
 - décodage de Viterbi
 - lien entre réseaux bayésiens dynamiques et HMMs
- données incomplètes
 - algorithme EM (Viterbi, Baum-Welch)

références

- M. Sigelle, Bases de la Reconnaissance des Formes: Chaînes de Markov et Modèles de Markov Cachés, chapitre 7, Polycopié Telecom ParisTech, 2012.
- L. Likforman-Sulem, E. Barney Smith, Reconnaissance des Formes: théorie et pratique sous matlab, Ellipses, TechnoSup, 2013.
- L. Rabiner, A tutorial on Hidden Markov Models and selected applications in Speech Recognition, proc. of the IEEE, 1989.