CES Data Science

Modèles de Markov Cachés

Mai 2016

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

Plan

- Chaînes de Markov
  - modèles stochastiques
  - paramètres
- Modèles de Markov Cachés
  - discrets/continus
  - apprentissage
  - décodage

applications

- HMMs
  - reconnaissance de la parole
  - reconnaissance de l'écriture
  - reconnaissance d'objects, de visages dans les videos,…
  - Natural Language Processing (NLP): étiquetage morpho-syntaxique

PARTIE I: MODÈLES DE MARKOV

Modèle stochastique

- processus aléatoire à temps discret
  - suite de variables aléatoires q1, q2, …., qT
  - instants entiers t=1, 2, ….T
- q_t: variable aléatoire d'état observé au temps t
  - notée q(t) ou q_t
  - q(t) prend ses valeurs dans {1,2, ….Q} (nombre fini d'états)
  - → $P(q_t=i)$ probabilité d'observer l'état i au temps t

Modèle stochastique

- évolution du processus
  - état initial q1
  - transitions entre états
    - q1→ q2…→ qt t<= T
  - modèle: connaître la probabilité de chaque transition
  - calcul d'une séquence (observée) d'états

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

$= P(q_T | q_1, q_2, \dots q_{T-1}) P(q_{T-1} | q_1, q_2, \dots q_{T-2}) P(q_1, q_2, \dots q_{T-2})$

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

Chaîne de Markov à temps discret

- processus stochastique
  - séquence de variables aléatoires appelées états
  - espace d'états fini et discret
- propriété de Markov d'ordre k
  - $P(q_t | q_1, q_2, \ldots q_{t-1}) = P(q_t | q_{t-k} \ldots q_{t-1})$
  - k=1 ou 2 en pratique
- *cas k=1*
  - $P(q_t | q_1, q_2, \ldots q_{t-1}) = P(q_t | q_{t-1})$
  - $P(q_1, q_2, \ldots q_T) = P(q_1)P(q_2 / q_1)P(q_3 / q_2) \ldots\ldots P(q_T | q_{T-1})$
  - → probabilités de transition entre états

Chaîne de Markov stationnaire

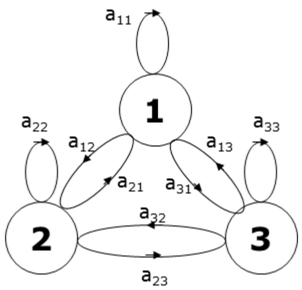
- probabilités de transition ne dépendent pas du temps
  - $P(q_t = j | q_{t-1} = i) = P(q_{t+k} = j | q_{t+k-1} = i) = a_{ij}$
  - a_{ij}= probabilité de passer de l'état i à l'état *j*
- définition: modèle d'une chaîne de Markov stationnaire
- matrice des probabilités de transitions
  - $A=[a_{ij}]$ $i=1,\ldots Q, j=1,\ldots. Q$
- vecteur des probabilités initiales
  - $\Pi = [\pi_i]$ $i=1,\ldots Q$
  - $\pi_i = P(q_1 = i)$
- contraintes : $0 <= \pi_i <= 1$ $0 <= a_{ij} <= 1$

$$\sum_{i=1}^{Q} \pi_i = 1 \qquad \sum_{j=1}^{Q} a_{ij} = 1$$

topologie du modèle: ergodique / gauche droite

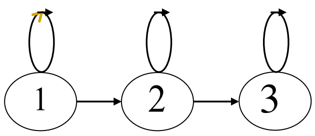
- modèle ergodique

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



- modèle gauche droite

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

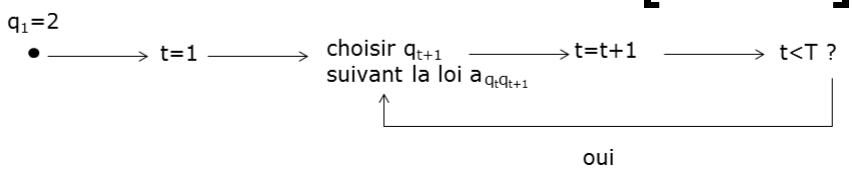


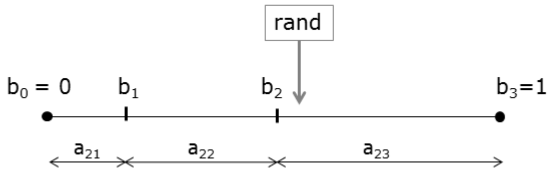
générer une séquence d'états

- on part de l'état q_1= 2
- générer séquence d'états de longueur T

suivant chaîne de Markov (matrice A)

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



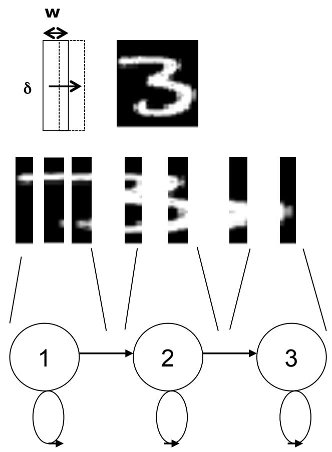


$$F_{Q_{t+1}|q_t=i}(j)=P(Q_{t+1}\le j\,|\,q_t=i,\lambda)$$

PARTIE II: MODÈLES DE MARKOV CACHÉS

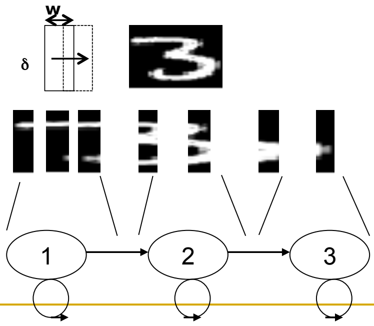
Modèles de Markov Cachés

- une classe de forme
  - modèle λ
- combinaison de 2 processus stochastiques
  - un observé
  - un caché
- on n'observe pas la séquence d'états $q = q_1\, q_2 \ldots q_T$
- on observe la séquence d'observations
  $o = o_1\, o_2 \ldots o_T$
- les observations sont générées (émises) par les états



Processus stochastiques

- variables d'états
  - notées $q_1, q_2, \ldots, q_T$ (q_t ou $q(t)$)
  - instants entiers t=1, 2, ….T
  - q_t prend ses valeurs dans {1,2, ….Q} (nombre fini d'états)
- variables d'observations
  - discrètes ou continues

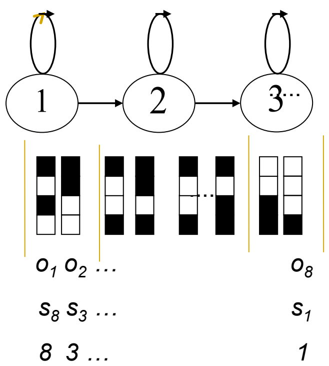


HMMs discrets

- soit un ensemble de Q états discrets {1, 2 … Q}
- un ensemble de N symboles discrets
  {s1, s2,…sN} → {1, 2 … N}
- on observe la séquence
  $o = o_1 o_2, , o_t \; \ldots . o_T$
- $o_t \in \{s1, \ldots sN\}$
- elle correspond à la séquence

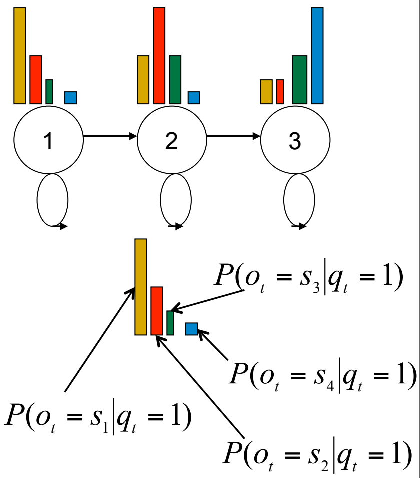
  d'états (cachée) $q = q_1 q_2, , q_t \; \ldots . q_T$

  q_t: état à l'instant t, $q_t \in \{1, \ldots Q\}$



HMMs discrets

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



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

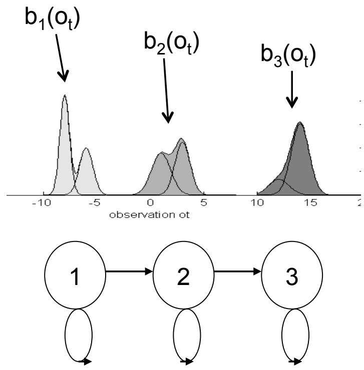
$$A = \{a_{ij}\} = P(q_t = j | q_{t-1} = i)$$

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

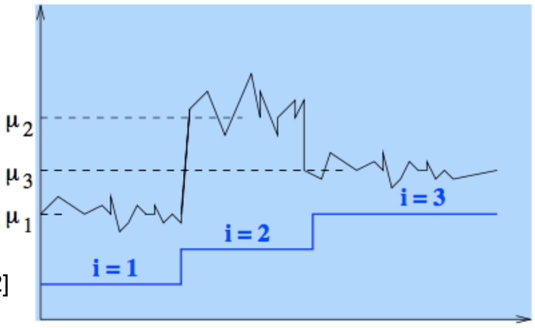
modèles de Markov cachés continus

- 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,..Q

→ gaussienne ou mélange gaussiennes



observations continues, scalaires:modèle Gaussien



[Sigelle 2012]

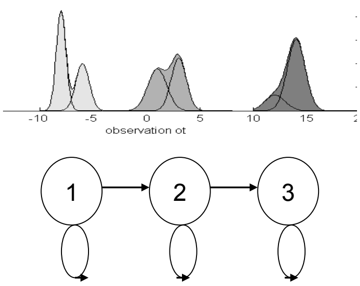
$$P(o_t \ / \ q_t = i, \ \lambda) = \frac{1}{\sqrt{2\pi}\sigma_i} \ \exp - \frac{(o_t - \mu_i)^2}{2\sigma_i{}^2}$$

modèle: inclut μi et σi, i=1,2, 3

mélange de gaussiennes

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

observations continues (scalaires ou vectorielles)



c_{ik}: poids de la kième loi gausssienne
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,..M

hypothèses fondamentales

- indépendance des observations conditionnellement aux états

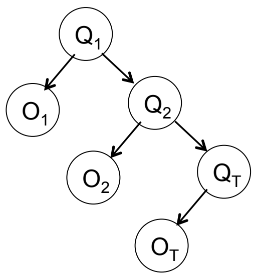
$$P(o_1,..o_t...o_T | q_1...q_t...q_T,\lambda) = \prod_{t=1}^{T} P(o_t | q_t,\lambda)$$

- chaîne de Markov stationnaire (transitions entre états)

$P(q_1, q_2, ...q_T)= P(q_1)P(q_2 / q_1)P(q_3 / q_2) \; ..... \; P(q_T / q_{T-1})$

hypothèses fondamentales

- indépendance des observations conditionnellement aux états



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

- chaîne de Markov stationnaire (transitions entre états)

$P(q_1, q_2, ...q_T)= P(q_1)P(q_2 / q_1)P(q_3 / q_2) \; ..... \; P(q_T / q_{T-1})$

hypothèses fondamentales

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

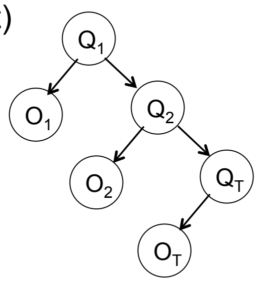
$$P(o_1,..o_t...o_T,q_1...q_t...q_T\,|\lambda) = \pi_{q_1} b_{q_1}(o_1)\prod_{t=2}^{T} a_{q_{t-1},q_t}\, P(o_t\,|q_t,\lambda)$$

$$= \pi_{q_1} b_{q_1}(o_1)\prod_{t=2}^{T} a_{q_{t-1},q_t}\, b_{q_t}(o_t)$$

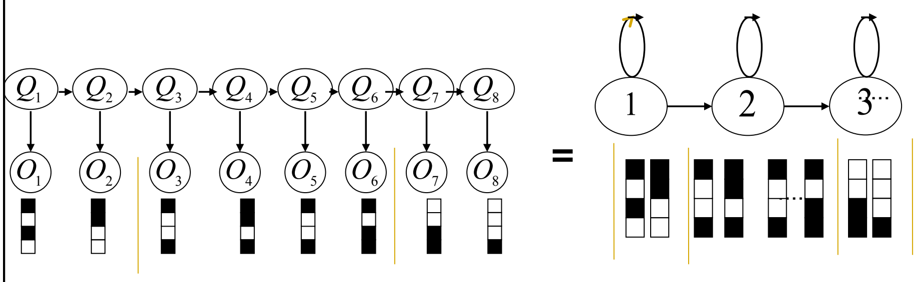
$$= P(o_1,..o_t...o_T\,|q_1...q_t...q_T,\lambda)P(q_1...q_t...q_T)$$

HMM / réseau bayésien

- un HMM est un cas particulier de réseau Bayésien (modèle graphique)
- les variables d'observations sont indépendantes connaissant leur variable parent (état)



HMM= special case of DBN



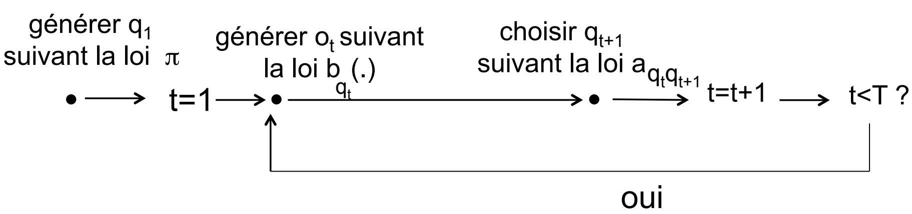
- HMM: Hidden Markov Model
- RBD: réseau Bayésien Dynamique de type arbre
- 1 state variable + 1 observation variable at each time step t

$(Q_t)_{1\le t\le T}$: state variable (hidden)

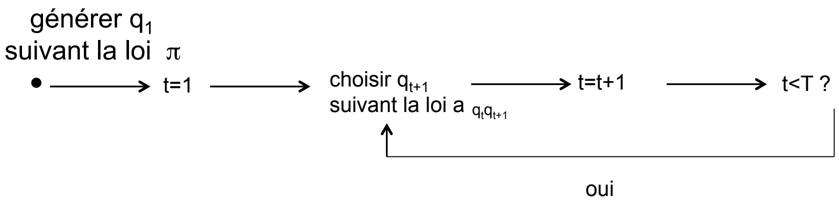
$(O_t)_{1\le t\le T}$: observation variable generated by state variable

générer une séquence d'observations

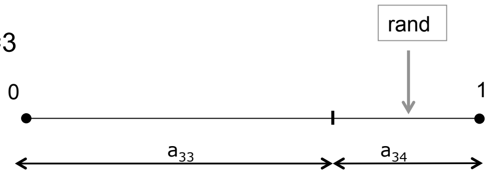
- générer la séquence d'états q1,….qT, puis générer la séquence observations à partir de chaque état
- ou générer q1 puis o1 (q1→ o1); générer q2 à partir de q1 (q1→ q2), puis o2 (q2→ o2), etc…



étape 1 : générer une séquence d'états



ex: $q_6=3$



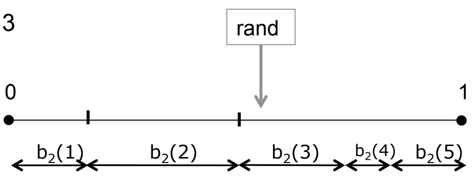
$a_{31}=0;\ a_{32}=0;$

→ $q_7=4$

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

étape 2 : générer les observations (discrètes)

- séquence états
  - $q1= 1;\ q_2= 1;\ q_3= 1;\ q_4= 2;\ q_5=2;\ q_6=3;$.......
- générer l'observation à t=4
  - $q_4=2;$
  - → $o_4= 3$



HMM pour la reconnaissance des formes

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

$$P(o_1,..o_t...o_T|\lambda_m)$$

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

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

algorithme de décodage de Viterbi

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

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

- 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 :

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

algorithme de Viterbi

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

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

- récurrence

$$P(q_1 q_2 \ldots q_t = i, q_{t+1} = j, o_1 o_2 \ldots o_t o_{t+1} | \lambda)$$
$$= P(o_{t+1}, q_{t+1} = j | o_1 \ldots o_t, q_1 .. q_{t-1} q_t = i, \lambda) P(o_1 \ldots o_t, q_1 .. q_{t-1} q_t = i | \lambda)$$
$$= P(o_{t+1} | q_{t+1} = j, \lambda) P(q_{t+1} = j | q_t = i, \lambda) P(o_1 \ldots o_t, q_1 .. q_{t-1} q_t = i | \lambda)$$
$$\max_{q1q2..qt} P(q_1 q_2 \ldots q_t = i, q_{t+1} = j, o_1 o_2 \ldots o_t o_{t+1} | \lambda) = \max_i b_j(o_{t+1}) a_{ij} \delta_t(i)$$

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

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

algorithme de décodage de Viterbi

- 1ere colonne: Initialisation

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

- colonnes 2 à T : récursion

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

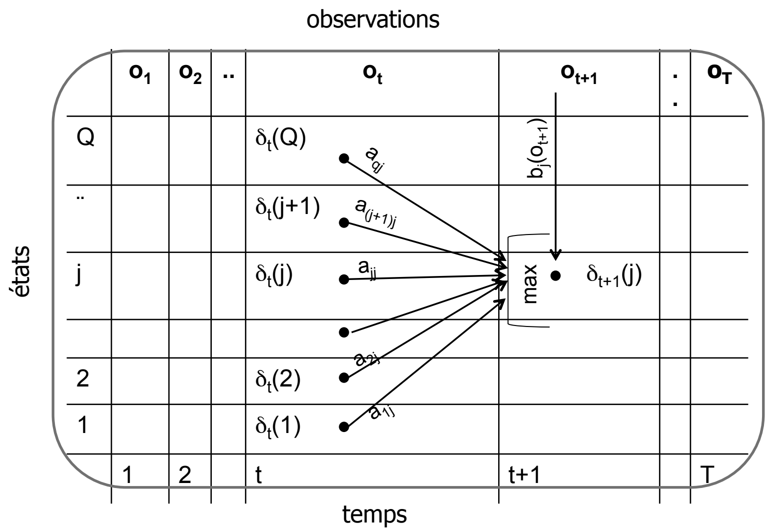
$$\varphi_{t+1}(j) = \arg\max_i a_{ij} \delta_t(i)$$ sauvegarde meilleur chemin (état précédent)

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

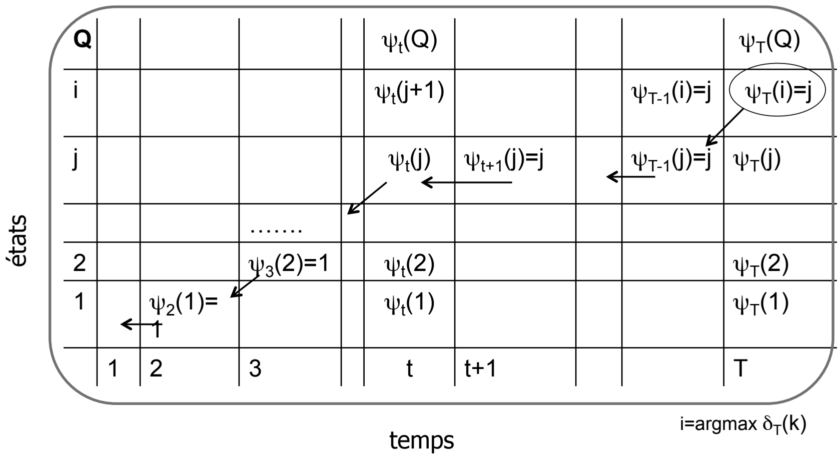
$$\hat{q}_T = \arg\max_j \delta_T(j)$$

- backtrack

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



observations
o1
o2
..
ot
ot+1
oT
Q
..
j
2
1
δt(Q)
δt(j+1)
δt(j)
δt(2)
δt(1)
aQj
a(j+1)j
ajj
a2j
a1j
bj(ot+1)
max
δt+1(j)
états
1
2
t
t+1
T
temps



Q
i
j
2
1
ψt(Q)
ψt(j+1)
ψt(j)
ψt+1(j)=j
ψT-1(i)=j
ψT-1(j)=j
ψT(Q)
ψT(i)=j
ψT(j)
....
ψ3(2)=1
ψ2(1)=
1
ψt(2)
ψt(1)
ψT(2)
ψT(1)
états
1
2
3
t
t+1
T
i=argmax δT(k)
temps

variables forward-backward

$$P(o|\lambda) = \sum_i P(o, q_t = i|\lambda)$$

$$P(o, q_t = i|\lambda) = P(o_1 \ldots o_t, q_t = i, o_{t+1} \ldots o_T|\lambda)$$

$$= P(o_{t+1} \ldots o_T|o_1 \ldots o_t, q_t = i, \lambda) P(o_1 \ldots o_t, q_t = i|\lambda)$$

$$= \underbrace{P(o_{t+1} \ldots o_T|q_t = i, \lambda)}_{\beta_t(i)} \underbrace{P(o_1 \ldots o_t, q_t = i|\lambda)}_{\alpha_t(i)}$$

$$= \beta_t(i)\alpha_t(i)$$

$\beta_t(i)$: variable backward

$\alpha_t(i)$: variable forward

$$P(o|\lambda) = \sum_{i=1}^{Q} \alpha_t(i)\beta_t(i)$$

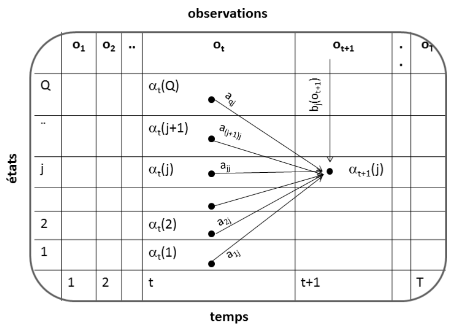
algorithme de décodage forward-backward

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

$$\alpha_1(j) = b_j(o_1)\pi_j$$

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

$$P(o|\lambda) = \sum_{j=1}^{Q} \alpha_T(j)$$



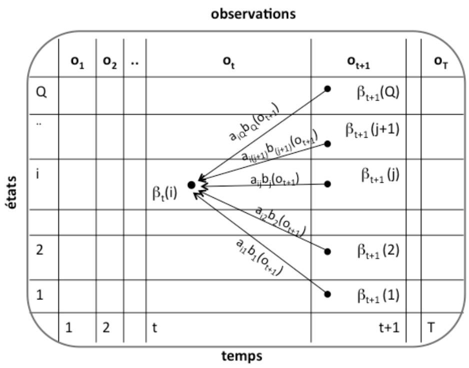
algorithme de décodage forward-backward

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

$$\beta_T(i) = 1$$

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

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



autres variables: γ et ξ

$$\gamma_t(i) = P(q_t = i|O) = \frac{\alpha_t(i)\beta_t(i)}{P(O)} \qquad i = 1,....Q$$

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{Q} \alpha_t(j)\beta_t(j)}$$

- $\gamma_t(i)$: probabilité a posteriori que l'observation o_t soit dans l'etat i.
- réalise un alignement 'soft' de la séquence d'observations O sur les états
- permet de calculer le taux d'occupation des états
  - state occupation probabilities (=occupation counts)

autres variables: γ et ξ

$$\xi_t(i,j) = P(q_t = i, q_{t+1} = j | o, \lambda) = \frac{P(o, q_t = i, q_{t+1} = j | \lambda)}{P(o|\lambda)}$$

- $\xi_t(i,j)$: probabilité que l'observation o_t soit dans l'etat i, et l'observation o_{t+1} soit dans l'etat j, connaissant toute la séquence o.

$$\xi_t(i,j) = P(q_t = i, q_{t+1} = j | o, \lambda) = \frac{\beta_{t+1}(j) b_j(o_{t+1}) a_{ij} \alpha_t(i)}{\sum_{k=1}^{Q} \alpha_t(k) \beta_t(k)}$$

PARTIE III: ESTIMATION DE PARAMETRES

Apprentissage en données complètes

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

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

Apprentissage en données complètes

- sur la base d'apprentissage totale

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

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

apprentissage données complètes

HMM continu, gaussienne mono-variee,

$$\widehat{\mu_i} = \frac{\sum_{l=1}^{L} \sum_{t=1}^{T^{(l)}} o_t^{(l)} \, \mathbb{1}_{q_t^{(l)}=i}}{\sum_{l=1}^{L} \sum_{t=1}^{T^{(l)}} \mathbb{1}_{q_t^{(l)}=i}}$$

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

Apprentissage en données incomplètes

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

Apprentissage en données incomplètes

- apprentissage Viterbi
- décodage par Viterbi
  - séquence états optimale

  → on se ramène au cas « données complètes »

apprentissage Baum-Welch

$$\hat{\pi}_i = \frac{\sum_{l=1}^{L} \gamma_1^{(l)}(i)}{L}$$

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

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

HMM discret

apprentissage Baum-Welch

gaussienne mono-variee,

$$\widehat{\mu_i} = \frac{\sum_{t=1}^{T} o_t \gamma_t(i)}{\sum_{t=1}^{T} \gamma_t(i)}$$

$$\widehat{(\sigma_i)^2} = \frac{\sum_{t=1}^{T} \left(o_t - \widehat{\mu_i}\right)^2 \gamma_t(i)}{\sum_{t=1}^{T} \gamma_t(i)}$$

conclusion

- chaînes de Markov
  - modèle de Markov observé

- modèles de Markov Cachés
  - modèle caché : émet des observations
  - approche générative pour la modélisation de séquences
  - apprentissage cas discret et données complètes
  - décodage de Viterbi
  - lien entre réseaux bayésiens dynamiques et HMMs
  - apprentissage cas discret 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.