CES Data Science

Modèles de Markov Cachés

Septembre 2015

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
  - Speech recognition
  - Handwriting recognition
  - Recognition of objects, faces in videos,…

modèle stochastique

- une classe de formes
  - est représentée par un *modèle*
- processus aléatoire à temps discret
  - change d'état aux instants entiers t=1, 2, ….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

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

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

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

Chaîne de Markov à temps discret

- espace d'états fini
- 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*
  - $P(q_t / q_1, q_2, \ldots q_{t-1}) = P(q_t / q_{t-1})$
  - $P(q_1, q_2, \ldots q_T) = P(q_1) P(q_2 / q_1) P(q_3 / q_2) \; \ldots.. \; P(q_T / q_{T-1})$
  - → *probabilités de transition entre états*

Chaîne de Markov stationnaire

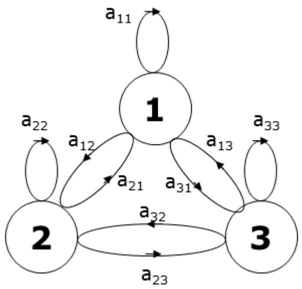
- probabilités de transition ne dépendent pas du temps
  - $P(\ q_t = i \ / \ q_{t-1} = j\ ) = P(\ q_{t+k} = \ i \ / \ q_{t+k-1} = j\ ) = a_{ij}$
  - a_{ij}= *probabilité de passer de l'état i à l'état j*
- définition: modèle d'une chaîne de Markov stationnaire
- *matrice des probabilités de transitions*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

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

topologie du modèle: ergodique / gauche droite

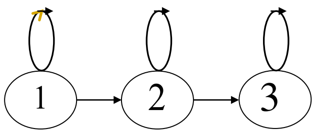
- modèle ergodique

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



- 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

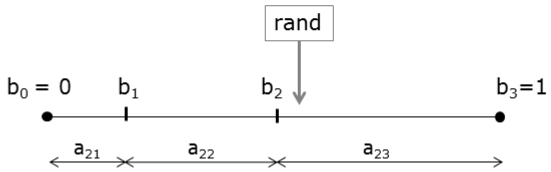
suivant chaîne de Markov (matrice A)

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

$q_1=2$

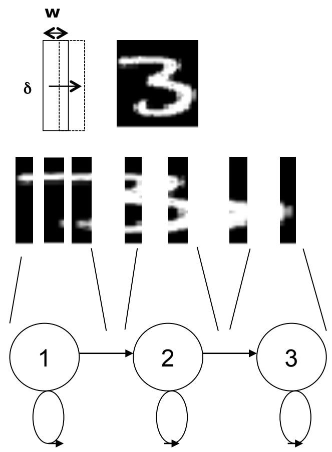
● → t=1 → choisir q_{t+1} suivant la loi $a_{q_t q_{t+1}}$ → t=t+1 → t<T ?

oui



Modèles de Markov Cachés

- 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 par les états



HMMs discrets

- soit un ensemble de Q états discrets {1, 2 … Q}
- un ensemble de N symboles discrets

  {s1, s2,…sN} → {1, 2 … N}

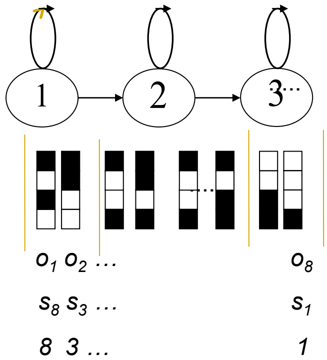
- on observe la séquence

  $o = o_1 o_2, , o_t \; .... o_T$

- $o_t \in \{s1, \ldots sN\}$
- elle correspond à la séquence

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

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



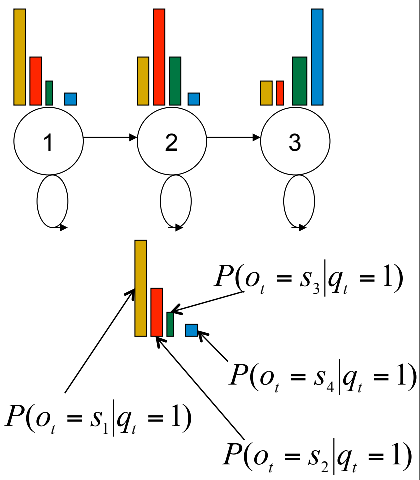
HMMs discrets

- 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)

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

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

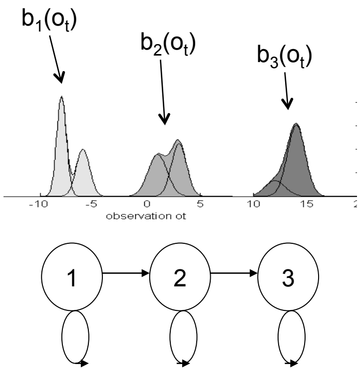
$$B = \{b_{ki}\} = P(o_t = s_k | q_t = i)$$



HMMs continus

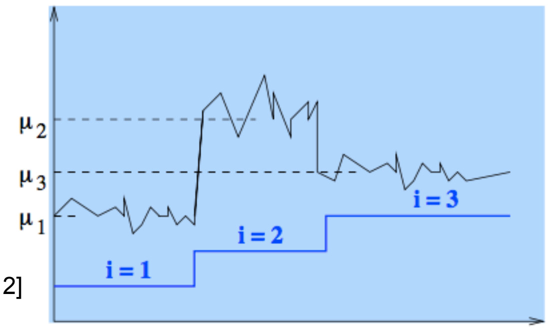
- λ: continuous HMM defined by
- π initial probability vector
- A: state transition matrix
- $b_i(o_t)$: density probability function of observations in state i, i=1,..Q

(Gaussian or Gaussian mixture)



observations continues, scalaires:modèle Gaussien

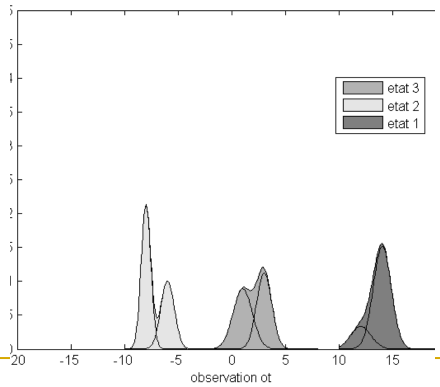
[Sigelle 2012]



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

lois gaussiennes

scalaires, multivariées ou mélange de gaussiennes



hypothèses fondamentales

- indépendance des observations conditionnellement aux états

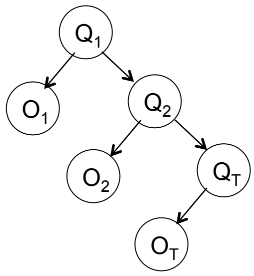
$$P(o_1,..o_t...o_T|q_1...q_t...q_T,\lambda) = \prod_{t=1}^{T} P(o_t|q_t,\lambda)$$

- chaîne de Markov stationnaire (transitions entre états)

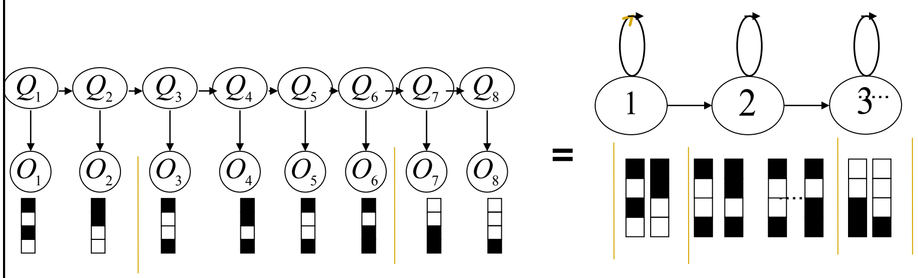
$$P(q_1, q_2, \ldots q_T) = P(q_1)P(q_2 / q_1)P(q_3 / q_2) \;\ldots.. P(q_T / q_{T-1})$$

HMM / réseau bayésien

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



HMM= special case of DBN



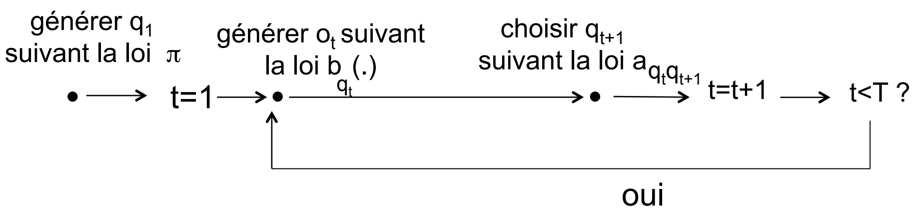
- HMM: Hidden Markov Model
- DBN: tree
- 1 state variable + 1 observation variable at each time step t

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

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

générer une séquence d'observations

- generating an observation sequence of length T
  - genererate a sequence of hidden states.
  - from each state, generate one observation.



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:

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

Apprentissage en données complètes

- pour chaque modèle λ, estimer les paramètres
- on a une base d'apprentissage
  - L séquences d'observation $o^{(l)}$, l=1….L
  - et séquences d'états associées
- pour une séquence o=o1….oT

et la séquence d'états q=q1…..qT associée

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

Apprentissage en données complètes

- sur la base d'apprentissage totale

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

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

Apprentissage en données incomplètes

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

calcul de la vraisemblance

- algorithme de décodage de Viterbi
- for observation séquence $o=o_1,\ldots o_T$

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

- instead of summing over all state sequences, search for the optimal state sequence :

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

- then estimate likelihood by :

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

décodage : 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 $o_1 \ldots o_t$.

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

- récurrence

$$P(q_1 q_2 \ldots q_t = i, q_{t+1} = j, o_1 o_2 \ldots o_t o_{t+1} | \lambda)$$

$$= P(o_{t+1}, q_{t+1} = j | o_1 \ldots o_t, q_1 \ldots q_t = i, \lambda) P(o_1 \ldots o_t, q_1 \ldots q_t = i | \lambda)$$

$$= P(o_{t+1} | q_{t+1} = j, \lambda) P(q_{t+1} = j | q_t = i, \lambda) P(o_1 \ldots o_t, q_1 \ldots q_t = i | \lambda)$$

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

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

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

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

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

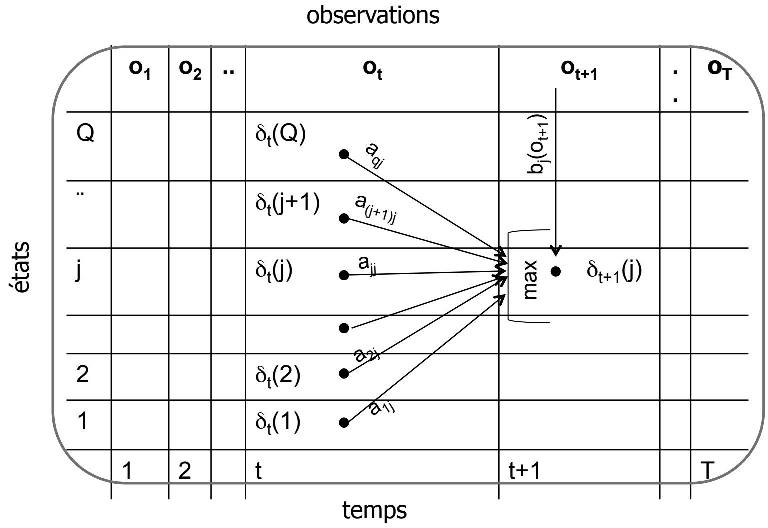
$$\varphi_{t+1}(j) = \arg\max_i a_{ij} \delta_t(i)$$ sauvegarde meilleur chemin (état précédent)

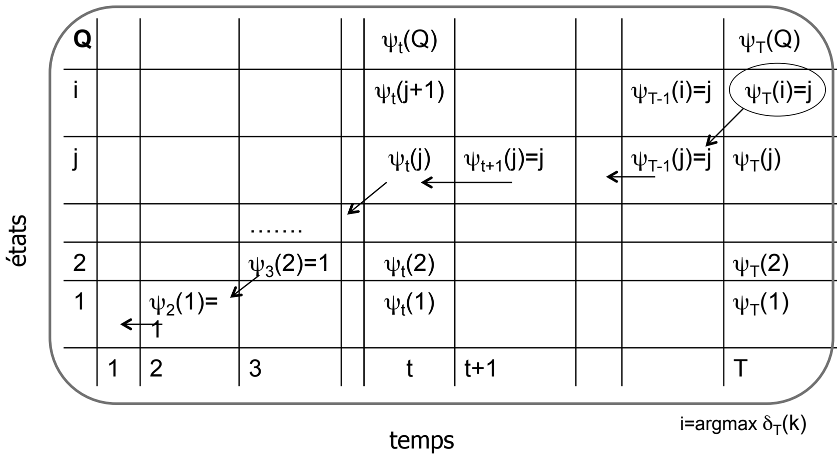
- terminaison $$P(o, \hat{q}) = \max_j \delta_T(j)$$

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

- 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
δt(Q)
aQj
δt(j+1)
a(j+1)j
bj(ot+1)
j
δt(j)
ajj
max
δt+1(j)
2
δt(2)
a2j
1
δt(1)
a1j
1
2
t
t+1
T
états
temps


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

variables forward-backward

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

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

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

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

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

$\beta_t(i)$: variable backward (analogue à λ)

$\alpha_t(i)$: variable forward (analogue à π)

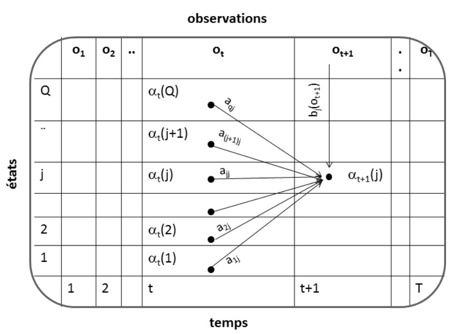
algorithme de décodage forward-backward

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

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

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

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



conclusion

- chaînes de Markov
- 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.