

## MDI341 Machine Learning Avancé

### Modèles de Markov Cachés

Mars 2020

Laurence Likforman-Sulem  
Telecom ParisTech/IDS  
[likforman@telecom-paristech.fr](mailto:likforman@telecom-paristech.fr)



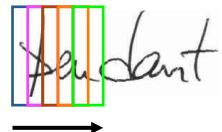
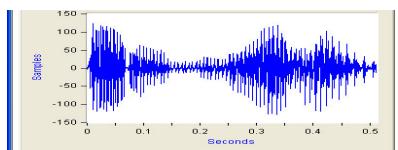
## Plan

- Chaînes de Markov
  - modèles stochastiques, paramètres
- Modèles de Markov Cachés
  - discrets/continus
  - Modèles génératifs
  - décodage : Viterbi, Baum-Welch
  - apprentissage: Viterbi, forward-backward

## applications

### ■ HMMs

- reconnaissance de la parole
- reconnaissance de l'écriture
- reconnaissance d'objets, de visages dans les vidéos,...
- Natural Language Processing (NLP): étiquetage morpho-syntaxique, correction orthographique



THE → TGE



Laurence Likforman-Telecom ParisTech

3

## PARTIE I: CHAINES DE MARKOV

4

## Modèle stochastique

- processus aléatoire à temps discret
  - ensemble de variables aléatoires  $q_1, q_2, \dots, q_T$
  - indexées aux instants entiers  $t=1, 2, \dots, T$
- notation
  - $q_t$ : variable aléatoire d'état observé au temps  $t$ 
    - notée  $q(t)$  ou  $q_t$
    - $q(t)$  prend ses valeurs dans espace fini d'états  $S$   
 $S=\{1,2, \dots, Q\}$
  - $P(q_t=i)$ : probabilité d'observer l'état  $i$  au temps  $t$

exemples états: indices de pollution, météo (beau, pluie, nuageux), fonction des mots d'un texte (verbe, nom, pronom;....)(NLP)

Laurence Likforman-Telecom ParisTech

5

## Modèle stochastique

- évolution du processus
  - état initial  $q_1$
  - suite (chaîne) de transitions entre états
    - $q_1 \rightarrow q_2 \dots \rightarrow q_t \quad t \leq T$
- calcul probabilité d'une séquence d'états
$$\begin{aligned} P(q_1, q_2, \dots, q_T) &= P(q_T | q_1, q_2, \dots, q_{T-1}) P(q_1, q_2, \dots, q_{T-1}) \\ &= P(q_T | q_1, q_2, \dots, q_{T-1}) P(q_{T-1} | q_1, q_2, \dots, q_{T-2}) P(q_1, q_2, \dots, q_{T-2}) \\ &= P(q_1) P(q_2 / q_1) P(q_3 / q_1, q_2) \dots P(q_T | q_1, q_2, \dots, q_{T-1}) \end{aligned}$$
- modèle: connaître la probabilité de chaque transition+proba initiale  $P(q_1)$

6

## Chaîne de Markov à temps discret

- propriété de Markov d'ordre k : dépendance limitée
  - $P(q_t | q_1, q_2, \dots, q_{t-1}) = P(q_t | q_{t-k}, \dots, q_{t-1})$
  - $k=1$  ou  $2$  en pratique
- cas  $k=1$ 
  - $P(q_t | q_1, q_2, \dots, q_{t-1}) = P(q_t | q_{t-1})$
  - $P(q_1, q_2, \dots, q_T) = P(q_1)P(q_2 | q_1)P(q_3 | q_2) \dots P(q_T | q_{T-1})$
  - $\rightarrow$  probabilités de transition entre états

Laurence Likforman-Telecom ParisTech

7

## 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  $j$
- modèle  $\lambda$  d'une chaîne de Markov stationnaire
  - matrice des probabilités de transitions
    - $A = [a_{ij}] \quad i=1, \dots, Q, j=1, \dots, Q$
  - vecteur des probabilités initiales
    - $\Pi = [\pi_i] \quad i=1, \dots, Q$
    - $\pi_i = P(q_1 = i)$
  - contraintes :  $0 \leq \pi_i \leq 1 \quad 0 \leq a_{ij} \leq 1$ 
$$\sum_{i=1}^Q \pi_i = 1 \quad \sum_{j=1}^Q a_{ij} = 1$$

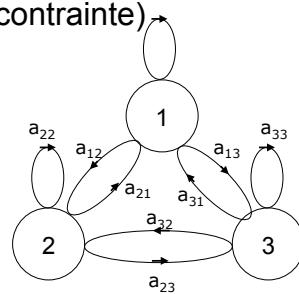
Laurence Likforman-Telecom ParisTech

8

### topologie du modèle: ergodique / gauche droite

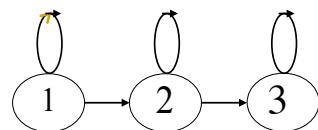
- modèle ergodique (sans contrainte)

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



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

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



9

### Chaîne de Markov stationnaire: mini TD

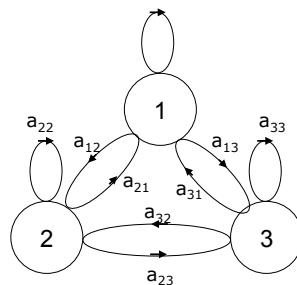
- Soit une chaîne à 3 états
  - 1: pluie (r), 2: nuages (c), 3: soleil (s)
- on observe  $q_1 = s$ , quelle est la probabilité d'observer pendant les 7 jours suivants les temps (états)

s    s s r r s c s

      $t=1$      $t=2$

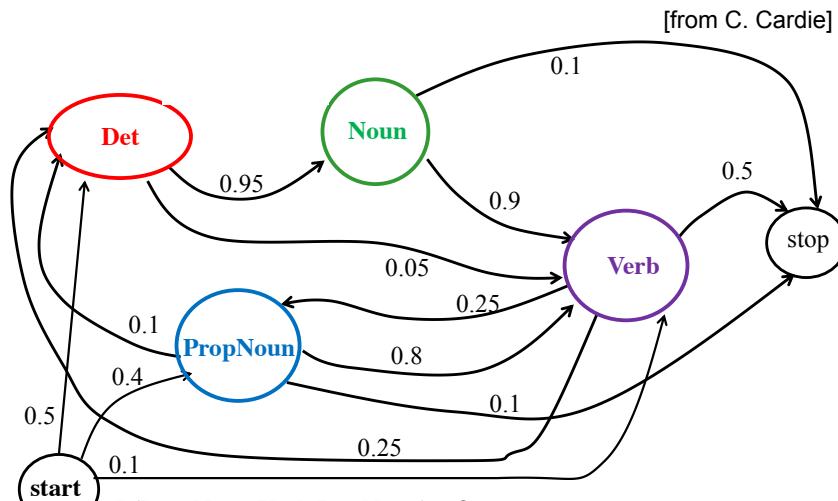
- modèle ergodique

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



10

## Mini-TD: POS part-of-speech tagging



Probabilité de la séquence: Nom propre-verbe -déterminant - Nom) ?

11

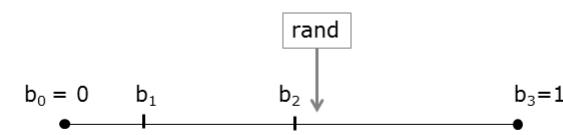
## Générer une séquence d'états

- on part de l'état  $q_1 = 2$
- générer séquence d'états de longueur T suivant chaîne de Markov (matrice A)

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

$q_1=2$   
 $\bullet \longrightarrow t=1 \longrightarrow \text{choisir } q_{t+1} \text{ suivant la loi } a_{q_t q_{t+1}}$   
 $t=t+1 \longrightarrow t < T ?$

oui

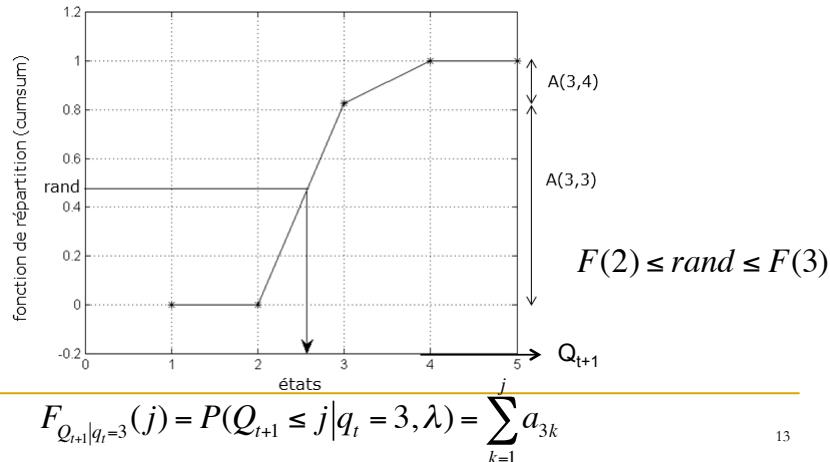


$$F_{Q_{t+1}|q_t=2}(j) = P(Q_{t+1} \leq j | q_t = 2, \lambda) = \sum_{k=1}^j a_{2k}$$

12

### Méthode d'inversion de la fonction de répartition

$$A = \begin{bmatrix} 0.1 & 0.9 & 0 & 0 \\ 0 & 0.15 & 0.85 & 0 \\ 0 & 0 & 0.8 & 0.2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{matrice de transitions modèle } \lambda$$



### générer une séquence d'états: mini-TD

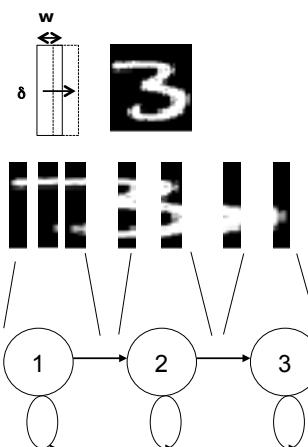
- on donne
- générer séquence d'états de longueur  $T=5$  suivant chaîne de Markov (matrice  $A$ )
  - $\pi = [0.35 \ 0.65]$        $A = \begin{bmatrix} 0.35 & 0.65 \\ 0.2 & 0.8 \end{bmatrix}$
  - *on tire les nombres aléatoires suivants:*
  - $u1 = 0.92$  ( $q1$ )
  - $u2 = 0.31$
  - $u3 = 0.1$
  - $u4 = 0.4$
  - $u5 = 0.01$

## PARTIE II: MODÈLES DE MARKOV CACHÉS

15

### Modèles de Markov Cachés

- une classe de forme
  - modèle  $\lambda$
- 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

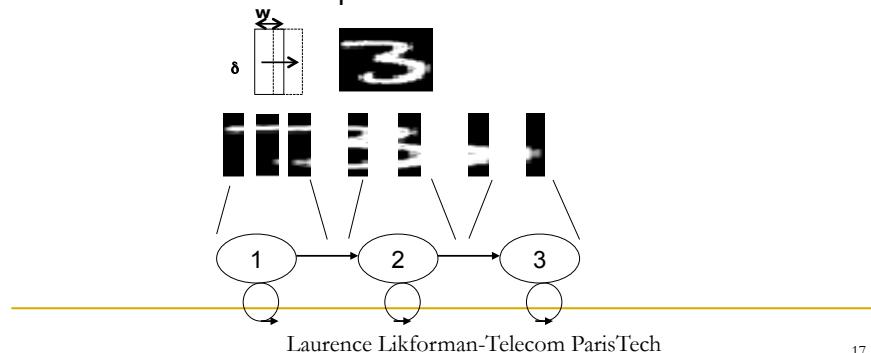


Laurence Likforman-Telecom ParisTech

16

## Processus stochastiques

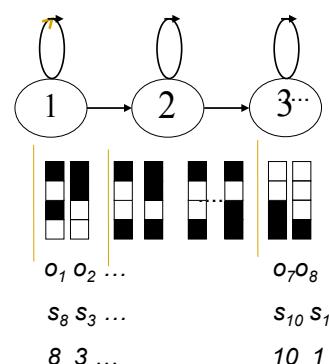
- variables d'états
  - $q_t$  prend ses valeurs dans  $\{1, 2, \dots, Q\}$  (nombre fini d'états)
  - évolution: probabilités de transition
- variables d'observations
  - discrètes ou continues
  - évolution: émission par les états



17

## 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  $o = o_1 o_2 o_3 \dots o_T$ 
  - $o = s_8 s_3 s_{13} s_6 s_8 s_5 s_{10} s_1$
  - $o = 8 \ 3 \ 13 \ 6 \ 8 \ 5 \ 10 \ 1$
- $q$  correspond à séquence d'états (cachés)
  - $q = q_1 q_2 q_3 \dots q_T$
  - $q = 1 \ 1 \ 2 \ 2 \ 2 \ 2 \ 3 \ 3$



18

## HMMs discrets

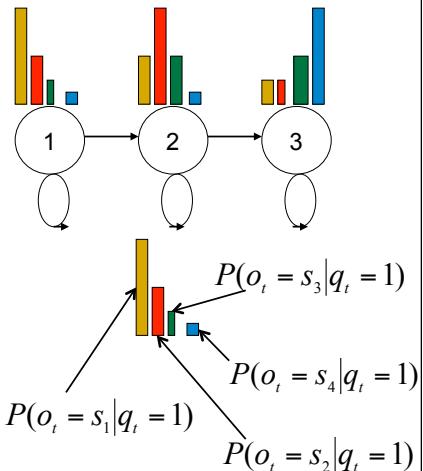
- HMM  $\lambda$  discret est défini par

- $\pi$  vecteur probabilités initiales
- $A$ : matrice transition
- $B$  : matrice des probabilités d'observation des symboles (dans les états)

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

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

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

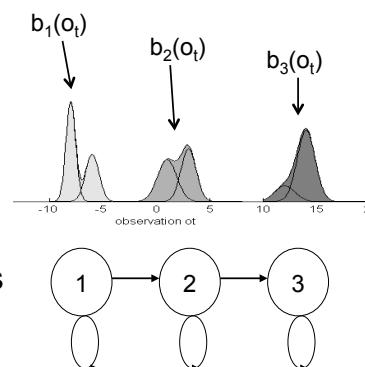


Laurence Likforman-Telecom  
ParisTech

19

## modèles de Markov cachés continus

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

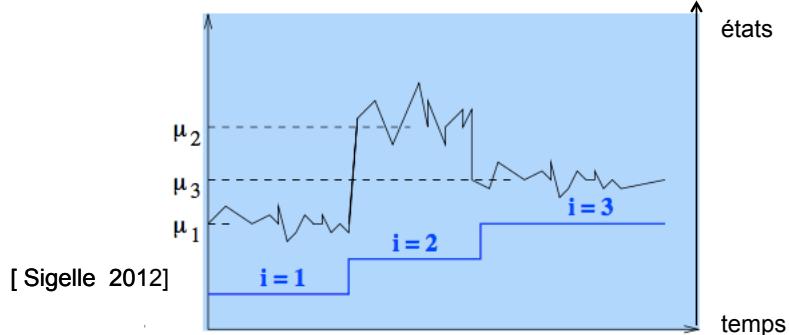


L. Likforman - Telecom ParisTech

20

## modèle d'observations Gaussien

observation continue scalaire



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

modèle: inclut  $\mu_i$  et  $\sigma_i$ ,  $i=1,2,3$

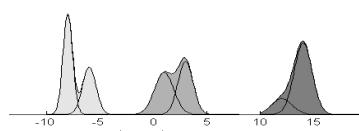
Laurence Likforman-Telecom ParisTech

21

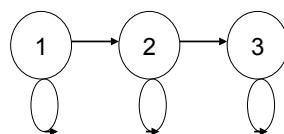
## mélange de gaussiennes

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

observations continues (scalaires ou vectorielles)



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



modèle  $\lambda$ : inclut  $c_{ik}$ ,  $\mu_{ik}$  et  $\Sigma_{ik}$ ,  $i=1,2,3$  et  $k=1,..M$

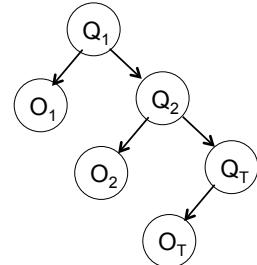
L. Likforman - Telecom ParisTech

22

## 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)

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

Laurence Likforman-Telecom ParisTech

23

## hypothèses fondamentales

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

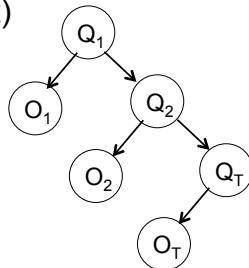
$$\begin{aligned} P(o_1, \dots, o_T, q_1, \dots, q_T | \lambda) &= \pi_{q_1} b_{q_1}(o_1) \prod_{t=2}^T a_{q_{t-1}, q_t} P(o_t | q_t, \lambda) \\ &= \pi_{q_1} b_{q_1}(o_1) \prod_{t=2}^T a_{q_{t-1}, q_t} b_{q_t}(o_t) \\ &= P(o_1, \dots, o_T | q_1, \dots, q_T, \lambda) P(q_1, \dots, q_T) \end{aligned}$$

Laurence Likforman-Telecom ParisTech

24

## HMM / réseau bayésien

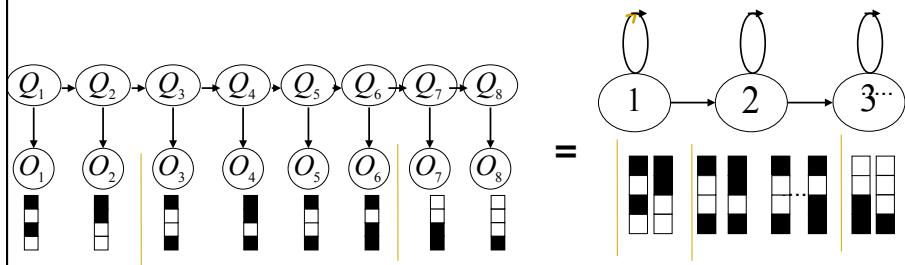
- un HMM est un cas particulier de réseau Bayésien (modèle graphique)
- les variables d'observations sont indépendantes connaissant leur variable parent (état)



Laurence Likforman-Telecom  
ParisTech

25

## HMM= cas particulier de DBN



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

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

$(O_t)_{1 \leq t \leq T}$ : variable observée « générée » par variable d'état

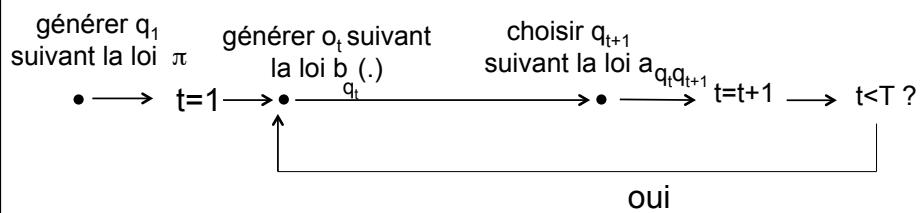
26

## PARTIE III. MODELE GENERATIF

27

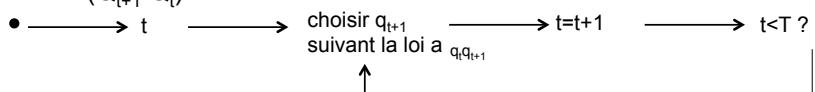
### générer une séquence d'observations

- 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...

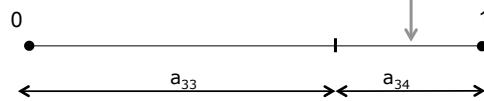


## étape 1 : générer une séquence d'états

générer  $q_{t+1}$  suivant la loi  
 $P(Q_{t+1}/Q_t)$



ex:  $q_6=3$



$\rightarrow q_7=4$

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

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

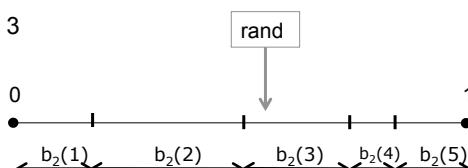
## étape 2 : générer les observations (discrètes)

### ■ 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 $t=4$

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



$$B = \begin{bmatrix} 0.3 & 0 & 0.1 & 0.2 \\ 0.1 & 0.7 & 0.2 & 0.1 \\ 0.5 & 0.1 & 0.2 & 0.1 \\ 0.1 & 0.1 & 0.3 & 0.3 \\ 0 & 0.1 & 0.2 & 0.3 \end{bmatrix}$$

symboles

← états →

## PARTIE IV : DÉCODAGE

31

### HMM pour la reconnaissance des formes

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

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

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

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

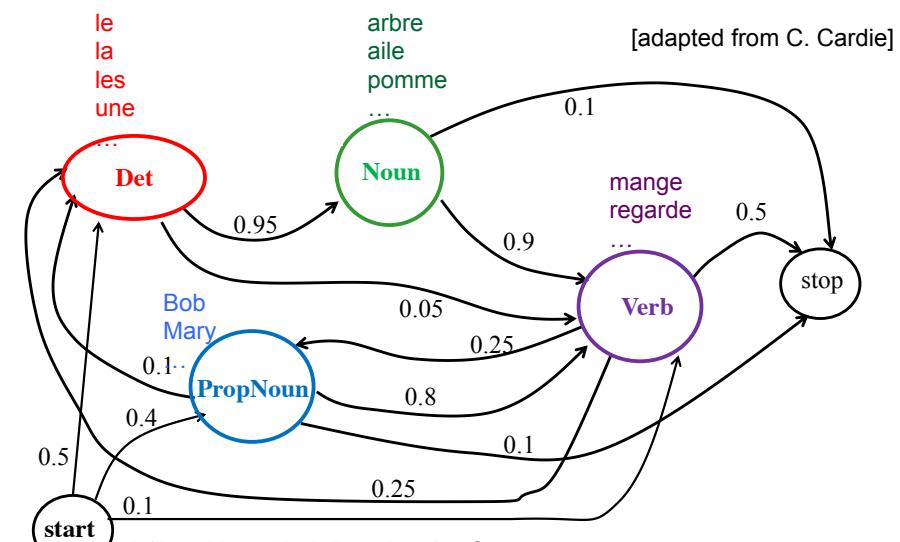
32

## HMM pour étiquetage morpho-syntaxique

- observations: mots
- séquence d'observation : suite de mots
- états cachés: Nom, pronom, verbe, etc....
- 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....

33

## Mini-TD: POS part-of-speech tagging



34

## algorithme de décodage de Viterbi

- vraisemblance pour séquence observations  $o=o_1, \dots, o_T$

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

- 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 :

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

- segmentation en états obtenue implicitement

Laurence Likforman-Telecom  
ParisTech

35

## 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  $o_1, \dots, o_t$ .

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

- 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

36

## 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

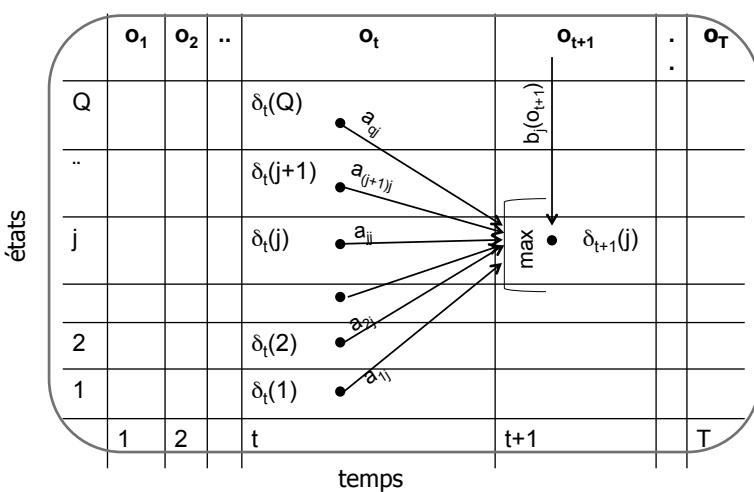
$$\hat{q}_t = \varphi(\hat{q}_{t+1}) \quad t = T-1, T-2, \dots, 1$$

Laurence Likforman-Telecom  
ParisTech

37

## calcul des deltas

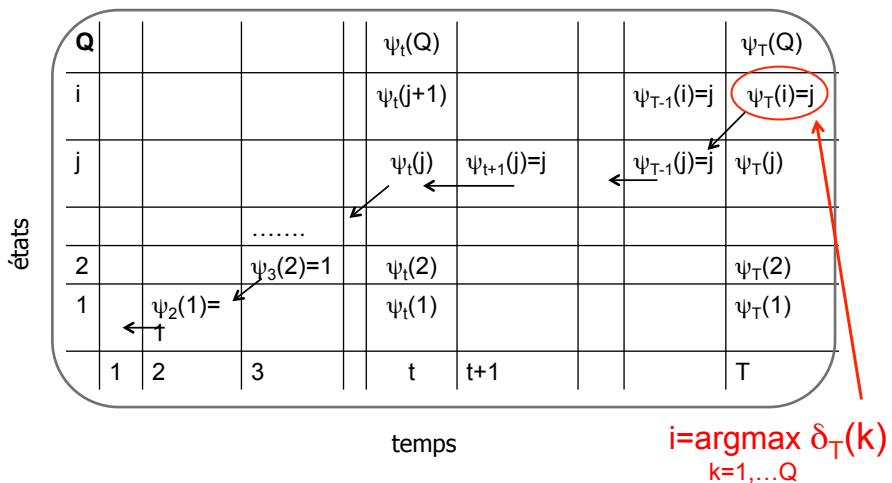
observations



Laurence Likforman-Telecom  
ParisTech

38

## backtrack



Laurence Likforman-Telecom  
ParisTech

39

## application décodage Viterbi

- POS tagging (Part of Speech)
- « Bob mange la pomme »  
→ ‘Nom propre’ ‘Verbe’ ‘déterminant’ ‘Nom’
- connaître la séquence états cachés optimale

Laurence Likforman-Telecom  
ParisTech

40

### Mini TD

Soit le modèle HMM  $\lambda$  défini par:

$$A = \begin{bmatrix} 0.3 & 0.5 & 0.2 \\ 0 & 0.3 & 0.7 \\ 0 & 0 & 1 \end{bmatrix} \quad \pi = \begin{bmatrix} 0.6 \\ 0.4 \\ 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 & 0 \\ 0.5 & 0.5 \\ 0 & 1 \end{bmatrix} \quad \text{états} \leftarrow \text{observations} \rightarrow$$

calculer la vraisemblance :  $P(aabb|\lambda)$

41

### variables forward-backward

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

$\beta_t(i)$ : variable backward

$\alpha_t(i)$ : variable forward

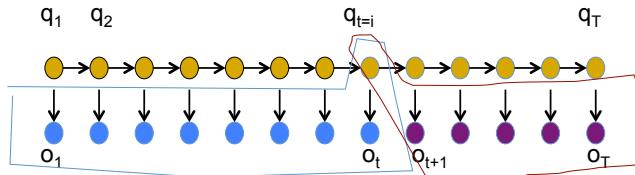
$$P(o|\lambda) = \sum_{i=1}^Q \alpha_t(i) \beta_t(i) \quad \text{on prend généralement } t=1 \text{ ou } t=T$$

Laurence Likforman-Telecom ParisTech

42

## variables forward-backward

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



$$\alpha_t(i) = P(o_1, \dots, o_t, q_t = i)$$

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

$\beta_t(i)$ : variable backward

$\alpha_t(i)$ : variable forward

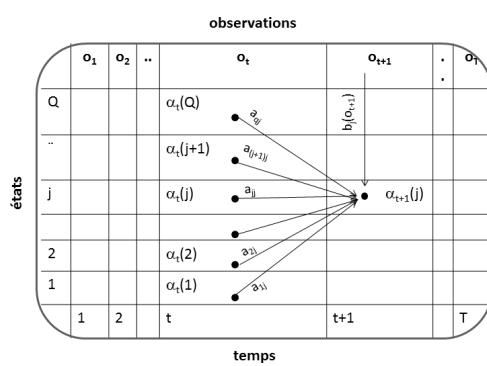
Laurence Likforman-Telecom  
ParisTech

43

## décodage forward-backward : algorithme forward

- calcul exact de la vraisemblance  $P(o | \text{modèle})$ : Baum-Welch

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



Laurence Likforman-Telecom  
ParisTech

44

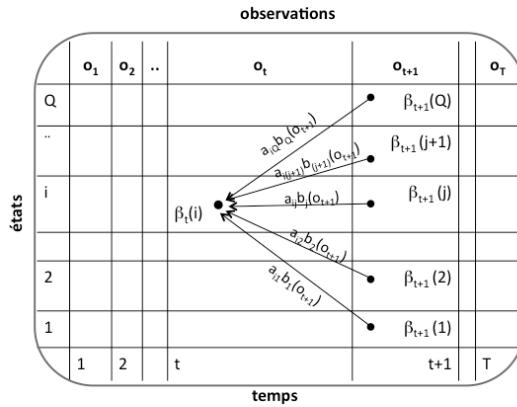
## décodage forward-backward : algorithme backward

- calcul exact de la vraisemblance  $P(O|\text{modèle})$ : Baum-Welch

$$\beta_T(i) = 1$$

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

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



Laurence Likforman-Telecom  
ParisTech

45

## autres variables: $\gamma$ et $\xi$

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

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

- $\gamma_t(i)$  : probabilité a posteriori que l'observation  $o_t$  soit dans l'état  $i$ .
- peut servir aussi pour décodage local:  $\hat{q}_t = \arg \max_j \gamma_t(j)$
- 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)

46

## autres variables: $\gamma$ et $\xi$

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

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

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

Laurence Likforman-Sulem

47

## PARTIE III: ESTIMATION DES PARAMETRES

48

## Apprentissage en données complètes

- pour chaque modèle  $\lambda$ , 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  $q=q_1 \dots q_T$

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

49

## 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,

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

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

51

## Apprentissage en données incomplètes

- estimer les paramètres, modèle  $\lambda$
- 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 en données incomplètes

- apprentissage Viterbi
  - décodage par Viterbi
  - séquence états optimale
  - on est ramené au cas « données complètes »

Laurence Likforman-Telecom ParisTech

53

## apprentissage Baum-Welch

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

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

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

HMM discret

54

## apprentissage Baum-Welch

gaussienne mono-variee,

$$\hat{\mu}_i = \frac{\sum_{t=1}^T o_t \gamma_t(i)}{\sum_{t=1}^T \gamma_t(i)}$$

$$\widehat{(\sigma_i)^2} = \frac{\sum_{t=1}^T (o_t - \hat{\mu}_i)^2 \gamma_t(i)}{\sum_{t=1}^T \gamma_t(i)}$$

55

## algorithme EM: expectation maximization

algorithme itératif

1) initialisation

→ 2) calcul des variables  $\alpha, \beta, \gamma, \xi$ ,  
3) mise à jour des paramètres

$$a_{ij}^{(n+1)} = \frac{\sum_{l=1}^L \sum_{t=1}^{T^{(l)}-1} P(q_t^{(l)} = i, q_{t+1}^{(l)} = j / o^{(l)}, \lambda^{(n)})}{\sum_{l=1}^L \sum_{t=1}^{T^{(l)}-1} P(q_t^{(l)} = i / o^{(l)}, \lambda^{(n)})}$$
$$\mu_i^{(n+1)} = \frac{\sum_{l=1}^L \sum_{t=1}^{T^{(l)}} o_t^{(l)} P(q_t^{(l)} = i / o^{(l)}, \lambda^{(n)})}{\sum_{l=1}^L \sum_{t=1}^{T^{(l)}} P(q_t^{(l)} = i / o^{(l)}, \lambda^{(n)})}$$

56

## conclusion

- chaînes de Markov
  - modèle de Markov observé
- modèles de Markov Cachés
  - approche générative pour la modélisation de séquences
    - modèle caché : émet des observations
  - lien entre réseaux bayésiens dynamiques et HMMs
- décodage
  - décodage de Viterbi
  - décodage Baum-Welch
- apprentissage
  - apprentissage en données complètes
  - apprentissage en données incomplètes
    - algorithme EM (Viterbi, Baum-Welch)

Laurence Likforman-Telecom ParisTech

57

## 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.

Laurence Likforman-Telecom  
ParisTech

58