Modèles de Markov pour le TAL

Carlos Ramisch

Sources

Ces supports réutilisent du matériel de :

- Diapos d'Alexis Nasr - Statistique Inférentielle 2015-2016
- L. R. Rabiner, 1989, *A Tutorial on HMM and Selected Applications in Speech Recognition*,
- D. Jurafsky and J. H. Martin, 2009, *Speech and Language Processing*, chapter 6

Programme

1. Processus stochastiques
2. Chaînes de Markov
3. POS tagging
4. Chaînes de Markov Cachées (HMM)

Processus stochastique

Processus stochastique (ou processus aléatoire)

Séquence $q_1, q_2 \dots q_T$ de variables aléatoires fondées sur le même ensemble fondamental Ω.

- Valeurs possibles des variables aléatoires $S_1 \dots S_N$
  $\rightarrow$ **états** possibles du processus
- $q_t \rightarrow$ état du processus au temps t (observation au temps t)
- Généralement, les variables aléatoires ne sont pas indépendantes les unes des autres.

Processus stochastique

Entièrement déterminé par :

1. loi de probabilité de la première variable aléatoire q_1 du processus lors de la première observation
2. pour $t > 1$ la probabilité conditionnelle :

$$P(q_t = S_j | q_1 = S_k, \ldots, q_{t-1} = S_i), \quad 1 \leq i, j, k \leq N$$

Chaîne de Markov

Une **chaîne de Markov** est un type particulier de processus stochastique qui vérifie deux conditions :

- **Hypothèse de Markov :** L'état au temps t dépend **uniquement** de son état au temps $t-1$:

$$P(q_t = S_j | q_1 = S_k, \ldots, q_{t-1} = S_i) = P(q_t = S_j | q_{t-1} = S_i)$$

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

$$\forall t, 0 < t \leq T, \quad P(q_t = S_j | q_{t-1} = S_i) = a_{ij}$$

Représentation matricielle

- On peut représenter une chaîne de Markov par la matrice de transition de l'état S_i à l'état S_j

$$A = \{a_{ij}\}$$

- Avec les propriétés, pour tout $1 \leq i, j \leq N$:

$$a_{ij} \geq 0$$

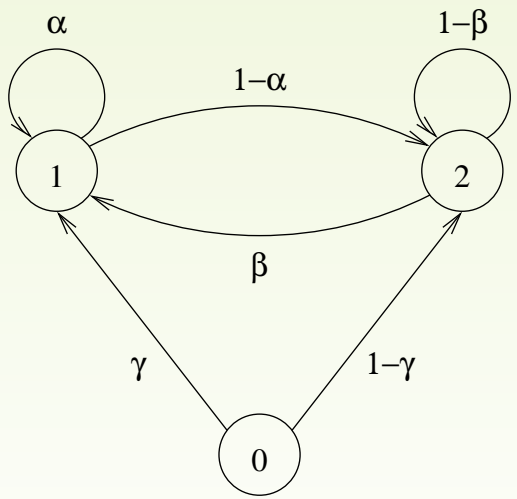
$$\sum_{j=1}^{N} a_{ij} = 1$$

- On a besoin aussi d'une distribution initiale :

$$\pi_i = P(q_1 = S_i), \forall 1 \leq i \leq N$$

Représentation graphique

- Graphe ou automate fini
- Les valeurs possibles $S_1 \ldots S_N$ de chaque état sont les noeuds/états
- Les transitions a_{ij} sont les étiquettes des arcs/transitions



Exemple 1 : météo I

- État S_1 = pluie
- État S_2 = couvert
- État S_3 = soleil

$$A = \{a_{ij}\} = \begin{bmatrix} 0.4 & 0.3 & 0.3 \\ 0.2 & 0.6 & 0.2 \\ 0.1 & 0.1 & 0.8 \end{bmatrix} \qquad \pi = [0 \quad 0 \quad 1]$$

1. Dessinez la représentation graphique du modèle
2. Calculez la probabilité de l'observation
   $O = \{S_3, S_3, S_3, S_1, S_1, S_3, S_2, S_3\}$
3. Quelle est la probabilité d'avoir d jours de soleil consécutifs ?
   $P(\{S_3, S_3, \ldots, S_j, j \neq 3\} | q_1 = S_3)$

Exemple 2 : modèles à n-grammes I

- Processus stochastique : production de mots dans une phrase
- Phrase = séquence de T mots $q_1 = w_1 \dots q_T = w_T$ (w_1^T)

$$P(w_1^T) = P(w_1) \times P(w_2|w_1) \times P(w_3|w_1^2) \dots P(w_T|w_1^{T-1})$$

$$= P(w_1) \times \prod_{k=2}^{T} P(w_k|w_1^{k-1})$$

- Hypothèse de Markov : historique récent

$$P(w_k|w_1^{k-1}) \approx P(w_k|w_{k-m+1}^{k-1})$$

- Pour $m = 2$

$$P(w_k|w_1^{k-1}) \approx P(w_k|w_{k-1})$$

Exemple 2 : modèles à n-grammes II

- Ainsi, $P(w_1^T)$ est simplifiée

$$P(w_1^T) = P(w_1) \times \prod_{k=2}^{T} P(w_k|w_{k-1}) = P(w_1) \times \prod_{k=2}^{T} \frac{P(w_{k-1}, w_k)}{P(w_{k-1})}$$

- Estimation des probabilités : maximum de vraisemblance

$$P(w_j^k) = \frac{c(w_j^k)}{N}$$

- $N =$ nb. de tokens du corpus, $c(\cdot)$ est le nb. d'occurrences
- Au final,

$$P(w_1^T) = \frac{c(w_1)}{N} \times \prod_{k=2}^{T} \frac{\frac{c(w_{k-1}, w_k)}{N}}{\frac{c(w_{k-1})}{N}} = \frac{c(w_1)}{N} \times \prod_{k=2}^{T} \frac{c(w_{k-1}, w_k)}{c(w_{k-1})}$$

Exemple 2 : modèles à n-grammes III

- Généralisation à m quelconque :

$$P(w_1^T) = \frac{c(w_1)}{N} \times \prod_{k=2}^{T} \frac{\frac{c(w_{k-m+1}^k)}{N}}{\frac{c(w_{k-m+1}^{k-1})}{N}} = \frac{c(w_1)}{N} \times \prod_{k=2}^{T} \frac{c(w_{k-m+1}^k)}{c(w_{k-m+1}^{k-1})}$$

POS tagging stochastique I

1. Étant donné une phrase $O = w_1^T$, quelle est sa séquence d'étiquettes $Q = q_1^T$?
2. Plusieurs étiquettes possibles pour un mot w_t
   - $\text{souris}_{[\text{nc}]}$
   - $\text{souris}_{[\text{v}]}$
3. Donner comme résultat la séquence optimale Q^* obtenu comme suit :

$$Q^* = \underset{\forall Q}{\text{argmax}}\, P(Q|O)$$

4. Appliquer la règle de la chaîne

$$P(Q|O) = \frac{P(Q,O)}{P(O)} = \frac{P(O|Q)P(Q)}{P(O)} \approx P(O|Q)P(Q)$$

POS tagging stochastique II

5. Décomposer les modèles mot à mot

$$P(Q|O) = \prod_{i=1}^{T} P(w_i|q_i)P(q_i|q_1^{i-1})$$

6. Simplifier le modèle de transition à l'aide de l'hypothèse de Markov

$$P(q_i|q_1^{i-1}) \approx P(q_i|q_{i-m+1}^{i-1})$$

7. Par exemple, pour $m = 2$, on choisit Q^* qui maximise

$$\prod_{i=1}^{T} P(w_i|q_i)P(q_i|q_{i-1})$$

POS tagging stochastique III

- Consulter le lexique pour construire le treillis d'étiquettes possibles
- Estimer les probabilités à partir d'un corpus annoté
- Utiliser l'algorithme de Viterbi pour trouver la solution optimale
- Modèles spéciaux pour traiter les mots inconnus

Exercice : étiqueteur simple I

- Corpus annoté :
  - je/CL porte/V
  - je/CL la/P fais/V
  - la/D porte/N
- Phrase à étiqueter :
  - $O = O_1 O_2 O_3 =$ je la porte

1. Sachant $N = 5$ et $T = 3$, il y a combien de séquences d'étiquettes $Q = q_1 q_2 q_3$ possibles au total ?
2. Dessinez le modèle de Markov représentant les transitions
3. Dessinez le treillis des solutions possibles considérant les paires mot/étiquette O_i / q_i observées dans le corpus annoté
4. Calculez les probabilités de tous les chemins du treillis par MLE (maximum de vraissemblance)

Modèles de Markov Cachés (HMM)

- 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 (*observables*) émis par les états selon une loi de probabilité.
- Au vu d'une séquence d'observation 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).
- $Q = q_1, q_2, \ldots, q_T \rightarrow$ évolution des états du HMM
  $O = O_1, O_2, \ldots, O_T \rightarrow$ suite des symboles émis par le HMM.

HMM : formalisation

Modèle de Markov à états cachés (HMM)

- N états S_i du modèle
- M symboles observables V_k
- Probabilités de transition $A = \{a_{ij}\} = P(q_{t+1} = S_j | q_t = S_i)$
- Probabilités d'émission $B = \{b_j(k)\} = P(O_t = v_k | q_t = S_j)$
- Probabilités initiales $\pi = \{\pi_i\} = P(q_1 = S_i)$

Un HMM est représenté sous forme compacte :

$$\lambda = (A, B, \pi)$$

Exemple HMM

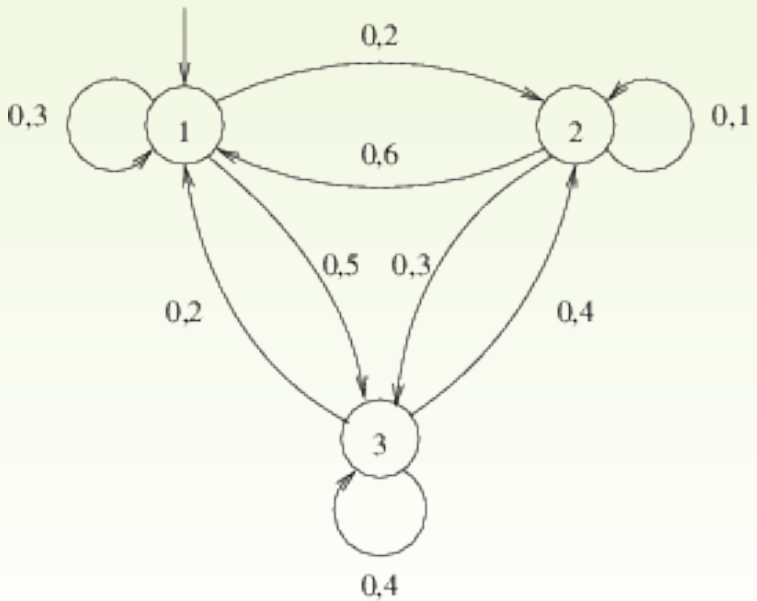
$\lambda_1 = \langle \{1,2,3\}, \{a,b,c\}, \pi, A, B \rangle$ avec :

$$\begin{array}{lll} b_a(1) = 0,6 & b_a(2) = 0 & b_a(3) = 0,3 \\ b_b(1) = 0,2 & b_b(2) = 0,5 & b_b(3) = 0 \\ b_c(1) = 0,2 & b_c(2) = 0,5 & b_c(3) = 0,7 \end{array} \quad \text{et} \quad \begin{array}{lll} a_{11} = 0,3 & a_{21} = 0,6 & a_{31} = 0,2 \\ a_{12} = 0,2 & a_{22} = 0,1 & a_{32} = 0,4 \\ a_{13} = 0,5 & a_{23} = 0,3 & a_{33} = 0,4 \end{array}$$

et

$$\pi_1 = 1 \quad \pi_2 = 0 \quad \pi_3 = 0$$

représentation graphique



Trois problèmes

1. Calcul de la probabilité d'une séquence d'observations O :

$$P(O|\lambda) = \sum_{\forall Q} P(O, Q|\lambda)$$

2. Calcul du chemin le plus probable :

$$Q* = \arg\max_{Q} P(Q|O, \lambda)$$

3. Estimation des paramètres du HMM :

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

Problème 1 - probabilité $P(O|\lambda)$

$$P(O|\lambda) = \sum_{\forall Q} P(O, Q|\lambda)$$

$$= \sum_{q_1 \dots q_T} \pi_{q_1} b_{q_1}(O_1) a_{q_1 q_2} b_{q_2}(O_2) \dots a_{q_{T-1} q_T} b_{q_T}(O_T)$$

- Énumérer toutes les solutions est prohibitif
- $2TN^T$ opérations

Idée : programmation dynamique

Définissions une matrice représentant la probabilité d'une observation partielle :

$$\alpha_t(i) = P(O_1 \dots O_t, q_t = S_i|\lambda)$$

On peut remplir $\alpha_t(i)$ itérativement avec l'algorithme *forward*

Algorithme de calcul de $P(O)$ - forward

1. Initialisation :

$$\alpha_1(i) = \pi_i b_i(O_1),\ 1 \leq i \leq N$$

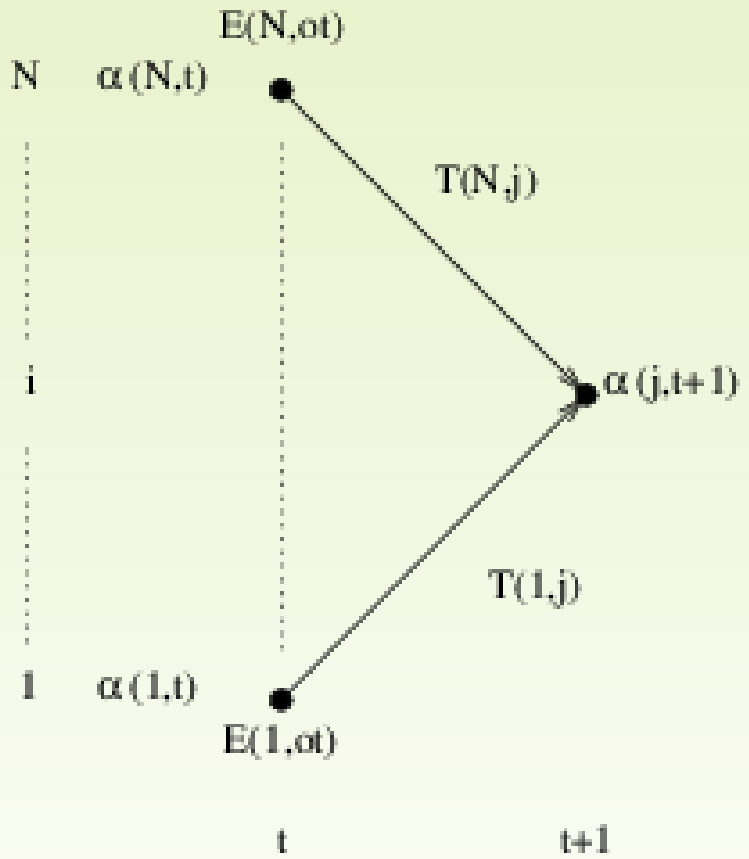
2. Etape récursive :

$$\alpha_t(j) = \sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij} b_j(O_t),\ 2 \leq t \leq T,\ 1 \leq j \leq N$$

3. Calcul de la probabilité totale :

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i)$$

Calcul de $\alpha_{t+1}(j)$



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.

Exercice : forward étiqueteur I

- Corpus annoté :
  - je/CL porte/V
  - je/CL la/P fais/V
  - la/D porte/N
- Phrase à étiqueter :
  - $O =$ je la porte

1. Calculez $P(O = \textit{je la porte}|\lambda)$ à l'aide de la matrice $\alpha_t(i)$

Calcul backward

- On définit la variable $\beta_t(i)$ de la façon suivante :

$$\beta_t(i) = P(O_{t+1} \dots O_T | q_t = S_i, \lambda)$$

Attention : $\alpha_t(i) = P(O_1 \dots O_t, q_t = S_i | \lambda)$

Algorithme de calcul de $P(O)$ - backward

1. Initialisation :

$$\beta_T(i) = 1,\ 1 \leq i \leq N$$

2. Étape récursive :

$$\beta_t(i) = \sum_{j=1}^{N} \beta_{t+1}(j) a_{ij} b_j(O_{t+1}),\ 1 \leq t \leq T-1,\ 1 \leq i \leq N$$

3. Calcul de la probabilité totale :

$$P(O|\lambda) = \sum_{i=1}^{N} \pi_i \beta_1(i)$$

Combinaison *backward* et *forward* I

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

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_t(i)\beta_t(i) \ \forall t \ 1 \leq t \leq T$$

- Ce résultat est établi en utilisant la formule des probabilités totales :

$$P(O|\lambda) = \sum_{i=1}^{N} P(O, q_t = S_i|\lambda)$$

Combinaison *backward* et *forward* II

- Chacun des termes de la somme peut être exprimée en fonction des probabilités forward et backward :

$$
\begin{aligned}
P(O, q_t = S_i) &= P(O_1 \dots O_T, q_t = S_i | \lambda) \\
&= P(O_1 \dots O_t, q_t = S_i, O_{t+1} \dots O_T | \lambda) \\
&= P(O_1 \dots O_t, q_t = S_i | \lambda) \times P(O_{t+1} \dots O_T | O_1 \dots O_t, q_t = S_i, \lambda) \\
&= P(O_1 \dots O_t, q_t = S_i | \lambda) \times P(O_{t+1} \dots O_T | q_t = S_i, \lambda) \\
&= \alpha_t(i)\beta_t(i)
\end{aligned}
$$

Problème 2 - Chemin le plus probable

- Étant donné un HMM λ et une séquence d'observations $O = O_1 \dots O_T$, déterminer la séquence d'états $Q* = q_1, q_2, \dots q_T$ la plus probable ayant pu générer O.
- Solution naïve : déterminer toutes les séquences d'états ayant pu générer O, puis calculer leurs probabilités afin de déterminer la plus probable.
- Méthode coûteuse $\rightarrow$ existe N^T chemins possibles.
- Solution : utiliser le treillis (algorithme de Viterbi)

Algorithme de Viterbi

1. Initialisation du treillis :

$$\delta(j,1) = \pi_j b_j(O_1),\ 1 \leq j \leq N$$

$$\psi(j,1) = 0$$

2. Etape récursive :

$$\delta(j,t+1) = \max_{1 \leq i \leq N} \delta(i,t) a_{ij} b_j(O_{t+1}),\ 1 \leq t < T,\ 1 \leq j \leq N$$

stockage du meilleur état précédent :

$$\psi(j,t+1) = \arg \max_{1 \leq i \leq N} \delta(i,t) a_{ij} b_j(O_{t+1}),\ 1 \leq t < T,\ 1 \leq j \leq N$$

Algorithme de Viterbi (pseudo-code)

```
def viterbi(x[]):
  score = [][]
  backtrack = [][]
  for i in 1 .. n:
    for y in labels:
      score[i][y] = max_y2 score[i-1][y2] * A(y, y2) * B(y, x[i])
      backtrack[i][y] = y2

  output = []
  i = n
  while i > 1:
    output[i] = backtrack[i][y2]
    y2 = backtrack[i][y2]
    i --
  return output
```

Exercice : étiqueteur Viterbi I

- Corpus annoté :
  - je/CL porte/V
  - je/CL la/P fais/V
  - la/D porte/N
- Phrase à étiqueter :
  - $O =$ je la porte

1. Calculer la séquence la plus probable avec Viterbi

Bibliographie

- C. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*
- D. Jurafsky and J. H. Martin, *Speech and Language Processing*
- H. Baayen, *Word Frequency Distributions*