Modelos de Markov Ocultos (HMM)

А. А. Марковъ (1886).

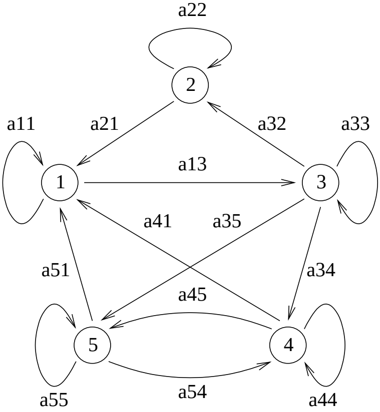
Miguel A. Alonso Jorge Graña Jesús Vilares

Departamento de Computación, Facultad de Informática, Universidade da Coruña

Índice

Cadenas de Markov

Estados $Q = \{1, 2, \ldots, N\}$, Instantes de tiempo $t = 1, 2, \ldots, T$



Propiedades

Propiedad del horizonte limitado

Una cadena de Markov de orden n es la que utiliza n estados previos para predecir el siguiente estado.
Para cadenas de orden 1, $n = 1$:

$$P(q_t = j | q_{t-1} = i, q_{t-2} = k, \ldots) = P(q_t = j | q_{t-1} = i)$$

Propiedades

Propiedad del tiempo estacionario

$P(q_t = j | q_{t-1} = i)$ es independiente del tiempo. Matriz $A = \{a_{ij}\}$ de probabilidades de transición independientes del tiempo:

$$a_{ij} = P(q_t = j | q_{t-1} = i) = P(j|i), \qquad 1 \leq i, j \leq N,$$

$$a_{ij} \geq 0, \qquad \forall i, j,$$

$$\sum_{j=1}^{N} a_{ij} = 1, \qquad \forall i.$$

Vector $\pi = \{\pi_i\}$ de probabilidad de ser el estado inicial:

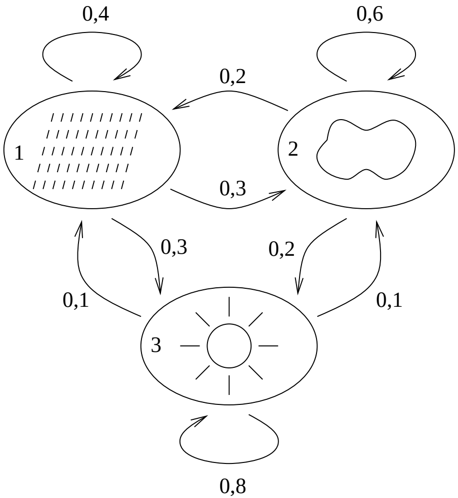
$$\pi_i = P(q_1 = i), \qquad \pi_i \geq 0, \qquad 1 \leq i \leq N,$$

$$\sum_{i=1}^{N} \pi_i = 1.$$

Ejemplo de cadena de Markov

$$A = \{a_{ij}\} = \begin{bmatrix} 0,4 & 0,3 & 0,3 \\ 0,2 & 0,6 & 0,2 \\ 0,1 & 0,1 & 0,8 \end{bmatrix}$$

$$\pi_i = \frac{1}{3}, \quad 1 \leq i \leq 3$$



Probabilidad de observar una secuencia de estados

$$P(o_1, o_2, \ldots, o_T) = \pi_{o_1} \prod_{t=1}^{T-1} a_{o_t o_{t+1}}$$

$$= P(q_1 = o_1)P(q_2 = o_2|q_1 = o_1)P(q_3 = o_3|q_2 = o_2)\ldots P(q_T = o_T|q_{T-1} = o_{T-1})$$

En el ejemplo, la probabilidad de observar la secuencia de estados *nublado-soleado-nublado-lluvia* (2, 3, 2, 1):

$$P(2, 3, 2, 1) = P(2)P(3|2)P(2|3)P(1|2)$$

$$= \pi_2 \times a_{23} \times a_{32} \times a_{21}$$

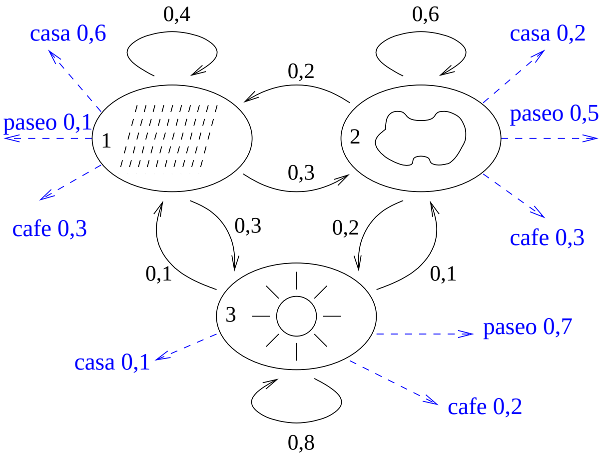
$$= \frac{1}{3} \times 0,2 \times 0,1 \times 0,2$$

$$= 0,00133333.$$

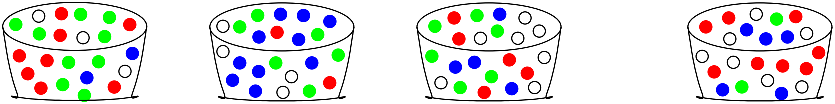
Modelos de Markov Ocultos

- En las cadenas de Markov cada estado se corresponde de manera determinista con un único suceso observable
- En los HMM **la observación es una función probabilística del estado**.
- Los HMM son un modelo doblemente estocástico, ya que uno de los procesos no se puede observar directamente (está oculto), sino que se puede observar sólo a través de otro conjunto de procesos estocásticos, los cuales producen la secuencia de observaciones.

Ejemplo de HMM



Otro ejemplo de HMM



Urna 1	Urna 2	Urna 3		Urna N
$P(color\ 1) = b_{11}$	$P(color\ 1) = b_{21}$	$P(color\ 1) = b_{31}$	$\dots$	$P(color\ 1) = b_{N1}$
$P(color\ 2) = b_{12}$	$P(color\ 2) = b_{22}$	$P(color\ 2) = b_{32}$	$\dots$	$P(color\ 2) = b_{N2}$
$P(color\ 3) = b_{13}$	$P(color\ 3) = b_{23}$	$P(color\ 3) = b_{33}$	$\dots$	$P(color\ 3) = b_{N3}$
$\vdots$	$\vdots$	$\vdots$		$\vdots$
$P(color\ M) = b_{1M}$	$P(color\ M) = b_{2M}$	$P(color\ M) = b_{3M}$	$\dots$	$P(color\ M) = b_{NM}$

Relación con etiquetación

- Bolas = palabras
- Urnas = etiquetas
- Secuencia de bolas observadas = frase

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

- Q es el conjunto de estados $\{1, 2, \ldots, N\}$.
  El estado actual en el instante de tiempo t se denota q_t.
  (instante de tiempo = posición de cada palabra).

- V es el conjunto de los sucesos observables $\{v_1, v_2, \ldots, v_M\}$.
  (M= tamaño del diccionario; cada v_k es una palabra distinta).

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

- $\pi = \{\pi_i\}$ es la distribución de probabilidad del estado inicial

$$\pi_i = P(q_1 = i), \qquad \pi_i \geq 0, \qquad 1 \leq i \leq N$$

$$\sum_{i=1}^{N} \pi_i = 1.$$

- $A = \{a_{ij}\}$ es la distribución de probabilidad de las transiciones entre estados

$$a_{ij} = P(q_t = j | q_{t-1} = i) = P(j|i), \qquad 1 \leq i, j \leq N, \qquad 1 \leq t \leq T$$

$$\sum_{j=1}^{N} a_{ij} = 1, \qquad \forall i.$$

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

- $B = \{b_j(v_k)\}$ es el **conjunto de probabilidades de emisión**, la distribución de probabilidad de los sucesos observables

$$b_j(v_k) = P(o_t = v_k | q_t = j) = P(v_k | j),$$

$$b_j(v_k) \geq 0, \quad 1 \leq j \leq N,\ 1 \leq k \leq M,\ 1 \leq t \leq T$$

$$\sum_{k=1}^{M} b_j(v_k) = 1, \qquad \forall j$$

Preguntas fundamentales

- Dados $O = (o_1, o_2, \ldots, o_T)$ y $\mu = (\pi, A, B)$
  ¿cómo calcular de una manera eficiente $P(O|\mu)$?
  (la probabilidad de la secuencia O dado el modelo μ)

- Dados $O = (o_1, o_2, \ldots, o_T)$ y $\mu = (\pi, A, B)$
  ¿cómo elegir la secuencia de estados $S = (q_1, q_2, \ldots, q_T)$ óptima?
  (la secuencia de estados que mejor *explica* la de observaciones)

- Dado $O = (o_1, o_2, \ldots, o_T)$
  ¿cómo estimar los parámetros del modelo μ que maximizan $P(O|\mu)$?
  (el modelo que mejor *explica* los datos observados)

Solución ineficiente para $P(O|\mu)$

- Entrada: $O = (o_1, o_2, \ldots, o_T)$ y $\mu = (\pi, A, B)$
- Enumerar todas las posibles secuencias de estados de longitud T
  Existen N^T secuencias distintas $S = (q_1, q_2, \ldots, q_T)$

$$P(S|\mu) = \pi_{q_1}\, a_{q_1 q_2}\, a_{q_2 q_3} \ldots a_{q_{T-1} q_T}$$

$$P(O|S,\mu) = \prod_{t=1}^{T} P(o_t|q_t,\mu) = b_{q_1}(o_1)\, b_{q_2}(o_2) \ldots b_{q_T}(o_T)$$

$$P(O,S|\mu) = P(S|\mu)\, P(O|S,\mu)$$

$$P(O|\mu) = \sum_{S} P(S|\mu)\, P(O|S,\mu)$$

- Ineficiencia: $(2T-1)N^T$ multiplicaciones y $N^T - 1$ sumas

Procedimiento hacia adelante

$$\alpha_t(i) = P(o_1, o_2, \ldots, o_t, q_t = i|\mu),$$

1. Inicialización:

$$\alpha_1(i) = \pi_i\, b_i(o_1), \qquad 1 \leq i \leq N.$$

2. Recurrencia:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i)\, a_{ij}\right] b_j(o_{t+1}), \quad t = 1, 2, \ldots, T-1, \quad 1 \leq j \leq N.$$

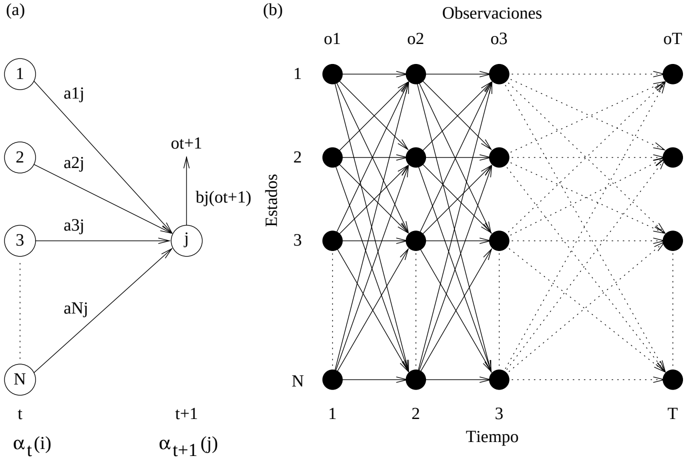
3. Terminación:

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

Eficiencia: $\mathcal{O}(N^2 T)$ operaciones

(a) Detalle de la secuencia de operaciones necesarias para el cálculo de $\alpha_{t+1}(j)$

(b) Enrejado de T observaciones y N estados



Procedimiento hacia atrás

$$\beta_t(i) = P(o_{t+1}, o_{t+2}, \ldots, o_T | q_t = i, \mu),$$

1. Inicialización:

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

2. Recurrencia:

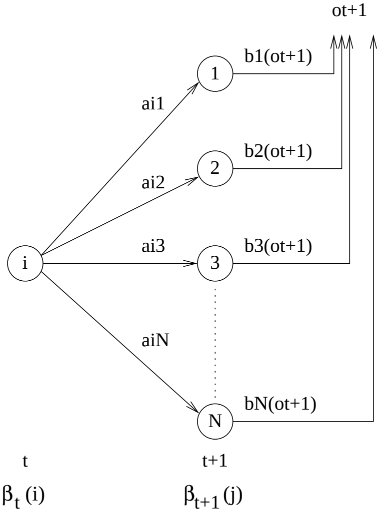
$$\beta_t(i) = \sum_{j=1}^{N} a_{ij}\, \beta_{t+1}(j)\, b_j(o_{t+1}), \quad t = T-1, T-2, \ldots, 1, \quad 1 \leq i \leq N$$

3. Terminación:

$$P(O|\mu) = \sum_{i=1}^{N} \beta_1(i)\, \pi_i\, b_i(o_1)$$

Eficiencia: $\mathcal{O}(N^2 T)$ operaciones

Detalle de la secuencia de operaciones necesarias para el cálculo de $\beta_{t+1}(j)$



Solución ineficiente para $P(S|O,\mu)$

- Selección de los estados que son **individualmente** más probables en cada instante de tiempo

$$\gamma_t(i) = P(q_t = i|O,\mu)$$

$$= \frac{P(q_t = i, O|\mu)}{P(O|\mu)} = \frac{P(q_t = i, O|\mu)}{\sum_{j=1}^{N} P(q_t = j, O|\mu)} = \frac{\alpha_t(i)\,\beta_t(i)}{\sum_{j=1}^{N} \alpha_t(j)\,\beta_t(j)}$$

- Reconstrucción de la secuencia más probable:

$$q_t^* = \arg \max_{1 \leq i \leq N} [\gamma_t(i)], \qquad 1 \leq t \leq T$$

- Inconsistencia: podría ocurrir que dos estados i y j aparecieran contiguos en la secuencia óptima aún cuando $a_{ij} = 0$

Solución eficiente para $P(S|O,\mu)$: el algoritmo de **Viterbi**

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

$\delta_t(i)$ almacena la probabilidad del mejor camino que termina en el estado i, teniendo en cuenta las t primeras observaciones

$$\delta_{t+1}(j) = \left[\max_{1\leq i\leq N} \delta_t(i)\, a_{ij}\right] b_j(o_{t+1})$$

La secuencia de estados se construye a través de una traza, que se almacena en las variables $\psi_t(j)$, que recuerda el argumento que maximizó esta ecuación para cada instante t y para cada estado j.

1. Inicialización:

$$\delta_1(i) = \pi_i \, b_i(o_1), \qquad 1 \leq i \leq N.$$

2. Recurrencia:

$$\delta_{t+1}(j) = \left[\max_{1 \leq i \leq N} \delta_t(i)\, a_{ij} \right] b_j(o_{t+1}), \quad t = 1, 2, \ldots, T-1, \quad 1 \leq j \leq N.$$

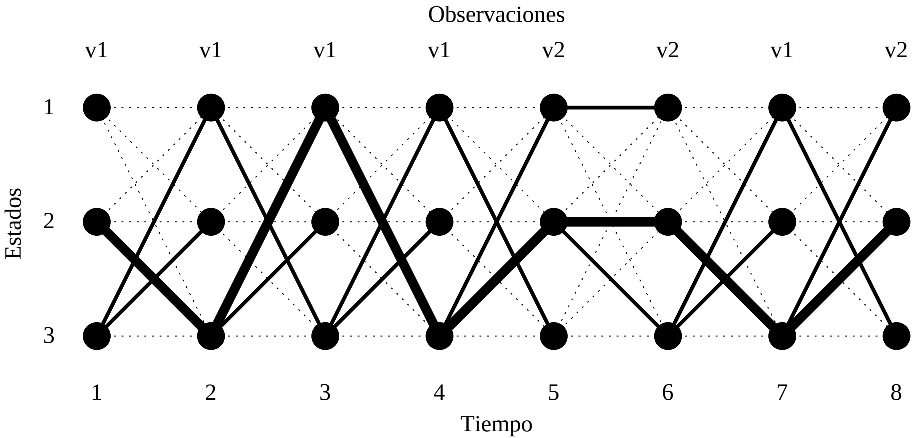
$$\psi_{t+1}(j) = \arg \max_{1 \leq i \leq N} \delta_t(i)\, a_{ij}, \quad t = 1, 2, \ldots, T-1, \quad 1 \leq j \leq N.$$

3. Terminación:

$$q_T^* = \arg \max_{1 \leq i \leq N} \delta_T(i).$$

4. Construcción hacia atrás de la secuencia de estados:

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \quad t = T-1, T-2, \ldots, 1.$$


Observaciones
v1
v1
v1
v1
v2
v2
v1
v2
Estados
1
2
3
1
2
3
4
5
6
7
8
Tiempo

Ejemplo de funcionamiento del algoritmo de Viterbi

Dado el modelo $\mu = (\pi, A, B)$ con $\quad Q = \{1, 2, 3\}, \quad V = \{v_1, v_2\}$,

$$\pi = \begin{bmatrix} 0,25 \\ 0,50 \\ 0,25 \end{bmatrix} \qquad A = \begin{bmatrix} 0,25 & 0,25 & 0,50 \\ 0 & 0,25 & 0,75 \\ 0,50 & 0,50 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 0,50 & 0,50 \\ 0,25 & 0,75 \\ 0,75 & 0,25 \end{bmatrix}$$

los cálculos para encontrar la secuencia de estados más probable dada la observación $O = (v_1, v_1, v_1, v_1, v_2, v_2, v_1, v_2)$ de longitud $T = 8$ son

$$\begin{aligned} \delta_1(1) &= \pi_1\, b_1(v_1) = (0,25)(0,50) \\ \delta_1(2) &= \pi_2\, b_2(v_1) = (0,50)(0,25) \\ \delta_1(3) &= \pi_3\, b_3(v_1) = (0,25)(0,75) \end{aligned}$$

Ejemplo de funcionamiento del algoritmo de Viterbi

Dado el modelo $\mu = (\pi, A, B)$ con $\quad Q = \{1, 2, 3\}, \quad V = \{v_1, v_2\},$

$$\pi = \begin{bmatrix} 0,25 \\ 0,50 \\ 0,25 \end{bmatrix} \qquad A = \begin{bmatrix} 0,25 & 0,25 & 0,50 \\ 0 & 0,25 & 0,75 \\ 0,50 & 0,50 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 0,50 & 0,50 \\ 0,25 & 0,75 \\ 0,75 & 0,25 \end{bmatrix}$$

los cálculos para encontrar la secuencia de estados más probable dada la observación $O = (v_1, v_1, v_1, v_1, v_2, v_2, v_1, v_2)$ de longitud $T = 8$ son

$$\begin{array}{ll}
\delta_2(1) = \max\,[\delta_1(1)\,a_{11}, \delta_1(2)\,a_{21}, \underline{\delta_1(3)\,a_{31}}]\; b_1(v_1) = (0,25)\,(0,50)^2\,(0,75) & \psi_2(1) = 3 \\
\delta_2(2) = \max\,[\delta_1(1)\,a_{12}, \delta_1(2)\,a_{22}, \underline{\delta_1(3)\,a_{32}}]\; b_2(v_1) = (0,25)^2\,(0,50)\,(0,75) & \psi_2(2) = 3 \\
\delta_2(3) = \max\,[\delta_1(1)\,a_{13}, \underline{\delta_1(2)\,a_{23}}, \delta_1(3)\,a_{33}]\; b_3(v_1) = (0,25)\,(0,50)\,(0,75)^2 & \psi_2(3) = 2
\end{array}$$

Ejemplo de funcionamiento del algoritmo de Viterbi

Dado el modelo $\mu = (\pi, A, B)$ con $\quad Q = \{1, 2, 3\}, \quad V = \{v_1, v_2\},$

$$\pi = \begin{bmatrix} 0,25 \\ 0,50 \\ 0,25 \end{bmatrix} \qquad A = \begin{bmatrix} 0,25 & 0,25 & 0,50 \\ 0 & 0,25 & 0,75 \\ 0,50 & 0,50 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 0,50 & 0,50 \\ 0,25 & 0,75 \\ 0,75 & 0,25 \end{bmatrix}$$

los cálculos para encontrar la secuencia de estados más probable dada la observación $O = (v_1, v_1, v_1, v_1, v_2, v_2, v_1, v_2)$ de longitud $T = 8$ son

$$\begin{array}{ll}
\delta_3(1) = \max\,[\delta_2(1)\,a_{11}, \delta_2(2)\,a_{21}, \underline{\delta_2(3)\,a_{31}}]\; b_1(v_1) = (0,25)\,(0,50)^3\,(0,75)^2 & \psi_3(1) = 3 \\
\delta_3(2) = \max\,[\delta_2(1)\,a_{12}, \delta_2(2)\,a_{22}, \underline{\delta_2(3)\,a_{32}}]\; b_2(v_1) = (0,25)^2\,(0,50)^2\,(0,75)^2 & \psi_3(2) = 3 \\
\delta_3(3) = \max\,[\underline{\delta_2(1)\,a_{13}}, \delta_2(2)\,a_{23}, \delta_2(3)\,a_{33}]\; b_3(v_1) = (0,25)\,(0,50)^3\,(0,75)^2 & \psi_3(3) = 1
\end{array}$$

Ejemplo de funcionamiento del algoritmo de Viterbi

Dado el modelo $\mu = (\pi, A, B)$ con $\quad Q = \{1, 2, 3\}, \quad V = \{v_1, v_2\}$,

$$\pi = \begin{bmatrix} 0,25 \\ 0,50 \\ 0,25 \end{bmatrix} \qquad A = \begin{bmatrix} 0,25 & 0,25 & 0,50 \\ 0 & 0,25 & 0,75 \\ 0,50 & 0,50 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 0,50 & 0,50 \\ 0,25 & 0,75 \\ 0,75 & 0,25 \end{bmatrix}$$

los cálculos para encontrar la secuencia de estados más probable dada la observación $O = (v_1, v_1, v_1, v_1, v_2, v_2, v_1, v_2)$ de longitud $T = 8$ son

$$\begin{array}{ll}
\delta_4(1) = \max\left[\delta_3(1)\, a_{11}, \delta_3(2)\, a_{21}, \underline{\delta_3(3)\, a_{31}}\right] b_1(v_1) = (0,25)\,(0,50)^5\,(0,75)^2 & \psi_4(1) = 3 \\
\delta_4(2) = \max\left[\delta_3(1)\, a_{12}, \delta_3(2)\, a_{22}, \underline{\delta_3(3)\, a_{32}}\right] b_2(v_1) = (0,25)^2\,(0,50)^4\,(0,75)^2 & \psi_4(2) = 3 \\
\delta_4(3) = \max\left[\underline{\delta_3(1)\, a_{13}}, \delta_3(2)\, a_{23}, \delta_3(3)\, a_{33}\right] b_3(v_1) = (0,25)\,(0,50)^4\,(0,75)^3 & \psi_4(3) = 1
\end{array}$$

Ejemplo de funcionamiento del algoritmo de Viterbi

Dado el modelo $\mu = (\pi, A, B)$ con $\quad Q = \{1, 2, 3\}, \quad V = \{v_1, v_2\},$

$$\pi = \begin{bmatrix} 0,25 \\ 0,50 \\ 0,25 \end{bmatrix} \qquad A = \begin{bmatrix} 0,25 & 0,25 & 0,50 \\ 0 & 0,25 & 0,75 \\ 0,50 & 0,50 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 0,50 & 0,50 \\ 0,25 & 0,75 \\ 0,75 & 0,25 \end{bmatrix}$$

los cálculos para encontrar la secuencia de estados más probable dada la observación $O = (v_1, v_1, v_1, v_1, v_2, v_2, v_1, v_2)$ de longitud $T = 8$ son

$$\begin{array}{ll}
\delta_5(1) = \max\left[\delta_4(1)\, a_{11}, \delta_4(2)\, a_{21}, \underline{\delta_4(3)\, a_{31}}\right] b_1(v_2) = (0,25)\,(0,50)^6\,(0,75)^3 & \psi_5(1) = 3 \\
\delta_5(2) = \max\left[\delta_4(1)\, a_{12}, \delta_4(2)\, a_{22}, \underline{\delta_4(3)\, a_{32}}\right] b_2(v_2) = (0,25)\,(0,50)^5\,(0,75)^4 & \psi_5(2) = 3 \\
\delta_5(3) = \max\left[\underline{\delta_4(1)\, a_{13}}, \delta_4(2)\, a_{23}, \delta_4(3)\, a_{33}\right] b_3(v_2) = (0,25)^2\,(0,50)^6\,(0,75)^2 & \psi_5(3) = 1
\end{array}$$

Ejemplo de funcionamiento del algoritmo de Viterbi

Dado el modelo $\mu = (\pi, A, B)$ con $\quad Q = \{1, 2, 3\}, \quad V = \{v_1, v_2\}$,

$$\pi = \begin{bmatrix} 0,25 \\ 0,50 \\ 0,25 \end{bmatrix} \qquad A = \begin{bmatrix} 0,25 & 0,25 & 0,50 \\ 0 & 0,25 & 0,75 \\ 0,50 & 0,50 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 0,50 & 0,50 \\ 0,25 & 0,75 \\ 0,75 & 0,25 \end{bmatrix}$$

los cálculos para encontrar la secuencia de estados más probable dada la observación $O = (v_1, v_1, v_1, v_1, v_2, v_2, v_1, v_2)$ de longitud $T = 8$ son

$$\begin{array}{ll}
\delta_6(1) = \max\left[\underline{\delta_5(1)\, a_{11}}, \delta_5(2)\, a_{21}, \delta_5(3)\, a_{31}\right] b_1(v_2) = (0,25)^2\,(0,50)^7\,(0,75)^3 & \psi_6(1) = 1 \\
\delta_6(2) = \max\left[\delta_5(1)\, a_{12}, \underline{\delta_5(2)\, a_{22}}, \delta_5(3)\, a_{32}\right] b_2(v_2) = (0,25)^2\,(0,50)^5\,(0,75)^5 & \psi_6(2) = 2 \\
\delta_6(3) = \max\left[\delta_5(1)\, a_{13}, \underline{\delta_5(2)\, a_{23}}, \delta_5(3)\, a_{33}\right] b_3(v_2) = (0,25)^2\,(0,50)^5\,(0,75)^5 & \psi_6(3) = 2
\end{array}$$

Ejemplo de funcionamiento del algoritmo de Viterbi

Dado el modelo $\mu = (\pi, A, B)$ con $Q = \{1, 2, 3\}$, $V = \{v_1, v_2\}$,

$$\pi = \begin{bmatrix} 0,25 \\ 0,50 \\ 0,25 \end{bmatrix} \qquad A = \begin{bmatrix} 0,25 & 0,25 & 0,50 \\ 0 & 0,25 & 0,75 \\ 0,50 & 0,50 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 0,50 & 0,50 \\ 0,25 & 0,75 \\ 0,75 & 0,25 \end{bmatrix}$$

los cálculos para encontrar la secuencia de estados más probable dada la observación $O = (v_1, v_1, v_1, v_1, v_2, v_2, v_1, v_2)$ de longitud $T = 8$ son

$$\begin{array}{ll}
\delta_7(1) = \max\,[\delta_6(1)\,a_{11}, \delta_6(2)\,a_{21}, \underline{\delta_6(3)\,a_{31}}]\; b_1(v_1) = (0,25)^2\,(0,50)^7\,(0,75)^5 & \psi_7(1) = 3 \\
\delta_7(2) = \max\,[\delta_6(1)\,a_{12}, \delta_6(2)\,a_{22}, \underline{\delta_6(3)\,a_{32}}]\; b_2(v_1) = (0,25)^3\,(0,50)^6\,(0,75)^5 & \psi_7(2) = 3 \\
\delta_7(3) = \max\,[\delta_6(1)\,a_{13}, \underline{\delta_6(2)\,a_{23}}, \delta_6(3)\,a_{33}]\; b_3(v_1) = (0,25)^2\,(0,50)^5\,(0,75)^7 & \psi_7(3) = 2
\end{array}$$

Ejemplo de funcionamiento del algoritmo de Viterbi

Dado el modelo $\mu = (\pi, A, B)$ con $Q = \{1, 2, 3\}$, $V = \{v_1, v_2\}$,

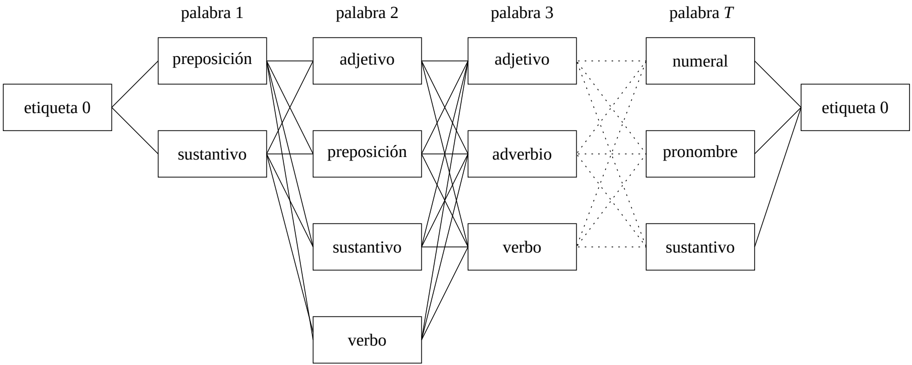
$$\pi = \begin{bmatrix} 0,25 \\ 0,50 \\ 0,25 \end{bmatrix} \qquad A = \begin{bmatrix} 0,25 & 0,25 & 0,50 \\ 0 & 0,25 & 0,75 \\ 0,50 & 0,50 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 0,50 & 0,50 \\ 0,25 & 0,75 \\ 0,75 & 0,25 \end{bmatrix}$$

los cálculos para encontrar la secuencia de estados más probable dada la observación $O = (v_1, v_1, v_1, v_1, v_2, v_2, v_1, v_2)$ de longitud $T = 8$ son

$$\begin{array}{ll}
\delta_8(1) = \max\left[\delta_7(1)\, a_{11}, \delta_7(2)\, a_{21}, \underline{\delta_7(3)\, a_{31}}\right] b_1(v_2) = (0,25)^2\,(0,50)^7\,(0,75)^7 & \psi_8(1) = 3 \\
\delta_8(2) = \max\left[\delta_7(1)\, a_{12}, \delta_7(2)\, a_{22}, \underline{\delta_7(3)\, a_{32}}\right] b_2(v_2) = (0,25)^2\,(0,50)^6\,(0,75)^8 & \psi_8(2) = 3 \\
\delta_8(3) = \max\left[\underline{\delta_7(1)\, a_{13}}, \delta_7(2)\, a_{23}, \delta_7(3)\, a_{33}\right] b_3(v_2) = (0,25)^3\,(0,50)^8\,(0,75)^5 & \psi_8(3) = 1
\end{array}$$

$q_8^* = 2$ y al reconstruir hacia atrás la secuencia de estados obtenemos $S = (2, 3, 1, 3, 2, 2, 3, 2)$

El algoritmo de Viterbi aplicado a la etiquetación



Viterbi para HMM de orden 2 (trigramas)

1. Inicialización:

$$\delta_1(i) = \pi_i\, b_i(o_1), \quad 1 \leq i \leq N.$$

$$\delta_2(i,j) = \delta_1(i)\, a_{ij}\, b_j(o_2), \quad 1 \leq i,j \leq N.$$

2. Recurrencia:

$$\delta_{t+1}(j,k) = \left[\max_{1\leq i\leq N} \delta_t(i,j)\, a_{ijk}\right] b_k(o_{t+1}), \quad t = 2,3,\ldots,T-1, \quad 1 \leq j,k$$

$$\psi_{t+1}(j,k) = \arg\max_{1\leq i\leq N} \delta_t(i,j)\, a_{ijk}, \quad t = 2,3,\ldots,T-1, \quad 1 \leq j,k \leq N.$$

3. Terminación:

$$(q^*_{T-1}, q^*_T) = \arg\max_{1\leq j,k\leq N} \delta_T(j,k).$$

4. Construcción hacia atrás de la secuencia de estados:

$$q^*_t = \psi_{t+2}(q^*_{t+1}, q^*_{t+2}), \quad t = T-2, T-3, \ldots, 1.$$

Algoritmo de Viterbi con logaritmos y sumas

0. Preproceso: $\tilde{\pi}_i = \log(\pi_i)$, $\tilde{a}_{ij} = \log(a_{ij})$, $\tilde{b}_i(o_t) = \log[b_i(o_t)]$
1. Inicialización:

$$\tilde{\delta}_1(i) = \log[\delta_1(i)] = \tilde{\pi}_i + \tilde{b}_i(o_1), \quad 1 \leq i \leq N.$$

2. Recurrencia:

$$\tilde{\delta}_{t+1}(j) = \log[\delta_{t+1}(j)] = \left[\max_{1\leq i\leq N}[\tilde{\delta}_t(i) + \tilde{a}_{ij}]\right] + \tilde{b}_j(o_{t+1}), \quad t = 1, 2, \ldots, T-$$

$$\psi_{t+1}(j) = \arg\max_{1\leq i\leq N}[\tilde{\delta}_t(i) + \tilde{a}_{ij}], \quad t = 1, 2, \ldots, T-1, \quad 1 \leq j \leq N.$$

3. Terminación:

$$q_T^* = \arg\max_{1\leq i\leq N} \tilde{\delta}_T(i).$$

4. Construcción hacia atrás de la secuencia de estados:

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \quad t = T-1, T-2, \ldots, 1.$$

Estimación de parámetros

Encontrar un modelo $\mu = (\pi, A, B)$ que maximice $P(O|\mu)$

- Estimación no supervisada

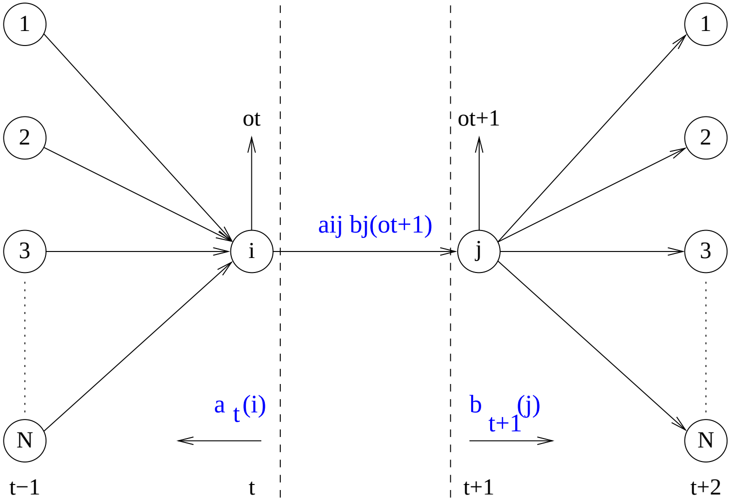
- Estimación supervisada

Estimación no supervisada: Algoritmo de Baum-Welch

- El algoritmo de Baum-Welch es un caso especial del algoritmo **EM** (Expectation-Maximization, maximización de la esperanza)
- $\xi_t(i,j) = P(q_t = i, q_{t+1} = j | O, \mu)$ es la probabilidad de estar en el estado i en el instante t y en el estado j en el instante $t+1$

$$\xi_t(i,j) = \frac{P(q_t = i, q_{t+1} = j, O|\mu)}{P(O|\mu)} = \frac{\alpha_t(i)\, a_{ij}\, b_j(o_{t+1})\, \beta_{t+1}(j)}{P(O|\mu)}$$

$$= \frac{\alpha_t(i)\, a_{ij}\, b_j(o_{t+1})\, \beta_{t+1}(j)}{\displaystyle\sum_{k=1}^{N}\sum_{l=1}^{N} \alpha_t(k)\, a_{kl}\, b_l(o_{t+1})\, \beta_{t+1}(l)}.$$


1
2
3
N
ot
i
aij bj(ot+1)
j
ot+1
1
2
3
N
a t(i)
b t+1(j)
t−1
t
t+1
t+2

Retomamos $\gamma_t(i)$ y lo relacionamos con $\xi_t(i,j)$:

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j).$$

Interpretación:

$$\sum_{t=1}^{T-1} \gamma_t(i) = \text{número esperado de transiciones desde el estado } i$$

$$\sum_{t=1}^{T-1} \xi_t(i,j) = \text{número esperado de transiciones desde el estado } i \text{ al estado } j$$

Utilizando estas fórmulas, se puede dar un método general para reestimar los parámetros de un HMM.

$$\bar{\pi}_i = \text{frecuencia esperada de estar en el estado } i \text{ en el el instante } 1 \ = \gamma_1(i)$$

$$\bar{a}_{ij} = \frac{\text{n}^{\text{o}} \text{ esperado de transiciones desde el estado } i \text{ al estado } j}{\text{n}^{\text{o}} \text{ esperado de transiciones desde el estado } i} = \frac{\displaystyle\sum_{t=1}^{T-1} \xi_t(i,j)}{\displaystyle\sum_{t=1}^{T-1} \gamma_t(i)}$$

$$\bar{b}_j(v_k) = \frac{\text{n}^{\text{o}} \text{ esperado de veces en el estado } j \text{ observando el símbolo } v_k}{\text{n}^{\text{o}} \text{ esperado de veces en el estado } j}$$

$$= \frac{\sum_{t=1}^{T} \gamma_t(j) \text{ tal que } o_t = v_k}{\sum_{t=1}^{T} \gamma_t(j)}$$

Algoritmo iterativo

- Definimos un modelo incial $\mu = (\pi, A, B)$
- Calculamos $\bar{\mu} = (\bar{\pi}, \bar{A}, \bar{B})$ mediante las ecuaciones de la transparencia anterior
- Reemplazamos μ por $\bar{\mu}$ y repetimos la reestimación de los parámetros un cierto número de veces, hasta que no se aprecie ninguna ganancia significativa entre $P(O|\bar{\mu})$ y $P(O|\mu)$

Problemas

- Muy sensible a las condiciones de inicialización del modelo
- Una inicialización incorrecta puede llevar a un máximo local
- Solución para π y A: estimación inicial equiprobable ligeramente modificada
- Solución para B:
  - **Método de Jelinek**, regla de Bayes suponiendo que todas las etiquetas que aparecen en el diccionario para una palabra dada son equiprobables
  - **Método de Kupiec**, agrupa las palabras en clases de ambigüedad

Estimación de parámetros supervisada

- **Corpus etiquetado**
- Estimación de máxima verosimilitud (maximum likelihood)

$$\pi_i = \frac{\text{nº de frases que comienzan por etiqueta } t^i}{\text{nº de frases}}$$

$$a_{ij} = \frac{C(t^i t^j)}{C(t^i)}$$

$$b_j(w^k) = \frac{C(w^k \mid t^j)}{C(t^j)}$$

Técnicas de suavizado (smoothing)

Los fenómenos que no aparecen en el corpus de entreniento dan lugar a ceros: necesidad de suavizar los parámetros.
Generalmente el suavizado es lo que marca la diferencia de rendimiento entre etiquetadores probabilísticos.

- Suavizado de Laplace (add-one)
- Descuento de Good-Turing
- Interpolación
- Backoff

Suavizado de Laplace (add-one)

- Añadir 1 a todas las cuentas $P(x) = \frac{C(x)}{N}$ donde x es un *token* (palabra, etiqueta, bigrama (de palabras o etiquetas), trigrama, ...) y N es el numero total de tokens en el texto

$$P_{Laplace}(x) = \frac{C(x) + 1}{N + V}$$

  donde V es el tamaño del *vocabulario* de tokens
- Alternativamente

$$C^*(x) = (C(x) + 1)\frac{N}{N + V}$$

- Inconveniente: Si V es grande y/o N pequeño deriva demasiada masa de probabilidad a tokens con pocas o ninguna apariciones

Suavizado de Good-Turing

- Usa las cuentas de los tokens que han sido observados una sola vez para estimar la cuenta de aquellos que nunca han sido observados
- Generaliza este razonamiento a los tokens observados c veces
- N_c es el número de tokens (etiquetas, bigramas, ...) con cuenta c:

$$N_c = \sum_{x:C(x)=c} 1$$

- Reemplaza la cuenta c por la cuenta suavizada c^*

$$c^* = (c+1)\frac{N_{c+1}}{N_c}$$

- En la práctica se aproxima N_0 mediante N, sólo se usa c^* para valores pequeños de c y se usan aproximaciones para el caso de que $N_{c+1} = 0$

Interpolación

$$P(z \mid x, y) = \lambda_1 P(z \mid x, y) + \lambda_2 P(z \mid y) + \lambda_3 P(z)$$

$$\sum_i \lambda_i = 1$$

Problema: calcular valores adecuados para λ_i

Backoff

$$P_{katz}(z \mid x, y) = \begin{cases} P^*(z \mid x, y) & \text{if } C(x, y, z) > 0 \\ \alpha(x, y)\ P_{katz}(z \mid y) & \text{else if } C(y, z) > 0 \\ P^*(z) & \text{otherwise} \end{cases}$$

$$P_{katz}(z \mid y) = \begin{cases} P^*(z \mid y) & \text{if } C(y, z) > 0 \\ \alpha(y)\ P_{katz}(z) & \text{otherwise} \end{cases}$$

donde los P^* son probabilizadas suavizadas al estilo Good-Turing y los valores de los α deben ser calculados para garantizar que la masa total de probabilidad sea 1.

Tratamiento de las palabras desconocidas

- Las probabilidades de emitir una palabra deconocida $l_1 \dots l_n$ con etiqueta t se determinana en función de sus terminaciones

$$P(l_{n-i+1},\dots,l_n|t) = \frac{P(l_{n-i+1},\dots,l_n)P(t|l_{n-i+1},\dots,l_n)}{P(t)}$$

$$P(t|l_{n-i+1},\dots,l_n) = \frac{\hat{p}(t|l_{n-i+1},\dots,l_n) + \theta_i\, P(t|l_{n-i+2},\dots,l_n)}{1+\theta_i}$$

$$\hat{p}(t|l_{n-i+1},\dots,l_n) = \frac{C(t,l_{n-i+1},\dots,l_n)}{C(l_{n-i+1},\dots,l_n)}$$

Árbol de sufijos

Las etiquetas posibles para cada sufijo se determinan a partir del árbol de letras de los sufijos

⋮	⋮
no	Wn
requiere	V3spi0
conocimiento	Scms
sobre	P
las	Dfp
funciones	Scfp
suplementarias	Afp0
de	P
⋮	⋮