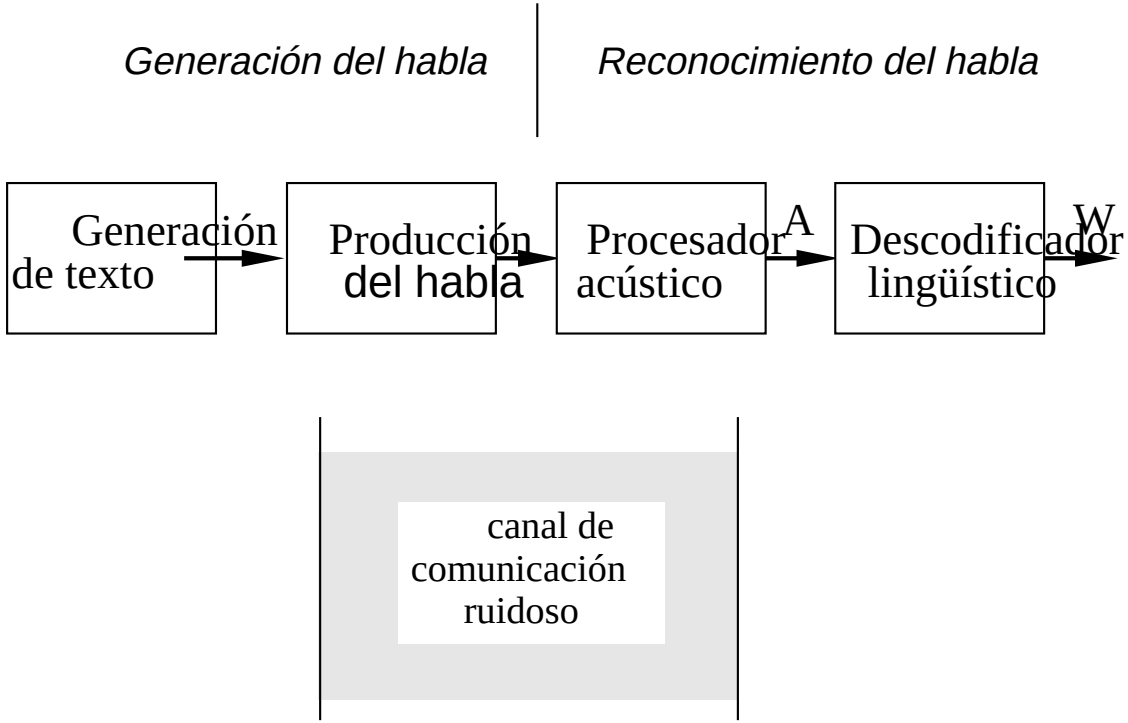
MIT

Modelos ocultos de Markov

- Introducción
- Formulación del problema
- Algoritmo de avance-retroceso
- Búsqueda de Viterbi
- Estimación del parámetro de Baum-Welch
- Otras consideraciones
  - Secuencias de observación múltiple
  - Modelos telefónicos para reconocimiento de voz continuo
  - Modelos ocultos de Markov (HMMs) de densidad continua
  - Cuestiones de implementación

Enfoque teórico de información al ASR (Reconocimiento automático del habla/voz)



El reconocimiento se consigue al maximizar la probabilidad de la cadena lingüística $\boldsymbol{W}$, dadas las pruebas acústicas $\boldsymbol{A}$, ej., elegir la secuencia lingüística $\hat{\boldsymbol{W}}$ tal que

$$P(\hat{\boldsymbol{W}}|\boldsymbol{A}) = \max_{\boldsymbol{W}} P(\boldsymbol{W}|\boldsymbol{A})$$

Enfoque teórico de información al ASR

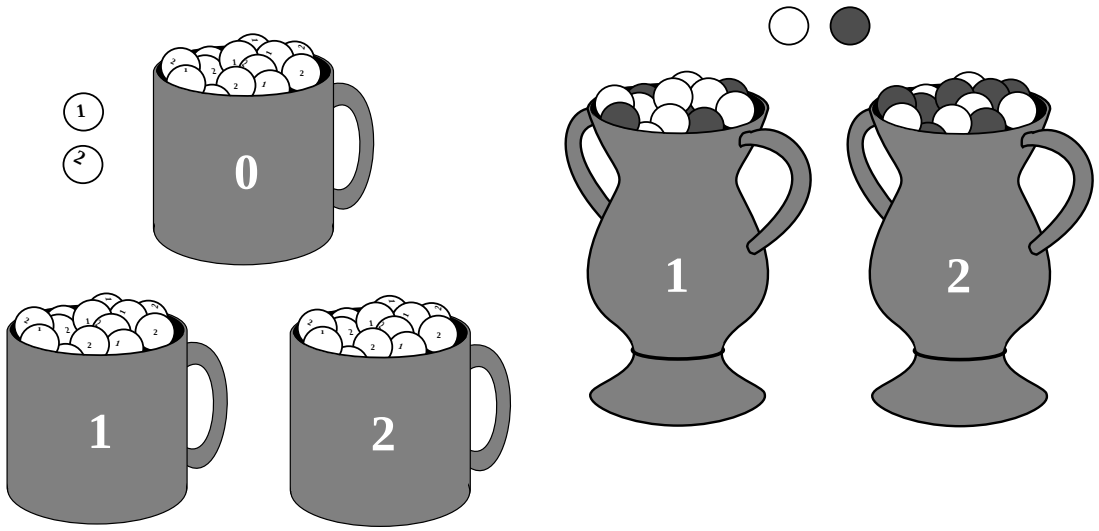
- A partir de la regla de Bayes:

$$P(\boldsymbol{W}|\boldsymbol{A}) = \frac{P(\boldsymbol{A}|\boldsymbol{W})P(\boldsymbol{W})}{P(\boldsymbol{A})}$$

- Los modelos ocultos de Markov (HMM) se ocupan de la cantidad $P(\boldsymbol{A}|\boldsymbol{W})$
- Cambio en la notación:

$$\begin{aligned} \boldsymbol{A} &\rightarrow \boldsymbol{O} \\ \boldsymbol{W} &\rightarrow \lambda \\ P(\boldsymbol{A}|\boldsymbol{W}) &\rightarrow P(\boldsymbol{O}|\lambda) \end{aligned}$$

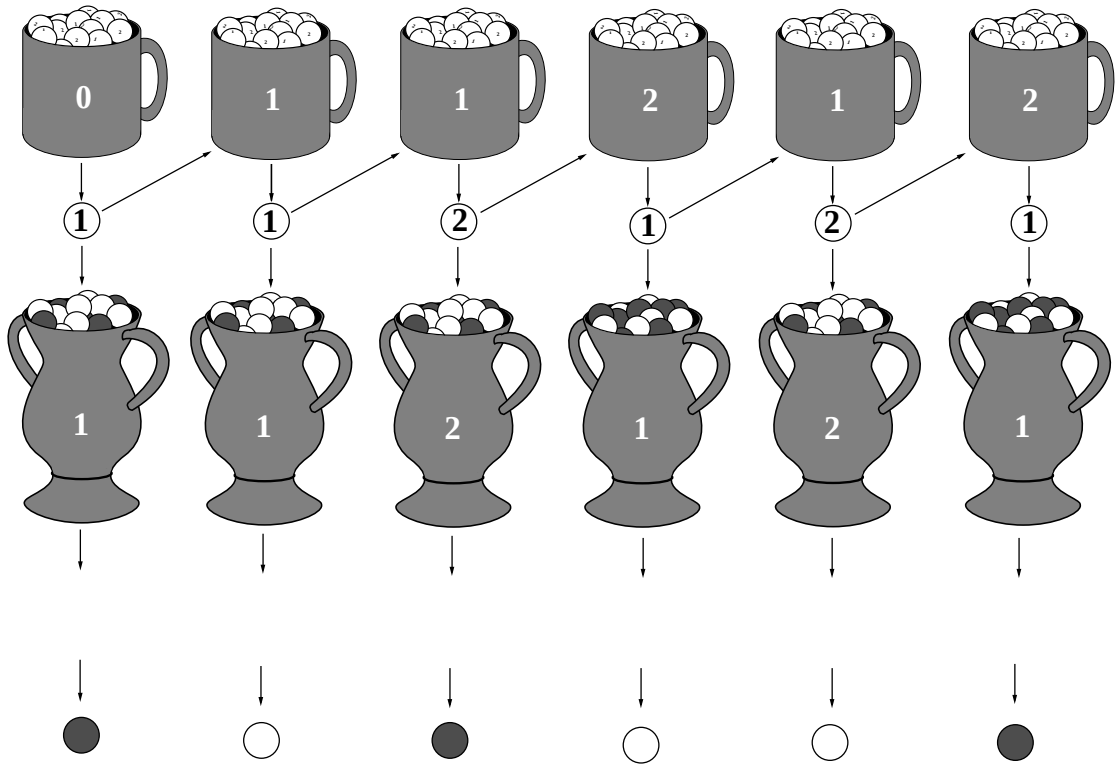
HMM: Un ejemplo



- Considere 3 tazas, cada una conteniendo mezclas de piedras con *estados* 1 y 2
- Las fracciones para la taza ith son a_{i1} y a_{i2}, y $a_{i1} + a_{i2} = 1$
- Considere 2 urnas, cada una con mezclas de bolas blancas y negras.
- Las fracciones para la urna ith son $b_i(B)$ y $b_i(W)$; $b_i(B) + b_i(W) = 1$
- El vector de parámetro para este modelo es:

$$\lambda = \{a_{01}, a_{02}, a_{11}, a_{12}, a_{21}, a_{22}, b_1(B), b_1(W), b_2(B), b_2(W)\}$$

HMM: Un ejemplo (continuación)



Secuencia de observación: $\boldsymbol{O} = \{B, W, B, W, W, B\}$

Secuencia de estado: $\boldsymbol{Q} = \{1, 1, 2, 1, 2, 1\}$

Objetivo: Dado el modelo λ y la secuencia de observación $\boldsymbol{O}$, ¿cómo se puede determinar la secuencia de estado subyacente $\boldsymbol{Q}$?

Elementos de un modelo oculto de Markov discreto

- N: número de estados del modelo
  - estados, $\boldsymbol{s} = \{s_1, s_2, \ldots, s_N\}$
  - estado en tiempo t, $q_t \in \boldsymbol{s}$
- M: número de símbolos de observación (ej., observaciones discretas)
  - símbolos de observación, $\boldsymbol{v} = \{v_1, v_2, \ldots, v_M\}$
  - observación en tiempo t, $o_t \in \boldsymbol{v}$
- $\boldsymbol{A} = \{a_{ij}\}$: distribución de la probabilidad de la transición del estado
  - $a_{ij} = P(q_{t+1} = s_j | q_t = s_i), \ 1 \leq i, j \leq N$
- $\boldsymbol{B} = \{b_j(k)\}$: distribución de la probabilidad del símbolo de observación del estado j
  - $b_j(k) = P(v_k \text{ at } t | q_t = s_j), \ 1 \leq j \leq N, 1 \leq k \leq M$
- $\pi = \{\pi_i\}$: distribución del estado inicial
  - $\pi_i = P(q_1 = s_i), \ 1 \leq i \leq N$

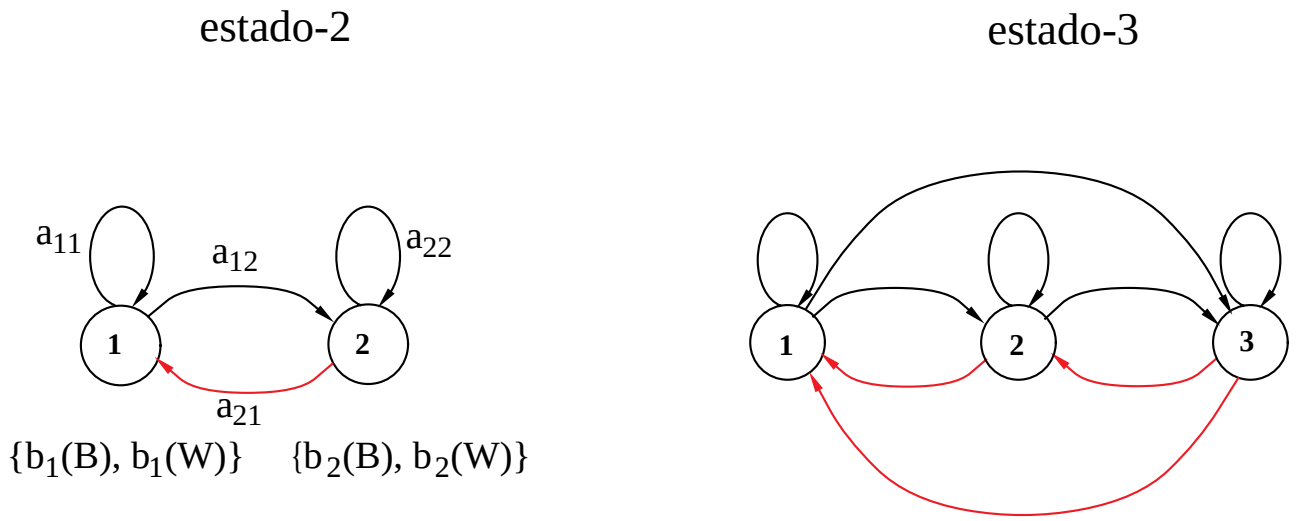
Desde una perspectiva notacional, un HMM se escribe típicamente como: $\lambda = \{\boldsymbol{A}, \boldsymbol{B}, \pi\}$

HMM: Un ejemplo (continuación)

Para nuestro ejemplo simple:

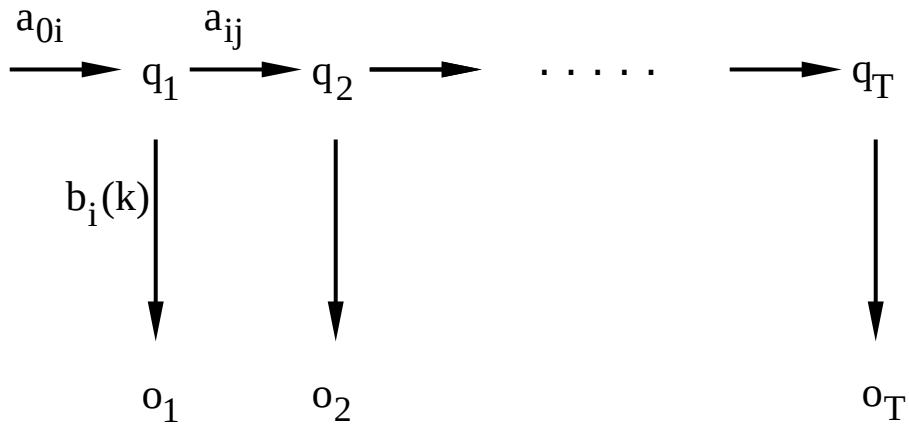
$$\pi = \{a_{01}, a_{02}\}, \quad \boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \text{ and } \quad \boldsymbol{B} = \begin{bmatrix} b_1(B) & b_1(W) \\ b_2(B) & b_2(W) \end{bmatrix}$$

Diagrama de estado

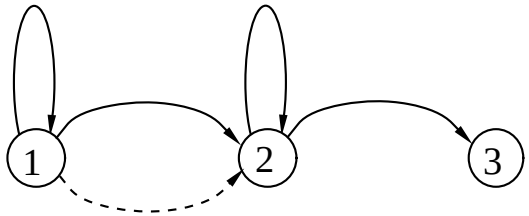


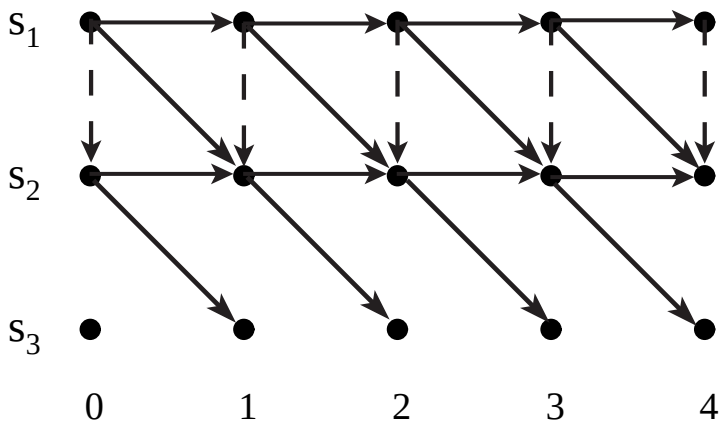
Generación de observaciones de HMM

1. Elegir un estado inicial $q_1 = s_i$, basado en la distribución del estado inicial, π
2. Para $t = 1$ a T:
   - Elegir $o_t = v_k$ en función de la distribución de probabilidad del símbolo en el estado s_i, $b_i(k)$
   - Transición a un nuevo estado $q_{t+1} = s_j$ según la distribución de probabilidad de la transición de estado para el estado s_i, a_{ij}
3. Incrementar en 1, volver al paso 2 si $t \leq T$; de lo contrario, terminar



Representación del diagrama de estado de Trellis





La línea con guiones representa una transición *nula*, en la que no se genera ningún símbolo de observación.

Tres problemas básicos de HMM

1. **Puntuación:** Dada una secuencia de observación $\boldsymbol{O} = \{o_1, o_2, ..., o_T\}$ y un modelo $\lambda = \{\boldsymbol{A}, \boldsymbol{B}, \pi\}$, ¿cómo calculamos $P(\boldsymbol{O} \mid \lambda)$, la probabilidad de la secuencia de observación?

   ==> Algoritmo de avance-retroceso

2. **Ajuste:** Dada una secuencia de observación $\boldsymbol{O} = \{o_1, o_2, ..., o_T\}$, ¿cómo elegimos una secuencia de estado $\boldsymbol{Q} = \{q_1, q_2, ..., q_T\}$ que de algún modo sea optimal?

   ==> Algoritmo de Viterbi

3. **Entrenamiento:** ¿Cómo ajustamos los parámetros del modelo $\lambda = \{A, B, \pi\}$ para maximizar $P(\boldsymbol{O} \mid \lambda)$?

   ==> Procedimientos de reestimación de Baum-Welch

Cómputo de $P(\boldsymbol{O}|\lambda)$

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

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

- Considere la secuencia del estado *fijo*: $\boldsymbol{Q} = q_1 q_2 \ldots q_T$

$$P(\boldsymbol{O}|\boldsymbol{Q}, \lambda) = b_{q_1}(o_1)b_{q_2}(o_2)\ldots b_{q_T}(o_T)$$

$$P(\boldsymbol{Q}|\lambda) = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} \ldots a_{q_{T-1} q_T}$$

  Por tanto:

$$P(\boldsymbol{O}|\lambda) = \sum_{q_1, q_2, \ldots, q_T} \pi_{q_1} b_{q_1}(o_1) a_{q_1 q_2} b_{q_2}(o_2) \ldots a_{q_{T-1} q_T} b_{q_T}(o_T)$$

- Cálculo requerido $\approx 2T \cdot NT$ (existen NT secuencias)
  Para $N = 5, T = 100 \Rightarrow 2 \cdot 100 \cdot 5100 \approx 1072$ cómputos

El algoritmo de avance

- Definamos la variable de avance $\alpha_t(i)$, como la probabilidad de la secuencia de observación parcial hasta el tiempo t y estado s_i en el tiempo t, dado el modelo, ej.:

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

- Se puede demostrar fácilmente que:

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

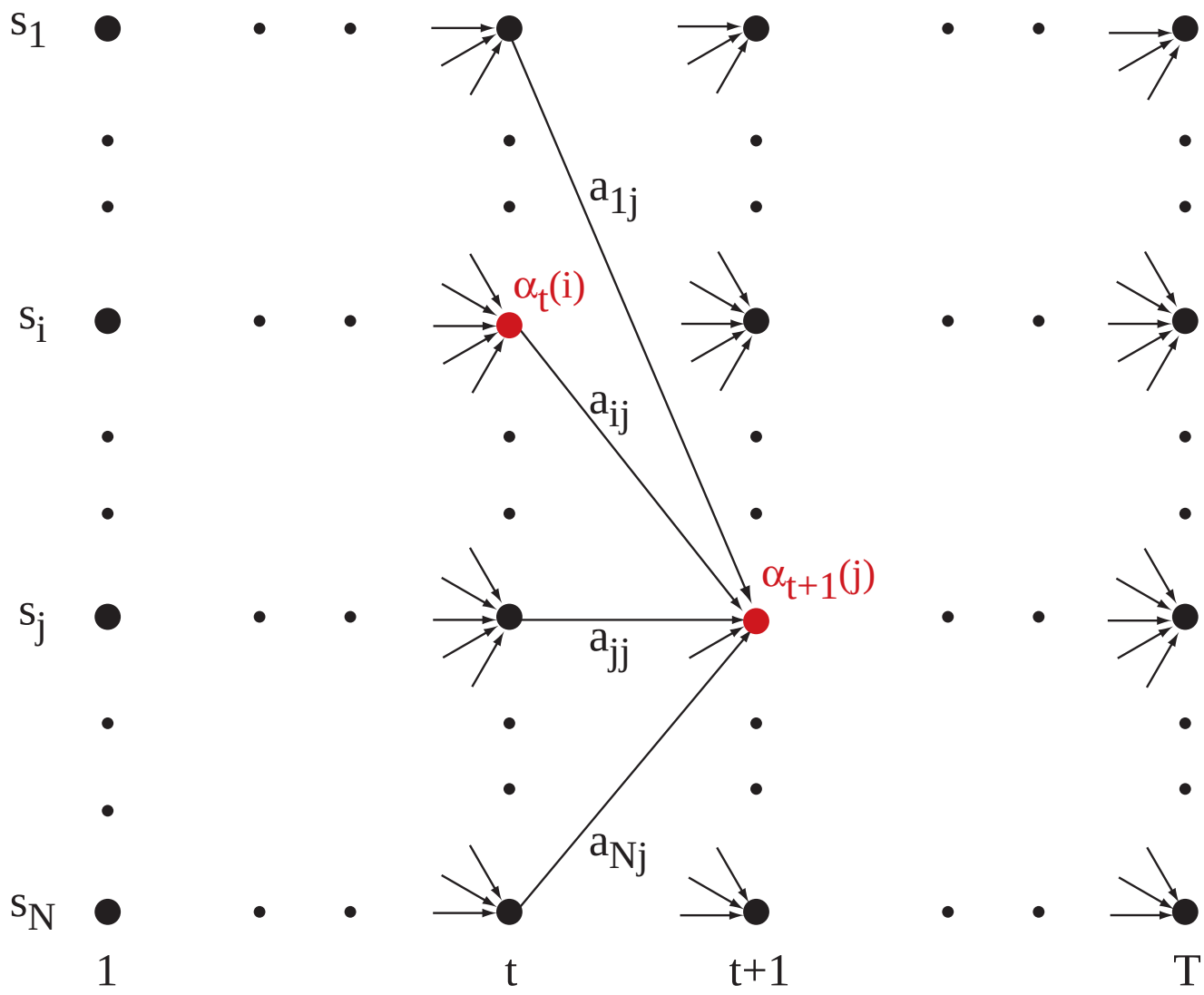
$$P(\boldsymbol{O}|\lambda) = \sum_{i=1}^{N} \alpha_T(i)$$

- Por inducción :

$$\alpha_{t+1}(j) = [\sum_{i=1}^{N} \alpha_t(i)\, a_{ij}] b_j(o_{t+1}), \qquad \begin{matrix} 1 \le t \le T-1 \\ 1 \le j \le N \end{matrix}$$

- El cálculo está en el orden de $N2T$.
  Para $N = 5, T = 100 \Rightarrow 100 \cdot 52$ cómputos, en vez de 1072

Ilustración del algoritmo de avance



El algoritmo de retroceso

- Del mismo modo, definamos la variable de retroceso $\beta_t(i)$, como la probabilidad de la secuencia de observación parcial desde el tiempo $t+1$ hasta el final, dado el estado s_i en tiempo t y el modelo, ej.,

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

- Puede demostrarse fácilmente que:

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

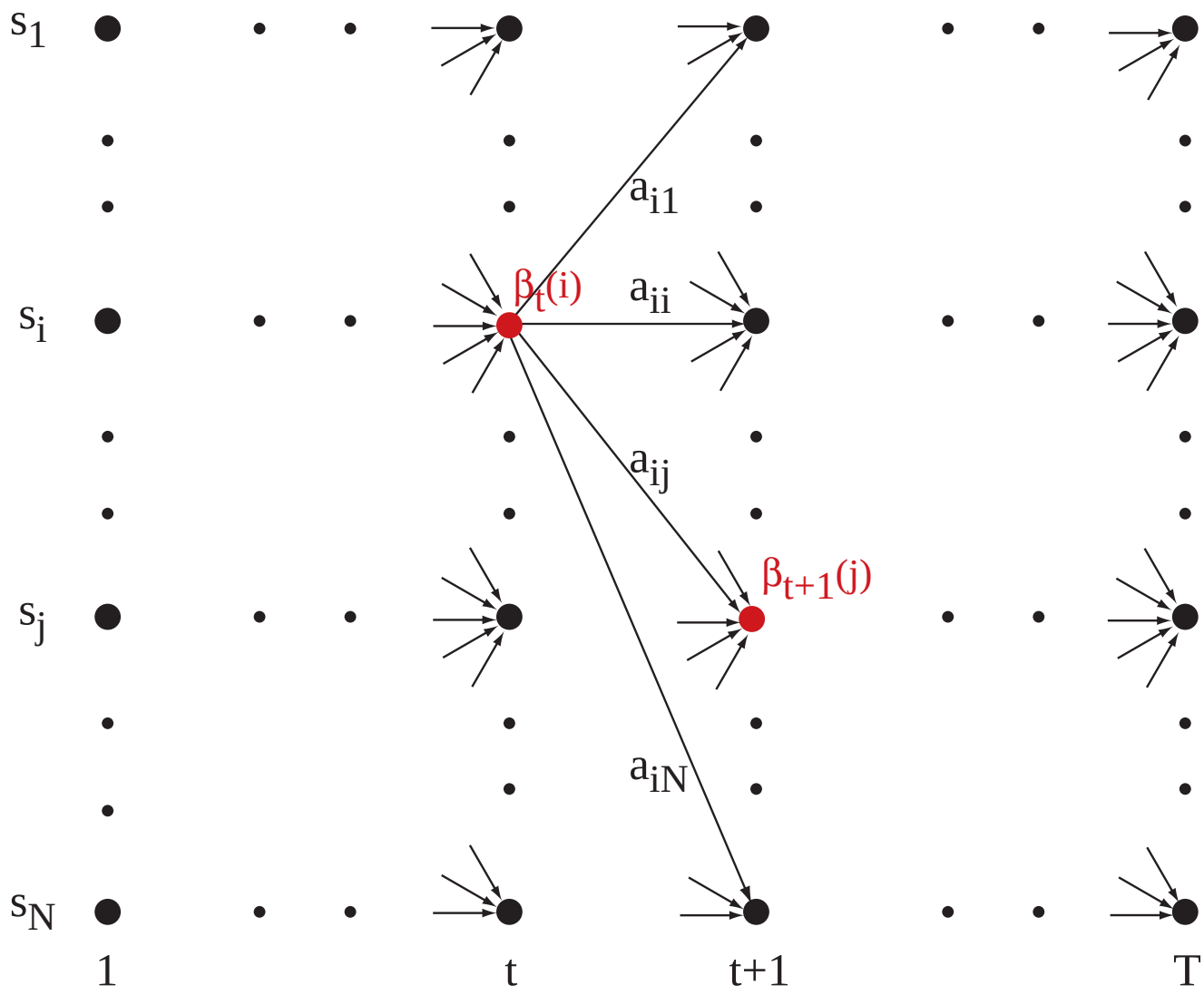
y:

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

- Por inducción:

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(o_{t+1}) \beta_{t+1}(j), \qquad \begin{array}{c} t = T-1, T-2, \ldots, 1 \\ 1 \le i \le N \end{array}$$

Ilustración del procedimiento de retroceso



Hallando secuencias optimales de estado

- Un criterio selecciona estados q_t, que son *individualmente* los más probables
  - Esto maximiza el número esperado de estados correctos
- Definamos $\gamma_t(i)$ como la probabilidad de estar en el estado si en el tiempo t, dada la secuencia de observación y el modelo, ej..

$$\gamma_t(i) = P(q_t = s_i|\boldsymbol{O}, \lambda) \qquad \sum_{i=1}^{N} \gamma_t(i) = 1, \qquad \forall t$$

- Luego el estado individualmente más probable q_t, en tiempo t es:

$$q_t = \underset{1 \le i \le N}{\operatorname{argmax}} \ \gamma_t(i) \qquad 1 \le t \le T$$

- Observe que se puede demostrar que:

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{P(\boldsymbol{O}|\lambda)}$$

Hallando secuencias optimales de estado

- El criterio de optimalidad individual presenta el problema de que la secuencia del estado optimal puede no obedecer a las restricciones de transición de estado.
- Otro criterio de optimalidad consiste en elegir la secuencia de estado que maximice $P(\boldsymbol{Q}, \boldsymbol{O}|\lambda)$; Esto se puede hallar mediante el algoritmo de *Viterbi* .
- Definamos $\delta_t(i)$ como la probabilidad más alta a lo largo de una trayectoria simple en tiempo t, que da cuenta de las primeras observaciones t , ej..

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

- Por inducción: $\delta_{t+1}(j) = [\max_i \delta_t(i) a_{ij}] b_j(o_{t+1})$
- Para recuperar la secuencia de estado, debemos seguir la pista de la secuencia de estado que proporcionó la mejor trayectoria, en tiempo t, al estaso s_i
  - Hacemos esto en una matriz aparte $\psi_t(i)$

El algoritmo de Viterbi

1. Inicialización:

$$\begin{aligned}\delta_1(i) &= \pi_i b_i(o_1), \qquad 1 \le i \le N \\ \psi_1(i) &= 0\end{aligned}$$

2. Recursión:

$$\begin{aligned}\delta_t(j) &= \max_{1\le i\le N}[\delta_{t-1}(i)a_{ij}]b_j(o_t), \qquad 2 \le t \le T \qquad 1 \le j \le N \\ \psi_t(j) &= \operatorname*{argmax}_{1\le i\le N}[\delta_{t-1}(i)a_{ij}], \qquad 2 \le t \le T \qquad 1 \le j \le N\end{aligned}$$

3. Terminación:

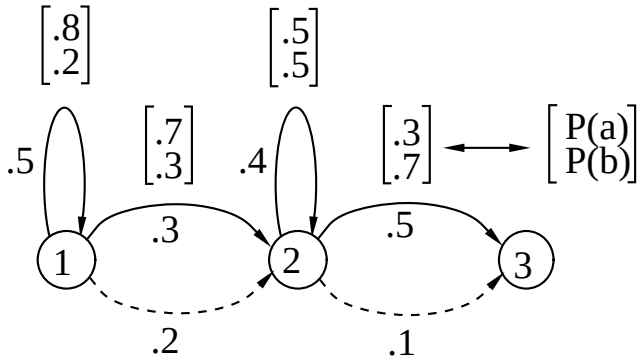
$$\begin{aligned}P^* &= \max_{1\le i\le N}[\delta_T(i)] \\ q_T^* &= \operatorname*{argmax}_{1\le i\le N}[\delta_T(i)]\end{aligned}$$

4. Trayectoria inversa (secuencia de estado):

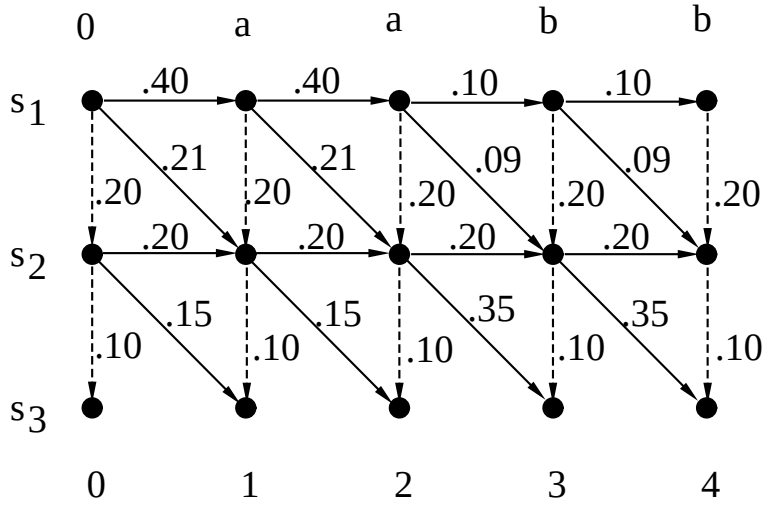
$$q_t^* = \psi_{t+1}(q_{t+1}^*), \qquad t = T-1, T-2, \ldots, 1$$

Cómputo $\approx N^2 T$

El algoritmo de Viterbi: Un ejemplo

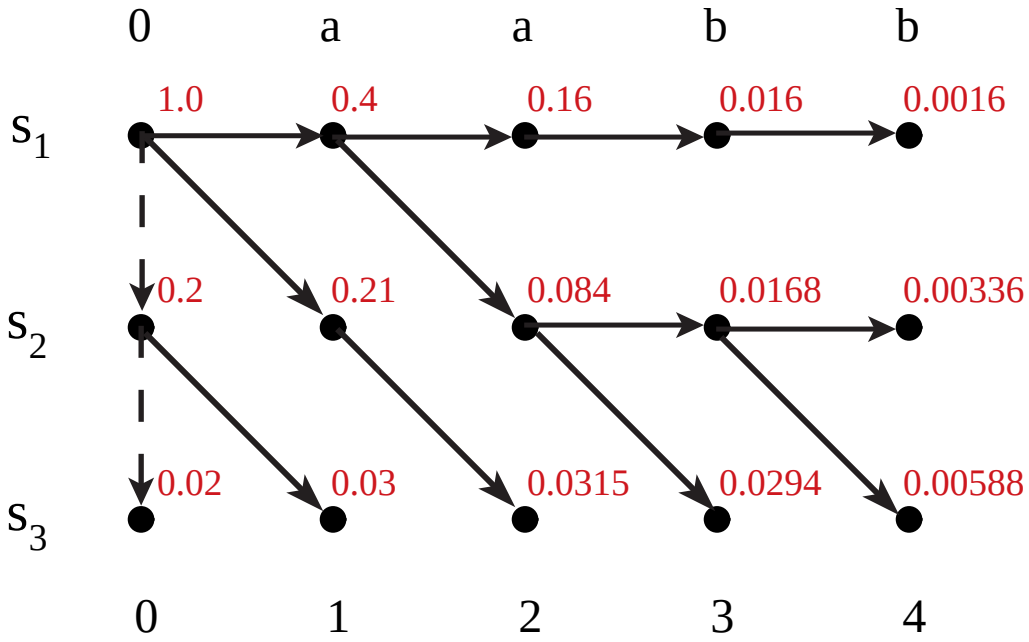


O={a a b b}



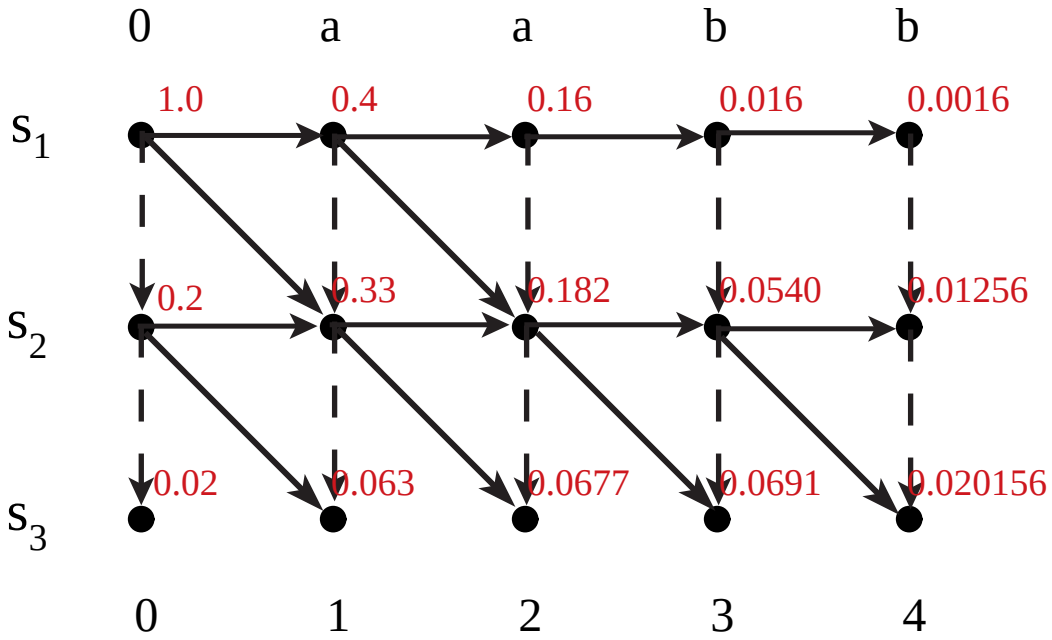
El algoritmo de Viterbi: Un ejemplo (continuación)

	0	*a*	*aa*	*aab*	*aabb*
s_1	1.0	s_1, a .4	s_1, a .16	s_1, b .016	s_1, b .0016
s_2	$s_1, 0$ .2	$s_1, 0$ .08 s_1, a .21 s_2, a .04	$s_1, 0$ .032 s_1, a .084 s_2, a .042	$s_1, 0$ .0032 s_1, b .0144 s_2, b .0168	$s_1, 0$ .00032 s_1, b .00144 s_2, b .00336
s_3	$s_2, 0$ .02	$s_2, 0$ .021 s_2, a .03	$s_2, 0$ .0084 s_2, a .0315	$s_2, 0$ .00168 s_2, b .0294	$s_2, 0$ .000336 s_2, b .00588



Correspondencia mediante el algoritmo de avance-retroceso

	0	a	aa	aab	$aabb$
s_1	1.0	s_1, a .4	s_1, a .16	s_1, b .016	s_1, b .0016
s_2	$s_1, 0$ .2	$s_1, 0$ .08 s_1, a .21 s_2, a .04	$s_1, 0$ .032 s_1, a .084 s_2, a .066	$s_1, 0$ .0032 s_1, b .0144 s_2, b .0364	$s_1, 0$ .00032 s_1, b .00144 s_2, b .0108
s_3	$s_2, 0$ .02	$s_2, 0$ .033 s_2, a .03	$s_2, 0$ .0182 s_2, a .0495	$s_2, 0$ .0054 s_2, b .0637	$s_2, 0$ .001256 s_2, b .0189



Reestimación de Baum-Welch

- La reestimación de Baum-Welch emplea el algoritmo de EM para determinar los parámetros de ML
- Definir $\xi_t(i,j)$ como la probabilidad de estar en el estado si en tiempo t y en el estado sj en tiempo $t+1$, dado el modelo y la secuencia de observación

$$\xi_t(i,j) = P(q_t = s_i, q_{t+1} = s_j | \boldsymbol{O}, \lambda)$$

- Luego:

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

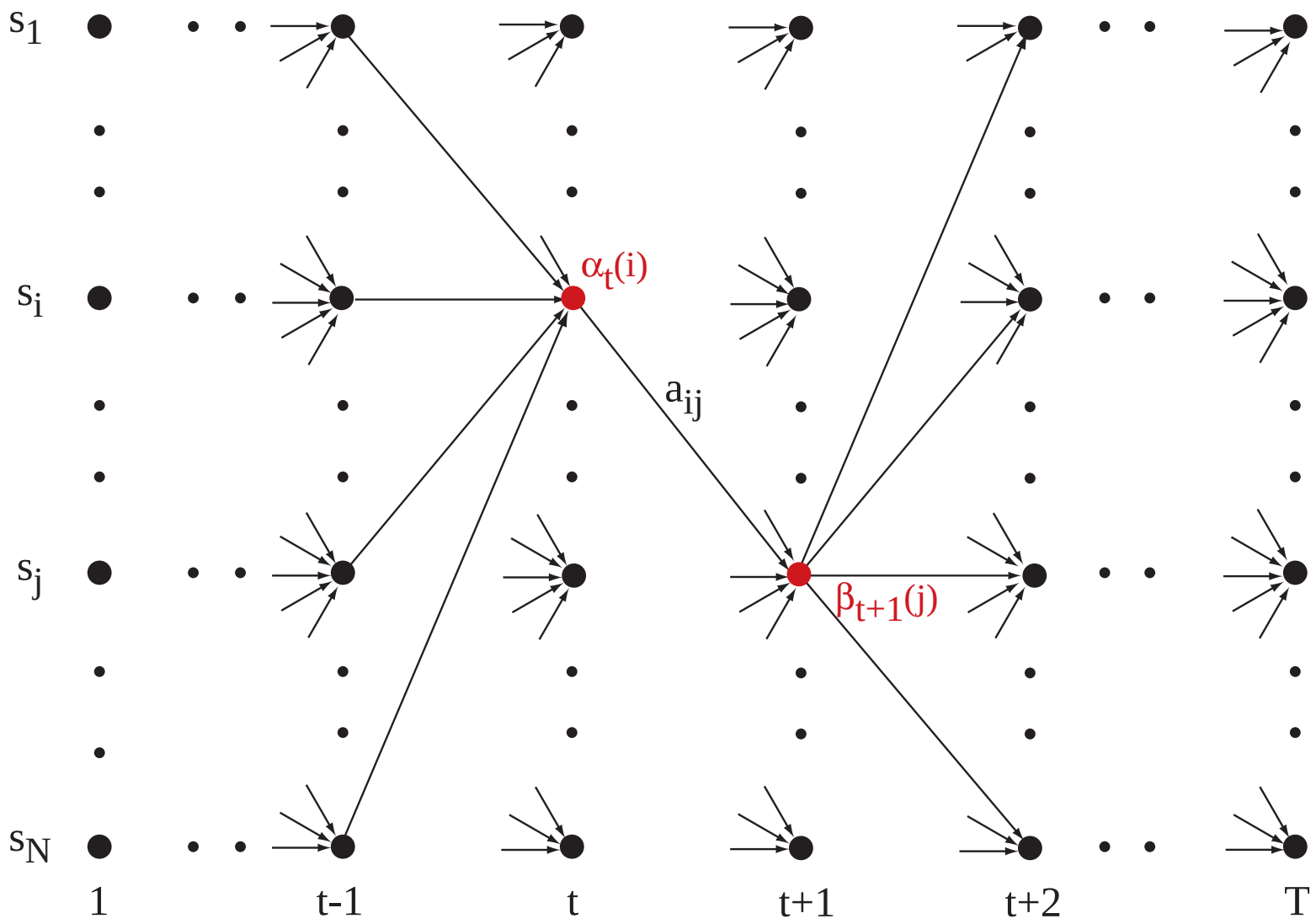
$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j)$$

- Sumando $\gamma_t(i)$ y $\xi_t(i,j)$, obtenemos:

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

$$\sum_{t=1}^{T-1} \xi_t(i,j) = \text{número esperado de transiciones desde } s_i \text{ a } s_j$$

Procedimientos de reestimación de Baum-Welch



Fórmulas de reestimación de Baum-Welch

$$\bar{\pi} = \text{número esperado de tiempos en el estado } s_i \text{ en } t = 1$$
$$= \gamma_1(i)$$

$$\bar{a}_{ij} = \frac{\text{número esperado de transiciones desde el estado si a sj}}{\text{número esperado de transiciones desde el estado } s_i}$$
$$= \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$$

$$\bar{b}_j(k) = \frac{\text{número esperado de tiempos en el estado } sj \text{ con símbolo } vk}{\text{número esperado de tiempos en el estado } sj}$$
$$= \frac{\sum_{t=1}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}$$
$$o_t = v_k$$

Fórmulas de reestimación de Baum-Welch

- Si $\lambda = (\boldsymbol{A}, \boldsymbol{B}, \pi)$ es el modelo inicial, y $\bar{\lambda} = (\bar{\boldsymbol{B}}, \bar{\boldsymbol{A}}, \bar{\pi})$ es el modelo reestimado. Se puede demostrar entonces que:

  1. El modelo inicial λ, define un punto crítico de la función de probabilidad, en cuyo caso $\bar{\lambda} = \lambda$, o
  2. El modelo $\bar{\lambda}$ es más probable que λ en el sentido de que $P(\boldsymbol{O}|\bar{\lambda}) > P(\boldsymbol{O}|\lambda)$, ej., hemos encontrado un nuevo modelo $\bar{\lambda}$, a partir del cual es más probable que la secuencia de observación se haya producido.

- Por tanto, podemos mejorar la probabilidad de que $\boldsymbol{O}$ sea observado a partir del modelo, si utilizamos iterativamente $\bar{\lambda}$ en lugar de λ y repetimos la reestimación hasta que se alcance algún punto restrictivo. El modelo resultante es conocido como el HMM con máxima probabilidad.

Secuencias de observación múltiple

- En reconocimiento de voz se utilizan normalmente los HMM de izquierda a derecha. Estos HMM no pueden entrenarse mediante una secuencia de observación simple, porque únicamente se encuentran disponibles un pequeño número de observaciones para entrenar a cada estado. Para obtener estimaciones fiables de parámetros del modelo, se deben emplear las secuencias de observación múltiples. En este caso, el procedimiento de reestimación debe ser modificado.

- Denotemos el conjunto de las secuencias de observación K como

$$\boldsymbol{O} = \{\boldsymbol{O}^{(1)}, \boldsymbol{O}^{(2)}, \ldots, \boldsymbol{O}^{(K)}\}$$

donde $\boldsymbol{O}^{(k)} = \{o^{(k)}_1, o^{(k)}_2 \ldots, o^{(k)}_{T_k}\}$ es la secuencia de observación k-th.

- Suponga que las secuencias de observación son mutuamente independientes, y que queremos calcular los parámetros con el fin de maximizar

$$P(\boldsymbol{O} \mid \lambda) = \prod_{k=1}^{K} P(\boldsymbol{O}^{(k)} \mid \lambda) = \prod_{k=1}^{K} P_k$$

Secuencias de observación múltiple (continuación)

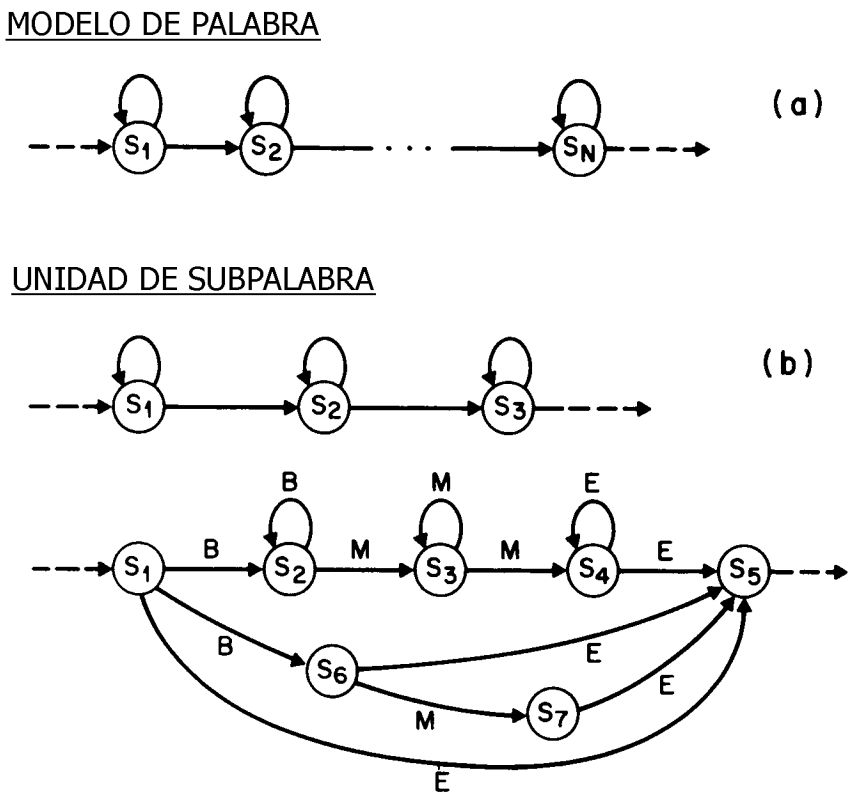
- Dado que las fórmulas de reestimación están basadas en la frecuencia de aparición de varios eventos, podemos modificarlos mediante la adición de frecuencias individuales de aparición para cada secuencia.

$$\bar{a}_{ij} = \frac{\sum_{k=1}^{K}\sum_{t=1}^{T_k-1}\xi_t^k(i,j)}{\sum_{k=1}^{K}\sum_{t=1}^{T_k-1}\gamma_t^k(i)} = \frac{\sum_{k=1}^{K}\frac{1}{P_k}\sum_{t=1}^{T_k-1}\alpha_t^k(i)a_{ij}b_j(o_{t+1}^{(k)})\beta_{t+1}^k(j)}{\sum_{k=1}^{K}\frac{1}{P_k}\sum_{t=1}^{T_k-1}\alpha_t^k(i)\beta_t^k(i)}$$

$$\bar{b}_j(\ell) = \frac{\sum_{k=1}^{K}\sum_{\substack{t=1\\ o_t^{(k)}=v_\ell}}^{T_k}\gamma_t^k(j)}{\sum_{k=1}^{K}\sum_{t=1}^{T_k}\gamma_t^k(j)} = \frac{\sum_{k=1}^{K}\frac{1}{P_k}\sum_{\substack{t=1\\ o_t^{(k)}=v_\ell}}^{T_k}\alpha_t^k(i)\beta_t^k(i)}{\sum_{k=1}^{K}\frac{1}{P_k}\sum_{t=1}^{T_k}\alpha_t^k(i)\beta_t^k(i)}$$

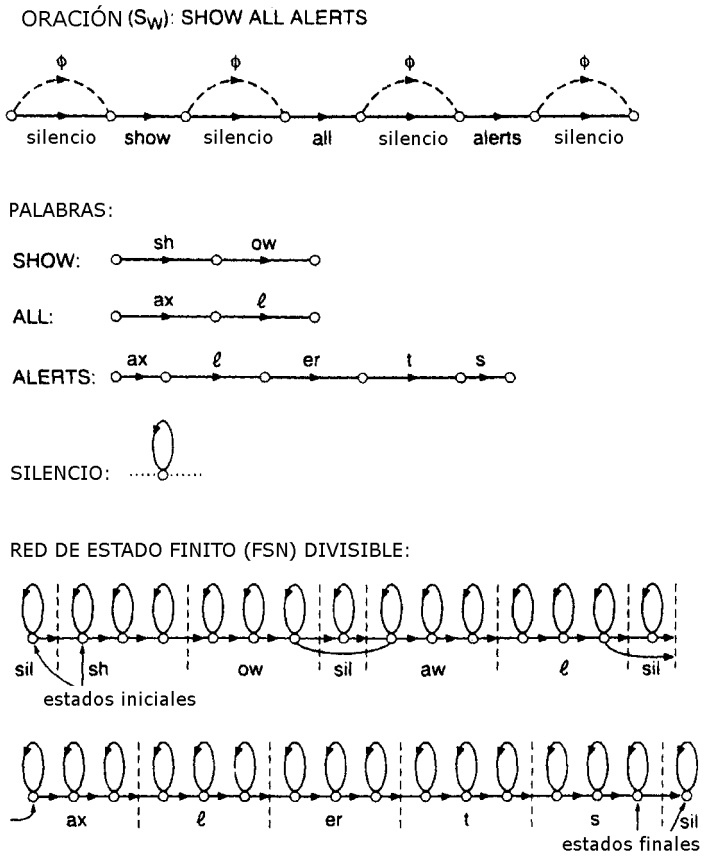
Los HMM en entornos telefónicos

- Los HMM basados en palabras son adecuados para el reconocmiento de pequeños vocabularios. Para el reconocimiento automático (ASR) de extensos vocabularios, los modelos basados en subpalabras (ej., entornos telefónicos) son más apropiados.



HMMs en entornos telefónicos (continuación)

- Los modelos telefónicos pueden presentar muchos estados, y las palabras se forman a partir de una concatenación de modelo telefónicos.



Modelos ocultos de Markov de densidad continua

- Un HMM de *densidad continua* sustituye las probabilidades de observación discreta $b_j(k)$, mediante un PDF continuo $b_j(\boldsymbol{x})$
- Una práctica común es representar $b_j(\boldsymbol{x})$ como una mezcla de gaussianas:

$$b_j(\boldsymbol{x}) = \sum_{k=1}^{M} c_{jk} N[\boldsymbol{x}, \mu_{jk}, \boldsymbol{\Sigma}_{jk}] \qquad 1 \le j \le N$$

donde c_{jk} es el peso de mezcla

$$c_{jk} \ge 0 \quad (1 \le j \le N, 1 \le k \le M, \text{ y } \sum_{k=1}^{M} c_{jk} = 1, 1 \le j \le N),$$

N es la densidad normal, y
μ_{jk} y $\boldsymbol{\Sigma}_{jk}$ son el vector de medias y la matriz de covarianza asociados con el estado j y la mezcla k.

Variaciones en el modelado acústico

- Los HMM *semicontinuos* computan primero un libro de código de VQ de tamaño M
  - El libro de código de VQ se modela entonces como una familia de PDF gaussianos.
  - Cada palabra e código está representada por un PDF gaussiano, y se puede utilizar junto con otros para modelar los vectores acústicos.
  - Desde la perspectiva de los CD-HMM (HMM de densidad continua), esto equivale a utilizar el mismo conjunto de mezclas M para modelar todos los estados.
  - Por tanto, se le conoce normalmente como un HMM de *mezcla enlazada.*
- Los tres métodos se han utilizado en muchas tareas de reconocimiento de voz, con diversos resultados.
- Para extensos vocabularios, en el reconocimiento de voz continuo con una cantidad suficiente (ej., decenas de horas) de entrenamiento de datos, los sistemas CD-HMM actualmente producen el mejor rendimiento, pero con un aumento considerable el el cómputo.

MIT

Cuestiones de implementación

- Escalamiento: para prevenir la generación de un valor inferior al mínimo aceptable.
- *Entrenamiento de K-medias segmentales*: para entrenar probabilidades de observación, ejecutando en primer lugar el alineamiento de Viterbi.
- Estimaciones iniciales de λ: para facilitar modelos robustos.
- Recorte: para reducir el cómputo de búsqueda.

Referencias

- X. Huang, A. Acero y H. Hon, *Spoken Language Processing*, Prentice-Hall, 2001.
- F. Jelinek, *Statistical Methods for Speech Recognition.* MIT Press, 1997.
- L. Rabiner y B. Juang, *Fundamentals of Speech Recognition*, Prentice-Hall, 1993.