Modelos Ocultos de Markov

José Luis Ruiz Reina
Franciso J. Martín Mateos
Carmen Graciani Díaz

Dpto. Ciencias de la Computación e Inteligencia Artificial
Universidad de Sevilla

Contenido

Tiempo e Incertidumbre

- En este tema vamos a ver razonamiento probabilístico en situaciones que discurren *a lo largo del tiempo* o en *secuencia*:
- Ejemplos:
  - Reconocimiento del habla
  - Movimiento de robots
  - Procesamiento de textos
  - Bioinformática
- Situaciones que transcurren en el tiempo:
  - El tiempo se considera en instantes discretos: $t = 1, 2, 3, \ldots$
  - Cada instante se modela mediante un conjunto de variables aleatorias, algunas *observables* y otras no.
  - Las relaciones entre dichas variables, así como las *transiciones* entre un instante y el siguiente, están afectadas de *incertidumbre*, que será modelada mediante distribuciones de probabilidad condicionada.

Cadenas de Markov

- Es el modelo más simple que veremos.
- Supongamos que la situación en cada instante t se describe mediante una única variable aleatoria X_t, que tienen un número finito de posibles valores que llamaremos *estados*: $s_1, s_2, \ldots, s_n$
  - En cada instante t, la variable X_t toma exactamente uno de sus posibles estados.
- Ejemplo:
  - Supongamos que el tiempo de cada día puede ser descrito como **`calor`**, **`frío`** o **`lluvia`**
  - Cadena de Markov: la v.a. X_t sería el tiempo del día t, y $\{$**`calor`**, **`frío`**, **`lluvia`**$\}$ el conjunto de posibles estados de la variable.

Propiedad de Markov

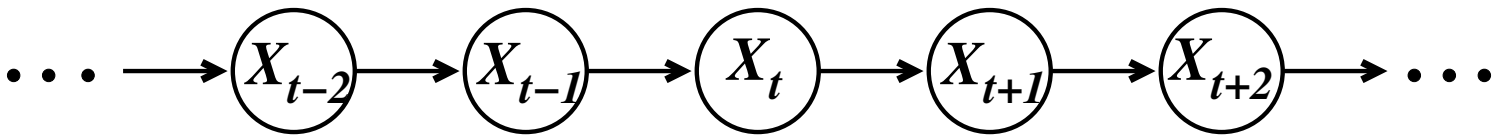
- Tenemos ahora que detallar cómo cambia, a lo largo del tiempo, el mundo que se describe.
- Asumiremos la siguiente propiedad, llamada *propiedad de Markov* (A. Markov, 1856-1922):

$$\boldsymbol{P}(X_t|X_1,\ldots,X_{t-1}) = \boldsymbol{P}(X_t|X_{t-1})$$

  - En una cadena de Markov, el estado actual sólo depende del estado inmediatamente anterior.
- Asumiremos también que la distribución $\boldsymbol{P}(X_t|X_{t-1})$ es independiente de t
  - Es decir, el modelo probabilístico que describe la manera de transitar entre un instante y el siguiente, no cambia con el tiempo
  - Es lo que se denomina un *proceso estacionario*

Cadena de Markov y redes bayesianas

- Vista como red bayesiana, una cadena de Markov tiene la siguiente estructura:



  - Cada nodo tiene una tabla de probabilidad correspondiente a $\boldsymbol{P}(X_t|X_{t-1})$
  - La misma tabla en todos los nodos
  - Excepto en el instante inicial X_1, cuya tabla es $\boldsymbol{P}(X_1)$
- Recordar: la estructura de la red implica que, dada la inmediatamente anterior, cada variables es independiente de todas las anteriores

Componentes de una cadena de Markov

- El conjunto de estados $s_1, \ldots, s_n$ (posibles valores de cada X_t)
- La tabla de probabilidad $\boldsymbol{P}(X_t|X_{t-1})$, dada por una *matriz de probabilidades de transición* $A = (a_{ij})$
  - Donde cada $a_{ij} = P(X_t = s_j|X_{t-1} = s_i)$
  - Es una matriz de tamaño $n \times n$, donde n es el número de posibles estados
  - Se cumple que $\sum_{1 \leq j \leq n} a_{ij} = 1$
- Además para el instante inicial, debemos especificar un vector $\pi = (\pi_i)$, donde $\pi_i = P(X_1 = s_i)$ (probabilidades iniciales para cada estado).

Ejemplo de cadena de Markov

- Estados: $\{\texttt{calor}, \texttt{frío}, \texttt{lluvia}\}$ (supondremos que $s_1 = \texttt{calor}$, $s_2 = \texttt{frío}$ y $s_3 = \texttt{lluvia}$)
- Matriz de probabilidades de transición:

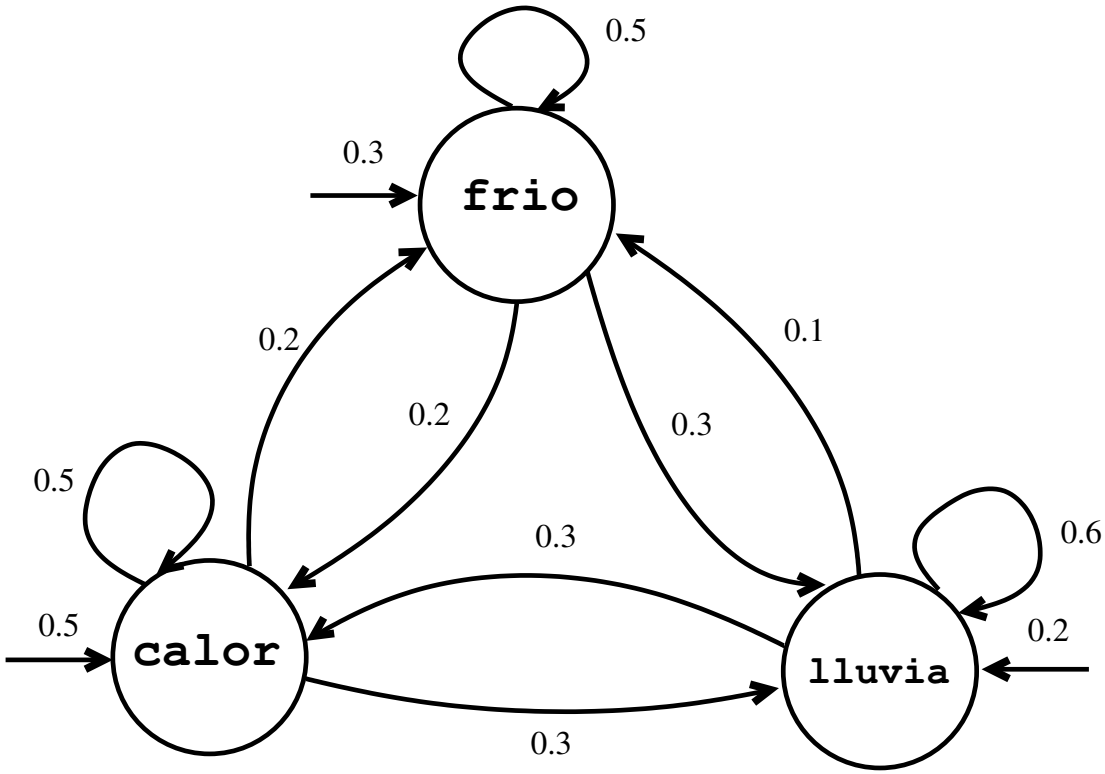
$$a_{11} = P(X_t = \texttt{calor} | X_{t-1} = \texttt{calor}) = 0.5$$
$$a_{12} = P(X_t = \texttt{frío} | X_{t-1} = \texttt{calor}) = 0.2$$
$$a_{13} = P(X_t = \texttt{lluvia} | X_{t-1} = \texttt{calor}) = 0.3$$
$$a_{21} = P(X_t = \texttt{calor} | X_{t-1} = \texttt{frío}) = 0.2$$
$$a_{22} = P(X_t = \texttt{frío} | X_{t-1} = \texttt{frío}) = 0.5$$
$$a_{23} = P(X_t = \texttt{lluvia} | X_{t-1} = \texttt{frío}) = 0.3$$
$$a_{31} = P(X_t = \texttt{calor} | X_{t-1} = \texttt{lluvia}) = 0.3$$
$$a_{32} = P(X_t = \texttt{frío} | X_{t-1} = \texttt{lluvia}) = 0.1$$
$$a_{33} = P(X_t = \texttt{lluvia} | X_{t-1} = \texttt{lluvia}) = 0.6$$

- Probabilidades iniciales:

$$\pi_1 = P(X_1 = \texttt{calor}) = 0.5$$
$$\pi_2 = P(X_1 = \texttt{frío}) = 0.3$$
$$\pi_3 = P(X_1 = \texttt{lluvia}) = 0.2$$

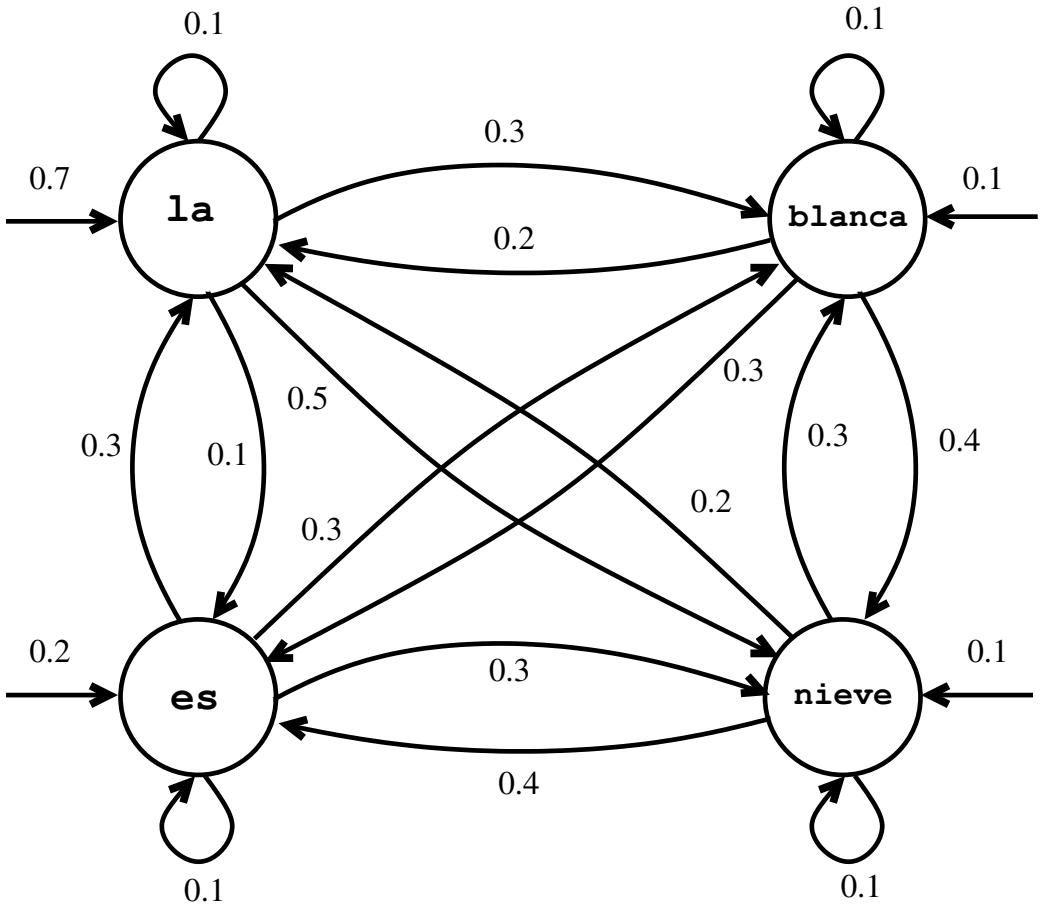
Representación gráfica de una cadena de Markov

- Es usual representar una cadena de Markov como un grafo dirigido
  - Los nodos son los estados
  - Los arcos están etiquetados con la probabilidad de pasar de un estado a otro



Otro ejemplo de cadena de Markov

- Secuencias formadas con las palabras **"la"**, **"nieve"**, **"es"** y **"blanca"**



Calculando probabilidades en una cadena de Markov

- Dos tipos de preguntas:
  - ¿Cuál es la probabilidad de que ocurra una determinada secuencia de estados $Q = q_1 \cdots q_t$? (es decir, $P(X_1 = q_1, X_2 = q_2, \ldots, X_t = q_t)$, abreviado como $P(q_1 q_2 \cdots q_t)$)
  - ¿Cuál es la probabilidad de que en el instante t se llegue al estado q? (Es decir, $P(X_t = q)$)

Probabilidad de una secuencia de estados

- ¿$P(q_1 q_2 \cdots q_t)$?
  - Aplicando la regla de la cadena, es igual a:

    $$P(X_t = q_t | X_1 = q_1, \ldots, X_{t-1} = q_{t-1}) \cdots \cdots P(X_2 = q_2 | X_1 = q_1) \cdot P(X_1 = q_1)$$

  - Por la propiedad de Markov, eso es igual a:

    $$P(X_t = q_t | X_{t-1} = q_{t-1}) \cdots P(X_2 = q_2 | X_1 = q_1) \cdot P(X_1 = q_1)$$

  - Luego $P(q_1 q_2 \cdots q_t) = \pi_{q_1} \cdot a_{q_1 q_2} \cdot a_{q_2 q_3} \cdots a_{q_{t-2} q_{t-1}} a_{q_{t-1} q_t}$
  - Es decir, la probabilidad de una secuencia es el producto de las correspondientes probabilidades de transitar de cada estado al siguiente.
- Ejemplos:
  - $P(\texttt{calor calor lluvia frío}) = 0.5 \cdot 0.5 \cdot 0.3 \cdot 0.1 = 0.0075$
  - $P(\texttt{la nieve es blanca}) = 0.7 \cdot 0.5 \cdot 0.4 \cdot 0.3 = 0.042$
  - $P(\texttt{blanca es nieve la}) = 0.1 \cdot 0.3 \cdot 0.3 \cdot 0.2 = 0.0018$

Probabilidad de llegar a un estado

- Notación: $P(X_t = q) = p_t(q)$ ¿Cómo calculamos $p_t(q)$?
- Una primera idea: $p_t(q) = P(X_t = q) = \sum_Q P(Q)$, donde tenemos un sumando por cada secuencia $Q = q_1 \cdots q_t$ que acaba en el estado q ($q_t = q$).
  - Por ejemplo, $p_3(\texttt{calor}) = P(\texttt{calor calor calor}) + P(\texttt{calor frío calor}) + P(\texttt{calor lluvia calor}) + \cdots + P(\texttt{lluvia frío calor}) + P(\texttt{lluvia lluvia calor})$
  - Ineficiente: n^{t-1} sumandos
- Una alternativa más eficiente: calcular recursivamente los valores $p_t(s_i)$ para todos los estados $1 \leq i \leq n$
  - $p_1(s_i) = P(X_1 = s_i) = \pi_i, 1 \leq i \leq n$
  - $p_t(s_j) = P(X_t = s_j) = \sum_{1 \leq i \leq n} P(X_t = s_j, X_{t-1} = s_i) =$
    $$\sum_{1 \leq i \leq n} (P(X_t = s_j | X_{t-1} = s_i) \cdot P(X_{t-1} = s_i)) = \sum_{1 \leq i \leq n} (a_{ij} \cdot p_{t-1}(s_i))$$
  - Complejidad: $O(t \cdot n^2)$

Ejemplo de cálculo de probabilidad de llegar a un estado

- $p_1(\texttt{calor}) = 0.5$
- $p_1(\texttt{frío}) = 0.3$
- $p_1(\texttt{lluvia}) = 0.2$
- $p_2(\texttt{calor}) = 0.5 \cdot p_1(\texttt{calor}) + 0.2 \cdot p_1(\texttt{frío}) + 0.3 \cdot p_1(\texttt{lluvia}) = 0.37$
- $p_2(\texttt{frío}) = 0.2 \cdot p_1(\texttt{calor}) + 0.5 \cdot p_1(\texttt{frío}) + 0.1 \cdot p_1(\texttt{lluvia}) = 0.27$
- $p_2(\texttt{lluvia}) = 0.3 \cdot p_1(\texttt{calor}) + 0.3 \cdot p_1(\texttt{frío}) + 0.6 \cdot p_1(\texttt{lluvia}) = 0.36$
- $p_3(\texttt{calor}) = 0.5 \cdot p_2(\texttt{calor}) + 0.2 \cdot p_2(\texttt{frío}) + 0.3 \cdot p_2(\texttt{lluvia}) = 0.347$
- $\ldots$

Modelos ocultos de Markov

- Muchas situaciones reales no pueden modelarse mediante cadenas de Markov
  - Problema: el valor de X_t (el estado) no es observable directamente
  - Lo que podemos observar depende del estado, aunque con cierta incertidumbre
- Ejemplos:
  - Etiquetado léxico
  - Movimiento de robots
  - Reconocimiento del habla
  - Bioinformática

Modelos ocultos de Markov: propiedades asumidas

- En un modelo oculto de Markov, para cada instante t tenemos:
  - una variable aleatoria X_t, con posibles valores $\{s_1, \ldots, s_n\}$ (estados, no observable directamente)
  - otra variable aleatoria E_t, con posibles valores $\{v_1, \ldots, v_m\}$ (percepciones, observable directamente)
- Propiedades asumidas:
  - *Propiedad de Markov*:

$$\boldsymbol{P}(X_t|X_1, X_2, \ldots, X_{t-1}) = \boldsymbol{P}(X_t|X_{t-1})$$

  - *Independencia de las percepciones:*

$$\boldsymbol{P}(E_t|X_1, \ldots, X_t, E_1, \ldots, E_{t-1}) = \boldsymbol{P}(E_t|X_t)$$

- En otras palabras:
  - En cada instante, el estado depende sólo del estado en el instante inmediatamente anterior
  - En cada instante, lo observado depende sólo del estado en ese instante

Ejemplo 1 de modelo oculto de Markov (J. Eisner)

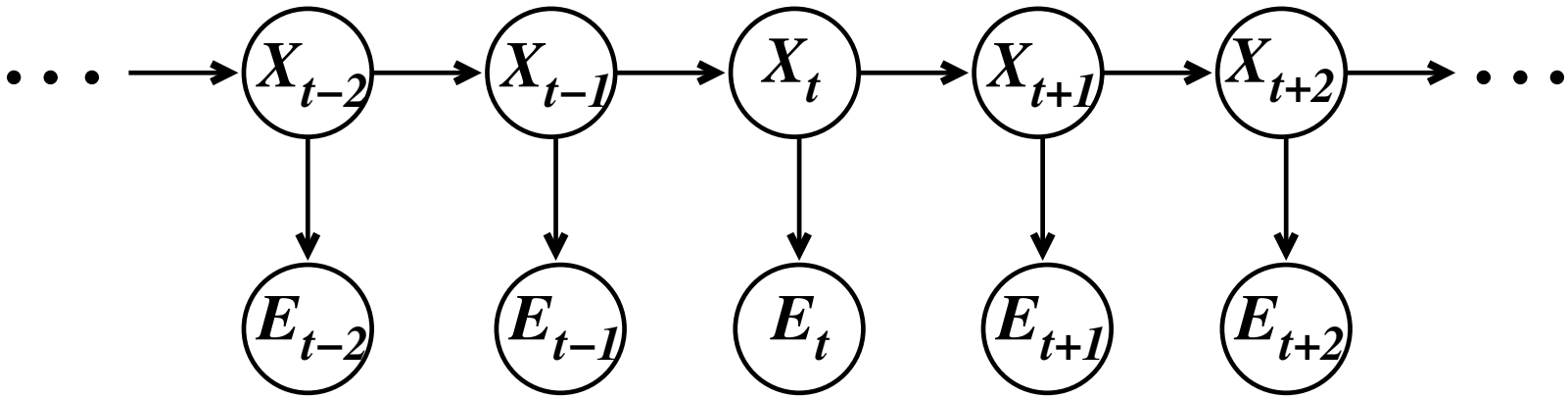
- Supongamos que en el año 2534, un científico quiere estudiar el tiempo que hubo en el año 2020, pero no es capaz de encontrar datos (por simplificar supondremos sólo dos tipos de días: *c* (calor) y *f* (frío)). Pero ha encontrado un diario de la época, en el que una persona describe cuántos helados tomó cada día de 2020 (1,2 ó 3).
  - Estados: variable X_t que indica el tiempo en el día t (posibles estados *c* y *f*)
  - Percepciones: variable E_t, número de helados comidos el día t (1, 2 ó 3).
- En este ejemplo, es razonable asumir que se cumple la propiedad de Markov y la de la independencia de las percepciones:
  - El tiempo que haga un día depende sólo del tiempo del día anterior.
  - La cantidad de helados comidos sólo depende (probabilísticamente) del tiempo en *ese mismo* día

Ejemplo 2 (Russell & Norvig)

- Un guardia de seguridad trabaja en unas oficinas subterráneas, sin conexión con el exterior. Cada día, no puede saber si está lloviendo o no, pero ve llegar al director de la oficina, que puede o no llevar paraguas
  - Estados: variable X_t que indica si llueve o no en el día t (con posibles estados l y $\neg l$)
  - Percepciones: variable E_t, que indica si el director lleva o no paraguas (con posibles valores u y $\neg u$)
- Nuevamente, es razonable asumir en este caso las propiedades de un modelo oculto de Markov

Modelos ocultos de Markov y redes bayesianas

- Vista como red bayesiana, una modelo oculto de Markov tiene la siguiente estructura:



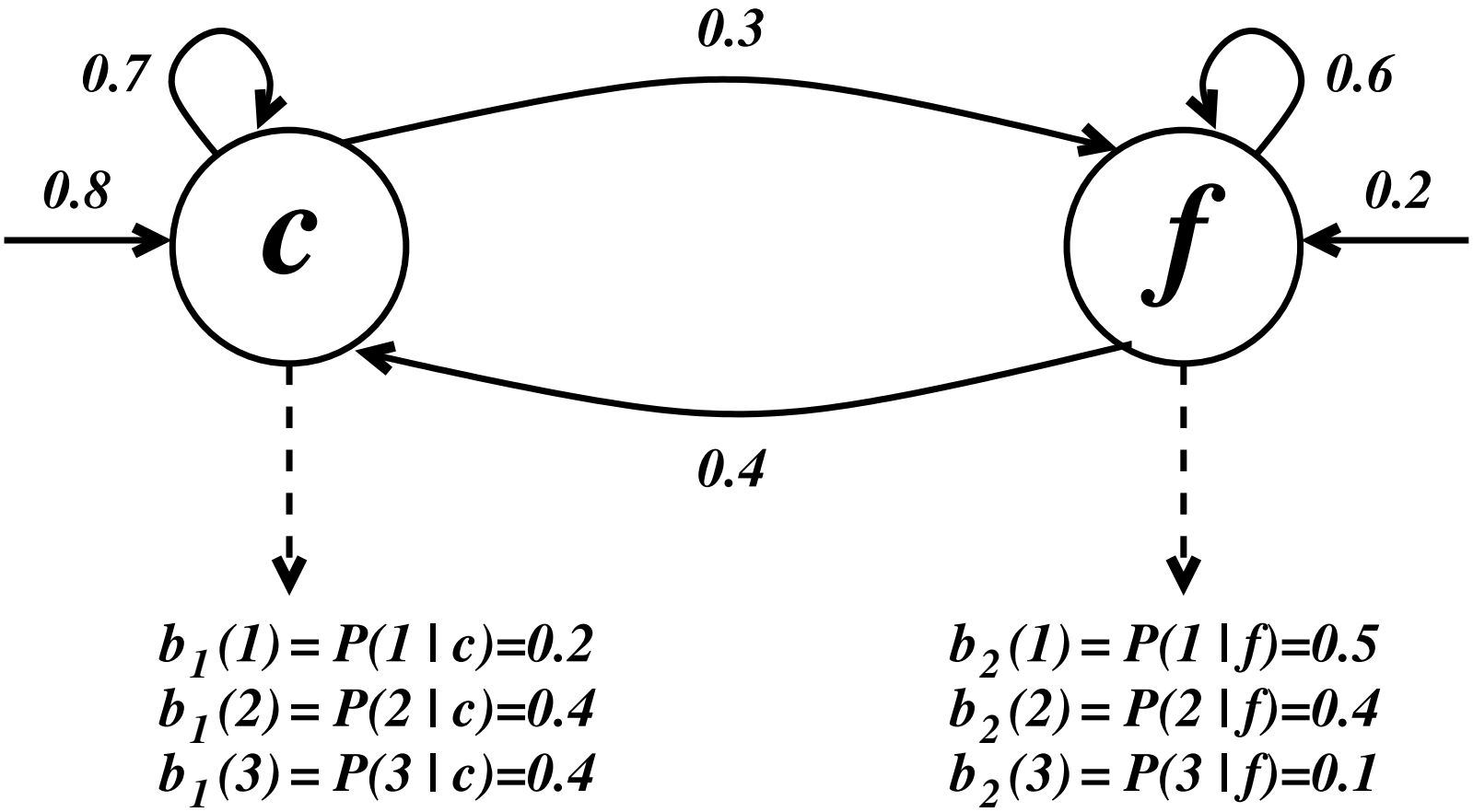
  - Cada nodo X_t tiene una tabla de probabilidad correspondiente a $\boldsymbol{P}(X_t|X_{t-1})$ (la misma tabla en todos los nodos)
  - Excepto en el instante inicial X_1, cuya tabla es $\boldsymbol{P}(X_1)$
  - Cada nodo E_t tiene una tabla de probabilidad correspondiente a $\boldsymbol{P}(E_t|X_t)$ (la misma tabla en todos los nodos)

Componentes de un modelo oculto de Markov

- Por un lado, la parte correspondiente a la cadena de Markov de los estados (*modelo de transición*):
  - Para cada $t > 0$, una variable X_t con posibles valores $s_1, \ldots, s_n$ (estados)
  - La tabla de probabilidad $\boldsymbol{P}(X_t|X_{t-1})$, dada por una matriz $A = (a_{ij})$, donde $a_{ij} = P(X_t = s_j|X_{t-1} = s_i)$ (probabilidad de pasar de s_i a s_j)
  - Vector $\pi = (\pi_i)$, donde $\pi_i = P(X_1 = s_i)$ (probabilidades *a priori* para cada estado).
- Por otro lado, la parte correspondiente a las percepciones (*modelo sensor*):
  - Para cada $t > 0$, una variable aleatoria E_t, con posibles valores $\{v_1, \ldots, v_m\}$
  - La tabla de probabilidad $\boldsymbol{P}(E_t|X_t)$, dada por una matriz de probabilidades de percepción $B = b_i(v_j)$, donde $b_i(v_j) = P(E_t = v_j|X_t = s_i)$ (probabilidad de observar v_j cuando el estado es s_i)

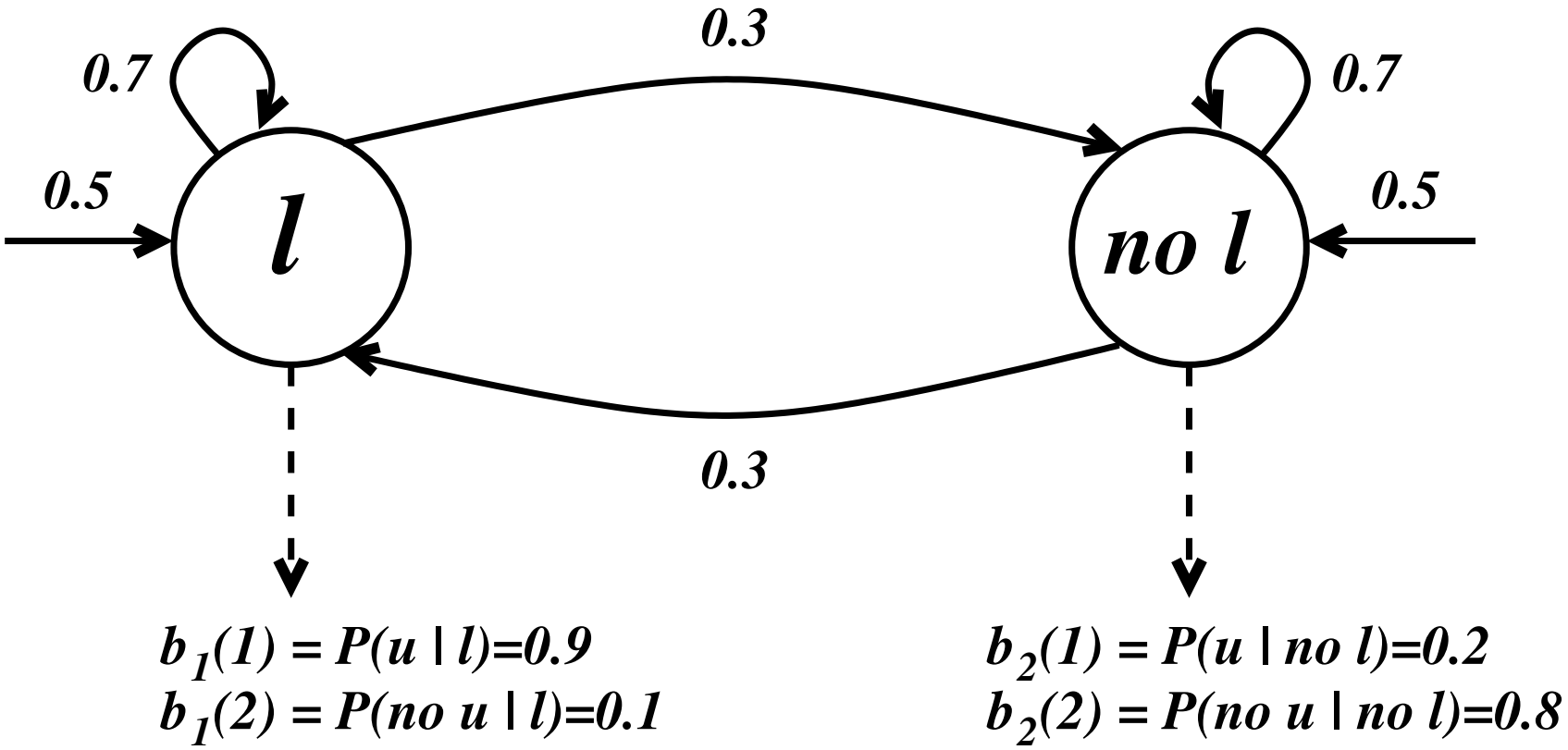
Representación gráfica, ejemplo 1

Suponiendo $s_1 = c$ y $s_2 = f$



Representación gráfica, ejemplo 2

Suponiendo $s_1 = l$ y $s_2 = no\ l$



Ejemplos de aplicaciones reales

- Reconocimiento del habla
  - Estados: las distintas sílabas
  - Percepciones: fonemas (con posible *ruido*)
- Localización de robots
  - Estados: posibles posiciones de un robot
  - Percepciones: datos de proximidad obtenidos de los sensores

Problemas principales a resolver en MOMs

- **Filtrado:** dada una secuencia de percepciones $o_1, \ldots, o_t$ y un estado q, ¿cuál es la probabilidad de que $X_t = q$?
  - Ejemplo: si sabemos que la secuencia de helados comidos hasta el cuarto día es $3, 1, 3, 2$ ¿cuál es la probabilidad de el cuarto día sea un día frío?
- **Explicación más verosímil** (o **Decodificación**): habiendo observado una secuencia $o_1, \ldots, o_t$ ¿cuál es la correspondiente secuencia de estados más probable?
  - Ejemplo: si sabemos que la secuencia de helados comidos hasta el sexto día es $1, 1, 2, 2, 3, 3$ ¿cuál es la secuencia más probable del tiempo en los seis primeros días?
- **Aprendizaje:** habiendo observado una secuencia $o_1, \ldots, o_t$ y conociendo los estados del modelo, pero no las matrices de transición y de percepciones, aprender dichas probabilidades

Filtrado

- Supongamos un modelo oculto de Markov:
  - Con variables aleatorias X_t (estado) y E_t (percepción)
  - Con n estados $\{s_1, \ldots, s_n\}$ y m posibles percepciones $\{v_1, \ldots, v_m\}$
  - Con matriz de transición A y matriz de percepción B
- Supongamos que tenemos la secuencia $O = o_1 o_2 \cdots o_t$, percepciones hasta el instante t
- **Filtrado**: dado un estado q, queremos calcular la probabilidad de que habiendo observado la secuencia O hasta el instante t, el estado del sistema en ese último instante sea q
- Es decir, queremos calcular $P(X_t = q | o_1 o_2 \cdots o_t)$

Filtrado: una primera simplificación

- Téngase en cuenta que:

$$\boldsymbol{P}(X_t|o_1 o_2 \cdots o_t) = \alpha \cdot \boldsymbol{P}(X_t, o_1 o_2 \cdots o_t)$$

- Luego bastará calcular $P(X_t = s_i, o_1 o_2 \cdots o_t)$ para todos los estados s_i $(1 \leq i \leq n)$, y luego normalizar
- Es decir, la probabilidad de que hasta el instante t hayamos observado la secuencia O **y además** el estado en el instante t sea s_i

Una manera ineficiente de hacer filtrado

- Una forma de calcular $P(X_t = q, O)$ es hacer $\sum_Q P(Q, O)$, donde tenemos un sumando por cada secuencia $Q = q_1 \cdots q_t$ que acaba en el estado q ($q_t = q$).
  - Supongamos dada una secuencia de estados $Q = q_1 \cdots q_t$ y una secuencia de percepciones $O = o_1 \cdots o_t$
  - Entonces (*¿por qué?*):

$$P(Q, O) = \pi_{q_1} \cdot a_{q_1 q_2} \cdots a_{q_{t-1} q_t} \cdot b_{q_1}(o_1) \cdots b_{q_t}(o_t)$$

- Ineficiente: ¡¡ n^{t-1} sumandos !!
- Alternativa eficiente (similar a lo que se hace en cadenas de Markov):
  - Calcular $P(X_t = s_i, o_1 \cdots o_t)$, de manera recursiva, para todos los estados $1 \leq i \leq n$

Cálculo recursivo para filtrado

- Si llamamos $\alpha_t(s_j)$ a $P(X_t = s_j, o_1 \cdots o_t)$, tenemos:
  - Inicio:
    $\alpha_1(s_i) = P(X_1 = s_i, o_1) = P(o_1|X_1 = s_i) \cdot P(X_1 = s_i) = b_i(o_1) \cdot \pi_i$
  - Recursión:

$$
\begin{aligned}
\alpha_t(s_j) &= P(X_t = s_j, o_1 \cdots o_t) \\
&= P(o_t|X_t = s_j, o_1 \cdots o_{t-1}) \cdot P(X_t = s_j, o_1 \cdots o_{t-1}) \\
&= P(o_t|X_t = s_j) \cdot \sum_{1 \le i \le n} P(X_t = s_j, X_{t-1} = s_i, o_1 \cdots o_{t-1}) \\
&= P(o_t|X_t = s_j) \cdot \sum_{1 \le i \le n} (P(X_t = s_j|X_{t-1} = s_i, o_1 \cdots o_{t-1}) \cdot \\
&\qquad P(X_{t-1} = s_i, o_1 \cdots o_{t-1})) \\
&= P(o_t|X_t = s_j) \cdot \sum_{1 \le i \le n} (P(X_t = s_j|X_{t-1} = s_i) \cdot \\
&\qquad P(X_{t-1} = s_i, o_1 \cdots o_{t-1})) \\
&= b_j(o_t) \sum_{1 \le i \le n} (a_{ij} \cdot \alpha_{t-1}(s_i))
\end{aligned}
$$

Algoritmo de "avance"(*forward*)

- **Entrada:**
  - Un modelo oculto de Markov
  - Una secuencia de percepciones $O = o_1 \cdots o_t$
- **Salida:**
  - Valores $\alpha_t(s_i) = P(X_t = s_j, o_1 \cdots o_t)$, para cada estado s_i
- **Procedimiento:**
  - Inicio: $\alpha_1(s_i) = b_i(o_1) \cdot \pi_i$, para $1 \leq i \leq n$
  - Para k desde 2 a t:
    - Para j desde 1 a n:
      - Hacer $\alpha_k(s_j) = b_j(o_k) \sum_{1 \leq i \leq n} (a_{ij} \cdot \alpha_{k-1}(s_i))$
  - Devolver los $\alpha_t(s_i)$, para $1 \leq i \leq n$

Nótese que la recursión es similar a la que se ha visto para cadenas de Markov, pero en este caso se "actualiza" con la probabilidad de la percepción, $b_j(o_k)$

Algoritmo de avance para el ejemplo 1

- Para el ejemplo 1, supongamos que hasta el cuarto día la secuencia de percepciones es $O = (3, 1, 3, 2)$ (es decir $o_1 = 3$, $o_2 = 1$, $o_3 = 3$ y $o_4 = 2$). Calculemos $\alpha_4(s_1)$ y $\alpha_4(s_2)$ (donde $s_1 = c$ y $s_2 = f$)
  - Paso 1:

$$\alpha_1(s_1) = b_1(o_1) \cdot \pi_1 = 0.4 \cdot 0.8 = 0.32$$
$$\alpha_1(s_2) = b_2(o_1) \cdot \pi_2 = 0.1 \cdot 0.2 = 0.02$$

  - Paso 2:

$$\begin{aligned}\alpha_2(s_1) &= b_1(o_2) \cdot [a_{11} \cdot \alpha_1(s_1) + a_{21} \cdot \alpha_1(s_2)] \\ &= 0.2 \cdot [0.7 \cdot 0.32 + 0.4 \cdot 0.02] = 0.0464\end{aligned}$$
$$\begin{aligned}\alpha_2(s_2) &= b_2(o_2) \cdot [a_{12} \cdot \alpha_1(s_1) + a_{22} \cdot \alpha_1(s_2)] \\ &= 0.5 \cdot [0.3 \cdot 0.32 + 0.6 \cdot 0.02] = 0.054\end{aligned}$$

Algoritmo de avance para el ejemplo 1

- Para el ejemplo 1, supongamos que hasta el cuarto día la secuencia de percepciones es $O = (3, 1, 3, 2)$ (es decir $o_1 = 3$, $o_2 = 1$, $o_3 = 3$ y $o_4 = 2$). Calculemos $\alpha_4(s_1)$ y $\alpha_4(s_2)$ (donde $s_1 = c$ y $s_2 = f$)
  - Paso 3:

$$\begin{aligned}\alpha_3(s_1) &= b_1(o_3) \cdot [a_{11} \cdot \alpha_2(s_1) + a_{21} \cdot \alpha_2(s_2)] \\ &= 0.4 \cdot [0.7 \cdot 0.0464 + 0.4 \cdot 0.054] = 0.021632\end{aligned}$$

$$\begin{aligned}\alpha_3(s_2) &= b_2(o_3) \cdot [a_{12} \cdot \alpha_2(s_1) + a_{22} \cdot \alpha_2(s_2)] \\ &= 0.1 \cdot [0.3 \cdot 0.0464 + 0.6 \cdot 0.054] = 0.004632\end{aligned}$$

  - Paso 4:

$$\begin{aligned}\alpha_4(s_1) &= b_1(o_4) \cdot [a_{11} \cdot \alpha_3(s_1) + a_{21} \cdot \alpha_3(s_2)] \\ &= 0.4 \cdot [0.7 \cdot 0.021632 + 0.4 \cdot 0.004632] = 0.00679808\end{aligned}$$

$$\begin{aligned}\alpha_4(s_2) &= b_2(o_4) \cdot [a_{12} \cdot \alpha_3(s_1) + a_{22} \cdot \alpha_3(s_2)] \\ &= 0.4 \cdot [0.3 \cdot 0.021632 + 0.6 \cdot 0.004632] = 0.00370752\end{aligned}$$

Algoritmo de avance para el ejemplo 2

- Para el ejemplo 2, supongamos que los dos primeros días observamos paraguas y el tercero no (es decir, $o_1 = u$, $o_2 = u$ y $o_3 = \neg u$). Calculemos la probabilidad de lluvia en el tercer día (los estados son $s_1 = l$ y que $s_2 = \neg l$)
  - Paso 1:

$$\alpha_1(s_1) = b_1(o_1) \cdot \pi_1 = 0.9 \cdot 0.5 = 0.45$$
$$\alpha_1(s_2) = b_2(o_1) \cdot \pi_2 = 0.2 \cdot 0.5 = 0.1$$

  - Paso 2:

$$\begin{aligned}\alpha_2(s_1) &= b_1(o_2) \cdot [a_{11} \cdot \alpha_1(s_1) + a_{21} \cdot \alpha_1(s_2)] \\ &= 0.9 \cdot [0.7 \cdot 0.45 + 0.3 \cdot 0.1] = 0.3105 \\ \alpha_2(s_2) &= b_2(o_2) \cdot [a_{12} \cdot \alpha_1(s_1) + a_{22} \cdot \alpha_1(s_2)] \\ &= 0.2 \cdot [0.3 \cdot 0.45 + 0.7 \cdot 0.1] = 0.0410\end{aligned}$$

Algoritmo de avance para el ejemplo 2

- Para el ejemplo 2, supongamos que los dos primeros días observamos paraguas y el tercero no (es decir, $o_1 = u$, $o_2 = u$ y $o_3 = \neg u$). Calculemos la probabilidad de lluvia en el tercer día (los estados son $s_1 = l$ y que $s_2 = \neg l$)
  - Paso 3:

$$\begin{aligned}\alpha_3(s_1) &= b_1(o_3) \cdot [a_{11} \cdot \alpha_2(s_1) + a_{21} \cdot \alpha_2(s_2)] \\ &= 0.1 \cdot [0.7 \cdot 0.3105 + 0.3 \cdot 0.0410] = 0.022965 \\ \alpha_3(s_2) &= b_2(o_3) \cdot [a_{12} \cdot \alpha_2(s_1) + a_{22} \cdot \alpha_2(s_2)] \\ &= 0.8 \cdot [0.3 \cdot 0.3105 + 0.7 \cdot 0.0410] = 0.097480\end{aligned}$$

- Nota: ¿tenemos ya calculada la probabilidad de que el tercer día sea de lluvia dado que se han observado paraguas en los dos primeros días y sin paraguas el tercero?
  - Aún no, queda un pequeño paso (*¿cuál?*)

Idea gráfica del cálculo que se realiza

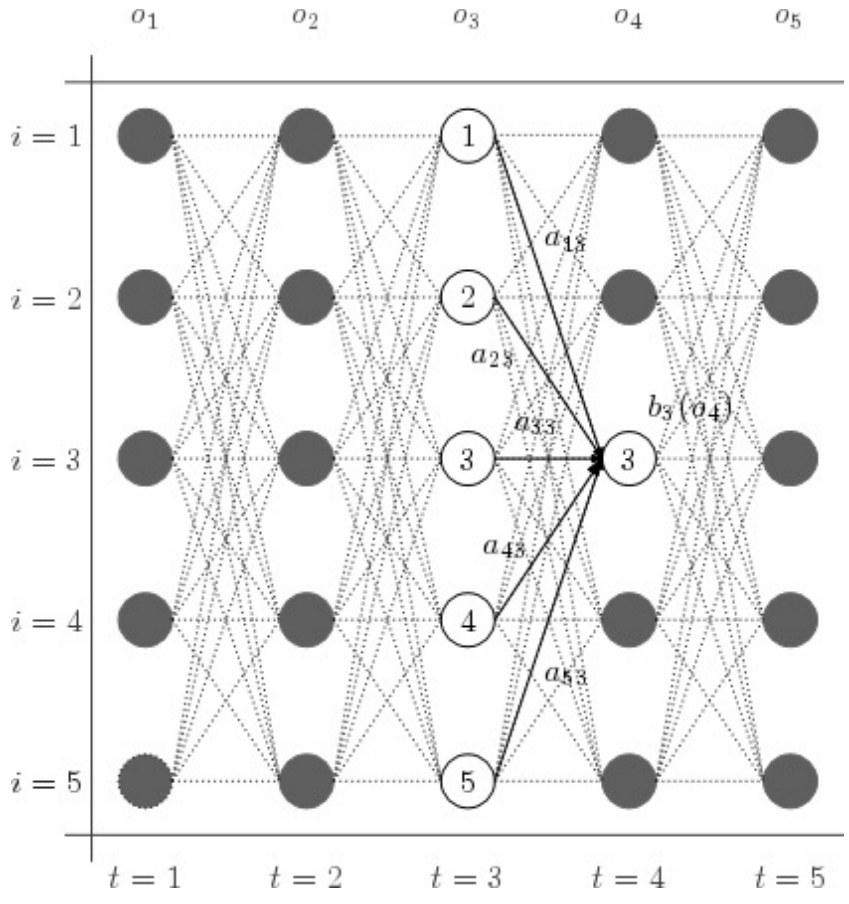


Imagen tomada de wikipedia

¿Cómo usamos los $\alpha_t(s_j)$? (I)

- Recordar que $\alpha_t(s_j) = P(X_t = s_j, o_1 \cdots o_t)$
- Los $\alpha_t(s_j)$ que calcula el algoritmo de avance se usan para:
  - Calcular la probabilidad (incondicional) de una secuencia dada de percepciones
  - Calcular la probabilidad de que tengamos un estado en el instante t, dada la secuencia de percepciones hasta ese instante (lo que hemos llamado *filtrado*)
- Cálculo de la probabilidad de una secuencia:
  - $P(o_1 \cdots o_t) = \sum_{1 \leq i \leq n} P(X_t = s_i, o_1 \cdots o_t) = \sum_{1 \leq i \leq n} \alpha_t(s_i)$
  - En el ejemplo 1, la probabilidad de la secuencia de percepciones $O = (3, 1, 3, 2)$ es $P(O) = \alpha_4(s_1) + \alpha_4(s_2) = 0.00679808 + 0.00370752 = 0.0105056$

¿Cómo usamos los $\alpha_t(s_j)$? (II)

- Cálculo de la probabilidad de que en el instante t estemos en un estado s_j, condicionado a que se ha observado la secuencia $o_1 \cdots o_t$ (*filtrado*):
  - $\boldsymbol{P}(X_t|o_1 \cdots o_t)$ se obtiene normalizando $\boldsymbol{P}(X_t, o_1 \cdots o_t)$
  - Es decir, normalizando los $\alpha_t(s_i)$, $1 \leq i \leq n$, se obtienen las correspondientes probabilidades de filtrado
  - En el ejemplo 2, la probabilidad de que el tercer día sea lluvioso, dado que se ha observado $O = (u, u, \neg u)$ (los dos primeros días con paraguas y el tercero no) se obtiene:
    - Normalizamos $\langle \alpha_3(s_1), \alpha_3(s_2) \rangle = \langle 0.190667939723525, 0.809332060276475 \rangle$, obteniendo $\approx \langle 0.19, 0.81 \rangle$
    - Por tanto, $P(X_3 = l|uu\neg u) \approx 0.19$

Smoothing

¿cuál es el estado más probable para alguna de las posiciones intermedias?

- Supongamos que tenemos la secuencia $O = o_1 o_2 \cdots o_t$ de percepciones hasta el instante t.
- Dado un estado q queremos calcular la probabilidad de que, habiendo observado la secuencia O hasta el instante t, el estado del sistema en el instante k ($1 \leq k < t$) sea q;

$$P(X_k = q|O) = \frac{P(X_k = q, O)}{P(O)} = \frac{P(X_k = q, O)}{\sum_{1 \leq i \leq n} \alpha_t(s_i)}.$$

- Se tiene que

$$\begin{aligned} P(X_k = q, O) &= P(X_k = q, o_1, \ldots, o_k) P(o_{k+1}, \ldots, o_t | X_k = q, o_1, \ldots, o_k) \\ &= P(X_k = q, o_1, \ldots, o_k) P(o_{k+1}, \ldots, o_t | X_k = q) \\ &= \alpha_k(q) P(o_{k+1}, \ldots, o_t | X_k = q) \end{aligned}$$

Cálculo recursivo para "smoothing" (I)

Si llamamos $\beta_k(s_j)$ a $P(o_{k+1}, \ldots, o_t | X_k = s_j)$, tenemos:

- Inicio:

$$\begin{aligned}\beta_{t-1}(s_i) &= P(o_t | X_{t-1} = s_i) = \sum_{1 \le j \le n} P(o_t, X_t = s_j | X_{t-1} = s_i) \\ &= \sum_{1 \le j \le n} P(o_t | X_t = s_j, X_{t-1} = s_i) P(X_t = s_j | X_{t-1} = s_i) \\ &= \sum_{1 \le j \le n} P(o_t | X_t = s_j) a_{ij} = \sum_{1 \le j \le n} b_j(o_t) a_{ij}\end{aligned}$$

Cálculo recursivo para "smoothing" (II)

- Recursión:

$$\begin{aligned}
\beta_r(s_i) &= P(o_{r+1},\ldots,o_t|X_r=s_i) = \sum_{1\le j\le n} P(o_{r+1},\ldots,o_t,X_{r+1}=s_j|X_r=s_i)\\
&= \sum_{1\le j\le n} P(o_{r+1},\ldots,o_t|X_{r+1}=s_j,X_r=s_i)P(X_{r+1}=s_j|X_r=s_i)\\
&= \sum_{1\le j\le n} P(o_{r+1},o_{r+2},\ldots,o_t|X_{r+1}=s_j)a_{ij}\\
&= \sum_{1\le j\le n} P(o_{r+1}|X_{r+1}=s_j)P(o_{r+2},\ldots,o_t|X_{r+1}=s_j,o_{r+1})a_{ij}\\
&= \sum_{1\le j\le n} b_j(o_{r+1})P(o_{r+2},\ldots,o_t|X_{r+1}=s_j)a_{ij} = \sum_{1\le j\le n} b_j(o_{r+1})\beta_{r+1}(s_j)a_{ij}
\end{aligned}$$

Se suele iniciar la recursión considerando: $\beta_t(s_i)=1$

Algoritmo de "retroceso"(*backward*)

- **Entrada:**
  - Un modelo oculto de Markov
  - Una secuencia de percepciones $O = o_1 \cdots o_t$
  - Un entero k tal que $1 \leq k < t$.
- **Salida:**
  - Valores $\beta_k(s_i) = P(o_{k+1}, \ldots, o_t | X_k = s_i)$, para cada estado s_i
- **Procedimiento:**
  - Inicio: $\beta_t(s_i) = 1$, para $1 \leq i \leq n$
  - Para r desde $t-1$ a k:
    - Para j desde 1 a n:
      - Hacer $\beta_r(s_j) = \sum_{1 \leq i \leq n} b_i(o_{r+1}) \beta_{r+1}(s_i) a_{ji}$
  - Devolver los $\beta_k(s_i)$, para $1 \leq i \leq n$

Algoritmo de retroceso para el ejemplo 1

- Para el ejemplo 1, supongamos que hasta el cuarto día la secuencia de percepciones es $O = (3, 1, 3, 2)$ (es decir $o_1 = 3$, $o_2 = 1$, $o_3 = 3$ y $o_4 = 2$). Calculemos $\beta_2(s_2)$ y $\beta_2(s_2)$ (donde $s_1 = c$ y $s_2 = f$)
  - Paso 1:

$$\begin{aligned}\beta_3(s_1) &= b_1(o_4) \cdot a_{11} + b_2(o_4) \cdot a_{12} \\ &= 0.4 \cdot 0.7 + 0.4 \cdot 0.3 = 0.4 \\ \beta_3(s_2) &= b_1(o_4) \cdot a_{21} + b_2(o_4) \cdot a_{22} \\ &= 0.4 \cdot 0.4 + 0.4 \cdot 0.6 = 0.4\end{aligned}$$

  - Paso 2:

$$\begin{aligned}\beta_2(s_1) &= b_1(o_3) \cdot \beta_3(s_1) \cdot a_{11} + b_2(o_3) \cdot \beta_3(s_2) \cdot a_{12} \\ &= 0.4 \cdot 0.4 \cdot 0.7 + 0.1 \cdot 0.4 \cdot 0.3 = 0.124 \\ \beta_2(s_2) &= b_1(o_3) \cdot \beta_3(s_1) \cdot a_{21} + b_2(o_3) \cdot \beta_3(s_2) \cdot a_{22} \\ &= 0.4 \cdot 0.4 \cdot 0.4 + 0.1 \cdot 0.4 \cdot 0.6 = 0.088\end{aligned}$$

Algoritmo de retroceso para el ejemplo 1

- Para el ejemplo 1, supongamos que hasta el cuarto día la secuencia de percepciones es $O = (3, 1, 3, 2)$ (es decir $o_1 = 3$, $o_2 = 1$, $o_3 = 3$ y $o_4 = 2$). ¿Cuál es la probabilidad de cada uno de los estados el segundo día?

  A partir de los ejemplos de avance y retroceso tenemos que:

  - $\alpha_2(s_1) = 0.0464$ y $\alpha_2(s_2) = 0.054$
  - $\beta_2(s_1) = 0.124$ y $\beta_2(s_2) = 0.088$

  Multiplicando y normalizando:

  - $P(X_2 = s_1|O) = .5476698$
  - $P(X_2 = s_2|O) = .4523302$

Algoritmo "forward-backward"

- **Entrada:**
  - Un modelo oculto de Markov
  - Una secuencia de percepciones $O = o_1 \cdots o_t$
  - Un entero k tal que $1 \leq k \leq t$.
- **Salida:**
  - Valores $P(X_k = s_i, o_1, \ldots, o_t)$, para cada estado s_i
- **Procedimiento:**
  - Inicio: $\beta_t(s_i) = 1$ y $\alpha_1(s_i) = b_i(o_1) \cdot \pi_i$, para $1 \leq i \leq n$
  - Para r desde $t-1$ a k:
    - Para j desde 1 a n:
      - Hacer $\beta_r(s_j) = \sum_{1 \leq i \leq n} b_i(o_{r+1}) \beta_{r+1}(s_i) a_{ji}$
  - Para r desde 2 a k:
    - Para j desde 1 a n:
      - Hacer $\alpha_r(s_j) = b_j(o_r) \sum_{1 \leq i \leq n} (a_{ij} \cdot \alpha_{r-1}(s_i))$
  - Devolver $\alpha_k(s_i)\beta_k(s_i)$, para $1 \leq i \leq n$

Decodificación (explicación más verosímil)

- Nuevamente, tenemos un modelo oculto de Markov y una secuencia de percepciones $O = o_1 \cdots o_t$
- Se trata ahora de calcular la secuencia de estados $Q = q_1 \cdots q_t$ que mejor explica las percepciones
- Es decir, buscamos:

$$\underset{Q}{argmax}\, P(Q|O)$$

- Podríamos calcular $P(Q|O)$ (ó $P(Q, O)$) para todas las secuencias posibles de estados Q, y tomar aquella que da el máximo
  - Ineficiente ($O(n^t)$)
- Cómo con el filtrado, emplearemos un método más eficiente, basado en recursión

Decodificación: método recursivo

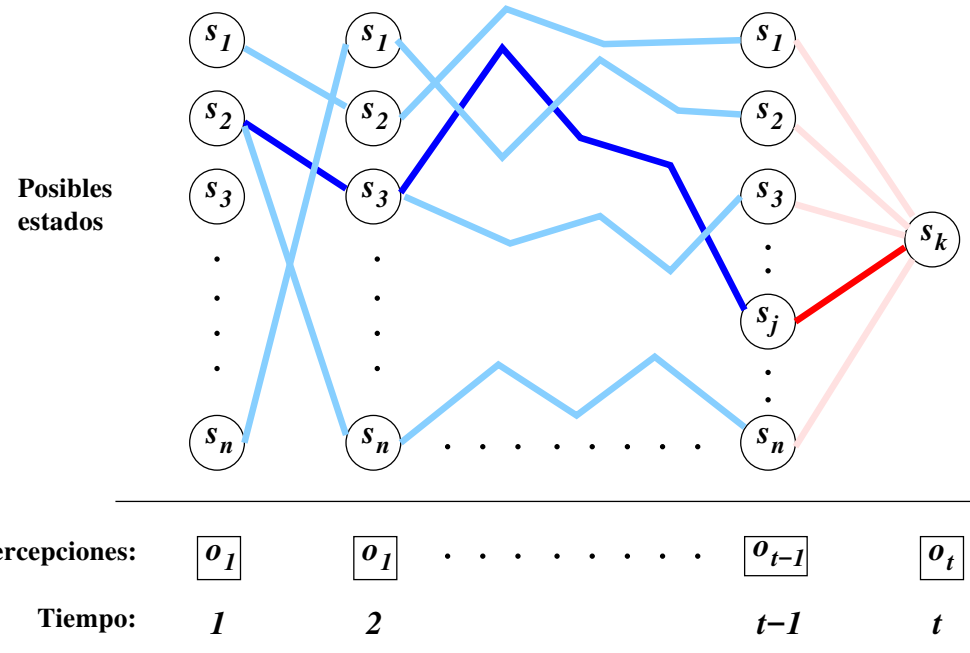
- Dada una secuencia de estados $q_1 \cdots q_t$ tal que $q_t = s_j$, podemos calcular la probabilidad de pasar por esa secuencia de estados, habiendo además observado $o_1 \cdots o_t$.
- Notaremos $\nu_t(s_j)$ a la máxima de esas probabilidades, entre todas esas secuencias de estados. Es decir:

$$\nu_t(s_j) = \max_{q_1 q_2 \cdots q_{t-1}} P(o_1 \cdots o_t, q_1 \cdots q_{t-1}, X_t = s_j)$$

- Cada $\nu_t(s_j)$ se puede calcular recursivamente, en función de los $\nu_{t-1}(s_i), 1 \leq i \leq n$ del instante anterior

Decodificación: método recursivo

- La idea principal de la recursión es que la secuencia más probable de estados hasta llegar a $X_t = s_j$, se compone de la secuencia más probable de llegar hasta un cierto estado s_i en el instante $t-1$, seguido de un paso de transición desde tal s_i a s_j. Por tanto, sólo hay que "buscar" entre los caminos más probables hasta el instante $t-1$ y ver desde cuál se obtiene la probabilidad máxima dando un paso más:



Decodificación: método recursivo

- Inicio: $\nu_1(s_i) = P(o_1, X_1 = s_i) = P(o_1|X_1 = s_i) \cdot P(X_1 = s_i) = b_i(o_1)\pi_i$
- Recursión:

$$
\begin{aligned}
\nu_t(s_j) &= \max_{q_1 q_2 \cdots q_{t-1}} P(o_1 \cdots o_t, q_1 \cdots q_{t-1}, X_t = s_j) \\
&= \max_{q_1 q_2 \cdots q_{t-1}} \big(P(o_t|o_1 \cdots o_{t-1}, q_1 \cdots q_{t-1}, X_t = s_j) \cdot \\
&\qquad P(o_1 \cdots o_{t-1}, q_1 \cdots q_{t-1}, X_t = s_j) \big) \\
&= P(o_t|X_t = s_j) \cdot \max_{q_1 q_2 \cdots q_{t-1}} P(o_1 \cdots o_{t-1}, q_1 \cdots q_{t-1}, X_t = s_j) \\
&= P(o_t|X_t = s_j) \cdot \\
&\quad \max_{i=1,\ldots,n} \big(P(X_t = s_j|X_{t-1} = s_i) \cdot \\
&\qquad \max_{q_1 q_2 \cdots q_{t-2}} P(o_1 \cdots o_{t-1}, q_1 \cdots q_{t-2}, X_{t-1} = s_i) \big) \\
&= b_j(o_t) \max_{i=1,\ldots,n} (a_{ij} \cdot \nu_{t-1}(s_i))
\end{aligned}
$$

Decodificación: método recursivo

- Se trata de un esquema recursivo análogo al del algoritmo de avance usado para filtrado, excepto que en lugar de sumatorios, se toman máximos
- En este caso, además de calcular las probabilidades máximas, debemos llevar la cuenta de la *secuencia de estados* que se corresponden con esas probabilidades
  - Se consigue almacenando, para cada estado e instante de tiempo, un puntero $pr_t(s_j)$ al estado inmediatamente anterior en la secuencia más probable
- El método descrito se denomina *algoritmo de Viterbi*

Algoritmo de Viterbi

- **Entrada:**
  - Un modelo oculto de Markov
  - Una secuencia de percepciones $O = o_1 \cdots o_t$
- **Salida:**
  - La secuencia de estados Q que maximiza $P(Q|O)$
- **Procedimiento:**
  - Inicio: Para $i = 1, \ldots, n$, hacer $\nu_1(s_i) = b_i(o_1) \cdot \pi_i$ y $pr_1(s_i) = null$,
  - Para k desde 2 a t:
    - Para j desde 1 a n:
      - Hacer $\nu_k(s_j) = b_j(o_k) \max_{i=1,\ldots,n} (a_{ij} \cdot \nu_{k-1}(s_i))$
      - Hacer $pr_k(s_j) = \operatorname{argmax}_{i=1,\ldots,n} (a_{ij} \cdot \nu_{k-1}(s_i))$
  - Hacer $s = \operatorname{argmax}_{i=1,\ldots,n} \nu_t(s_i)$
  - Devolver la secuencia de estados que lleva hasta s (siguiendo hacia atrás los punteros, comenzando en $pr_t(s)$)

Algoritmo de Viterbi para el ejemplo 1

- Para el ejemplo 1, supongamos que hasta el cuarto dia la secuencia de percepciones es $O = (3, 1, 3, 2)$. Calculemos la secuencia de estados más probable con esas percepciones (los estado son $s_1 = c$ y $s_2 = f$)
  - Paso 1:

    $\nu_1(s_1) = b_1(o_1) \cdot \pi_1 = 0.4 \cdot 0.8 = 0.32$

    $pr_1(s_1) = null$

    $\nu_1(s_2) = b_2(o_1) \cdot \pi_2 = 0.1 \cdot 0.2 = 0.02$

    $pr_1(s_2) = null$

Algoritmo de Viterbi para el ejemplo 1

- Para el ejemplo 1, supongamos que hasta el cuarto dia la secuencia de percepciones es $O = (3, 1, 3, 2)$. Calculemos la secuencia de estados más probable con esas percepciones (los estado son $s_1 = c$ y $s_2 = f$)
  - Paso 2:

$$\begin{aligned}
\nu_2(s_1) &= b_1(o_2) \cdot max\{a_{11} \cdot \nu_1(s_1), a_{21} \cdot \nu_1(s_2)\} \\
&= 0.2 \cdot max\{0.7 \cdot 0.32, 0.4 \cdot 0.02\} \\
&= 0.2 \cdot max\{\underline{0.224}, 0.008\} = 0.0448 \\
pr_2(s_1) &= s_1 \\
\nu_2(s_2) &= b_2(o_2) \cdot max\{a_{12} \cdot \nu_1(s_1), a_{22} \cdot \nu_1(s_2)\} \\
&= 0.5 \cdot max\{0.3 \cdot 0.32, 0.6 \cdot 0.02\} \\
&= 0.5 \cdot max\{\underline{0.096}, 0.012\} = 0.048 \\
pr_2(s_2) &= s_1
\end{aligned}$$

Algoritmo de Viterbi para el ejemplo 1

- Para el ejemplo 1, supongamos que hasta el cuarto dia la secuencia de percepciones es $O = (3, 1, 3, 2)$. Calculemos la secuencia de estados más probable con esas percepciones (los estado son $s_1 = c$ y $s_2 = f$)
  - Paso 3:

$$\begin{aligned}
\nu_3(s_1) &= b_1(o_3) \cdot max\{a_{11} \cdot \nu_2(s_1), a_{21} \cdot \nu_2(s_2)\} \\
&= 0.4 \cdot max\{0.7 \cdot 0.0448, 0.4 \cdot 0.048\} \\
&= 0.4 \cdot max\{\underline{0.03136}, 0.0192\} = 0.012544 \\
pr_3(s_1) &= s_1 \\
\nu_3(s_2) &= b_2(o_3) \cdot max\{a_{12} \cdot \nu_2(s_1), a_{22} \cdot \nu_2(s_2)\} \\
&= 0.1 \cdot max\{0.3 \cdot 0.0448, 0.6 \cdot 0.048\} \\
&= 0.1 \cdot max\{0.01344, \underline{0.0288}\} = 0.00288 \\
pr_3(s_2) &= s_2
\end{aligned}$$

Algoritmo de Viterbi para el ejemplo 1

- Para el ejemplo 1, supongamos que hasta el cuarto dia la secuencia de percepciones es $O = (3, 1, 3, 2)$. Calculemos la secuencia de estados más probable con esas percepciones (los estado son $s_1 = c$ y $s_2 = f$)
  - Paso 4:

$$\begin{aligned}
\nu_4(s_1) &= b_1(o_4) \cdot max\{a_{11} \cdot \nu_3(s_1), a_{21} \cdot \nu_3(s_2)\} \\
&= 0.4 \cdot max\{0.7 \cdot 0.012544, 0.4 \cdot 0.00288\} \\
&= 0.4 \cdot max\{\underline{0.0087808}, 0.001152\} = 0.00351232 \\
pr_4(s_1) &= s_1 \\
\nu_4(s_2) &= b_2(o_4) \cdot max\{a_{12} \cdot \nu_3(s_1), a_{22} \cdot \nu_3(s_2)\} \\
&= 0.4 \cdot max\{0.3 \cdot 0.012544, 0.6 \cdot 0.00288\} \\
&= 0.4 \cdot max\{\underline{0.0037632}, 0.001728\} = 0.00150528 \\
pr_4(s_2) &= s_1
\end{aligned}$$

Algoritmo de Viterbi para el ejemplo 1

- Para el ejemplo 1, supongamos que hasta el cuarto dia la secuencia de percepciones es $O = (3, 1, 3, 2)$. Calculemos la secuencia de estados más probable
con esas percepciones (los estado son $s_1 = c$ y $s_2 = f$)
  - Paso final:
    - El estado que da la probabilidad máxima en $t = 4$ es s_1, con probabilidad 0.00351232
    - Reconstruimos el camino hasta s_1:
$pr_4(s_1) = s_1, pr_3(s_1) = s_1, pr_2(s1) = s_1$
    - Luego la secuencia de estados más probable es
(s_1, s_1, s_1, s_1) (es decir, cuatro días seguidos de calor)

Algoritmo de Viterbi para el ejemplo 2

- Para el ejemplo 2, supongamos que la secuencia de percepciones hasta el tercer día es $O = (u, u, \neg u)$ (paraguas los dos primeros días y el tercero no). Calculemos la secuencia de estados más probable, con esas percepciones (los estados son $s_1 = l$ y $s_2 = \neg l$)
  - Paso 1:

    $\nu_1(s_1) = b_1(o_1) \cdot \pi_1 = 0.9 \cdot 0.5 = 0.45$

    $pr_1(s_1) = null$

    $\nu_1(s_2) = b_2(o_1) \cdot \pi_2 = 0.2 \cdot 0.5 = 0.1$

    $pr_1(s_2) = null$

Algoritmo de Viterbi para el ejemplo 2

- Para el ejemplo 2, supongamos que la secuencia de percepciones hasta el tercer día es $O = (u, u, \neg u)$ (paraguas los dos primeros días y el tercero no). Calculemos la secuencia de estados más probable, con esas percepciones (los estados son $s_1 = l$ y $s_2 = \neg l$)
  - Paso 2:

$$\begin{aligned}
\nu_2(s_1) &= b_1(o_2) \cdot max\{a_{11} \cdot \nu_1(s_1), a_{21} \cdot \nu_1(s_2)\} \\
&= 0.9 \cdot max\{0.7 \cdot 0.45, 0.3 \cdot 0.1\} \\
&= 0.9 \cdot max\{\underline{0.315}, 0.03\} = 0.2835 \\
pr_2(s_1) &= s_1 \\
\nu_2(s_2) &= b_2(o_2) \cdot max\{a_{12} \cdot \nu_1(s_1), a_{22} \cdot \nu_1(s_2)\} \\
&= 0.2 \cdot max\{0.3 \cdot 0.45, 0.7 \cdot 0.1\} \\
&= 0.2 \cdot max\{\underline{0.135}, 0.07\} = 0.027 \\
pr_2(s_2) &= s_1
\end{aligned}$$

Algoritmo de Viterbi para el ejemplo 2

- Para el ejemplo 2, supongamos que la secuencia de percepciones hasta el tercer día es $O = (u, u, \neg u)$ (paraguas los dos primeros días y el tercero no). Calculemos la secuencia de estados más probable, con esas percepciones (los estados son $s_1 = l$ y $s_2 = \neg l$)
  - Paso 3:

$$\begin{aligned}
\nu_3(s_1) &= b_1(o_3) \cdot max\{a_{11} \cdot \nu_2(s_1), a_{21} \cdot \nu_2(s_2)\} \\
&= 0.1 \cdot max\{0.7 \cdot 0.2835, 0.3 \cdot 0.027\} \\
&= 0.1 \cdot max\{\underline{0.19845}, 0.0081\} = 0.019845 \\
pr_3(s_1) &= s_1 \\
\nu_3(s_2) &= b_2(o_3) \cdot max\{a_{12} \cdot \nu_2(s_1), a_{22} \cdot \nu_2(s_2)\} \\
&= 0.8 \cdot max\{0.3 \cdot 0.2835, 0.7 \cdot 0.027\} \\
&= 0.8 \cdot max\{\underline{0.08505}, 0.0189\} = 0.06804 \\
pr_3(s_2) &= s_1
\end{aligned}$$

Algoritmo de Viterbi para el ejemplo 2

- Para el ejemplo 2, supongamos que la secuencia de percepciones hasta el tercer día es $O = (u, u, \neg u)$ (paraguas los dos primeros días y el tercero no). Calculemos la secuencia de estados más probable, con esas percepciones (los estados son $s_1 = l$ y $s_2 = \neg l$)
  - Paso final:
    - El estado que da la probabilidad máxima en $t = 3$ es s_2, con probabilidad 0.06804
    - Reconstruimos el camino hasta s_1:
      $$pr_3(s_2) = s_1, pr_2(s_1) = s_1$$
    - Luego la secuencia de estados más probable es (s_1, s_1, s_2) (es decir, los dos primeros días con lluvia y el tercero no)

El problema de los números muy bajos

- Problema en los dos algoritmos que hemos visto:
  - A medida que aumenta el número de percepciones, tanto los $\alpha_k(s_j)$ como los $\nu_k(s_j)$ son números cada vez más cercanos a cero
- Para evitar esto en el algoritmo de avance, es usual que en cada paso, se aplique normalización de los $\alpha_k(s_j)$
- ¿Cambia esto el resultado en el algoritmo de avance?
  - Estaríamos calculando en cada paso t las probabilidades de filtrado $P(X_k = s_j | o_1 \cdots o_k)$ (en lugar de $P(X_k = s_j, o_1 \cdots o_k)$)
  - Si quisieramos saber, la probabilidad $P(o_1 \cdots o_t)$, ésta se calcularía como la inversa del producto de los factores de normalización usados en cada paso (*¿por qué?*)

El problema de los números muy bajos en Viterbi

- En el algoritmo de Viterbi también podríamos usar normalización (y no cambiaría el resultado final)
- Sin embargo, puesto que no realiza sumas, es usual tratar este problema mediante el uso de logaritmos (*log-probabilidades*)
- Si se toman logaritmos, los productos que realiza el algoritmo se transforman en sumas
- Esto no afecta al cálculo de los máximos, ya que el logaritmo es una función creciente
- Nótese que las log-probabilidades de las tablas de transición y de las de percepción se pueden calcular al principio

Algoritmo de Viterbi (log-modificado)

- **Entrada:**
  - Un modelo oculto de Markov
  - Una secuencia de percepciones $O = o_1 \cdots o_t$
- **Salida:**
  - La secuencia de estados Q que maximiza $P(Q|O)$
- **Procedimiento:**
  - Inicio: Para $i = 1, \ldots, n$, hacer $\hat{\nu}_1(s_i) = log(b_i(o_1)) + log(\pi_i)$ y $pr_1(s_i) = null$,
  - Para k desde 2 a t:
    - Para j desde 1 a n:
      - Hacer $\hat{\nu}_k(s_j) = log(b_j(o_t)) + \max_{i=1,\ldots,n} (log(a_{ij}) + \hat{\nu}_{k-1}(s_i))$
      - Hacer $pr_k(s_j) = \operatorname{argmax}_{i=1,\ldots,n} (log(a_{ij}) + \hat{\nu}_{k-1}(s_i))$
  - Hacer $s = \operatorname{argmax}_{i=1,\ldots,n} \hat{\nu}_t(s_i)$
  - Devolver la secuencia de estados que lleva hasta s (siguiendo hacia atrás los punteros, comenzando en $pr_t(s)$)

Aprendizaje de MOMs

- El tercer problema importante en MOMs es el aprendizaje:
  - Conocemos el número de posibles estados y de percepciones, pero no conocemos ni las probabilidades de transición ni de percepción.
  - Sólo tenemos una secuencia de percepciones $o_1 \cdots o_t$, usualmente con t muy grande.
- El algoritmo que se usa para este tipo de problema es el denominado *algoritmo de Baum-Welch*
  - Es un algoritmo de tipo EM
- Otra posibilidad es hacer una aprendizaje ML
  - Sólo es válido si también tenemos la secuencia de estados correspondiente a las percepciones $q_1 \cdots q_t$

Aprendizaje ML

- $\pi_i = P(X_1 = s_i) = \dfrac{L_{s_i}}{\sum_{1 \leq k \leq n} L_{s_k}}$ siendo L_{s_k} el número de veces que el estado inicial es s_k.
- $a_{ij} = P(X_t = s_j | X_{t-1} = s_i) = \dfrac{N_{s_i s_j}}{\sum_{1 \leq k \leq n} N_{s_i s_k}}$ siendo $N_{s_i s_k}$ el número de veces que el estado s_k sigue al estado s_i
- $b_i(v_j) = P(E_t = v_j | X_t = s_i) = \dfrac{M_{s_i v_j}}{\sum_{1 \leq k \leq m} M_{s_i v_k}}$ siendo $M_{s_i v_k}$ el número de veces que el estado es s_i y se observa v_k (en la misma posición)

Estimación para ejemplo 1

Conocemos las secuencias de percepciones y estados:

f	f	f	f	f	c	f	c	c		c	f	f	c	c	c	c	c	c
1	2	1	1	2	3	2	3	3		2	2	1	1	2	3	2	2	3

f	f	f	f	f	f	c	c	c
2	1	3	2	1	1	2	3	2

Calculamos las frecuencias y estimamos las probabilidades

$$L_f = 2, L_c = 1$$
$$\pi_f = 2/3, \pi_c = 1/3$$

N	f	c
f	10	4
c	2	8

A	f	c
f	$5/7$	$2/7$
c	$1/5$	$4/5$

M	1	2	3
f	7	4	1
c	1	6	6

B	1	2	3
f	$7/12$	$1/3$	$1/12$
c	$1/13$	$6/13$	$6/13$

Algoritmo tipo EM

- Inicio: Parámetros iniciales del modelo estadístico
- Paso **E**: Calcular ciertos valores esperados a partir de los parámetros del modelo
- Paso **M**: Estimar los parámetros del modelo a partir de los valores esperados anteriores.
- Repetir los dos pasos anteriores hasta algún criterio de convergencia (o limitando el número de veces)

Algoritmo Baum-Welch

- **Entrada:**
  - Un modelo oculto de Markov, $\lambda = (\pi, A, B)$
  - Una secuencia de percepciones $O = o_1 \cdots o_t$
- **Salida:**
  - Un modelo oculto de Markov, λ' tal que $P(O|\lambda') \geq P(O|\lambda)$
- **Procedimiento:**
  - Repetir hasta un determinado criterio de convergencia:
    - $\lambda' = \lambda$
    - Paso **E**xpectation
    - Paso **M**aximization (que da lugar a un modelo λ)

Algoritmo Baum-Welch: Paso **E**

Calculamos los valores de las probabilidades de las transiciones y los estados (conocida una secuencia de percepciones)

- $\xi_k(i,j) = P(X_k = s_i, X_{k+1} = s_j | o_1 \cdots o_t)$

$$\xi_k(i,j) = \frac{P(X_k = s_i, X_{k+1} = s_j, o_1 \cdots o_t)}{P(o_1 \cdots o_t)}$$

$$= \frac{P(X_k = s_i, o_1 \cdots o_k) P(X_{k+1} = s_j, o_{k+1} \cdots o_t | X_k = s_i, o_1 \cdots o_k)}{P(o_1 \cdots o_t)}$$

$$= \frac{\alpha_k(s_i) P(X_{k+1} = s_j, o_{k+1} | X_k = s_i, o_1 \cdots o_k) P(o_{k+2} \cdots o_t | X_{k+1} = s_j, X_k = s_i, o_1 \cdots o_{k+1})}{\sum_{1 \le i \le n} \alpha_t(s_i)}$$

$$= \frac{\alpha_k(s_i) P(X_{k+1} = s_j, o_{k+1} | X_k = s_i) P(o_{k+2} \cdots o_t | X_{k+1} = s_j)}{\sum_{1 \le i \le n} \alpha_t(s_i)}$$

$$= \frac{\alpha_k(s_i) P(X_{k+1} = s_j | X_k = s_i) P(o_{k+1} | X_{k+1} = s_j, X_k = s_i) \beta_{k+1}(s_j)}{\sum_{1 \le i \le n} \alpha_t(s_i)} = \frac{\alpha_k(s_i) a_{ij} b_j(o_{k+1}) \beta_{k+1}(s_j)}{\sum_{1 \le i \le n} \alpha_t(s_i)}$$

Algoritmo Baum-Welch: Paso **E**

Calculamos los valores de las probabilidades de las transiciones y los estados (conocida una secuencia de percepciones)

- $\gamma_k(i) = P(X_k = s_i | o_1 \cdots o_t)$

$$\gamma_k(i) = \frac{P(X_k = s_i, o_1 \cdots o_t)}{P(o_1 \cdots o_t)} = \frac{\alpha_k(s_i)\beta_k(s_i)}{\sum_{1 \leq i \leq n} \alpha_t(s_i)}$$

Algoritmo Baum-Welch: Paso **M**

Estimamos los parámetros del modelo:

- $\pi_i = \gamma_1(i)$
- $a_{ij} = \dfrac{\sum_{1 \leq k \leq t} \xi_k(i,j)}{\sum_{1 \leq k \leq t} \gamma_k(i)}$
- $b_i(v_j) = \dfrac{\sum_{1 \leq k \leq t \wedge o_k = v_j} \gamma_k(i)}{\sum_{1 \leq k \leq t} \gamma_k(i)}$

Baum et al. demostraron que si a partir de un modelo $\lambda' = (\pi', A', B')$ calculamos el modelo λ utilizando el procedimiento anterior (pasos **E** y **M**) entonces se tiene que:

- o bien, $\lambda = \lambda'$
- o bien, $P(O|\lambda) > P(O|\lambda')$

Bibliografía

- Jurafsky, D. y Martin, J.H. *Speech and Language Processing* (Second Edition) (Prentice-Hall, 2009)
  - Cap. 6: “Hidden Markov and Maximum Entropy Models ”
- Russell, S. y Norvig, P. *Artificial Intelligence (A modern approach)* (Third edition) (Prentice Hall, 2009)
  - Cap. 15 (hasta 15.3): “Probabilistic reasoning over time”
- Russell, S. y Norvig, P. *Inteligencia Artificial (Un enfoque moderno)* (Segunda edición) (Pearson Educación, 2004)
  - Cap. 15 (hasta 15.3): “Razonamiento probabilista en el tiempo”