UNIVERSIDAD
DE GRANADA

MODELOS DE MARKOV OCULTOS Y APLICACIONES A LA BIOLOGÍA

XUSHENG ZHENG

Trabajo Fin de Grado

Doble Grado en Ingeniería Informática y Matemáticas

Tutores

Lidia Fernández Rodríguez

FACULTAD DE CIENCIAS

E.T.S. INGENIERÍAS INFORMÁTICA Y DE TELECOMUNICACIÓN

Granada, a 30 de mayo de 2023

ÍNDICE GENERAL

RESUMEN

Occaecati expedita cumque est. Aut odit vel nobis praesentium dolorem sed eligendi. Inventore molestiae delectus voluptatibus consequatur. Et cumque quia recusandae fugiat earum repellat porro. Earum et tempora vel voluptas. At sed animi qui hic eaque velit.

Saepe deleniti aut voluptatem libero dolores illum iusto iusto. Explicabo dolor quia id enim molestiae praesentium sit. Odit enim doloribus aut assumenda recusandae. Eligendi officia nihil itaque. Quas fugiat aliquid qui est.

Quis amet sint enim. Voluptatem optio quia voluptatem. Perspiciatis molestiae ut laboriosam repudiandae nihil.

1

PRELIMINARES

En este capítulo vamos a presentar las herramientas básicas para llevar a cabo nuestra teoría. Empezaremos recordando conceptos básicos de probabilidad, presentaremos algunos resultados relacionados con las matrices y finalmente introduciremos técnicas que se utilizan en la estadística y en la computación.

1.1 CONCEPTOS BÁSICOS DE PROBABILIDAD

Comenzamos presentado los elementos necesarios para considerar una probabilidad:

Definición 1.1. Sea Ω un conjunto arbitrario, diremos que una familia no vacía de subconjuntos de Ω, $\mathcal{A} \subseteq \mathcal{P}(\Omega)$, es una σ-álgebra si:

- Es cerrada para complementarios: $\forall A \in \mathcal{A}$, $\Omega \setminus A \in \mathcal{A}$.
- Es cerrada para uniones numerables: si $A_n \in \mathcal{A}$, $\forall n \in \mathbb{N} \implies \bigcup_{n\in\mathbb{N}} A_n \in \mathcal{A}$.

A la dupla $(\Omega, \mathcal{A})$ se le conoce como espacio medible.

Definición 1.2. Sea $(\Omega, \mathcal{A})$ un espacio medible, $P : \mathcal{A} \longrightarrow [0,1]$ es una probabilidad si:

- $P(A) \geq 0$, $\forall A \in \mathcal{A}$.
- $P(\Omega) = 1$.
- Dada una secuencia $\{A_n\}_{n\in\mathbb{N}} \subseteq \mathcal{A}$ con $A_i \not\parallel A_j$, $i \neq j$ entonces:

$$P\left(\bigcup_{n\in\mathbb{N}} A_n\right) = \sum_{n\in\mathbb{N}} P(A_n)$$

A la terna $(\Omega, \mathcal{A}, P)$ se le conoce como espacio probabilístico.

Definición 1.3. Sea $(\Omega, \mathcal{A}, P)$ un espacio probabilístico y $A \in \mathcal{A}$ con $P(A) > 0$. Sea $B \in \mathcal{A}$, se define la probabilidad condicionada de B dado A como:

$$P(B|A) = \frac{P(B \cap A)}{P(A)}$$

Relacionado con esta definición presentamos el siguiente teorema que nos será útil:

Teorema 1.4 (Teorema de probabilidad total)**.** Sea $(\Omega, \mathcal{A}, P)$ un espacio probabilístico y $\{A_n\}_{n\in\mathbb{N}} \subseteq \mathcal{A}$ una partición de Ω con $P(A_n) > 0$, $\forall n \in \mathbb{N}$. Sea $B \in \mathcal{A}$, entonces:

$$P(B) = \sum_{n\in\mathbb{N}} P(A_n)P(B|A_n)$$

Demostración. Por ser $\{A_n\}$ una partición de Ω, aplicando la definición de probabilidad condicionada:

$$P(B) = P\left(\bigcup_{n\in\mathbb{N}} B \cap A_n\right) = \sum_{n\in\mathbb{N}} P(B \cap A_n) = \sum_{n\in\mathbb{N}} P(A_n)P(B|A_n)$$

□

Definición 1.5. Sea $(\Omega, \mathcal{A}, P)$ un espacio probabilístico y $(\Omega', \mathcal{A}')$ un espacio medible. Una función $X : (\Omega, \mathcal{A}, P) \longrightarrow (\Omega', \mathcal{A}')$ es una variable aleatoria si:

$$X^{-1}(B) \in \mathcal{A}, \quad \forall B \in \mathcal{A}'$$

1.2 RESULTADOS SOBRE MATRICES

Para el estudio de las cadenas de Markov, necesitamos de antemano algunos resultados sobre matrices. Por comodidad, usaremos a lo largo de este trabajo la notación fila para representar los vectores. Los contenidos de esta sección se basa principalmente en [17].

Definición 1.6. Sea una matriz $A = [a_{ij}]$, diremos que A es:

- **no negativa** si $a_{ij} \geq 0$, $\forall i, j$.
- **positiva** si $a_{ij} \geq 0$, $\forall i, j$ y existe $a_{ij} > 0$ para al menos un par de índices i, j.
- **estrictamente positiva** si $a_{ij} > 0$, $\forall i, j$.

Definición 1.7. Sea $A = [a_{ij}]$ una matriz cuadrada de dimensión N, diremos que A es una **matriz estocástica** si:

$$a_{ij} \in [0, 1], \quad \forall i, j \in \{1, ..., N\},$$
$$\sum_{j=1}^{N} a_{ij} = 1, \quad \forall i \in \{1, ..., N\}.$$

Una forma de caracterizar a las matrices estocásticas es mediante la siguiente proposición:

Proposición 1.8. Sea A una matriz cuadrada positiva de dimensión $N \times N$:

1. A es estocástica si y solo si 1 es un valor propio de A^T con vector propio $\mathbf{1} = \begin{pmatrix} 1 & 1 & \dots & 1 \end{pmatrix}$.
2. si A es estocástica, entonces para todo valor propio λ, se cumple que $|\lambda| \leq 1$.

Demostración.

1. Es suficiente con observar que la condición de estocasticidad para una matriz positiva A es equivalente a que $\mathbf{1} \cdot A^T = \mathbf{1}$.
2. Sea $v = (v_1, \dots, v_N)$ un vector propio asociado a λ (a izquierda pues estamos usando la notación fila), por ser A positiva y estocástica, se verifica:

$$\begin{aligned} |\lambda| \sum_{j=1}^{N} |v_j| &= \sum_{j=1}^{N} |\lambda v_j| = \sum_{j=1}^{N} |(vA)_j| = \sum_{j=1}^{N} \left| \sum_{i=1}^{N} a_{ij} v_i \right| \\ &\leq \sum_{j=1}^{N} \sum_{i=1}^{N} a_{ij} |v_i| = \sum_{i=1}^{N} \left(\sum_{j=1}^{N} a_{ij} \right) |v_i| = \sum_{i=1}^{N} |v_i| \end{aligned}$$

   Puesto que $\sum_{r=1}^{N} |v_r| > 0$, tenemos que $|\lambda| \leq 1$. □

Observación. De la primera afirmación de la proposición 1.8, podemos ver que el producto de matrices estocásticas sigue siendo estocástica. En efecto, si A y B son dos matrices estocásticas entonces $\mathbf{1} \cdot (AB)^T = \mathbf{1} \cdot B^T \cdot A^T = \mathbf{1} \cdot A^T = \mathbf{1}$.

Presentamos a continuación la definición de un grafo dirigido y el grafo de una matriz no negativa:

Definición 1.9. Un grafo dirigido G es un par (V, L) donde $V = \{v_1, \dots, v_n\}$ es un conjunto finito de elementos llamados **nodos** (o **vértices**) y $L = \{l_1, \dots, l_m\} \subseteq V \times V$ es un conjunto de pares ordenados de dichos nodos llamados **arcos** (o **aristas**).

Definición 1.10. Sea $G = (V, L)$ un grafo dirigido, un **camino dirigido** C desde v_{i_0} a v_{i_p} es una secuencia de nodos $C = \{v_{i_0}, v_{i_1}, \dots, v_{i_p}\}$ tal que $v_{i_k} \in V$ para todo $k = 0, \dots, p$ y $(v_{i_{k-1}}, v_{i_k}) \in L$ para todo $k = 1, \dots, p$.

Definición 1.11. Sea $G = (V, L)$ un grafo dirigido, diremos que:

- un nodo $v_i \in V$ está **conectado** con un nodo $v_j \in V$ si existe un camino dirigido de v_i a v_j.
- un nodo $v_i \in V$ está **fuertemente conectado** con un nodo $v_j \in V$ si ambos están conectados.

Un grafo dirigido es **fuertemente conectado** si sólo tiene un nodo o todos sus nodos están fuertemente conectados entre ellos.

Definición 1.12. Sea A una matriz cuadrada no negativa de dimensión N, el grafo dirigido asociado a A es de la forma $G_A = (V_A, L_A)$ donde:

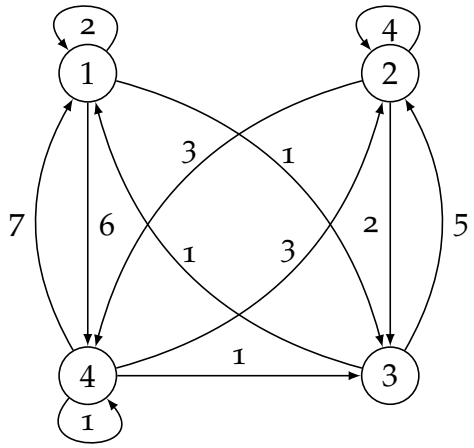
$$V_A = \{1, \ldots, N\}$$

$$L_A = \{(i, j) \in V_A \times V_A \mid a_{ij} > 0\}$$

Ejemplo 1.1. Dada la matriz:

$$A = \begin{pmatrix} 2 & 0 & 1 & 6 \\ 0 & 4 & 2 & 3 \\ 1 & 5 & 0 & 0 \\ 7 & 3 & 1 & 1 \end{pmatrix}$$

Podemos representar su grafo asociado:



El grafo de esta matriz es fuertemente conectado pues desde cualquier nodo podemos encontrar un camino dirigido a los otros nodos, incluido a sí mismo.

2

INTRODUCCIÓN A LAS CADENAS DE MARKOV

La cadena de Markov fue introducida por el matemático ruso Andréi Márkov en 1906. Desde su definición, se ha utilizado para describir procesos importantes en la teoría de probabilidad. Hoy en día, tiene aplicaciones en campos como la biología, la economía, la química o por ejemplo en algoritmos para Internet (PageRank).

En este capítulo se va a introducir la teoría de cadenas de Markov, avanzado progresivamente hacia las cadenas de Markov ocultas. Las fuentes principales de este capítulo son [19, Capítulo 4], [3, Capítulos 2 y 3] y [17, Capítulo 6].

2.1 PROPIEDAD DE MARKOV

Sea un conjunto finito $\mathbb{S} = \{s_1, ..., s_N\}$, definimos un proceso estocástico sobre $\mathbb{S}$ como una sucesión $\{\mathcal{X}_0, \mathcal{X}_1, \mathcal{X}_2, ...\}$, o $\{\mathcal{X}_t\}_{t=0}^{\infty}$ para abreviar, donde cada $\mathcal{X}_t$ es una variable aleatoria que toma valores en $\mathbb{S}$.

A pesar de que el índice t puede representar cualquier magnitud, lo más común es que represente el tiempo. En este caso, la noción de "pasado" y "futuro" aparecen de forma natural, esto es, si $t < t'$, entonces $\mathcal{X}_t$ es una variable "pasada" para $\mathcal{X}_{t'}$, mientras que $\mathcal{X}_{t'}$ es una variable "futura" para $\mathcal{X}_t$. Sin embargo, esto no sucede siempre así: por ejemplo, si el proceso estocástico corresponde a la secuencia del genoma de un organismo, el conjunto $\mathbb{S}$ estará formado por los cuatro símbolos para las subunidades de nucleótidos $\{A, C, G, T\}$ y las secuenciaciones tienen un significado más espacial que temporal.

Definición 2.1. Un proceso estocástico $\{\mathcal{X}_t\}_{t=0}^{\infty}$ se dice que posee **la propiedad de Markov**, o que es una **cadena de Markov**, si para todo $t \geq 1$ y $(u_0, ..., u_{t-1}, u_t) \in \mathbb{S}^{t+1}$ se tiene que:

$$P[\mathcal{X}_t = u_t | \mathcal{X}_0 = u_0, ..., \mathcal{X}_{t-1} = u_{t-1}] = P[\mathcal{X}_t = u_t | \mathcal{X}_{t-1} = u_{t-1}] \tag{2.1}$$

Es decir, un proceso con la propiedad de Markov es aquel en el que la probabilidad de que ocurra un determinado suceso en el instante t sólo depende de lo que ocurrió en el instante $t-1$ y no de los estados previos.

Introducimos la notación $\mathcal{X}_j^k = (\mathcal{X}_j, \mathcal{X}_{j+1}, \ldots, \mathcal{X}_k)$ para denotar los estados $\mathcal{X}_i$ con $j \leq i \leq k$. Con esta notación, podemos reescribir la definición 1.1 como sigue: un proceso estocástico $\{\mathcal{X}_t\}$ es una **cadena de Markov** si, para todo $(u_0, ..., u_{t-1}, u_t) \in \mathbb{S}^{t+1}$ es cierto que:

$$P[\mathcal{X}_t = u_t | \mathcal{X}_0^{t-1} = (u_0, \ldots, u_{t-1})] = P[\mathcal{X}_t = u_t | \mathcal{X}_{t-1} = u_{t-1}] \tag{2.2}$$

Tenemos que, por definición de probabilidad condicionada, para cualquier proceso estocástico $\{\mathcal{X}_t\}$ y cualquier secuencia $(u_0, ..., u_{t-1}, u_t) \in \mathbb{S}^{t+1}$:

$$P[\mathcal{X}_0^t = (u_0, \ldots, u_t)] = P[\mathcal{X}_0 = u_0] \cdot \prod_{i=0}^{t-1} P[\mathcal{X}_{i+1} = u_{i+1} | \mathcal{X}_0^i = (u_0, \ldots, u_i)]$$

Sin embargo, si consideramos una cadena de Markov, entonces la fórmula anterior se reduce a:

$$P[\mathcal{X}_0^t = (u_0, \ldots, u_t)] = P[\mathcal{X}_0 = u_0] \cdot \prod_{i=0}^{t-1} P[\mathcal{X}_{i+1} = u_{i+1} | \mathcal{X}_i = u_i] \tag{2.3}$$

En (2.3) vemos la importancia del valor:

$$P[\mathcal{X}_{t+1} = u | \mathcal{X}_t = v]$$

al que podemos identificar como una función de tres variables: el estado "actual" $v \in \mathbb{S}$, el estado "siguiente" $u \in \mathbb{S}$ y el "tiempo actual" $t \in \mathbb{N}_0$. Así, teniendo en cuenta que $\mathbb{S} = \{s_1, ..., s_N\}$, definimos la probabilidad de transición:

$$a_{ij}(t) := P[\mathcal{X}_{t+1} = s_j | \mathcal{X}_t = s_i], \quad \forall t \in \mathbb{N}_0. \tag{2.4}$$

Por tanto, $a_{ij}(t)$ es la probabilidad de realizar una transición desde el estado actual s_i al estado siguiente s_j en el instante t.

Definición 2.2. Sea $\mathcal{X}_t$ una cadena de Markov, la matriz cuadrada de dimensión N, $A(t) = [a_{ij}(t)]$, es la **matriz de transición** de $\mathcal{X}_t$ en el instante t. Una cadena de Markov es **homogénea** si $A(t)$ es constante para todo $t \in \mathbb{N}_0$; en otro caso, es **no homogénea**.

Sea $\mathcal{X}_t$ una cadena de Markov que toma valores en un conjunto finito $\mathbb{S} = \{s_1, ..., s_N\}$ y sea $A(t)$ su matriz de transición en el instante t. Puesto que los elementos de una fila i de $A(t)$ son todas las probabilidades de realizar una transición desde el estado s_i, deben sumar 1 y por lo tanto, $A(t)$ es una matriz estocástica para todo t.

Para continuar con el estudio de las cadenas de Markov definimos el siguiente conjunto:

Definición 2.3. El **N-símplex estándar** es el subconjunto de $\mathbb{R}^{N+1}$ dado por:

$$\Delta^N = \{(t_1, ..., t_{N+1}) \in \mathbb{R}^{N+1} \mid \sum_{i=1}^{N+1} t_i = 1 \text{ y } t_i \geq 0 \text{ para todo } i\}$$

Puesto que para todo t, $\sum_{i=1}^{N} P[\mathcal{X}_t = s_i] = 1$, podemos representar estas probabilidades con un vector $c^t \in \Delta^{N-1}$, siendo $c^t = (P[\mathcal{X}_t = s_1], \ldots, P[\mathcal{X}_t = s_N])$.

Teorema 2.4. Sea $\{\mathcal{X}_t\}$ una cadena de Markov con valores en $\mathbb{S} = \{s_1, ..., s_N\}$ y sea $A(t)$ su matriz de transición en el instante t. Supongamos que $\mathcal{X}_0$ se distribuye de acuerdo con $c^0 = (c_1^0, \ldots, c_N^0) \in \Delta^{N-1}$, esto es:

$$P[\mathcal{X}_0 = s_i] = c_i^0, \quad \forall i \in \{1, ..., N\}$$

Entonces para todo $t \in \mathbb{N}$, $\mathcal{X}_t$ se distribuye de acuerdo con:

$$c^t = c^0 A(0) A(1) \cdots A(t-1) \tag{2.5}$$

Demostración. Vamos a demostrarlo por inducción. Para $t = 1$, por el teorema de la probabilidad total tenemos que:

$$P[\mathcal{X}_1 = s_j] = \sum_{i=1}^{N} P[\mathcal{X}_0 = s_i] \cdot P[\mathcal{X}_1 = s_j | \mathcal{X}_0 = s_i] = \sum_{i=1}^{N} c_i^0 \cdot a_{ij}(0)$$

Esto es, el producto del vector c^0 con la columna j-ésima de $A(0)$. Luego:

$$c^1 = (P[\mathcal{X}_1 = s_1], \ldots, P[\mathcal{X}_1 = s_N]) = c^0 A(0)$$

Supongamos cierta la relación (2.5) para $t-1$, es decir, $c^{t-1} = c^0 A(0) A(1) \cdots A(t-2)$. Utilizando de nuevo el mismo teorema se tiene:

$$P[\mathcal{X}_t = s_j] = \sum_{i=1}^{N} P[\mathcal{X}_{t-1} = s_i] \cdot P[\mathcal{X}_t = s_j | \mathcal{X}_{t-1} = s_i] = \sum_{i=1}^{N} c_i^{t-1} \cdot a_{ij}(t-1)$$

que es el producto del vector c^{t-1} con la columna j-ésima de $A(t-1)$. Aplicando la hipótesis de inducción:

$$c^t = c^{t-1} A(t-1) = c^0 A(0) A(1) \cdots A(t-2) A(t-1) \qquad \square$$

Ejemplo 2.1. En este ejemplo presentamos una variación del juego de cartas "black-jack". En este caso, tenemos un dado de cuatro caras con valores 0, 1, 2 y 3, y con probabilidad uniforme en cada lanzamiento. Un jugador lanza el dado de forma repetida y $\mathcal{X}_t$ representa el valor acumulado tras t lanzamientos. Si el total es igual a nueve, el jugador gana; en otro caso se considera que pierde. Podemos asumir que el resultado de cada lanzamiento es independiente de los lanzamientos anteriores.

Tenemos entonces que $\{\mathcal{X}_t\}$ toma valores en el conjunto $\mathbb{S} := \{0, 1, ..., 8, W, L\}$ de cardinalidad 11. Sea $\mathcal{Y}_t$ el resultado del lanzamiento en el instante t:

$$P[\mathcal{Y}_t = 0] = P[\mathcal{Y}_t = 1] = P[\mathcal{Y}_t = 2] = P[\mathcal{Y}_t = 3] = 1/4$$

Examinemos ahora la variación de $\mathcal{X}_t$: puesto que el valor de $\mathcal{X}_t$ se va acumulando tras cada lanzamiento, tenemos que $\mathcal{X}_t = \mathcal{X}_{t-1} + \mathcal{Y}_t$. En el caso de que $\mathcal{X}_{t-1} + \mathcal{Y}_t = 9$, consideraremos $\mathcal{X}_t = W$ (ganar) y, si $\mathcal{X}_{t-1} + \mathcal{Y}_t > 9$, consideraremos $\mathcal{X}_t = L$ (perder). Si $\mathcal{X}_t = W$ o L, consideraremos que el juego está acabado y $\mathcal{X}_{t+1} = \mathcal{X}_t$. Estas observaciones se pueden resumir en las siguientes reglas:

- Si $\mathcal{X}_{t-1} \leq 5$:

$$P[\mathcal{X}_t = \mathcal{X}_{t-1}] = P[\mathcal{X}_t = \mathcal{X}_{t-1} + 1] = P[\mathcal{X}_t = \mathcal{X}_{t-1} + 2] \\ = P[\mathcal{X}_t = \mathcal{X}_{t-1} + 3] = 1/4$$

- Si $\mathcal{X}_{t-1} = 6$:

$$P[\mathcal{X}_t = 6] = P[\mathcal{X}_t = 7] = P[\mathcal{X}_t = 8] = P[\mathcal{X}_t = W] = 1/4$$

- Si $\mathcal{X}_{t-1} = 7$:

$$P[\mathcal{X}_t = 7] = P[\mathcal{X}_t = 8] = P[\mathcal{X}_t = W] = P[\mathcal{X}_t = L] = 1/4$$

- Si $\mathcal{X}_{t-1} = 8$:

$$P[\mathcal{X}_t = 8] = P[\mathcal{X}_t = W] = 1/4$$

$$P[\mathcal{X}_t = L] = 1/2$$

- Si $\mathcal{X}_{t-1} = W$ o L:

$$P[\mathcal{X}_t = \mathcal{X}_{t-1}] = 1$$

$\{\mathcal{X}_t\}$ es una cadena de Markov pues la distribución de $\mathcal{X}_t$ depende únicamente del valor de $\mathcal{X}_{t-1}$ y no de cómo se ha alcanzado dicho valor. Notemos que las probabilidades anteriores no dependen de t, con lo cual la matriz de transición de $\mathcal{X}_t$ es una matriz fija y $\mathcal{X}_t$ es homogénea.

La matriz de transición de $\mathcal{X}_t$ es entonces una matriz 11×11 dada por:

$$A = \begin{pmatrix}
1/4 & 1/4 & 1/4 & 1/4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1/4 & 1/4 & 1/4 & 1/4 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1/4 & 1/4 & 1/4 & 1/4 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1/4 & 1/4 & 1/4 & 1/4 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1/4 & 1/4 & 1/4 & 1/4 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1/4 & 1/4 & 1/4 & 1/4 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1/4 & 1/4 & 1/4 & 1/4 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/4 & 1/4 & 1/4 & 1/4 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/4 & 1/4 & 2/4 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}$$

Es natural que el juego comience con el valor inicial igual a cero. Por lo tanto, la distribución de $\mathcal{X}_0$ está representada por $c_0 \in \mathbb{R}^{11}$ con un 1 en la primera componente y ceros en el resto. Aplicando repetidamente la fórmula (2.5) obtendremos las distribuciones de $\mathcal{X}_1$, $\mathcal{X}_2$, etc. Así, sea c_t la distribución de $\mathcal{X}_t$, tenemos:

$$c_0 = \begin{pmatrix} 1 & 0 & \dots & 0 \end{pmatrix}$$

$$c_1 = c_0 A = \begin{pmatrix} 1/4 & 1/4 & 1/4 & 1/4 & 0 & \dots & 0 \end{pmatrix}$$

$$c_2 = c_1 A = \begin{pmatrix} 1/16 & 1/8 & 3/16 & 1/4 & 3/16 & 1/8 & 1/16 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Cabe destacar que, si examinamos la distribución c_t, observamos que $P[\mathcal{X}_t \in \{0 \dots 8\}]$ tiende a cero cuando $t \to \infty$. Esto es natural pues el juego terminará eventualmente en victoria (W) o en pérdida (L) y todos los otros estados son transitorios.

2.2 ESTADOS DE UNA CADENA DE MARKOV

A partir de ahora, vamos a centrarnos en el estudio de cadenas de Markov cuyas matrices de transición son constantes. En consecuencia, las probabilidades de transición son independientes del instante t. Nos referiremos a ellas directamente como cadenas de Markov, asumiendo homogeneidad.

Está claro que los estados juegan un papel importante en el estudio de las cadenas de Markov. Para describir el comportamiento de una cadena de Markov estudiaremos las propiedades de sus estados. Antes de hacer este análisis empezaremos observando cómo evoluciona una cadena de Markov tras n instantes.

Definición 2.5. Sea $\{\mathcal{X}_t\}$ una cadena de Markov, $s_i, s_j \in \mathbb{S}$, $n, m \in \mathbb{N}_0$, denotamos:

$$P_{ij}^{m,m+n} := P[\mathcal{X}_{n+m} = s_j | \mathcal{X}_m = s_i]$$

Si $n = 0$:

$$P_{ij}^{m,m} = P[\mathcal{X}_m = s_j | \mathcal{X}_m = s_i] = \delta_{ij} = \begin{cases} 1, & \text{si } i = j \\ 0, & \text{si } i \neq j \end{cases}$$

Teorema 2.6 (Ecuación de Chapman-Kolmogorov)**.** En condiciones anteriores, sea $r \in \mathbb{N}_0$:

$$P_{ij}^{m,m+n+r} = \sum_{s_k \in \mathbb{S}} P_{ik}^{m,m+n} P_{kj}^{m+n,m+n+r}$$

Demostración.

$$\begin{aligned} P_{ij}^{m,m+n+r} &= P[\mathcal{X}_{m+n+r} = s_j | \mathcal{X}_m = s_i] \\ &= \sum_{s_k \in \mathbb{S}} P[\mathcal{X}_{m+n+r} = s_j | \mathcal{X}_{m+n} = s_k, \mathcal{X}_m = s_i] P[\mathcal{X}_{m+n} = s_k | \mathcal{X}_m = s_i] \end{aligned}$$

Aplicando la propiedad de Markov:

$$P[\mathcal{X}_{m+n+r} = s_j | \mathcal{X}_{m+n} = s_k, \mathcal{X}_m = s_i] = P[\mathcal{X}_{m+n+r} = s_j | \mathcal{X}_{m+n} = s_k] = P_{kj}^{m+n,m+n+r}$$

Por lo tanto:

$$P_{ij}^{m,m+n+r} = \sum_{s_k \in \mathbb{S}} P_{ik}^{m,m+n} P_{kj}^{m+n,m+n+r} \qquad \square$$

Notemos que por ser $\{\mathcal{X}_t\}$ homogénea, $P_{ij}^{m,m+1}$ es independiente de m, por lo que aplicando inductivamente la ecuación de Chapman-Kolmogorov, tenemos que las probabilidades $P_{ij}^{m,m+n}$ son independientes de m.

Definición 2.7. Sea $\{\mathcal{X}_t\}$ una cadena de Markov, $s_i, s_j \in \mathbb{S}$, $n, m \in \mathbb{N}_0$, se definen las probabilidades de transición en n pasos como:

$$a_{ij}^{(n)} := P_{ij}^{m,m+n} = P[\mathcal{X}_{m+n} = s_j | \mathcal{X}_m = s_i],$$

y la matriz de las probabilidades de transición en n pasos como $A^{(n)} = [a_{ij}^{(n)}]$.

Lema 2.8. La matriz de transición en n pasos cumple que $A^{(n)} = A^n, \forall n \in \mathbb{N}$. Por la observación a la proposición 1.8, $A^{(n)}$ es estocástica.

Demostración. Expresando la ecuación de Chapman-Kolmogorov en forma matricial tenemos que $A^{(n+r)} = A^{(n)} A^{(r)}$. Por ser $A^{(1)} = A$:

$$A^{(n)} = A^{(n-1)} A = A^{(n-2)} A^2 = \cdots = A^n \qquad \square$$

2.2.1 *Estados accesibles y comunicables*

Definición 2.9. El estado s_j se dice **alcanzable** o **accesible** desde el estado s_i, representado por $i \longrightarrow j$, si existe $n \in \mathbb{N}_0$ tal que $a_{ij}^{(n)} > 0$. Dos estados s_i y s_j mutuamente alcanzables se dice que son **comunicables** y se representa por $i \longleftrightarrow j$.

La definición anterior tiene el significado siguiente: si s_j es accesible desde el estado s_i, entonces existirá un $n \in \mathbb{N}_0$ tal que $P[\mathcal{X}_{m+n} = s_j] > 0$ siempre que $\mathcal{X}_m = s_i$. Es decir, empezando desde el estado s_i, hay una probabilidad positiva de que, en un número finito de transiciones, alcancemos el estado s_j.

Teorema 2.10. La propiedad de comunicación, $\longleftrightarrow$, es una relación de equivalencia sobre el conjunto de estados $\mathbb{S}$.

Demostración.

- **Reflexividad:** $a_{ii}^{(0)} = \delta_{ii} = 1 > 0$. Por tanto, $i \longleftrightarrow i$.

- **Simetría:** si $i \longleftrightarrow j$, existen $n, m \in \mathbb{N}_0$ tales que $a_{ij}^{(n)}, a_{ji}^{(m)} > 0$, escogiendo los mismos n y m, tenemos que $j \longleftrightarrow i$.
- **Transitividad:** sean $i \longleftrightarrow j$ y $j \longleftrightarrow k$:
  - Por ser $i \longleftrightarrow j$, existen $n, m \in \mathbb{N}_0$ tales que $a_{ij}^{(n)}, a_{ji}^{(m)} > 0$.
  - Por ser $j \longleftrightarrow k$, existen $r, s \in \mathbb{N}_0$ tales que $a_{jk}^{(r)}, a_{kj}^{(s)} > 0$.

  Aplicando entonces la ecuación de Chapman-Kolmogorov tenemos que:
  - $a_{ik}^{(n+r)} = \sum_{s_l \in \mathbb{S}} a_{il}^{(n)} a_{lk}^{(r)} \geq a_{ij}^{(n)} a_{jk}^{(r)} > 0$
  - $a_{ki}^{(s+m)} = \sum_{s_l \in \mathbb{S}} a_{kl}^{(s)} a_{li}^{(m)} \geq a_{kj}^{(s)} a_{ji}^{(m)} > 0$

  Por lo tanto, $i \longleftrightarrow k$. □

Como resultado, podemos dividir el conjunto de los estados $\mathbb{S}$ en clases de equivalencia atendiendo a las comunicaciones entre estados. Esto nos lleva a la siguiente definición:

Definición 2.11. Sea $\{\mathcal{X}_t\}$ una cadena de Markov sobre un conjunto finito $\mathbb{S}$, $\{\mathcal{X}_t\}$ se dice **irreducible** si hay solo una clase de equivalencia sobre $\mathbb{S}$ mediante la relación $\longleftrightarrow$.

Es decir, una cadena de Markov es irreducible si todos los estados se comunican unos con otros. Tomando el ejemplo de "blackjack", podemos apreciar que:

- Un estado $s_i \in \{0 \dots 8\}$ no es comunicable con un estado s_j con valor inferior $\implies a_{ij}^{(n)} = 0, \forall n \in \mathbb{N}_0, j < i$.
- Los estados W y L no son modificables $\implies a_{Wi}^{(n)} = a_{Li}^{(n)} = 0, \forall n \in \mathbb{N}_0, i \in \{0 \dots 8, W \text{ o } L\}$

Por simetría, cada estado es únicamente comunicable consigo mismo. En consecuencia, hay 11 clases de equivalencia en $\mathbb{S}$, una por cada estado y la cadena de Markov no es irreducible.

2.2.2 *Periodicidad de una cadena de Markov*

Definición 2.12. Sea $a_{ii}^{(n)}$ la probabilidad de transición en n pasos al estado s_i desde s_i, el periodo $\lambda(i)$ de un estado s_i es el máximo común divisor de todos los $n \in \mathbb{N}$ con $a_{ii}^{(n)} > 0$, esto es:

$$\lambda(i) := m.c.d(\{n \in \mathbb{N} \mid a_{ii}^{(n)} > 0\})$$

Si $a_{ii}^{(n)} = 0$ para todo $n \in \mathbb{N}$, entonces definimos $\lambda(i) := 0$.

A continuación indicamos que la periodicidad también es una propiedad de clase. Esto es, si el estado s_i en una clase tiene periodo T, entonces todos los estados de esa clase tienen periodo T.

Teorema 2.13. Si $i \longleftrightarrow j$, entonces $\lambda(i) = \lambda(j)$, es decir, el periodo es constante en cada clase de equivalencia.

Demostración ([20]). Si $i = j$ el resultado es trivial. Supongamos que $i \neq j$, entonces existen $n, m \in \mathbb{N}$ tales que $a_{ij}^{(n)}, a_{ji}^{(m)} > 0$, por la ecuación de Chapman-Kolmogorov:

$$a_{ii}^{(n+m)} = \sum_{s_l \in \mathbb{S}} a_{il}^{(n)} a_{li}^{(m)} \geq a_{ij}^{(n)} a_{ji}^{(m)} > 0 \Longrightarrow \lambda(i) \,|\, (n+m)$$

Sea $s \in \mathbb{N}$ tal que $a_{jj}^{(s)} > 0$:

$$a_{ii}^{(n+m+s)} \geq a_{ij}^{(n)} a_{jj}^{(s)} a_{ji}^{(m)} > 0 \Longrightarrow \lambda(i) \,|\, (n+m+s)$$

Por lo tanto, $\lambda(i) \,|\, s$. Como s es arbitrario, $\lambda(i)$ es un divisor común de $\{n \in \mathbb{N} \,|\, a_{jj}^{(n)} > 0\}$ y por definición de periodo, $\lambda(i) \,|\, \lambda(j)$. Realizando la misma discusión intercambiando los papeles de i y j, obtenemos que $\lambda(j) \,|\, \lambda(i)$. Como consecuencia, $\lambda(i) = \lambda(j)$. □

Definición 2.14. Un estado s_i se dice **no periódico** o **aperiódico** si $\lambda(i) = 1$.

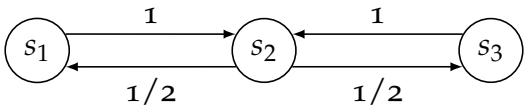
Definición 2.15. Una cadena de Markov se llama **no periódica** o **aperiódica** si todos sus estados son aperiódicos. En caso contrario, se dice **periódica**. Si la cadena de Markov es irreducible, entonces podemos hablar de **periodo de la cadena**.

La periodicidad es una propiedad que se puede apreciar fácilmente si representamos el grafo dirigido asociado a la matriz de transición. Veamos en el siguiente ejemplo:

Ejemplo 2.2. Sea una cadena de Markov con la siguiente matriz de transición:

$$A = \begin{pmatrix} 0 & 1 & 0 \\ 1/2 & 0 & 1/2 \\ 0 & 1 & 0 \end{pmatrix}$$

Si representamos el grafo asociado:



podemos ver que todos los estados tienen periodo 2. Es más, todos los estados son comunicables entre sí luego la cadena es irreducible. En definitiva, tenemos una cadena de Markov periódica con periodo 2.

2.2.3 *Tiempos de transición*

Al inicio de esta sección habíamos visto las probabilidades de transición en n pasos del estado s_i al estado s_j. El número de transiciones necesarias para ir del estado s_i al estado s_j, se denomina **tiempo de transición de s_i hasta s_j**.

Cuando $s_i = s_j$, este tiempo es justo el número de transiciones que se necesita para regresar al estado inicial s_i. En este caso, este tiempo lo denominaremos **tiempo de recurrencia para el estado s_i**.

Definición 2.16. Sea $\{\mathcal{X}_t\}$ una cadena de Markov. La variable aleatoria:

$$\tau_{ij} := \text{mín}\{t \in \mathbb{N} \mid \mathcal{X}_t = s_j \text{ dado } \mathcal{X}_0 = s_i\}$$

considerando mín $\emptyset = \infty$, representa el tiempo mínimo que la cadena necesita para ir desde s_i hasta s_j y se conoce como **tiempo de transición de s_i hasta s_j**. La variable τ_{ii} se llama **tiempo de recurrencia para el estado s_i**.

Definición 2.17. Sea $\{\mathcal{X}_t\}$ una cadena de Markov y $s_i, s_j \in \mathbb{S}$. Para todo $n \in \mathbb{N}$ se define la **probabilidad de tiempo de transición en n pasos** como la probabilidad de que $\{\mathcal{X}_t\}$ alcance s_j en n pasos partiendo desde s_i:

$$\begin{aligned} f_{ij}^{(0)} &:= 0 \\ f_{ij}^{(n)} &:= P[\tau_{ij} = n] = P[\mathcal{X}_n = s_j, \mathcal{X}_r \neq s_j, 1 \leq r \leq n-1 | \mathcal{X}_0 = s_i] \end{aligned}$$

Cuando $i = j$, hablamos de **probabilidad de tiempo recurrencia en n pasos**.

Como ya se dijo en la definición 2.16, los tiempos de transición son variables aleatorias y sus distribuciones dependen de las probabilidades de transición. En particular, $f_{ij}^{(n)}$ denota la probabilidad de que el tiempo de transición del estado s_i al s_j sea igual a n. Este tiempo de transición es n si la primera transición es del estado s_i a algún estado $s_k \neq s_j$ y el tiempo de transición del estado s_k al estado s_j es $n-1$.

Por lo tanto, estas probabilidades verifican la relación recursiva mostrada en la siguiente proposición:

Proposición 2.18. Las probabilidades de tiempo de transición en n pasos $f_{ij}^{(n)}$, con $n \in \mathbb{N}$, verifican la siguiente relación recursiva:

$$\begin{aligned} f_{ij}^{(1)} &= a_{ij} \\ f_{ij}^{(n)} &= \sum_{s_k \neq s_j} a_{ik} f_{kj}^{(n-1)}, \ n > 1 \end{aligned}$$

Para s_i y s_j fijos, las $f_{ij}^{(n)}$ son números no negativos tales que $\sum_{n=1}^{\infty} f_{ij}^{(n)} \leq 1$. Si esta suma es estrictamente menor que 1, significa que una cadena que al inicio se encuentra en el estado s_i puede no alcanzar nunca el estado s_j. Cuando la suma sí es igual a 1, las $f_{ij}^{(n)}$ pueden considerarse como la función de masa de probabilidad de la variable aleatoria tiempo de transición τ_{ij}.

Definición 2.19. Sea $\{\mathcal{X}_t\}$ una cadena de Markov, se denomina **probabilidad de tiempo de transición** a la probabilidad de que la cadena alcance s_j empezando por s_i, es decir:

$$f_{ij}^{*} := \sum_{n=1}^{\infty} f_{ij}^{(n)} \leq 1$$

Si consideramos la variable aleatoria τ_{ij}, entonces la probabilidad de que la cadena nunca alcance s_j, empezando desde s_i, es $P[\tau_{ij} = \infty] = 1 - f_{ij}^{*}$. Y está claro que f_{ii}^{*} es la probabilidad de que la cadena vuelva por lo menos una vez al estado s_i empezando por s_i. Para calcular estas probabilidades, podemos utilizar el siguiente resultado:

Teorema 2.20. Las probabilidades de tiempo de transición f_{ij}^{*} cumplen la ecuación:

$$f_{ij}^{*} = a_{ij} + \sum_{s_k \neq s_j} a_{ik} f_{kj}^{*} \tag{2.6}$$

Demostración. Por la proposición 2.18:

$$\begin{aligned} f_{ij}^{*} &= \sum_{n=1}^{\infty} f_{ij}^{(n)} = a_{ij} + \sum_{n=2}^{\infty} f_{ij}^{(n)} = a_{ij} + f_{ij}^{(2)} + f_{ij}^{(3)} + \dots \\ &= a_{ij} + \sum_{s_k \neq s_j} a_{ik} f_{kj}^{(1)} + \sum_{s_k \neq s_j} a_{ik} f_{kj}^{(2)} + \dots \\ &= a_{ij} + \sum_{s_k \neq s_j} a_{ik} \sum_{n=1}^{\infty} f_{kj}^{(n)} = a_{ij} + \sum_{s_k \neq s_j} a_{ik} f_{kj}^{*} \end{aligned}$$

□

Ejemplo 2.3. Volviendo al ejemplo de "blackjack", ya vimos que la cadena convergía a W o L con probabilidad 1. Calculemos ahora la probabilidad de ganar o de perder dado un estado inicial. Es claro que $f_{WW}^{*} = f_{LL}^{*} = 1$ pues una vez que se gana o se pierde no se modifica más el estado. Por el mismo motivo, tenemos que $f_{WL}^{*} = f_{LW}^{*} = 0$. Para calcular las otras probabilidades podemos proceder de forma recursiva sobre los posibles valores de mayor a menor y usando (2.6):

$$\begin{aligned} f_{8W}^{*} &= a_{8W} + \sum_{s_k \neq W} a_{8k} f_{kW}^{*} = a_{8W} + a_{88} f_{8W}^{*} + a_{8L} f_{LW}^{*} \\ &= \frac{1}{4} + \frac{1}{4} f_{8W}^{*} + \frac{2}{4} f_{LW}^{*} = \frac{1}{4} + \frac{1}{4} f_{8W}^{*} \end{aligned}$$

Despejando tenemos que:

$$f^*_{8W} = \frac{1}{3}$$

De forma similar:

$$f^*_{8L} = a_{8L} + \sum_{s_k \neq L} a_{8k} f^*_{kL} = a_{8L} + a_{88} f^*_{8L} + a_{8W} f^*_{WL}$$
$$= \frac{2}{4} + \frac{1}{4} f^*_{8L} + \frac{1}{4} f^*_{WL} = \frac{1}{2} + \frac{1}{4} f^*_{8L}$$

Despejando:

$$f^*_{8L} = \frac{2}{3}$$

No es de extrañar que $f^*_{8W} + f^*_{8L} = 1$, pues el juego siempre acabará ganando o perdiendo. Es más, para cualquier estado inicial s_i, es cierto que $f^*_{iW} + f^*_{iL} = 1$. Para calcular la probabilidad de ganar desde el estado 7:

$$f^*_{7W} = a_{7W} + \sum_{s_k \neq W} a_{7k} f^*_{kW} = a_{7W} + a_{77} f^*_{7W} + a_{78} f^*_{8W} + a_{7L} f^*_{LW}$$
$$= \frac{1}{4} \left(1 + \frac{1}{3} + f^*_{7W} \right) = \frac{1}{3} + \frac{1}{4} f^*_{7W}$$

Despejando:

$$f^*_{7W} = \frac{4}{9}$$

Si seguimos, obtenemos la siguiente tabla:

i	f^*_{iW}	f^*_{iL}
8	1/3	2/3
7	4/9	5/9
6	16/27	11/27
5	37/81	44/81
4	121/243	122/243
3	376/729	353/729
2	1072/2187	1115/2187
1	3289/6561	3272/6561
0	9889/19683	9794/19683

Podemos ver que la probabilidad de ganar es mayor o menor dependiendo de cada estado inicial. En particular, el estado 6 ofrece la mejor opción para ganar. Esto se debe a que desde el estado 6 no es posible perder en la siguiente ronda pero sí es posible ganar.

Ahora que conocemos la probabilidad de alcanzar un estado s_j desde un estado s_i, nos interesa saber cuánto tarda de media:

Definición 2.21. Sea $\{\mathcal{X}_t\}$ una cadena de Markov, el **tiempo de transición medio** del estado s_i al estado s_j se define por:

$$\mu_{ij} := E[\tau_{ij}] = \begin{cases} \infty, & \text{si } f_{ij}^* < 1 \\ \sum_{n=1}^{\infty} n f_{ij}^{(n)}, & \text{si } f_{ij}^* = 1 \end{cases}$$

Para el caso i=j, se llamará **tiempo medio de recurrencia** para el estado s_i.

Este tiempo representa el número medio de transiciones que se necesita para pasar de un estado s_i a otro estado s_j. Para calcularlo, podemos emplear el siguiente teorema:

Teorema 2.22. Supongamos que para todo $s_k \in \mathbb{S}$, $\mu_{kj} < \infty$. Entonces, si $s_i \in \mathbb{S}$, el tiempo de transición medio μ_{ij} satisface la ecuación:

$$\mu_{ij} = 1 + \sum_{s_k \neq s_j} a_{ik}\mu_{kj} \tag{2.7}$$

Demostración. Puesto que $\mu_{ij} < \infty$, debe ser $\mu_{ij} = \sum_{n=1}^{\infty} n f_{ij}^{(n)}$. Por la proposición 2.18:

$$\begin{aligned} \mu_{ij} &= a_{ij} + \sum_{n=2}^{\infty} n f_{ij}^{(n)} = a_{ij} + 2f_{ij}^{(2)} + 3f_{ij}^{(3)} + \dots \\ &= a_{ij} + \sum_{s_k \neq s_j} a_{ik} 2 f_{kj}^{(1)} + \sum_{s_k \neq s_j} a_{ik} 3 f_{kj}^{(2)} + \cdots = a_{ij} + \sum_{s_k \neq s_j} a_{ik} \sum_{n=2}^{\infty} n f_{kj}^{(n-1)} \\ &= a_{ij} + \sum_{s_k \neq s_j} a_{ik} \sum_{n=1}^{\infty} (n+1) f_{kj}^{(n)} = a_{ij} + \sum_{s_k \neq s_j} a_{ik} \sum_{n=1}^{\infty} \left(n f_{kj}^{(n)} + f_{kj}^{(n)} \right) \end{aligned}$$

Por hipótesis, $\mu_{kj} < \infty$, luego $\sum_{n=1}^{\infty} n f_{kj}^{(n)}$ es una serie convergente por ser una serie de términos no negativos y mayorada. De mismo modo, por definición de μ_{kj} tenemos que $f_{kj}^* = \sum_{n=1}^{\infty} f_{kj}^{(n)} = 1$ es convergente, y en consecuencia, la serie de sumas de términos es convergente con $\sum_{n=1}^{\infty} n f_{kj}^{(n)} + \sum_{n=1}^{\infty} f_{kj}^{(n)} = \sum_{n=1}^{\infty} \left(n f_{kj}^{(n)} + f_{kj}^{(n)} \right)$. Empleando esto:

$$\begin{aligned} \mu_{ij} &= a_{ij} + \sum_{s_k \neq s_j} a_{ik} \left(\sum_{n=1}^{\infty} n f_{kj}^{(n)} + \sum_{n=1}^{\infty} f_{kj}^{(n)} \right) = a_{ij} + \sum_{s_k \neq s_j} a_{ik} \left(\mu_{kj} + 1 \right) \\ &= a_{ij} + \sum_{s_k \neq s_j} a_{ik}\mu_{kj} + \sum_{s_k \neq s_j} a_{ik} = \sum_{s_k \in \mathbb{S}} a_{ik} + \sum_{s_k \neq s_j} a_{ik}\mu_{kj} \\ &= 1 + \sum_{s_k \neq s_j} a_{ik}\mu_{kj} \end{aligned}$$

□

Los conceptos anteriores son extensibles a subconjuntos de $\mathbb{S}$. Si tomamos el ejemplo de "blackjack", podemos considerar el subconjunto $E = \{W, L\}$. Si $\mathcal{X}_t \in E$, implica que el juego ha terminado. Está claro que desde cualquier estado s_i, $f_{iE}^* = 1$ y $\mu_{iE} < \infty$. Podemos reescribir (2.7) como:

$$\mu_{iE} = 1 + \sum_{s_k \notin E} a_{ik}\mu_{kE}$$

Es claro que $\mu_{WE} = \mu_{LE} = 1$. Si $\mathcal{X}_0 = 8$:

$$\mu_{8E} = 1 + a_{88}\mu_{8E} = 1 + \frac{1}{4}\mu_{8E}$$

Despejando:

$$\mu_{8E} = \frac{4}{3}$$

Para $\mathcal{X}_0 = 7$:

$$\mu_{7E} = 1 + a_{77}\mu_{7E} + a_{78}\mu_{8E} = 1 + \frac{1}{4}\mu_{7E} + \frac{1}{4}\frac{4}{3} = \frac{4}{3} + \frac{1}{4}\mu_{7E}$$

Despejando:

$$\mu_{7E} = \frac{16}{9}$$

Si seguimos procediendo de esta manera, obtenemos la siguiente tabla:

i	μ_{iE}	$\approx$
8	4/3	1.333
7	16/9	1.778
6	64/27	2.370
5	256/81	3.160
4	916/243	3.700
3	3232/729	4.433
2	11200/2187	5.121
1	37888/6561	5.775
0	126820/19683	6.443

que representa el número de rondas que se necesitan para terminar el juego iniciando desde cada estado.

2.2.4 *Estados recurrentes y transitorios*

Vamos a distinguir ahora entre diversos tipos de estados, lo cual nos va a permitir dividir el espacio de estados en varios grupos. Para ello vamos a utilizar la probabilidad de tiempo de transición f_{ij}^*:

Definición 2.23. Un estado $s_i \in \mathbb{S}$ se dice **recurrente** si y sólo si $f_{ii}^* = 1$, en otro caso, se llamará **transitorio**. Si todos los estados son recurrentes, entonces se hablará de una cadena de Markov recurrente, en otro caso, transitoria.

Un estado es transitorio si, después de haber entrado a este estado la cadena puede no regresar nunca a él. Por consiguiente, el estado s_i es transitorio si y sólo si existe un estado $s_j \neq s_i$ que es accesible desde s_i pero no viceversa. Así, si el estado s_i es transitorio y la cadena alcanza dicho estado, existe una probabilidad positiva de que la cadena se mueva al estado s_j y no regrese nunca al estado s_i.

La otra posibilidad es que iniciando en el estado s_i, la cadena siempre regrese a ese estado. En este caso, decimos que el estado es recurrente.

Estas propiedades también se pueden definir utilizando las probabilidades de transición en n pasos:

Proposición 2.24. Sea $s_i \in \mathbb{S}$:

- s_i es recurrente si y sólo si $\sum_{n=1}^{\infty} a_{ii}^{(n)} = \infty$.
- s_i es transitorio si y sólo si $\sum_{n=1}^{\infty} a_{ii}^{(n)} < \infty$.

Demostración. Para demostrarlo, primero veamos la relación que existe entre $a_{ij}^{(n)}$ y $f_{ij}^{(n)}$, recordemos que $a_{ij}^{(n)}$ es la probabilidad de que se produzca una transición de s_i a s_j en n pasos, lo cual incluye posibles transiciones en los pasos $1, 2, 3, \ldots n-1$. De esta forma:

$$a_{ij}^{(1)} = f_{ij}^{(1)}$$
$$a_{ij}^{(2)} = f_{ij}^{(2)} + f_{ij}^{(1)} a_{jj}^{(1)}$$
$$a_{ij}^{(3)} = f_{ij}^{(3)} + f_{ij}^{(2)} a_{jj}^{(1)} + f_{ij}^{(1)} a_{jj}^{(2)}$$

Así, en general:

$$a_{ij}^{(n)} = \sum_{k=1}^{n} f_{ij}^{(k)} a_{jj}^{(n-k)} = f_{ij}^{(n)} + f_{ij}^{(n-1)} a_{jj}^{(1)} \cdots + f_{ij}^{(1)} a_{jj}^{(n-1)} \tag{2.8}$$

Usando esto:

$$\begin{aligned} \sum_{n=1}^{\infty} a_{ii}^{(n)} &= \sum_{n=1}^{\infty} \left[f_{ii}^{(n)} + f_{ii}^{(n-1)} a_{ii}^{(1)} \cdots + f_{ii}^{(1)} a_{ii}^{(n-1)} \right] \\ &= \sum_{n=1}^{\infty} f_{ii}^{(n)} + \sum_{n=1}^{\infty} f_{ii}^{(n)} \sum_{k=1}^{\infty} a_{ii}^{(k)} = f_{ii}^* + f_{ii}^* \sum_{n=1}^{\infty} a_{ii}^{(n)} \end{aligned}$$

Despejando:

$$\sum_{n=1}^{\infty} a_{ii}^{(n)} = \frac{f_{ii}^*}{1 - f_{ii}^*}$$

Luego esta claro que:

- s_i es recurrente $\iff f_{ii}^* = 1 \iff \sum_{n=1}^{\infty} a_{ii}^{(n)} = \frac{f_{ii}^*}{1 - f_{ii}^*} = \infty.$
- s_i es transitorio $\iff f_{ii}^* < 1 \iff \sum_{n=1}^{\infty} a_{ii}^{(n)} = \frac{f_{ii}^*}{1 - f_{ii}^*} < \infty.$

Con esta proposición, ya podemos demostrar que las propiedades de recurrencia y transitoriedad son de clase:

Teorema 2.25. Sean $s_i, s_j \in \mathbb{S}$, si s_i es recurrente e $i \longrightarrow j$, entonces $f_{ji}^* = 1$, s_j es recurrente y $f_{ij}^* = 1$.

La siguiente demostración se puede consultar en [20, Página 38]:

Demostración. Si $i \longrightarrow j, \exists n \in \mathbb{N}$ tal que $a_{ij}^{(n)} > 0$. Por (2.8), $\exists k \in \{1, \ldots, n\}$ tal que $f_{ij}^{(k)} > 0$, luego $f_{ij}^* > 0$. Si fuese $f_{ji}^* < 1$, con probabilidad $1 - f_{ji}^* > 0$ partiendo desde s_j no pasaríamos nunca por s_i. Así que con probabilidad al menos $f_{ij}^*(1 - f_{ji}^*) > 0$ saliendo de s_i no volveríamos a pasar nunca por s_i, lo cual contradice con que s_i sea recurrente.

Puesto que $f_{ji}^* > 0$, existirá $m \in \mathbb{N}$ tal que $f_{ji}^{(m)} > 0$, por (2.8), $a_{ji}^{(m)} > 0$, luego también $j \longrightarrow i$ e $i \longleftrightarrow j$.

Veamos ahora que s_j ha de ser recurrente: puesto que $i \longleftrightarrow j$, existen $n, m \in \mathbb{N}$ tales que $a_{ij}^{(n)} > 0$ y $a_{ji}^{(m)} > 0$. Para todo $r \geq m + n$ se tiene:

$$a_{jj}^{(r)} \geq a_{ji}^{(m)} a_{ii}^{(r-m-n)} a_{ij}^{(n)}$$

Por lo tanto:

$$\sum_{r=1}^{\infty} a_{jj}^{(r)} \geq \sum_{r=1}^{n+m} a_{jj}^{(r)} + \sum_{r=n+m+1}^{\infty} a_{ji}^{(m)} a_{ii}^{(r-m-n)} a_{ij}^{(n)} = \sum_{r=1}^{n+m} a_{jj}^{(r)} + a_{ji}^{(m)} a_{ij}^{(n)} \sum_{r=1}^{\infty} a_{ii}^{(r)}$$

Puesto que s_i es recurrente, $\sum_{r=1}^{\infty} a_{ii}^{(r)} = \infty$ luego también $\sum_{r=1}^{\infty} a_{jj}^{(r)} = \infty$. Por la proposición anterior, s_j es recurrente.

Finalmente, $f_{ij}^* = 1$ es clara utilizando que $j \longrightarrow i$, s_j es recurrente y la primera parte de esta demostración. □

De la demostración, podemos apreciar también que si s_i es recurrente y s_j es transitoria entonces no es posible acceder desde s_i a s_j ($i \not\longrightarrow j$). En consecuencia:

Corolario 2.26. Sean $s_i, s_j \in \mathbb{S}$ con $i \longleftrightarrow j$, entonces o son ambos transitorios o son ambos recurrentes.

Vamos a presentar ahora un tipo especial de estado recurrente:

Definición 2.27. Un estado $s_i \in \mathbb{S}$ se llamará **absorbente** si $a_{ii} = 1$.

Si una cadena de Markov ha alcanzado un estado absorbente s_i, permanecerá allí para siempre pues $a_{ij} = 0$ para todo $s_j \neq s_i$. En consecuencia la clase de equivalencia $[s_i]$ estará formado únicamente por s_i.

Con los resultados anteriores seremos capaces de dividir el espacio de estados $\mathbb{S}$ en dos subconjuntos disjuntos: uno constituido por los estados transitorios y otro por los estados recurrentes. Los estados transitorios son inaccesibles desde los recurrentes. Los estados recurrentes se pueden dividir de manera única en clases de equivalencia mediante la relación de equivalencia $i \longleftrightarrow j$. Notemos que si s_i y s_j están en clases distintas entonces $i \not\to j$ y $j \not\to i$ por el teorema 2.25.

De acuerdo con este resultado, podemos hacer una reordenación de los estados de $\mathbb{S}$ (es decir, una reordenación de las filas y columnas de la matriz de transición) que coloque los estados transitorios al final y agrupe los estados de cada una de las clases de equivalencia de los estados recurrentes. Tendremos así, la siguiente estructura para la matriz de transición de una cadena de Markov:

$$\left(\begin{array}{cccc|c} P_1 & 0 & \dots & 0 & \\ 0 & P_2 & \dots & 0 & \\ \vdots & \vdots & \ddots & \vdots & 0 \\ 0 & 0 & \dots & P_r & \\ \hline \multicolumn{4}{c|}{R} & Q \end{array}\right)$$

donde las submatrices P_i están asociadas a cada clase de equivalencia, R proporciona las probabilidades para pasar desde los estados transitorios a los recurrentes y Q da las probabilidades entre estados transitorios.

2.3 COMPORTAMIENTO ASINTÓTICO DE UNA CADENA DE MARKOV

Aparte de las definiciones del sección anterior, que nos permiten calcular directamente probabilidades relacionadas con los estados, también nos interesa el comportamiento de una cadena de Markov a largo plazo. Para ello, vamos a estudiar la matriz de transición en n pasos cuando n tiende a infinito.

En primer lugar, vamos a presentar una distribución especial:

Definición 2.28. Sea A la matriz de transición de una cadena de Markov $\{\mathcal{X}_t\}$, diremos que $\pi \in \Delta^{N-1}$ es una distribución estacionaria si $\pi A = \pi$.

Supongamos que $\{\mathcal{X}_t\}$ es una cadena de Markov con matriz de transición A. Hemos visto que, dependiendo de la distribución inicial, el proceso resultante $\{\mathcal{X}_t\}$ evoluciona de forma distinta a lo largo del tiempo. Pero, si existe una distribución

estacionaria π y en un instante t' se da $c^{t'} = (P[\mathcal{X}_{t'} = s_1, \ldots, \mathcal{X}_{t'} = s_N]) = \pi$, entonces $\forall t \in \mathbb{N}, t \geq t', c^t = \pi$.

Notemos también que π es un vector propio a la izquierda de A con valor propio 1, que sabemos que siempre existe por la proposición 1.8. Una distribución estacionaria se puede entender como un punto de equilibrio de la cadena. Es posible que existan varias distribuciones estacionarias pero bajo ciertas condiciones, podemos afirmar que existe una única distribución estacionaria y que la cadena converge hacia ella. Para justificar estas condiciones, necesitamos introducir algunas propiedades de las cadenas irreducibles y aperiódicas (véase [11, Capítulo 4]):

Proposición 2.29. Sea $\{\mathcal{X}_t\}$ una cadena de Markov aperiódica, entonces existe $H \in \mathbb{N}$ tal que $a_{ii}^{(m)} > 0$ para todo estado $s_i \in \mathbb{S}$ y todo número natural $m \geq H$.

Para demostrarlo utilizaremos el siguiente lema de teoría de números:

Lema 2.30. Sea D un conjunto de enteros no negativos tal que:

1. es cerrado para la suma, es decir, si $a, b \in D \implies a + b \in D$
2. $m.c.d(D) = 1$

entonces D contiene a todos los enteros no negativos salvo un subconjunto finito. En consecuencia, existe $H \in \mathbb{N}$ tal que para todo número natural $m \geq H$, $m \in D$.

Demostración. Véase [11, Página 26]. □

Demostración (Proposición 2.29)**.** Para cada estado s_i, consideramos:

$$D_i = \{n \in \mathbb{N} \mid a_{ii}^{(n)} > 0\}$$

puesto que la cadena es aperiódica, todos los estados son aperiódicos y $m.c.d(D_i) = 1$. Sean t, s elementos de D_i, luego $a_{ii}^{(t)} > 0$ y $a_{ii}^{(s)} > 0$. Como:

$$a_{ii}^{(t+s)} \geq a_{ii}^{(t)} a_{ii}^{(s)} > 0 \implies t + s \in D_i \implies D_i \text{ es cerrado para la suma}$$

por el lema anterior, existe $H_i \in \mathbb{N}$ tal que $\forall m \geq H_i, m \in D_i$. Puesto que el espacio de estados $\mathbb{S}$ es finito, existe $H = \text{máx}\{H_i \mid i \in \{1, \ldots N\}\}$ tal que $\forall m \geq H, m \in \mathbb{N}, a_{ii}^{(m)} > 0, \forall s_i \in \mathbb{S}$. □

Proposición 2.31. Sea $\{\mathcal{X}_t\}$ una cadena de Markov irreducible y aperiódica, entonces existe $M \in \mathbb{N}$ tal que $a_{ij}^{(m)} > 0, \forall s_i, s_j \in \mathbb{S}$ y $\forall m \in \mathbb{N}, m \geq M$.

Demostración. Puesto que la cadena es aperiódica, por la proposición 2.29, existe $H \in \mathbb{N}$ tal que $a_{ii}^{(m)} > 0$ para todo estado $s_i \in \mathbb{S}$ y todo número natural $m \geq H$.

Además, como la cadena es irreducible, para todo par de estados s_i, s_j, existe $n_{ij} \in \mathbb{N}$ tal que $a_{ij}^{(n_{ij})} > 0$. Por lo tanto, para $m \geq H + n_{ij}$:

$$a_{ij}^{(m)} \geq a_{ii}^{(m-n_{ij})} a_{ij}^{(n_{ij})} > 0$$

Puesto que el espacio de los estados es finito, basta elegir $M = H + \text{máx}\{n_{ij} \mid i, j \in \{1, \dots N\}\}$. □

Definición 2.32. Una cadena de Markov $\{\mathcal{X}_t\}$ es **regular** si su matriz de transición A es primitiva. Esto es, existe $k \in \mathbb{N}$ tal que A^k tiene sólo elementos estrictamente positivos.

Corolario 2.33. Una cadena de Markov es regular si y sólo si es irreducible y aperiódica.

Demostración. Si la cadena es irreducible y aperiódica, entonces es regular como consecuencia directa de la proposición 2.31.

Supongamos que la cadena es regular, existe entonces $k \in \mathbb{N}$ tal que $a_{ij}^{(k)} > 0, \forall s_i, s_j \in \mathbb{S}$. En consecuencia, todos los estados son comunicables y la cadena es irreducible. Para demostrar que la cadena es aperiódica basta calcular el periodo de un estado s_i ahora que sabemos que es irreducible. Notemos que:

$$a_{ii}^{(3k)} = \sum_{s_j \in \mathbb{S}} a_{ij}^{(k)} a_{jj}^{(k)} a_{ji}^{(k)} > 0$$

Por otro lado, puesto que $\sum_{s_j \in \mathbb{S}} a_{ij}^{(k+1)} = 1$, existe $a_{ij}^{(k+1)} > 0$. Por lo tanto:

$$a_{ii}^{(3k+1)} = \sum_{s_j \in \mathbb{S}} a_{ij}^{(k+1)} a_{jj}^{(k)} a_{ji}^{(k)} > 0$$

Puesto $m.c.d(3k, 3k+1) = 1$, tenemos que la cadena es aperiódica. □

Las anteriores propiedades nos proporcionan una condición suficiente para que exista una única distribución estacionaria, que se muestra en el siguiente teorema:

Teorema 2.34. Si $\{\mathcal{X}_t\}$ es una cadena de Markov irreducible, entonces existe una única distribución estacionaria π asociada a $\{\mathcal{X}_t\}$. Si además la cadena es aperiódica, para cada distribución inicial c^0:

$$\lim_{t \to \infty} c^t = \lim_{t \to \infty} c^0 A^t = \pi$$

Por este teorema, podemos asegurar que si una cadena es irreducible, entonces existe un único punto de equilibrio. Es más, si la cadena es también aperiódica, entonces a largo plazo convergerá hacia dicho equilibrio, sea cual sea la distribución inicial. Para la demostración, necesitamos algunos resultados sobre matrices:

Definición 2.35. Dada A una matriz cuadrada real de dimensión $N \times N$, se llama grafo de A ($graf(A)$) a la gráfica dirigida sobre N nodos $\{\alpha_1, \alpha_2, \ldots, \alpha_N\}$ tal que si $a_{ij} > 0$ entonces existe una flecha desde α_i hacia α_j.

Definición 2.36. Se dice que $graf(A)$ está **fuertemente conectado** si para toda pareja de nodos α_i, α_j existe un camino que los conecta, es decir, se puede ir del nodo α_i a α_j en un número finito de pasos.

Definición 2.37. Sea A una matriz cuadrada positiva, se dice que A es **irreducible** si $graf(A)$ está fuertemente conectado.

Proposición 2.38. Sea $\{\mathcal{X}_t\}$ una cadena de Markov irreducible, entonces la matriz de transición asociada A es una matriz irreducible.

Demostración. Por ser la cadena irreducible, para todo par de estados s_i, s_j existe $n \in \mathbb{N}$ tal que $a_{ij}^{(n)} > 0$. Aplicando inductivamente la ecuación de Chapman-Kolmogorov, existe al menos una secuencia de productos con n términos $a_{ik} \cdots a_{lj} > 0$. En consecuencia, para todo par de nodos α_i, α_j existe un camino que los conecta. □

Teorema 2.39. Sea A una matriz irreducible, entonces:

1. Existe un valor propio positivo y simple λ_1 tal que para todo valor propio λ, se tiene que $\lambda_1 \geq \lambda$.
2. Existe un único vector propio asociado a λ_1 con todos las componentes estrictamente positivas.
3. Todos los valores propios con módulo igual a λ_1 son simples.

Demostración. Véase [17, Página 263, Teorema 7.13]. □

Con el anterior teorema y la proposición 1.8, tenemos que para una cadena de Markov irreducible el vector propio asociado a 1 es único y por lo tanto existe una única distribución estacionaria. Además, por el teorema 2.39, sabemos que cada componente de la distribución estacionaria es estrictamente positiva. Para probar la segunda parte del teorema 2.34 necesitamos algunos resultados más:

Teorema 2.40 (Teorema de Perron-Frobenius)**.** Sea A una matriz primitiva, entonces:

1. A tiene un valor propio λ_1 real, estrictamente positivo y dominante, esto es:

$$|\lambda_i| < \lambda_1, \ \forall \lambda_i \in \sigma(A) \setminus \{\lambda_1\}$$

   y $\rho(A) = \lambda_1$.
2. Se puede tomar un vector propio v_1 asociado al valor propio λ_1 con todas las componentes estrictamente positivas.

Demostración. Véase [17, Página 202]. □

Como consecuencia de este teorema tenemos el siguiente corolario:

Corolario 2.41. Sea A una matriz primitiva de dimensión $N \times N$, λ_1 su valor propio real, estrictamente positivo y dominante y v_1 vector propio asociado a λ_1. Entonces, sea X_0 un vector de dimensión N con todas las componentes no negativas y al menos una componente estrictamente positiva:

$$\lim_{n\to\infty} \frac{1}{||X_0 A^n||_1} X_0 A^n = \frac{1}{||v_1||_1} v_1$$

Demostración. Véase [17, Página 201, Teorema 5.19]. □

Si una cadena de Markov es irreducible y aperiódica, sabemos que es regular. Por lo tanto, usando el corolario anterior (pues $c^0 \in \Delta^{N-1} \subset \mathbb{R}^N$) tenemos que:

$$\lim_{t\to\infty} \frac{1}{||c^0 A^t||_1} c^0 A^t = \frac{1}{||\pi||_1} \pi$$

Puesto que para todo t, $c^t = c^0 A^t \in \Delta^{N-1}$ y $\pi \in \Delta^{N-1}$, sus normas tienen valor 1. Así, hemos probado la segunda afirmación del teorema 2.34. Veamos en el siguiente ejemplo [6, Página 102] la utilidad del teorema 2.34:

Ejemplo 2.4. Supongamos que una ciudad tiene tres cadenas de supermercados $\{A, B, C\}$. Considerando un determinado periodo de tiempo, observamos que, por diferentes razones como el precio, la calidad, etc, algunos habitantes deciden cambiar de cadena.

Para estudiar este cambio a largo plazo, se utiliza la cadena $\{\mathcal{X}_t\}$ = la cadena de supermercado escogida por el cliente en el día t. Se supone también que la proporción de clientes que cambian de supermercado al día es constante con la siguiente matriz de transición:

$$A = \begin{pmatrix} 0{,}8 & 0{,}1 & 0{,}1 \\ 0{,}2 & 0{,}7 & 0{,}1 \\ 0{,}1 & 0{,}3 & 0{,}6 \end{pmatrix}$$

Está claro que la cadena es regular pues la matriz de transición sólo contiene elementos positivos, el vector propio asociado a 1 es:

$$\pi = \begin{pmatrix} 0{,}45 & 0{,}35 & 0{,}2 \end{pmatrix}$$

Lo cual nos indica que a largo plazo, el 45 % de los clientes se quedarán en el supermercado A, el 35 % en el supermercado B y el 20 % en el supermercado C.

Notemos que para este resultado no ha sido necesario saber cual es la proporción de clientes que acuden a cada supermercado en el momento del que se inicia el estudio. Para comprobar la previsión anterior, podemos suponer una distribución inicial alejada de la distribución estacionaria:

$$c^0 = \begin{pmatrix} 0{,}1 & 0{,}2 & 0{,}7 \end{pmatrix}$$

Tras 5 días:

$$c^5 = c^0 A^5 = \begin{pmatrix} 0{,}3995 & 0{,}3848 & 0{,}2156 \end{pmatrix}$$

Tras 14 días:

$$c^{14} = c^0 A^{14} = \begin{pmatrix} 0{,}44937 & 0{,}3506 & 0{,}20003 \end{pmatrix}$$

3

MODELOS OCULTOS DE MARKOV

En este capítulo, estudiaremos un tipo especial de proceso estocástico llamado modelo oculto de Markov (HMM). Empezaremos introduciendo estos modelos, para después seguir discutiendo sobre los problemas y algoritmos que conllevan. En adelante, utilizaremos la abreviatura HMM para referirnos a los modelos ocultos de Markov.

Este capítulo se basa principalmente en [15], [18, Capítulo 2] y [16].

3.1 EXTENSIÓN A MODELOS OCULTOS DE MARKOV

Hasta ahora hemos considerado cadenas de Markov en las cuales cada estado es un evento observable (o material). Este modelo es demasiado restrictivo para aplicar a numerosos problemas en los que no podemos observar directamente los acontecimientos que nos interesan. Para estudiar este tipo de problema extendemos el concepto de modelo de Markov para incluir los casos en los que la observación es una función probabilística del estado. Como resultado, obtenemos un proceso estocástico conjunto formado por una cadena de Markov homogénea que no es observable (es decir, oculta) pero que produce una serie de consecuencias perceptibles mediante otro proceso estocástico. Es decir, tendremos una cadena $\{\mathcal{X}_t\}_{t=0}^{\infty}$ que representa los sucesos ocultos y un proceso $\{\mathcal{Y}_t\}_{t=0}^{\infty}$ que representa las consecuencias observadas de $\{\mathcal{X}_t\}$.

Para aclarar esta idea, consideramos el siguiente ejemplo:

Ejemplo 3.1. Un guardia de seguridad trabaja en una instalación subterránea, sin conexión con el exterior. Cada día, no puede saber si está lloviendo o no, pero por las mañanas ve llegar al director con o sin paraguas.

En este caso, $\mathcal{X}_t$ indica si llueve o no en el día t e $\mathcal{Y}_t$ indica si el director lleva o no paraguas. Está claro que $\mathcal{Y}_t$ es consecuencia directa de $\mathcal{X}_t$ y asumiendo que la posibilidad de que llueva en un día determinado depende únicamente del tiempo del día anterior, tenemos que $\{\mathcal{X}_t\}$ es una cadena de Markov homogénea.

Una representación común de la estructura de HMM es la siguiente:

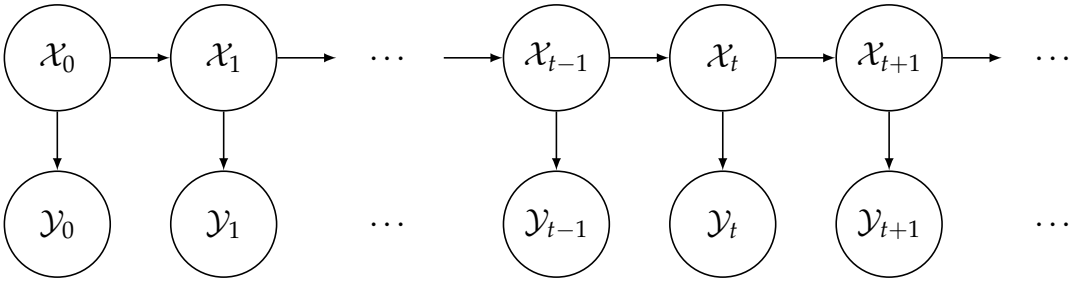


Figura 1: Estructura de un HMM

La representación anterior y el ejemplo 3.1 nos dan una idea de lo que es un HMM. Para concretarlo, damos la siguiente definición:

Definición 3.1. Sean $\{\mathcal{X}_t\}_{t=0}^{\infty}$ e $\{\mathcal{Y}_t\}_{t=0}^{\infty}$ procesos estocásticos que toman valores en conjuntos finitos $\mathbb{S} = \{s_1, \ldots, s_N\}$ y $\mathbb{V} = \{v_1, \ldots, v_M\}$ respectivamente. El proceso conjunto $\{(\mathcal{X}_t, \mathcal{Y}_t)\}$ es un modelo de Markov oculto si:

- $\{\mathcal{X}_t\}$ es una cadena de Markov homogénea.
- $P[\mathcal{Y}_t = y_t | \mathcal{X}_0 = x_0, \ldots, \mathcal{X}_t = x_t, \mathcal{Y}_0 = y_0, \ldots, \mathcal{Y}_{t-1} = y_{t-1}] = P[\mathcal{Y}_t = y_t | \mathcal{X}_t = x_t]$, es decir, la observación en el instante t depende únicamente del estado que se encuentra en dicho momento. Llamaremos a esta probabilidad la probabilidad de emisión.

En distintas fuentes, en lugar de dar una definición explícita de HMM nombran los elementos que lo caracterizan. Puesto que son de enorme importancia, vamos a presentarlos a continuación. Un HMM se caracteriza por:

1. El conjunto de estados $\mathbb{S} = \{s_1, \ldots, s_N\}$ que, a pesar de no ser observables, suelen conllevar un significado físico del problema.
2. El conjunto de posibles observaciones $\mathbb{V} = \{v_1, \ldots, v_M\}$ que corresponden a las salidas materiales del sistema.
3. La matriz de transición A asociada a $\{\mathcal{X}_t\}$ con:
$$a_{ij} = P[\mathcal{X}_{t+1} = s_j | \mathcal{X}_t = s_i]$$
4. La matriz de emisiones $B \in [0,1]^{N \times M}$ estocástica con:
$$b_{jk} = P[\mathcal{Y}_t = v_k | \mathcal{X}_t = s_j] \text{ para todo } t \geq 0$$
Para reducir la confusión, en adelante utilizaremos la notación $b_{s_j}(v_k)$ para referirnos a estas probabilidades.
5. Una distribución inicial $\pi \in \Delta^{N-1}$ tal que:
$$P[\mathcal{X}_0 = s_i] = \pi_i$$

Es frecuente (véase [15]) utilizar la notación:

$$\lambda = (A, B, \pi) \tag{3.1}$$

para representar un HMM.

3.2 LOS TRES PROBLEMAS BÁSICOS DE LOS HMMS

A partir de los conceptos anteriores, podemos identificar 3 entidades: el modelo, la secuencia de observaciones o de salidas y la secuencia de estados. Existen 3 problemas básicos de interés que involucran a estas entidades:

1. Dado un modelo, ¿cuál es la probabilidad de observar una secuencia particular de salidas? En este caso no nos interesa la secuencia de estados, tan sólo queremos conocer la probabilidad de que ocurran ciertas observaciones.
2. Dado un modelo y una secuencia de salidas, ¿cuál es la secuencia de estados más probable que generar dichas salidas?
3. Dada una secuencia de salidas y conocido el espacio de estados, ¿cuál es el modelo que maximiza la probabilidad de observar dichas salidas?

El **problema 1** es un problema de evaluación donde calculamos la probabilidad de observar una secuencia de salidas conocido el modelo. También nos permite conocer si el modelo se ajusta a dicha secuencia. Esto puede ser útil, por ejemplo, si estamos considerando varios modelos posibles. En ese caso, la solución del **problema 1** nos permitiría elegir el modelo que más se ajuste a las observaciones.

En el caso del ejemplo 3.1, un ejemplo del **problema 1** podría ser calcular la probabilidad de que el director lleve paraguas dos días seguidos y no en el tercero.

El **problema 2** es donde intentamos cubrir la parte oculta del modelo, es decir, encontrar la secuencia "correcta" de estados. Está claro que en realidad no existe una secuencia "correcta". Por ello, utilizaremos criterios de optimalidad para resolver este problema de la mejor manera posible.

En el caso del ejemplo 3.1, si se observa el paraguas en los dos primeros días y no en el tercero, parece lógico pensar que ha llovido en esos dos primeros días y no en el tercero. Veremos que es efectivamente así resolviendo este problema mediante el algoritmo de Viterbi.

En el **problema 3** pretendemos optimizar los parámetros del modelo dada una secuencia de salidas. La secuencia de observaciones usada para ajustar los parámetros se suele denominar secuencia de entrenamiento. El entrenamiento de HMM es importante, pues así podemos adaptar los parámetros a los datos percibidos. A partir de los resultados, podemos formular mejores modelos para describir fenómenos reales.

Como ejemplo, vamos considerar un problema de reconocimiento de voz, una de las aplicaciones más conocidas de HMM. Podemos utilizar las soluciones del **problema 3** para entrenar un HMM λ_0 (usando la notación (3.1)) que reconoce la pronunciación de la palabra "no" y otro HMM λ_1 que reconoce la pronunciación del "sí". Entonces dada la pronunciación de una palabra desconocida, podemos calcular la probabilidad de dicha pronunciación en cada uno de los dos modelos usando la solución del **problema 1**. Así, podemos determinar si la palabra se asemeja más al "sí" o al "no".

En las siguientes subsecciones vamos a intentar solucionar estos problemas siguiendo principalmente la metodología descrita en [15]. Notemos que, por ser $\{\mathcal{X}_t\}$ homogénea y por el hecho de que $\mathcal{Y}_t$ depende únicamente del estado en el instante t, el instante en el que se comienza a observar las salidas es indiferente. Por lo tanto, podemos suponer siempre que las observaciones inician en el instante $t = 0$.

3.2.1 *Solución al problema 1*

Queremos calcular la probabilidad de una secuencia de observaciones concreta, $O = (O_0, O_1, \ldots, O_r)$ conocido el modelo. La forma más directa de hacerlo es mediante enumeración de todas las posibles secuencias de estados de longitud $r+1$. Consideramos una de ellas:

$$Q = (q_0, q_1, \ldots, q_r) \in \mathbb{S}^{r+1}$$

siendo q_0 el estado inicial. Para facilitar la escritura, introducimos la siguiente notación:

$$\mathcal{Y}_k^l := (\mathcal{Y}_k, \mathcal{Y}_{k+1}, \ldots, \mathcal{Y}_{l-1}, \mathcal{Y}_l)$$

Además, dada la secuencia Q, para representar las probabilidades de transición a partir de los estados de la secuencia pondremos:

$$a_{q_i q_j} = P[\mathcal{X}_{t+1} = q_j | \mathcal{X}_t = q_i]$$

Usando la notación anterior, la probabilidad de la secuencia de observaciones O dada la secuencia de estados Q es:

$$P[\mathcal{Y}_0^r = O | \mathcal{X}_0^r = Q] = \prod_{t=0}^{r} P[\mathcal{Y}_t = O_t | \mathcal{X}_t = q_t]$$

donde aplicamos la independencia entre las observaciones. Por lo tanto:

$$P[\mathcal{Y}_0^r = O | \mathcal{X}_0^r = Q] = b_{q_0}(O_0) \cdot b_{q_1}(O_1) \cdots b_{q_r}(O_r)$$

Y la probabilidad de dicha secuencia de estados Q se puede calcular como:

$$P[\mathcal{X}_0^r = Q] = \pi_{q_0} \cdot a_{q_0 q_1} \cdot a_{q_1 q_2} \cdots a_{q_{r-1} q_r}$$

Es claro que:

$$P[\mathcal{Y}_0^r = O, \mathcal{X}_0^r = Q] = P[\mathcal{Y}_0^r = O | \mathcal{X}_0^r = Q] \cdot P[\mathcal{X}_0^r = Q]$$

Y la probabilidad de O se puede obtener sumando esta probabilidad mediante todas las posibles secuencias de estados:

$$P[\mathcal{Y}_0^r = O] = \sum_{Q \in \mathbb{S}^{r+1}} P[\mathcal{Y}_0^r = O | \mathcal{X}_0^r = Q] \cdot P[\mathcal{X}_0^r = Q]$$

$$= \sum_{(q_0,q_1,\ldots,q_r)\in S^{r+1}} \pi_{q_0} \cdot b_{q_0}(O_0) \cdot a_{q_0q_1} \cdot b_{q_1}(O_1) \cdots a_{q_{r-1}q_r} \cdot b_{q_r}(O_r)$$

Esta manera de calcular, involucra un orden de $2 \cdot (r+1) \cdot N^{(r+1)}$ operaciones, puesto que existen $N^{(r+1)}$ posibles secuencias de estados y para cada una de estas secuencias hay que realizar $2 \cdot (r+1)$ cálculos. Esto hace imposible calcular esta probabilidad, pues incluso para un modelo de 5 estados, si se quiere calcular la probabilidad de una secuencia con 100 observaciones ($r = 99$) se necesitarían $2 \cdot 100 \cdot 5^{100} \approx 10^{72}$ operaciones. Afortunadamente, existe una forma más eficiente de resolver el **problema 1** y es mediante el conocido como **algoritmo de avance-retroceso**.

Definición 3.2. Definimos la **variable de avance** $\alpha_t(i)$ como la probabilidad de observar la secuencia parcial $(O_0, O_1, \ldots, O_t)$ y que el estado en el instante t sea s_i:

$$\alpha_t(i) = P[\mathcal{Y}_0^t = (O_0, \ldots, O_t), \mathcal{X}_t = s_i]$$

En el instante inicial $t = 0$, para todo $i \in \{1, \ldots, N\}$:

$$\alpha_0(i) = P[\mathcal{Y}_0 = O_0, \mathcal{X}_0 = s_i] = P[\mathcal{Y}_0 = O_0 | \mathcal{X}_0 = s_i] \cdot P[\mathcal{X}_0 = s_i] = b_{s_i}(O_0) \cdot \pi_i$$

Suponiendo que conocemos las $\alpha_t(i)$ para todo i, podemos calcular fácilmente $\alpha_{t+1}(j)$, que es la probabilidad de observar $(O_0, O_1, \ldots, O_{t+1})$ y $\mathcal{X}_{t+1} = s_j$. Puesto que queremos que $\mathcal{X}_{t+1} = s_j$, primero calculamos la probabilidad de mantener la secuencia parcial $(O_0, \ldots, O_t)$ actualizado el estado, esto no es más que la suma de las variables de avance en t multiplicados por las probabilidades de transición:

$$\begin{aligned} &P[\mathcal{Y}_0^t = (O_0, \ldots, O_t), \mathcal{X}_{t+1} = s_j] = \\ &= \sum_{i=1}^{N} P[\mathcal{Y}_0^t = (O_0, \ldots, O_t), \mathcal{X}_t = s_i] \cdot P[\mathcal{X}_{t+1} = s_j | \mathcal{X}_t = s_i] \\ &= \sum_{i=1}^{N} \alpha_t(i) \cdot a_{ij} \end{aligned}$$

Dado que $\mathcal{Y}_{t+1}$ depende únicamente de $\mathcal{X}_{t+1}$, una vez conocida la suma anterior:

$$\begin{aligned} \alpha_{t+1}(j) &= P[\mathcal{Y}_0^{t+1} = (O_0, \ldots, O_t, O_{t+1}), \mathcal{X}_{t+1} = s_j] \\ &= P[\mathcal{Y}_0^t = (O_0, \ldots, O_t), \mathcal{X}_{t+1} = s_j] \cdot P[\mathcal{Y}_{t+1} = O_{t+1} | \mathcal{X}_{t+1} = s_j] \\ &= \left(\sum_{i=1}^{N} \alpha_t(i) \cdot a_{ij} \right) \cdot b_{s_j}(O_{t+1}) \end{aligned}$$

Luego podemos calcular las variables de avance de forma recursiva:

$$\alpha_{t+1}(j) = \left(\sum_{i=1}^{N} \alpha_t(i) \cdot a_{ij} \right) \cdot b_{s_j}(O_{t+1}), \qquad 0 \le t \le r-1, \quad 1 \le j \le N \tag{3.2}$$

Finalmente, notemos que:

$$P[\mathcal{Y}_0^r = O] = \sum_{i=1}^{N} P[\mathcal{Y}_0^r = O, \mathcal{X}_r = s_i] = \sum_{i=1}^{N} \alpha_r(i) \tag{3.3}$$

Si revisamos el cálculo de las variables de avance $\alpha_t(j)$, podemos ver que para cada estado se necesitan $2N$ operaciones en una etapa $t > 0$, puesto que hay N estados. Podemos concluir que el cálculo de todas las variables de avance requiere un orden de $2rN^2$ operaciones. Si $N = 5$ y $r = 99$, necesitaríamos alrededor de 5000 operaciones usando el algoritmo de avance, en comparación con 10^{72} operaciones que requiere el cálculo directo.

A continuación vamos a utilizar el sencillo caso del ejemplo 3.1 para poner en práctica lo que acabamos de ver:

Ejemplo 3.2. En primer lugar vamos a concretar el modelo: representamos el conjunto de estados como $\mathbb{S} = \{R, \neg R\}$ entendiendo R como lluvia. El conjunto de observaciones también tiene cardinalidad 2: $\mathbb{V} = \{U, \neg U\}$ entendiendo U como presencia de paraguas. Además, vamos a concretar los parámetros del modelo como sigue:

$$A = \begin{pmatrix} 0{,}7 & 0{,}3 \\ 0{,}3 & 0{,}7 \end{pmatrix}$$

$$B = \begin{pmatrix} 0{,}9 & 0{,}1 \\ 0{,}2 & 0{,}8 \end{pmatrix}$$

$$\pi = \begin{pmatrix} 0{,}5 & 0{,}5 \end{pmatrix}$$

Supongamos que queremos calcular la probabilidad de observar la secuencia $O = (O_0, O_1, O_2) = (U, U, \neg U)$, calculamos las variables de avance hasta $r = 2$ teniendo en cuenta que $s_1 = R$ y $s_2 = \neg R$:

- $t = 0$:

$$\alpha_0(1) = b_{s_1}(O_0) \cdot \pi_1 = 0{,}9 \cdot 0{,}5 = 0{,}45$$
$$\alpha_0(2) = b_{s_2}(O_0) \cdot \pi_2 = 0{,}2 \cdot 0{,}5 = 0{,}1$$

- $t = 1$:

$$\begin{aligned} \alpha_1(1) &= b_{s_1}(O_1) \cdot (\alpha_0(1) \cdot a_{11} + \alpha_0(2) \cdot a_{21}) \\ &= 0{,}9 \cdot (0{,}45 \cdot 0{,}7 + 0{,}1 \cdot 0{,}3) = 0{,}3105 \end{aligned}$$
$$\begin{aligned} \alpha_1(2) &= b_{s_2}(O_1) \cdot (\alpha_0(1) \cdot a_{12} + \alpha_0(2) \cdot a_{22}) \\ &= 0{,}2 \cdot (0{,}45 \cdot 0{,}3 + 0{,}1 \cdot 0{,}7) = 0{,}041 \end{aligned}$$

- $t = 2$:

$$\begin{aligned} \alpha_2(1) &= b_{s_1}(O_2) \cdot (\alpha_1(1) \cdot a_{11} + \alpha_1(2) \cdot a_{21}) \\ &= 0{,}1 \cdot (0{,}3105 \cdot 0{,}7 + 0{,}041 \cdot 0{,}3) = 0{,}022965 \end{aligned}$$
$$\begin{aligned} \alpha_2(2) &= b_{s_2}(O_2) \cdot (\alpha_1(1) \cdot a_{12} + \alpha_1(2) \cdot a_{22}) \\ &= 0{,}8 \cdot (0{,}3105 \cdot 0{,}3 + 0{,}041 \cdot 0{,}7) = 0{,}09748 \end{aligned}$$

Así, la probabilidad de que el director lleve paraguas dos días seguidos y no en el tercero es:

$$P[\mathcal{Y}_0^2 = (U, U, \neg U)] = \alpha_2(1) + \alpha_2(2) = 0{,}022965 + 0{,}09748 = 0{,}120445$$

La parte de retroceso del algoritmo no es necesaria para resolver **problema 1**, pero se va a usar en las soluciones a los **problemas 2 y 3**, así que vamos a presentarla aquí.

Definición 3.3. Definimos **la variable de retroceso** $\beta_t(i)$ como la probabilidad de observar la secuencia parcial $(O_{t+1}, O_{t+1}, \ldots, O_r)$ condicionada a que en el instante t, el estado sea s_i. Es decir:

$$\beta_t(i) = P[\mathcal{Y}_{t+1}^r = (O_{t+1}, O_{t+2}, \ldots, O_r) | \mathcal{X}_t = s_i]$$

Puesto que la secuencia de salidas acaba en O_r, $\beta_r(i)$ no se puede determinar usando la definición anterior. En este caso, se define:

$$\beta_r(i) = 1, \qquad \forall i \in \{1, \ldots, N\}$$

De forma similar a las variables de avance, podemos calcular $\beta_t(i)$ en base a $\beta_{t+1}(j)$. Puesto que conocemos éstos últimos, solo tenemos que preocuparnos por O_{t+1}. De nuevo, dado que $\mathcal{Y}_t$ depende únicamente de $\mathcal{X}_t$:

$$\begin{aligned}
&P[\mathcal{Y}_{t+1}^r = (O_{t+1}, O_{t+2}, \ldots, O_r) | \mathcal{X}_{t+1} = s_j] = \\
&= P[\mathcal{Y}_{t+1} = O_{t+1} | \mathcal{X}_{t+1} = s_j] \cdot P[\mathcal{Y}_{t+2}^r = (O_{t+2}, O_{t+3}, \ldots, O_r) | \mathcal{X}_{t+1} = s_j] \\
&= b_{s_j}(O_{t+1}) \cdot \beta_{t+1}(j)
\end{aligned}$$

Además, puesto que $\mathcal{X}_{t+1}$ depende de $\mathcal{X}_t$:

$$\begin{aligned}
\beta_t(i) &= P[\mathcal{Y}_{t+1}^r = (O_{t+1}, O_{t+2}, \ldots, O_r) | \mathcal{X}_t = s_i] \\
&= \sum_{j=1}^{N} P[\mathcal{X}_{t+1} = s_j | \mathcal{X}_t = s_i] \cdot P[\mathcal{Y}_{t+1}^r = (O_{t+1}, O_{t+2}, \ldots, O_r) | \mathcal{X}_{t+1} = s_j] \\
&= \sum_{j=1}^{N} a_{ij} \cdot b_{s_j}(O_{t+1}) \cdot \beta_{t+1}(j)
\end{aligned}$$

Luego también podemos calcular las variables de retroceso de forma recursiva:

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} \cdot b_{s_j}(O_{t+1}) \cdot \beta_{t+1}(j), \qquad 0 \leq t \leq r-1, \quad 1 \leq i \leq N \tag{3.4}$$

Para cada estado, se necesitan $3N - 1$ operaciones en una etapa con $0 \leq t \leq r - 1$. Dado que existen N estados, se requiere un orden de $3rN^2$ operaciones para calcular todas las variables de retroceso.

Veremos en los siguientes apartados, que las variables de avance y de retroceso serán usadas para resolver los **problemas 2 y 3**.

3.2.2 *Solución al problema 2*

Existen varias formas de resolver el **problema 2** en el que, a diferencia del **problema 1**, no es posible dar una solución exacta. Se trata ahora de encontrar una secuencia de estados "óptima" dada una secuencia de observaciones y un determinado modelo. En primer lugar, debemos definir lo que es una secuencia de estados óptima. Existen varios criterios de optimalidad, una de ellas consiste en escoger estados que son más probables individualmente. Con este criterio se pretende maximizar el número estimado de estados individuales correctos. Para implementar esta solución al **problema 2**, definimos la siguiente variable:

$$\gamma_t(i) = P[\mathcal{X}_t = s_i | \mathcal{Y}_0^r = (O_0, O_1, \ldots, O_r)]$$

que es la probabilidad de que el estado en el instante t sea s_i condicionada a observar la secuencia de salidas completa $O = (O_0, O_1, \ldots, O_r)$.

Proposición 3.4. $\gamma_t(i)$ se puede expresar en función de las variables de avance y de retroceso de la siguiente forma:

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

Demostración. Por definición de las variables:

$$\begin{aligned} \alpha_t(i) \cdot \beta_t(i) &= P[\mathcal{Y}_0^t = (O_0, \ldots, O_t), \mathcal{X}_t = s_i] \cdot P[\mathcal{Y}_{t+1}^r = (O_{t+1}, O_{t+2}, \ldots, O_r) | \mathcal{X}_t = s_i] \\ &= P[\mathcal{Y}_0^r = (O_0, O_1, \ldots, O_r), \mathcal{X}_t = s_i] \end{aligned}$$

Por lo tanto:

$$\begin{aligned} \frac{\alpha_t(i) \cdot \beta_t(i)}{\sum_{j=1}^{N} \alpha_t(j) \cdot \beta_t(j)} &= \frac{P[\mathcal{Y}_0^r = (O_0, O_1, \ldots, O_r), \mathcal{X}_t = s_i]}{\sum_{j=1}^{N} P[\mathcal{Y}_0^r = (O_0, O_1, \ldots, O_r), \mathcal{X}_t = s_j]} \\ &= \frac{P[\mathcal{Y}_0^r = (O_0, O_1, \ldots, O_r), \mathcal{X}_t = s_i]}{P[\mathcal{Y}_0^r = (O_0, O_1, \ldots, O_r)]} \end{aligned}$$

Aplicando la definición de probabilidad condicionada tenemos la igualdad del enunciado. □

Usando estas variables, podemos definir los estados más probables individualmente dada la secuencia de observaciones O:

Definición 3.5. Sea la secuencia de observaciones $O = (O_0, O_1, \ldots, O_r)$, definimos el estado más probable individualmente en el instante t como:

$$q_t = \underset{1 \leq i \leq N}{\arg\max} \{\gamma_t(i)\} = \{s_i \in \mathbb{S} \mid \forall s_j \in \mathbb{S} : \gamma_t(j) \leq \gamma_t(i)\} \tag{3.5}$$

A pesar de que (3.5) maximiza el número estimado de estados correctos, puede haber problemas con la secuencia de estados resultante. Por ejemplo, si existen estados inalcanzables desde una de ellas, la secuencia de estados "óptima" puede ser inválida. Esto se debe a que la solución proporcionada por (3.5) sólo determina los estados más probables en cada instante, sin tener en cuenta la probabilidad de existencia de la secuencia resultante en ningún momento.

Una posible solución a este problema consiste en modificar el criterio. Se pueden considerar secuencias de estados que maximizan el número estimado de parejas (q_t, q_{t+1}) o de ternas (q_t, q_{t+1}, q_{t+2}) de estados correctas. Estos criterios pueden ser razonables para ciertas aplicaciones concretas, pero el criterio más utilizado es el de encontrar la secuencia $Q = (q_0, q_1, \ldots, q_r)$ que maximiza $P[\mathcal{X}_0^r = Q | \mathcal{Y}_0^r = O]$. Lo cual es equivalente a maximizar $P[\mathcal{X}_0^r = Q, \mathcal{Y}_0^r = O]$.

Una técnica formal para encontrar dicha secuencia Q existe, se basa en métodos de programación dinámica y se llama **algoritmo de Viterbi**. En primer lugar definimos la variable de Viterbi:

$$\begin{aligned}\delta_t(i) :&= \max_{(q_0,q_1\cdots,q_{t-1})\in \mathbb{S}^t} P[\mathcal{X}_0^{t-1} = (q_0, q_1, \ldots, q_{t-1}), \mathcal{X}_t = s_i, \mathcal{Y}_0^t = (O_0, \ldots, O_t)] \\ &= \max_{(q_0,q_1\cdots,q_{t-1})\in \mathbb{S}^t} P[\mathcal{X}_0^t = (q_0, q_1, \ldots, q_{t-1}, s_i), \mathcal{Y}_0^t = (O_0, \ldots, O_t)]\end{aligned}$$

En cada instante, $\delta_t(i)$ nos proporciona la probabilidad de la secuencia de estados más probable $(q_0, q_1, \ldots, q_{t-1}, q_t)$ de longitud $t+1$ con $q_t = s_i$, habiendo además observado $(O_0, \ldots, O_t)$.

En el instante inicial $t = 0$, para todo $i \in \{1, \ldots, N\}$:

$$\delta_0(i) = P[\mathcal{X}_0 = s_i, \mathcal{Y}_0 = O_0] = P[\mathcal{X}_0 = s_i] \cdot P[\mathcal{Y}_0 = O_0 | \mathcal{X}_0 = s_i] = b_{s_i}(O_0) \cdot \pi_i$$

Notemos que cada $\delta_t(i)$ con $t > 0$ se puede calcular recursivamente en función de $\delta_{t-1}(j), 1 \leq j \leq N$. La idea de la recursividad se debe a la propiedad de Markov, pues la secuencia más probable de estados hasta llegar a $\mathcal{X}_t = s_i$ está compuesta por la secuencia más probable $(q_0, q_1 \ldots, q_{t-1})$ con q_{t-1} igual a un cierto estado s_j y la transición de s_j a s_i. Esta idea se puede observar con la Figura 2.

Por lo tanto, tenemos que hallar las secuencias más probables hasta $t-1$ y ver desde cuál se obtiene la probabilidad máxima dando un paso más:

$$\begin{aligned}\delta_t(i) &= \max_{(q_0,\cdots,q_{t-1})\in \mathbb{S}^t} P[\mathcal{X}_0^{t-1} = (q_0, \ldots, q_{t-1}), \mathcal{X}_t = s_i, \mathcal{Y}_0^t = (O_0, \ldots, O_t)] \\ &= \max_{(q_0,\cdots,q_{t-1})\in \mathbb{S}^t} \big(P[\mathcal{X}_0^{t-1} = (q_0, \ldots, q_{t-1}), \mathcal{X}_t = s_i, \mathcal{Y}_0^{t-1} = (O_0, \ldots, O_{t-1})] \\ &\qquad \cdot P[\mathcal{Y}_t = O_t | \mathcal{X}_0^{t-1} = (q_0, \ldots, q_{t-1}), \mathcal{X}_t = s_i, \mathcal{Y}_0^{t-1} = (O_0, \ldots, O_{t-1})]\big)\end{aligned}$$

Aplicando que $\mathcal{Y}_t$ depende únicamente de $\mathcal{X}_t$:

$$\delta_t(i) = \max_{(q_0,\dots,q_{t-1})\in\mathbb{S}^t} P[\mathcal{X}_0^{t-1} = (q_0,\dots,q_{t-1}), \mathcal{X}_t = s_i, \mathcal{Y}_0^{t-1} = (O_0,\dots,O_{t-1})] \cdot P[\mathcal{Y}_t = O_t | \mathcal{X}_t = s_i]$$

Utilizando la idea de recursión:

$$\begin{aligned}
&\max_{(q_0,\dots,q_{t-1})\in\mathbb{S}^t} P[\mathcal{X}_0^{t-1} = (q_0,\dots,q_{t-1}), \mathcal{X}_t = s_i, \mathcal{Y}_0^{t-1} = (O_0,\dots,O_{t-1})] \\
&= \max_{j=1,\dots,N} \big(P[\mathcal{X}_t = s_i | \mathcal{X}_{t-1} = s_j] \cdot \max_{(q_0,\dots,q_{t-2})\in\mathbb{S}^t} P[\mathcal{X}_0^{t-2} = (q_0,\dots,q_{t-2}), \\
&\quad \mathcal{X}_{t-1} = s_j, \mathcal{Y}_0^{t-1} = (O_0,\dots,O_{t-1})]\big) \\
&= \max_{j=1,\dots,N} \big(a_{ji} \cdot \delta_{t-1}(j)\big)
\end{aligned}$$

Por lo tanto:

$$\delta_t(i) = \max_{j=1,\dots,N} \big(a_{ji} \cdot \delta_{t-1}(j)\big) \cdot b_{s_i}(O_t) \tag{3.6}$$

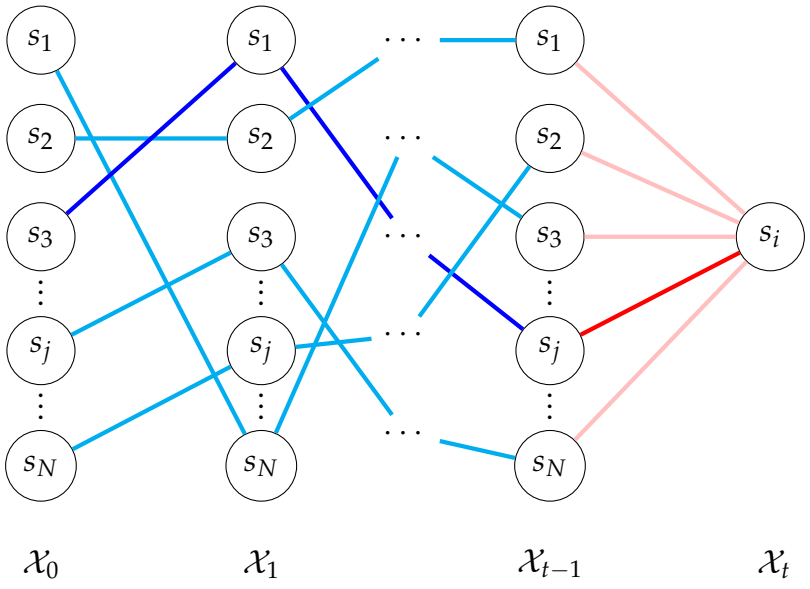


Figura 2: Recursión de algoritmo de Viterbi

Además, necesitamos tener constancia de los estados que forman parte de la secuencia óptima. Para ello, definimos $\psi_t(i)$ como el estado en el instante $t-1$ que maximiza $\delta_t(i)$. Resumiendo:

- En el instante inicial:

$$\delta_0(i) = b_{s_i}(O_0) \cdot \pi_i, \qquad 1 \le i \le N.$$

- Aplicando recursión:

$$\delta_t(i) = \max_{j=1,\dots,N} \left(a_{ji} \cdot \delta_{t-1}(j)\right) \cdot b_{s_i}(O_t), \qquad 1 \leq t \leq r,$$
$$\psi_t(i) = \arg\max_{j=1,\dots,N} \left(a_{ji} \cdot \delta_{t-1}(j)\right), \qquad 1 \leq i \leq N.$$

- Finalmente:

$$P^* = \max_{i=1,\dots,N} \delta_r(i)$$
$$q_r^* = \arg\max_{i=1,\dots,N} \delta_r(i)$$

- Para encontrar la secuencia de estados óptima:

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \quad t = r-1, r-2, \dots, 0$$

Notemos que el esquema recursivo del algoritmo de Viterbi es similar al del algoritmo de avance. La mayor diferencia se encuentra en que, en este caso, se toma el máximo en lugar de calcular una sumatoria.

Ejemplo 3.3. Bajo los parámetros del ejemplo 3.2, ahora podemos hallar la secuencia de estados más probable habiendo observado el director con paraguas en los dos primeros días y no en el tercero. Recordemos que $O = (O_0, O_1, O_2) = (U, U, \neg U)$ y los estados son $s_1 = L$ y $s_2 = \neg L$:

- $t = 0$:

$$\delta_0(1) = b_{s_1}(O_0) \cdot \pi_1 = 0{,}9 \cdot 0{,}5 = 0{,}45$$

$$\delta_0(2) = b_{s_2}(O_0) \cdot \pi_2 = 0{,}2 \cdot 0{,}5 = 0{,}1$$

- $t = 1$:

$$\begin{aligned}\delta_1(1) &= b_{s_1}(O_1) \cdot \max\{a_{11} \cdot \delta_0(1),\, a_{21} \cdot \delta_0(2)\} \\ &= 0{,}9 \cdot \max\{(0{,}7 \cdot 0{,}45),\, (0{,}3 \cdot 0{,}1)\} \\ &= 0{,}9 \cdot \max\{\underline{0{,}315}, 0{,}03\} = 0{,}2835 \\ \psi_1(1) &= L\end{aligned}$$

$$\begin{aligned}\delta_1(2) &= b_{s_2}(O_1) \cdot \max\{a_{12} \cdot \delta_0(1),\, a_{22} \cdot \delta_0(2)\} \\ &= 0{,}2 \cdot \max\{(0{,}3 \cdot 0{,}45),\, (0{,}7 \cdot 0{,}1)\} \\ &= 0{,}2 \cdot \max\{\underline{0{,}135}, 0{,}07\} = 0{,}027 \\ \psi_1(2) &= L\end{aligned}$$

- $t = 2$:

$$\begin{aligned}\delta_2(1) &= b_{s_1}(O_2) \cdot \max\{a_{11} \cdot \delta_1(1),\, a_{21} \cdot \delta_1(2)\} \\ &= 0{,}1 \cdot \max\{(0{,}7 \cdot 0{,}2835),\, (0{,}3 \cdot 0{,}027)\} \\ &= 0{,}1 \cdot \max\{\underline{0{,}19845}, 0{,}0081\} = 0{,}019845 \\ \psi_2(1) &= L\end{aligned}$$

$$\begin{aligned}\delta_2(2) &= b_{s_2}(O_2) \cdot \text{máx}\{a_{12} \cdot \delta_1(1),\, a_{22} \cdot \delta_1(2)\} \\ &= 0{,}8 \cdot \text{máx}\{(0{,}3 \cdot 0{,}2835),\, (0{,}7 \cdot 0{,}027)\} \\ &= 0{,}8 \cdot \text{máx}\{\underline{0{,}08505}, 0{,}0189\} = 0{,}06804 \\ \psi_2(2) &= L\end{aligned}$$

Puesto que $\delta_2(2) > \delta_2(1)$, tomamos $q_2^* = s_2 = \neg L$, reconstruyendo la secuencia tenemos que:

$$q_1^* = \psi_2(2) = L = s_1, \quad q_0^* = \psi_1(1) = L$$

Luego la secuencia de estados resultante de aplicar el algoritmo de Viterbi es $Q = (L, L, \neg L)$, lo cual coincide con nuestra intuición.

3.2.3 *Solución al problema 3*

El tercer problema, y el más complicado, es de tratar de determinar un método para ajustar los parámetros del modelo de manera que maximiza la probabilidad de una secuencia de observaciones dada.

Sin embargo, no existe ninguna forma analítica de resolver este problema de optimización. Dada una secuencia finita de observaciones $O = (O_0, \ldots, O_r)$, no existe una manera óptima de estimar los parámetros del modelo.

No obstante, podemos elegir $\lambda = (A, B, \pi)$ de forma que, con estos parámetros, $P[\mathcal{Y}_0^r = O]$ se maximice de forma local. Para alcanzar estos parámetros vamos a utilizar un algoritmo de esperanza-maximización llamado **algoritmo de Baum-Welch**. En primer lugar, dado un par de estados $s_i, s_j \in \mathbb{S}$, definimos:

$$\xi_t(i,j) := P[\mathcal{X}_t = s_i, \mathcal{X}_{t+1} = s_j | \mathcal{Y}_0^r = O], \qquad 0 \leq t \leq r-1$$

que es la probabilidad de que los estados en los instantes t y $t+1$ sean s_i y s_j respectivamente, condicionada a observar la secuencia de salidas completa $O = (O_0, O_1, \ldots, O_r)$.

Proposición 3.6. $\xi_t(i,j)$ se puede expresar en función de las variables de avance y de retroceso de la siguiente forma:

$$\xi_t(i,j) = \frac{\alpha_t(i) \cdot a_{ij} \cdot b_{s_j}(O_{t+1}) \cdot \beta_{t+1}(j)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) \cdot a_{ij} \cdot b_{s_j}(O_{t+1}) \cdot \beta_{t+1}(j)} \tag{3.7}$$

Demostración. Aplicando la definición de probabilidad condicionada:

$$\xi_t(i,j) = \frac{P[\mathcal{X}_t = s_i, \mathcal{X}_{t+1} = s_j, \mathcal{Y}_0^r = O]}{P[\mathcal{Y}_0^r = O]}$$

Veamos que los numeradores coinciden. Por definiciones de las variables de avance y de retroceso:

$$\begin{aligned}\alpha_t(i) \cdot a_{ij} \cdot b_{s_j}(O_{t+1}) \cdot \beta_{t+1}(j) &= P[\mathcal{Y}_0^t = (O_0, \ldots, O_t), \mathcal{X}_t = s_i] \cdot P[\mathcal{X}_{t+1} = s_j | \mathcal{X}_t = s_i] \\ &\quad \cdot P[\mathcal{Y}_{t+1} = O_{t+1} | \mathcal{X}_{t+1} = s_j] \\ &\quad \cdot P[\mathcal{Y}_{t+2}^r = (O_{t+2}, \ldots, O_r) | \mathcal{X}_{t+1} = s_j] \\ &= P[\mathcal{Y}_0^t = (O_0, \ldots, O_t), \mathcal{X}_t = s_i, \mathcal{X}_{t+1} = s_j, \\ &\quad \mathcal{Y}_{t+1} = O_{t+1}, \mathcal{Y}_{t+2}^r = (O_{t+2}, \ldots, O_r)] \\ &= P[\mathcal{Y}_0^r = (O_0, \ldots, O_r), \mathcal{X}_t = s_i, \mathcal{X}_{t+1} = s_j]\end{aligned}$$

Teniendo esto, podemos ver que los denominadores también coinciden:

$$\begin{aligned}P[\mathcal{Y}_0^r = O] &= \sum_{i=1}^{N} \sum_{j=1}^{N} P[\mathcal{Y}_0^r = O, \mathcal{X}_t = s_i, \mathcal{X}_{t+1} = s_j] \\ &= \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) \cdot a_{ij} \cdot b_{s_j}(O_{t+1}) \cdot \beta_{t+1}(j)\end{aligned}$$

□

Recordemos que previamente habíamos definido la variable $\gamma_t(i)$ como:

$$\gamma_t(i) = P[\mathcal{X}_t = s_i | \mathcal{Y}_0^r = O]$$

No es difícil ver a partir de las definiciones, que:

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j), \qquad 0 \leq t \leq r-1$$

Si sumamos $\gamma_t(i)$ con t desde 0 hasta r, obtenemos una cantidad que se puede interpretar como la cantidad esperada de visitas al estado s_i. Si excluimos $t = r$, se puede interpretar como el número esperado de transiciones que se realizan a partir de s_i. De forma similar, la suma de $\xi_t(i,j)$ con t de 0 a $r-1$ se puede interpretar como el número esperado de transiciones de s_i a s_j. En resumen:

$$\sum_{t=0}^{r} \gamma_t(i) = \text{número esperado de visitas a } s_i$$

$$\sum_{t=0}^{r-1} \gamma_t(i) = \text{número esperado de transiciones desde } s_i$$

$$\sum_{t=0}^{r-1} \xi_t(i,j) = \text{número esperado de transiciones de } s_i \text{ a } s_j$$

Usando lo anterior, podemos dar un método razonable para reestimar los parámetros de un HMM. Definimos:

$$\overline{\pi}_i = \gamma_0(i) = P[\mathcal{X}_0 = s_i | \mathcal{Y}_0^r = O]$$

$$\overline{a}_{ij} = \frac{\text{número esperado de transiciones de } s_i \text{ a } s_j}{\text{número esperado de transiciones desde } s_i}$$
$$= \frac{\sum_{t=0}^{r-1} \xi_t(i,j)}{\sum_{t=0}^{r-1} \gamma_t(i)}$$

$$\overline{b}_j(k) = \frac{\text{número esperado de visitas a } s_j \text{ habiendo observado } v_k}{\text{número esperado de visitas a } s_j}$$
$$= \frac{\sum_{\substack{t=0 \\ O_t = v_k}}^{r} \gamma_t(j)}{\sum_{t=0}^{r} \gamma_t(j)}$$

Si definimos el modelo actual como $\lambda = (A, B, \pi)$, y lo utilizamos para calcular los parámetros anteriores, podemos definir un modelo reestimado $\overline{\lambda} = (\overline{A}, \overline{B}, \overline{\pi})$. Entonces está probado que puede ocurrir una de las siguientes opciones [15]:

- el modelo inicial λ es un punto crítico de la función de probabilidad, en ese caso $\overline{\lambda} = \lambda$.
- la probabilidad de observar la secuencia de salidas O bajo el modelo $\overline{\lambda}$ es mayor que bajo λ. Es decir, $P[\mathcal{Y}_0^r = O|\overline{\lambda}] > P[\mathcal{Y}_0^r = O|\lambda]$.

Con el proceso anterior, podemos iterar reemplazando $\overline{\lambda}$ en el lugar de λ. De esta forma, podemos aumentar la probabilidad de observar O hasta llegar a un valor límite. Cabe destacar que este algoritmo es susceptible de caer en un máximo local y en diversos problemas de interés, pueden existir varios máximos locales.

Recordemos que el algoritmo de Baum-Welch se utiliza para reestimar los parámetros de $\lambda = (A, B, \pi)$ con la intención de maximizar $P[\mathcal{Y}_0^r = O|\lambda]$ con $O = (O_0, \cdots, O_r)$ una secuencia de observaciones conocida. Pero también podemos usarlo para determinar los estados de un espacio $\mathbb{S}$ al que sólo se conoce la cardinalidad del espacio: ante problemas en los que no tenemos suficiente información, podemos partir de un modelo inicial estimado y aplicar el algoritmo de Baum-Welch para construir un modelo ajustado mediante una secuencia de observaciones suficientemente amplia. A partir de los resultados, podemos dar un significado físico a los estados del espacio $\mathbb{S}$. Como ejemplo, examinamos el siguiente caso:

Ejemplo 3.4. Consideramos el trabajo de Cave y Neuwirth [4], el cual pretende estudiar las propiedades básicas del inglés. Para ello, tomaron el libro "Brown Corpus" con alrededor de un millón de palabras. Eliminaron las puntuaciones, números, caracteres especiales y convirtieron todas las letras a minúscula obteniendo 26 letras y el espacio, un total de 27 símbolos al que consideraron como posibles observaciones.

Tomando espacios de estados S de cardinalidad $N = 2, \ldots, 12$ sin especificar los significados de los estados, la aplicación del algoritmo de Baum-Welch les ha permitido llegar a conclusiones dependiendo de N. En particular, para $N = 2$ obtuvieron las siguientes probabilidades de observar cada letra estando en uno de los dos estados:

	s_1	s_2
a	0.000	0.133
b	0.022	0.000
c	0.063	0.000
d	0.056	0.000
e	0.000	0.218
f	0.037	0.000
g	0.015	0.010
h	0.074	0.000
i	0.000	0.150
j	0.000	0.000
k	0.009	0.000
l	0.060	0.000
m	0.041	0.000
n	0.140	0.000
o	0.000	0.136
p	0.030	0.001
q	0.001	0.000
r	0.087	0.000
s	0.105	0.000
t	0.157	0.019
u	0.000	0.045
v	0.016	0.000
w	0.020	0.000
x	0.002	0.000
y	0.004	0.018
z	0.001	0.000
espacio	0.060	0.269

Si analizamos estas probabilidades, podemos ver que los estados separan las vocales de los consonantes, de forma que s_1 representa el conjunto de consonantes y s_2 representa el conjunto de vocales.

Este resultado también se ha alcanzado utilizando la traducción del Quijote a inglés [1] y no debería de sorprender a una persona que tenga conocimiento sobre el idioma. Sin embargo, permite a personas que carecen de ese conocimiento llegar a conclusiones analizando los resultados obtenidos de un HMM asociado al problema.

3.3 MEJORA DE LAS SOLUCIONES

No es difícil darse cuenta de que las variables $\alpha_t(i), \beta_t(i)$ y $\delta_t(i)$ tienden a 0 cuando $t \to \infty$ por ser productos de probabilidades. Para evitar multiplicar por cero, necesitamos tomar medidas dependiendo del caso.

3.3.1 *Normalización de $\alpha_t(i)$ y $\beta_t(i)$*

En primer lugar, consideramos el cálculo de $\alpha_t(i)$. Recordemos que por (3.2):

$$\alpha_t(i) = \left(\sum_{j=1}^{N} \alpha_{t-1}(j) \cdot a_{ji} \right) \cdot b_{s_i}(O_t), \qquad 1 \leq t \leq r, \quad 1 \leq i \leq N$$

Para resolver el problema de que multiplicar por $\alpha_t(i)$ cuando éste tiende a 0, parece lógico normalizar $\alpha_t(i)$ dividiéndolo por la suma de las $\alpha_t(j)$ con $j \in \{1, \ldots, N\}$ y utilizar el resultado de la división. Sin embargo, necesitamos verificar que con esta reestimación, no estamos alterando el resultado del algoritmo de avance. Para ello, definimos las siguientes variables:

- Para $t = 0$, consideramos:

$$\tilde{\alpha}_0(i) = \alpha_0(i), \qquad c_0 = \frac{1}{\sum_{j=1}^{N} \tilde{\alpha}_0(j)}, \qquad \hat{\alpha}_0(i) = c_0 \cdot \tilde{\alpha}_0(i), \qquad 1 \leq i \leq N$$

- Para $1 \leq t \leq r$, $1 \leq i \leq N$:

$$\tilde{\alpha}_t(i) = \left(\sum_{j=1}^{N} \hat{\alpha}_{t-1}(j) \cdot a_{ji} \right) \cdot b_{s_i}(O_t), \qquad c_t = \frac{1}{\sum_{j=1}^{N} \tilde{\alpha}_t(j)}, \qquad \hat{\alpha}_t(i) = c_t \cdot \tilde{\alpha}_t(i)$$

Proposición 3.7. La variable $\hat{\alpha}_t(i)$ cumple que:

$$\hat{\alpha}_t(i) = \frac{\alpha_t(i)}{\sum_{j=1}^{N} \alpha_t(j)}, \qquad 0 \leq t \leq r, \qquad 1 \leq i \leq N$$

Demostración. Por definición $\hat{\alpha}_0(i) = c_0 \cdot \alpha_0(i)$, supongamos que:

$$\hat{\alpha}_t(i) = c_0 \cdot c_1 \cdots c_t \cdot \alpha_t(i) \tag{3.8}$$

Entonces:

$$\begin{aligned}
\hat{\alpha}_{t+1}(i) &= c_{t+1} \cdot \tilde{\alpha}_{t+1}(i) \\
&= c_{t+1} \cdot \left(\sum_{j=1}^{N} \hat{\alpha}_t(j) \cdot a_{ji} \right) \cdot b_{s_i}(O_{t+1}) \\
&= c_0 \cdot c_1 \cdots c_t \cdot c_{t+1} \cdot \left(\sum_{j=1}^{N} \alpha_t(j) \cdot a_{ji} \right) \cdot b_{s_i}(O_{t+1}) \\
&= c_0 \cdot c_1 \cdots c_t \cdot c_{t+1} \cdot \alpha_{t+1}(i)
\end{aligned}$$

Luego por inducción, (3.8) es cierto para todo $t \leq r$. En consecuencia, por (3.8) y definición de $\hat{\alpha}_t(i)$ y $\tilde{\alpha}_t(i)$:

$$\begin{aligned}
\tilde{\alpha}_t(i) &= \frac{\hat{\alpha}_t(i)}{c_t} = \frac{c_0 \cdots c_t \cdot \alpha_t(i)}{c_t} = c_0 \cdots c_{t-1} \cdot \alpha_t(i) \\
\hat{\alpha}_t(i) &= \frac{\tilde{\alpha}_t(i)}{\sum_{j=1}^{N} \tilde{\alpha}_t(j)} = \frac{c_0 \cdots c_{t-1} \cdot \alpha_t(i)}{c_0 \cdots c_{t-1} \cdot \sum_{j=1}^{N} \alpha_t(j)} = \frac{\alpha_t(i)}{\sum_{j=1}^{N} \alpha_t(j)}
\end{aligned}$$

□

Corolario 3.8. Se cumple que:

$$\hat{\alpha}_t(i) = P[\mathcal{X}_t = s_i | \mathcal{Y}_0^t = (O_0, \ldots, O_t)], \qquad 0 \leq t \leq r, \qquad 1 \leq i \leq N$$

Demostración. Por la proposición 3.7 y la definición de $\alpha_t(i)$:

$$\begin{aligned}
\hat{\alpha}_t(i) &= \frac{\alpha_t(i)}{\sum_{j=1}^{N} \alpha_t(j)} = \frac{P[\mathcal{Y}_0^t = (O_0, \ldots, O_t), \mathcal{X}_t = s_i]}{\sum_{j=1}^{N} P[\mathcal{Y}_0^t = (O_0, \ldots, O_t), \mathcal{X}_t = s_j]} \\
&= \frac{P[\mathcal{Y}_0^t = (O_0, \ldots, O_t), \mathcal{X}_t = s_i]}{P[\mathcal{Y}_0^t = (O_0, \ldots, O_t)]} = P[\mathcal{X}_t = s_i | \mathcal{Y}_0^t = (O_0, \ldots, O_t)]
\end{aligned}$$

□

Por lo tanto, $\hat{\alpha}_t(i)$ es el resultado de dividir $\alpha_t(i)$ por la suma de las $\alpha_t(j)$ con $j \in \{1, \ldots, N\}$ como queríamos. Además, con este proceso hemos evitado calcular directamente $\alpha_t(i)$. Ahora debemos comprobar que utilizando $\hat{\alpha}_t(i)$ podemos calcular $P[\mathcal{Y}_0^r = O]$, notemos que de la proposición 3.7 tenemos:

$$\sum_{j=1}^{N} \hat{\alpha}_t(j) = 1, \qquad \forall t \in \{0, \ldots, r\}$$

Entonces, aplicando (3.8):

$$1 = \sum_{j=1}^{N} \hat{\alpha}_r(j) = c_0 \cdot c_1 \cdots c_r \cdot \sum_{j=1}^{N} \alpha_r(j) = c_0 \cdot c_1 \cdots c_r \cdot P[\mathcal{Y}_0^r = O]$$

En consecuencia:

$$P[\mathcal{Y}_0^r = O] = \frac{1}{\prod_{t=0}^{r} c_t}$$

Para evitar dividir entre 0, podemos aplicar el logaritmo:

$$\log\left(P[\mathcal{Y}_0^r = O]\right) = -\sum_{t=0}^{r} \log c_t$$

Antes de pasarnos a la normalización de $\beta_t(i)$, vamos a pararnos a examinar la variable c_t. Tal como hemos definido c_t, uno puede pensar que depende de la variable $\tilde{\alpha}_t(i)$ y por lo tanto depende de $\alpha_t(i)$. Pero con la siguiente proposición vamos a ver que c_t es independiente de $\alpha_t(i)$:

Proposición 3.9. La variable c_t cumple que:

$$\begin{aligned} c_0 &= P[\mathcal{Y}_0 = O_0]^{-1} \\ c_t &= P[\mathcal{Y}_t = O_t | \mathcal{Y}_0^{t-1} = (O_0, \ldots, O_{t-1})]^{-1}, \qquad 1 \leq t \leq r \end{aligned}$$

Demostración. De la proposición 3.7 y (3.8) tenemos que:

$$\hat{\alpha}_t(i) = \frac{\alpha_t(i)}{\sum_{j=1}^{N} \alpha_t(j)} = c_0 \cdot c_1 \cdots c_t \cdot \alpha_t(i) \tag{3.9}$$

Para $t = 0$ tenemos:

$$\hat{\alpha}_0(i) = \frac{\alpha_0(i)}{\sum_{j=1}^{N} \alpha_0(j)} = c_0 \alpha_0(i)$$

Por lo tanto:

$$c_0 = \frac{1}{\sum_{j=1}^{N} \alpha_0(j)} = \frac{1}{\sum_{j=1}^{N} P[\mathcal{Y}_0 = O_0, \mathcal{X}_0 = s_j]} = P[\mathcal{Y}_0 = O_0]^{-1}$$

Para $t > 0$, utilizamos (3.9):

$$\begin{aligned} \prod_{k=0}^{t-1} c_k = \frac{1}{\sum_{j=1}^{N} \alpha_{t-1}(j)} &= \frac{1}{\sum_{j=1}^{N} P[\mathcal{Y}_0^{t-1} = (O_0, \ldots, O_{t-1}), \mathcal{X}_{t-1} = s_j]} \\ &= P[\mathcal{Y}_0^{t-1} = (O_0, \ldots, O_{t-1})]^{-1} \end{aligned}$$

Y en consecuencia:

$$c_t = \frac{P[\mathcal{Y}_0^{t-1} = (O_0, \ldots, O_{t-1})]}{\sum_{j=1}^{N} P[\mathcal{Y}_0^t = (O_0, \ldots, O_t), \mathcal{X}_t = s_j]} = \frac{P[\mathcal{Y}_0^{t-1} = (O_0, \ldots, O_{t-1})]}{P[\mathcal{Y}_0^t = (O_0, \ldots, O_t)]}$$

$$= \left(\frac{P[\mathcal{Y}_0^{t-1} = (O_0, \ldots, O_{t-1}), \mathcal{Y}_t = O_t]}{P[\mathcal{Y}_0^{t-1} = (O_0, \ldots, O_{t-1})]} \right)^{-1} = P[\mathcal{Y}_t = O_t | \mathcal{Y}_0^{t-1} = (O_0, \ldots, O_{t-1})]^{-1}$$

□

Así, c_t es la inversa de una probabilidad. Luego $c_t \geq 1$ y podemos utilizarla para normalizar $\beta_t(i)$ de mismo modo que con $\alpha_t(i)$. Recordemos que por (3.4):

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} \cdot b_{s_j}(O_{t+1}) \cdot \beta_{t+1}(j), \qquad 0 \leq t \leq r-1, \quad 1 \leq i \leq N$$

Definimos la variable normalizada $\hat{\beta}_t(i)$ de siguiente manera:

- Para $t = r$, consideramos:

$$\hat{\beta}_r(i) = c_r, \qquad 1 \leq i \leq N$$

- Para $0 \leq t \leq r-1$:

$$\hat{\beta}_t(i) = c_t \cdot \sum_{j=1}^{N} a_{ij} \cdot b_{s_j}(O_{t+1}) \cdot \hat{\beta}_{t+1}(j), \qquad 1 \leq i \leq N$$

No es difícil ver que la variable $\hat{\beta}_t(i)$ cumple lo siguiente:

$$\hat{\beta}_t(i) = c_t \cdot c_{t+1} \cdots c_r \cdot \beta_t(i), \qquad 0 \leq t \leq r, \qquad 1 \leq i \leq N \tag{3.10}$$

En efecto, para $t = r$ está claro que:

$$\hat{\beta}_r(i) = c_r = c_r \cdot \beta_r(i)$$

Supongamos cierto para $t > 0$, veamos que la proposición se cumple para $t-1$:

$$\begin{aligned}
\hat{\beta}_{t-1}(i) &= c_{t-1} \cdot \sum_{j=1}^{N} a_{ij} \cdot b_{s_j}(O_t) \cdot \hat{\beta}_t(j) \\
&= c_{t-1} \cdot \sum_{j=1}^{N} a_{ij} \cdot b_{s_j}(O_t) \cdot [c_t \cdot c_{t+1} \cdots c_r \cdot \beta_t(j)] \\
&= c_{t-1} \cdot c_t \cdot c_{t+1} \cdots c_r \cdot \sum_{j=1}^{N} a_{ij} \cdot b_{s_j}(O_t) \cdot \beta_t(j) \\
&= c_{t-1} \cdot c_t \cdot c_{t+1} \cdots c_r \cdot \beta_{t-1}(i)
\end{aligned}$$

Por lo tanto, $\hat{\beta}_t(i)$ cumple una propiedad inductiva similar a la de $\hat{\alpha}_t(i)$.

En realidad, cualquiera otra forma de normalización hubiese sido válida [5]. El motivo por el que utilizamos c_t para normalizar $\beta_t(i)$ es que de esta forma, podemos utilizar las variables normalizadas en (3.7) y en el cálculo de los parámetros reestimados en el algoritmo de Baum-Welch.

Con esta normalización, presentamos los pseudo-códigos que resuelven los **problema 1** y **3**:

Algoritmo 1 Algoritmo de avance-retroceso normalizado

Input: $\lambda = (A, B, \pi)$
Input: $O = (O_0, \dots, O_r)$ = secuencia de observaciones
Output: $\alpha_t(i), \beta_t(i), c_t, \forall i \in \{1, \dots N\}, \forall t \in \{0, \dots r\}$
Output: $logProb = \log\left(P[\mathcal{Y}_0^r = O|\lambda]\right)$

```
//Calcular α_0(i)
c_0 ← 0
for i ← 1 to N do
    α_0(i) = π_i · b_{s_i}(O_0)
    c_0 = c_0 + α_0(i)
end for

//Normalizar α_0(i)
c_0 ← 1/c_0
for i ← 1 to N do
    α_0(i) = c_0 · α_0(i)
end for

//Calcular α_t(i)
for t ← 1 to r do
    c_t ← 0
    for i ← 1 to N do
        α_t(i) ← 0
        for j ← 1 to N do
            α_t(i) ← α_t(i) + α_{t−1}(j) · a_{ji}
        end for
        α_t(i) ← α_t(i) · b_{s_i}(O_t)
        c_t ← c_t + α_t(i)
    end for

    //Normalizar α_t(i)
    c_t ← 1/c_t
    for i ← 1 to N do
        α_t(i) = c_t · α_t(i)
```

```
    end for
end for

//Inicializamos β_r(i) con los valores de c_r
for i ← 1 to N do
    β_r(i) = c_r
end for

//Calcular β_t(i)
for t ← r − 1 to 0 by −1 do
    for i ← 1 to N do
        β_t(i) ← 0
        for j ← 1 to N do
            β_t(i) ← β_t(i) + a_ij · b_{s_j}(O_{t+1}) · β_{t+1}(j)
        end for

        //Normalizar β_t(i)
        β_t(i) ← c_t · β_t(i)
    end for
end for

//Calcular log(P[Y_0^r = O])
logProb ← 0
for t ← 0 to r do
    logProb ← logProb + log(c_t)
end for
logProb ← −logProb
```

Algoritmo 2 Algoritmo de Baum-Welch

Input: $\lambda = (A, B, \pi)$
Input: $maxIters$ = número máximo de iteraciones
Input: $O = (O_0, \ldots, O_r)$ = secuencia de entrenamiento
Output: $\overline{\lambda} = (\overline{A}, \overline{B}, \overline{\pi})$

```
iters ← 0
oldLogProb ← −∞
Llamar a algoritmo 1

while iters < maxIters y logProb > oldLogProb do

    //Calcular ξ_t(i, j) y γ_t(i)
    for t ← 0 to r − 1 do
        for i ← 1 to N do
            γ_t(i) ← 0
```

```
            for j ← 1 to N do
                ξ_t(i,j) ← α_t(i) · a_ij · b_{s_j}(O_{t+1}) · β_{t+1}(j)
                γ_t(i) ← γ_t(i) + ξ_t(i,j)
            end for
        end for
    end for

    //Para γ_r(i) notemos que coinciden con α_r(i) normalizados
    for i ← 1 to N do
        γ_r(i) ← α_r(i)
    end for

    //Reestimación de π
    for i ← 1 to N do
        π̄_i ← γ_0(i)
    end for

    //Reestimación de A y B
    for i ← 1 to N do
        denom ← 0
        for t ← 0 to r − 1 do
            denom ← denom + γ_t(i)
        end for

        for j ← 1 to N do
            numerA ← 0
            for t ← 0 to r − 1 do
                numerA ← numerA + ξ_t(i,j)
            end for
            ā_ij ← numerA/denom
        end for

        denom ← denom + γ_r(i)
        for k ← 1 to M do
            numerB ← 0
            for t ← 0 to r do
                if O_t == v_k then
                    numerB ← numerB + γ_t(i)
                end if
            end for
            b̄_i(k) ← numerB/denom
        end for
    end for
```

54:
55: $\lambda \leftarrow \overline{\lambda}$
56: $oldLogProb \leftarrow logProb$
57: Llamar a algoritmo 1
58: $iters \leftarrow iters + 1$
59: **end while**
60:
61: **return** $\overline{\lambda} \leftarrow \lambda$

3.3.2 *Mejora del algoritmo de Viterbi*

En el caso del algoritmo de Viterbi, podemos tomar el logaritmo de $\delta_t(i)$. Al ser el logaritmo una función creciente, no afecta al cálculo de los máximos. De manera que:

- En el instante inicial:

$$\hat{\delta}_0(i) = \log(b_{s_i}(O_0) \cdot \pi_i), \qquad 1 \leq i \leq N.$$

- Aplicando recursión:

$$\hat{\delta}_t(i) = \max_{j=1,\dots,N} \left(\hat{\delta}_{t-1}(j) + \log(a_{ji})\right) + \log(b_{s_i}(O_t)), \qquad 1 \leq t \leq r,$$
$$\hat{\psi}_t(i) = \arg\max_{j=1,\dots,N} \left(\hat{\delta}_{t-1}(j) + \log(a_{ji})\right), \qquad 1 \leq i \leq N.$$

- Finalmente:

$$\hat{P}^* = \max_{i=1,\dots,N} \hat{\delta}_r(i)$$
$$\hat{q}_r^* = \arg\max_{i=1,\dots,N} \hat{\delta}_r(i)$$

- Para encontrar la secuencia de estados óptima:

$$\hat{q}_t^* = \hat{\psi}_{t+1}(\hat{q}_{t+1}^*), \quad t = r-1, r-2, \dots, 0$$

Así, tenemos el siguiente pseudo-código que resuelve el **problema 2**:

Algoritmo 3 Algoritmo de Viterbi

Input: $\lambda = (A, B, \pi)$
Input: $O = (O_0, \dots, O_r)$ = secuencia de observaciones
Output: $Q = (q_0, \dots, q_r)$ = secuencia de estados que maximiza $P[\mathcal{X}_0^r = Q | \mathcal{Y}_0^r = O]$
1: //Calcular $\delta_0(i)$
2: **for** $i \leftarrow 1$ **to** N **do**
3: $\quad \delta_0(i) \leftarrow \log(b_{s_i}(O_0) \cdot \pi_i)$
4: **end for**

5:
6: //Calcular $\delta_t(i)$ y $\psi_t(i)$
7: **for** $t \leftarrow 1$ **to** r **do**
8: | **for** $i \leftarrow 1$ **to** N **do**
9: | | $\delta_t(i) \leftarrow \max_{j=1,\dots,N} \left(\delta_{t-1}(j) + \log(a_{ji})\right) + \log(b_{s_i}(O_t))$
10: | | $\psi_t(i) \leftarrow \arg\max_{j=1,\dots,N} \left(\delta_{t-1}(j) + \log(a_{ji})\right)$
11: | **end for**
12: **end for**
13: //Obtener la secuencia óptima
14: $q_r = \arg\max_{i=1,\dots,N} \delta_r(i)$
15: **for** $t \leftarrow r-1$ **to** 0 **by** -1 **do**
16: | $q_t \leftarrow \psi_{t+1}(q_{t+1})$
17: **end for**

4

APLICACIONES A LA BIOLOGÍA

En este capítulo, introduciremos algunos problemas de biología computacional y de bioinformática que pueden ser resueltos mediante el uso de HMM. Empezaremos presentado algunos conceptos básicos de biología relacionados con nuestro estudio, discutiremos sobre la naturaleza y la importancia que tienen los problemas y cómo podemos adaptar el modelo y los algoritmos vistos en el capítulo anterior para resolverlos.

Para este capítulos, las fuentes principales son [7], [12] y [19, Capítulo 8].

4.1 NOCIONES BÁSICAS DE BIOLOGÍA

Para nuestro estudio necesitamos introducir algunas nociones básicas de biología, en concreto, presentaremos conceptos relacionados con el ADN y las proteínas. Cabe destacar que por el carácter que tiene este trabajo, no entraremos en detalle sobre estos conceptos y sólo vamos a exponer la información relevante para poder introducir los problemas. Por lo tanto, el lector puede encontrar en determinadas ocasiones, un falta de rigor en el sentido de biología. Por último, este apartado se basa esencialmente en [19, Capítulo 8] y [8, Apéndice A].

El material genético para la mayoría de los seres vivos es el ácido desoxirribonucleico, conocido generalmente como ADN. Consiste en un polímero (conjunto) de nucleótidos, en los que cada nucleótido está compuesto por un glúcido (la desoxirribosa), un grupo fosfato y una base nitrogenada de uno de los siguientes cuatro tipos: adenina, guanina, citosina y timina. En general, se denota a cada nucleótido por la letra inicial de la base que contiene, es decir, por A, G, C y T respectivamente.

Nucleótidos adyacentes en una de las cadenas del ADN se conectan mediante un vínculo químico entre el glúcido de uno y el grupo fosfato del siguiente. La estructura clásica de doble hélice del ADN se forma cuando se conectan las dos cadenas de nucleótidos mediante puentes de hidrógeno. Estas conexiones sólo se forman para pares de nucleótidos concretos (conocidos como par de bases): la adenina con la timina ($A \leftrightarrow T$) y la guanina con la citosina ($G \leftrightarrow C$). Por lo tanto, las dos cadenas

de ADN son complementarias pues si una cadena contiene una *A*, entonces estará conectada con una *T* en la cadena contraria. Análogamente, si una contiene una *C*, la otra contendrá una *G*.

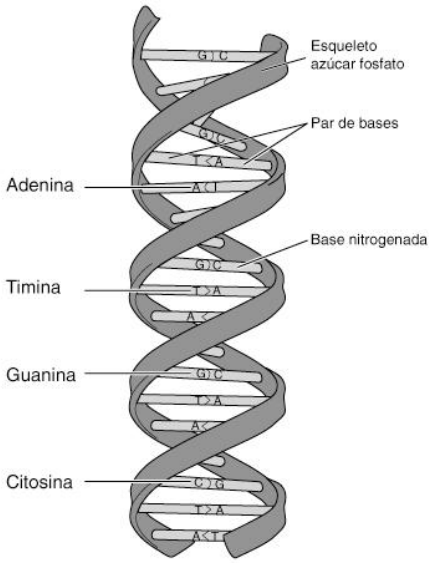


Figura 3: Estructura del ADN

La secuencia en la que constituyen las diferentes bases en una de las cadenas del ADN representa la información genética codificada en dicha cadena. Por la complementariedad de las cadenas, a partir de una de ellas se puede también determinar la información de la otra. Además, cada cadena tiene definida una dirección espacial contraria a la otra, de modo que sólo podemos interpretar una cadena en la dirección correspondiente.

A nivel celular, el ADN se organiza en cromosomas, cada uno de los cuales contiene un ADN que puede tener cientos de millones de par de bases. La mayoría de las células humanas contienen 23 pares de cromosomas, uno heredado del padre y el otro de la madre. Los dos cromosomas de un par son prácticamente idénticos, con la excepción del cromosoma sexual, para el cual existen dos tipos, X e Y. Casi todas las células del cuerpo humano contienen copias idénticas del conjunto completo de 23 pares de cromosomas. El conjunto total de ADN de un organismo se conoce como su genoma, cabe destacar que el genoma humano contiene a más de tres mil millones de pares de bases.

Un cromosoma humano está compuesto principalmente por ADN no codificante, cuya función apenas se está empezando a entender. Interpuesto en el ADN se encuentran genes que codifican proteínas. Estos genes representan aproximadamente el 2 % del genoma humano, sin embargo, son el foco de atención de los genetistas. Los genes a menudo están organizados en exones, que son secuencias que eventualmente serán utilizadas por la célula, alternado con intrones, que serán descartados en el

proceso de codificación. La información en estos genes se codificará en ARN (ácido ribonucleico), y en muchos casos, finalmente en proteínas.

En el proceso de codificación de un gen a un ARN, se utiliza la secuencia de ADN del gen (eliminano los intrones) como plantilla. Al igual que el ADN, el ARN está también compuesto por una serie de nucleótidos, pero con ciertas diferencias: el ARN está formado en general por una única cadena y sustituye la base nitrogenada uracilo (U) por la timina (T). Un caso particular de ARN, el ARNm (ARN mensajero), será finalmente transformado en proteína.

Una proteína está compuesta por una secuencia de aminoácidos, existen una gran variedad de aminoácidos pero sólo 20 aparecen en las proteínas. Cada uno de estos aminoácidos está representado por una o más secuencias de tres nucleótidos de ARN conocidas como codones. La combinación de cuatro posibles nucleótidos en grupos de tres resulta en $4^3 = 64$ codones, lo que significa que la mayoría de los aminoácidos están codificados por más de un codón. La función de una proteína depende finalmente tanto de su secuencia de aminoácidos como de la estructura tridimensional que ha adquirido de su transformación a partir de un ARNm.

4.2 SOFTWARE UTILIZADO

Antes de presentar las aplicaciones de HMM en la biología, presentamos los recursos software que vamos a utilizar para ilustrar algunos ejemplos. Como recurso principal se va a utilizar dos librerías de Python:

- **NumPy**: es una de las librerías más utilizadas de Python, proporciona la capacidad de tratar elementos matemáticos de forma sencilla y eficiente. En este caso se ha utilizado la versión más reciente hasta el momento, la 1.24.3. Se puede consultar la documentación en [14].
- **hmmlearn**: es una librería de códigos abiertos para Python que implementa modelos de Markov ocultos. Utiliza códigos escrito en C++ para los algoritmos, de forma que son más eficientes que si son implementados directamente en Python. También implementa modelos que no hemos visto en este trabajo. Se ha utilizado para este trabajo la versión 0.3.0. Se puede consultar la documentación en [9] y el código en [10]

A partir de estas herramientas se implementarán archivos de Jupyter Notebook que nos servirán para ilustrar mediante ejemplos algunas de las aplicaciones de HMM en la biología.

4.3 ISLAS CPG

En biología computacional, la predicción de genes es un problema en el que se busca identificar regiones codificadoras o genes en un ADN. Puesto que estas regiones poseen ciertas periodicidades y propiedades estadísticas, los HMM son utilizados para este problema. Considerando las estructuras de los genes como estados ocultos y los pares de bases del ADN como observaciones, la predicción de genes se puede solucionar aplicando el algoritmo de Viterbi. Este razonamiento es aplicable también en otros problemas de análisis biológico como la búsqueda de regiones funcionales, extracción de patrones, búsqueda de motivos de secuencia e identificación de islas CpG. Este último, será el objetivo de nuestro estudio en este apartado [13].

Las islas CpG son regiones de ADN con una gran concentración de dinucletidos CpG, que son citosinas (C) seguido de guaninas (G). Se definen formalmente como regiones de al menos 200 pares de bases con una proporción de C o G superior al de 50 % y con un ratio CpG de observado/esperado superior al de 60 %. Están íntimamente relacionadas con el inicio de un gen en numerosos genomas de mamíferos, por tanto la presencia de una isla CpG es importante en la predicción de genes. Podemos utilizar HMM para determinar si un fragmento corto de ADN proviene de una isla CpG o para encontrar todas las islas CpG en un segmento largo.

Existen diversos modelos que sirven para este problema, vamos a presentar un modelo casi trivial en el que consta de 2 estados (con los símbolos + y - presentando la pertenencia a una isla CpG o no):

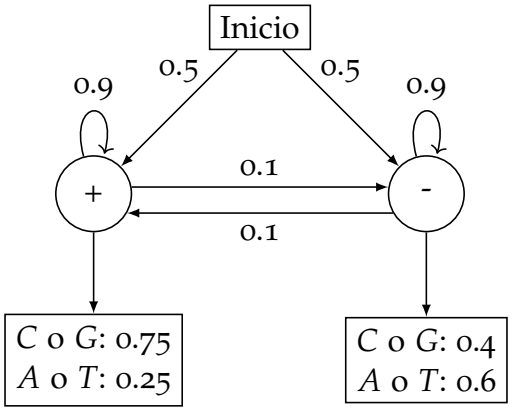


Figura 4: HMM sencillo para identificar islas CpG

Usando este modelo podemos utilizar el algoritmo de Viterbi para identificar islas CpG. Como ejemplo, consideramos la siguiente secuencia:

TCTCGCTGCCGCCAACCCTCGGCGCCGTCGGGTTCGCCGCGGCTCTGATAAG
TCCCGTTTATGGTACCCGGGCCGATCTCTGGTGGGAATCGGAGACCTGTGTAC
CCTGACGCATCCGTTTGTGTTCCCTACACGGCCGACGCAGACCGGGCGCGCG
GCGCCACCCAACGAAGCCCGGGTATGGCACGTGCCCCAGGCGGTGCCCTAC

CCGTATTTCGGGACAAGTTCCCGGATCGGGTGAAAGTTAACGGAAGGATGCC
AAGCAATAGCGGCCACAGGACCCGCCTGGCGACGCATGGACTGGATCCGGA
GGTCTGGCCAACAGTTGATTTCATGGGTTACAGCCCCGGTGTAGATCCCCTCA
TGGTCTCCCGAACCGATTAGTTTGAAAACTGTATCTCCTGGCCGCCTAACAGG
TATAAAGAGCCGGCTCACACTGGGGTGAGGGGGCGCGTGGCCCCCTT

Para determinar si proviene de una isla CpG aplicamos el algoritmo de Viterbi utilizando funcionalidades de **hmmlearn**. Obtenemos el siguiente secuencia de estados:

++----------++++++++
++
++
++++++++++++++----------++++++++++++++++
++++++++++++++++++++++++++++++++++----------++++++++++++++++
++++++++++++++++++++++----------++++++++++----------++++++++++++
++++++++++++++++++++++++++

Podemos usar este resultado para distinguir los estados correspondientes en la secuencia de observaciones:

TCTCGCTGCCGCCAACCCTCGGCGCCGTCGGGTTCGCCGCGGCTCTGATAAGT
CCCGTTTATGGTACCCGGGCCGATCTCTGGTGGGAATCGGAGACCTGTGTAC
CCTGACGCATCCGTTTGTGTTCCCTACACGGCCGACGCAGACCGGGCGCGCG
GCGCCACCCAACGAAGCCCGGGTATGGCACGTGCCCCAGGCGGTGCCCTAC
CCGTATTTCGGGACAAGTTCCCGGATCGGGTGAAAGTTAACGGAAGGATGCC
AAGCAATAGCGGCCACAGGACCCGCCTGGCGACGCATGGACTGGATCCGGA
GGTCTGGCCAACAGTTGATTTCATGGGTTACAGCCCCGGTGTAGATCCCCTCA
TGGTCTCCCGAACCGATTAGTTTGAAAACTGTATCTCCTGGCCGCCTAACAGG
TATAAAGAGCCGGCTCACACTGGGGTGAGGGGGCGCGTGGCCCCCTT

Donde el color azul representa que las letras provienen de un estado + y el color rojo representa que provienen de un estado - como resultado de aplicar el algoritmo de Viterbi. Si analizamos las secuencias en rojo, podemos ver que contienen una proporción de T o A superior al de 50 %. Pero en definitiva, tenemos una mayor proporción de estados + en la secuencia de estados resultante, lo que sugiere que la secuencia de observaciones puede provenir de una isla CpG.

4.4 ALINEAMIENTO DE PARES DE SECUENCIAS

En análisis de secuencias, muchas veces es importante comparar dos secuencias para determinar si están funcionalmente relacionadas. Por ejemplo, genes con funciones similares en diferentes organismos suelen tener secuencias de ADN muy similares. Si un gen nuevo es suficientemente similar a otro gen de otro organismo cuya función es conocida, entonces es razonable esperar que el nuevo gen ejerza la misma función. Lo mismo ocurre con las proteínas, si se descubre una nueva proteína del que se

desconoce su estructura tridimensional pero con una similitud sustancial entre su secuencia de aminoácidos y la secuencia de una proteína del que sí se conoce la estructura tridimensional, entonces es razonable esperar que la estructura de la nueva proteína sea de alguna forma similar al de la proteína conocida.

Para estudiar este problema, comenzamos estableciendo algunas notaciones. Vamos a considerar un par de secuencias, x e y de longitudes n y m respectivamente. Sea x_i el i-ésimo símbolo de x e y_j el j-ésimo símbolo de y, estos símbolos pertenecen a un cierto alfabeto $\mathcal{A}$ que en el caso del ADN serán las 4 bases $\{A, C, G, T\}$ y en el caso de las proteínas serán los 20 aminoácidos.

Si sólo consideramos las secuencias originales el problema sería trivial: sólo existe un único alineamiento en el caso de que $n = m$ y en otro caso, el problema se reduciría en encontrar la mejor posición para incluir una secuencia en otra. El problema real se tiene en cuenta posibles inserciones de huecos (*gaps*) en cualquiera de las dos secuencias, sin posibilidad de insertar hueco en ambas secuencias al mismo tiempo ni insertar dos huecos en diferentes secuencias de forma seguida, para obtener alineamientos entre símbolos iguales. Consideramos el siguiente ejemplo:

Ejemplo 4.1. Tomamos dos secuencias de ADN:

$$x = CACGAAT, y = AGTTCAA$$

Podemos considerar un alineamiento entre estas dos secuencias como sigue:

$$\begin{array}{c} C\ A - - - C\ G\ A\ A\ T \\ - A\ G\ T\ T\ C - A\ A - \end{array}$$

donde cada $-$ representa un hueco.

Como podemos apreciar en el ejemplo, las inserciones producen nuevos alineamientos a tener en cuenta. Para valorar los alineamientos, se define una matriz de puntuaciones que asigna para cada coincidencia de símbolos un valor. Esta matriz se conoce usualmente como la matriz de sustitución, un ejemplo de matriz de sustitución para secuencias de ADN podría ser la siguiente:

$$S = \begin{array}{c} \\ A \\ C \\ G \\ T \end{array} \begin{array}{c} \begin{array}{cccc} A & C & G & T \end{array} \\ \begin{pmatrix} 10 & -3 & -2 & 1 \\ -2 & 8 & 1 & -2 \\ -3 & 1 & 9 & -3 \\ 0 & -3 & -2 & 6 \end{pmatrix} \end{array}$$

En esta matriz, cada elemento representa la puntuación que obtendría una coincidencia entre el símbolo de la fila y el símbolo de la columna. Cada uno de estos valores, se puede calcular de la siguiente manera:

$$s(a, b) = \log\left(\frac{p_{ab}}{q_a \cdot q_b}\right)$$

siendo a y b elementos del alfabeto $\mathcal{A}$, p_{ab} la probabilidad de que se produzca un emparejamiento entre a y b y q_a la frecuencia relativa esperada de que se produzca un símbolo a en una secuencia.

Estos valores son relevantes pues afectan a la significación final del análisis: si tenemos un sistema de puntuación ajustado a la realidad, podremos afirmar con mayor seguridad las similitudes entre dos secuencias. Por esta razón, existen métodos para derivar estos valores a partir de datos conocidos y matrices de sustitución utilizadas ampliamente como las matrices PAM o BLOSUM.

Además de las coincidencias entre los símbolos del alfabeto, se tiene en cuenta también las coincidencias entre un símbolo y un hueco. Estas coincidencias tienen una puntuación negativa, lo que se conoce generalmente como penalizaciones por hueco (*gap penalties*). El coste asociado a una consecución de huecos de longitud g puede venir dado por una de las dos siguientes funciones lineares:

$$\gamma_1(g) = -gd$$

$$\gamma_2(g) = -d - (g-1)e$$

donde d se considera la penalización por iniciar la secuencia de huecos y e se considera la penalización por extender la secuencia, usualmente con un valor menor que d. Mientras que en γ_1 se trata todos los huecos por igual, en γ_2 se penaliza menos las secuencias de huecos con mayores longitudes. Esto es deseable cuando se prevé que las inserciones de varios huecos sean tan frecuentes como inserciones de un único hueco. En la práctica, los valores d y e se escogen empíricamente una vez elegido los valores de la matriz de sustitución.

Con estos elementos, podemos determinar el mejor alineamiento con huecos entre dos secuencias entendiéndolo como aquel con mayor puntuación dada la matriz de sustitución. Si consideramos las dos secuencias enteras, estaremos hablando del alineamiento global de pares de secuencias. Si lo que buscamos es el mejor alineamiento entre subsecuencias de x e y, entonces estaremos hablando del alineamiento local.

Para nuestro estudio, vamos a centrarnos en el problema de alineamiento global. Este problema se puede resolver empleando el algoritmo de Needleman-Wunsch, un algoritmo de programación dinámica que garantiza encontrar el alineamiento óptimo entre dos secuencias con posibles huecos. Pero también podemos utilizar un tipo específico de HMM para resolver este problema, los *Pair HMMs*.

4.4.1 *Pair HMM*

A diferencia de los HMMs que habíamos presentado hasta ahora, los *pair HMMs* generan dos salidas en cada estado en lugar de uno. Por esta razón, podemos utilizar este modelo para el problema de alineamiento de pares. Existen distintas variaciones de este modelo, vamos a presentar el modelo ilustrado en [7].

El espacio de estados consiste principalmente en tres estados: en el estado M se generan dos símbolos del alfabeto $\mathcal{A}$ mientras que en los estados X e Y se emite un símbolo sobre una de las dos de salidas dejando un hueco en la otra. Llamaremos p_{ab} a la probabilidad de que se emitan los símbolos a y b desde el estado M y q_a la probabilidad de emitir un símbolo a y un hueco desde los estados X e Y.

A los estado X e Y se les conocen como estados de inserción (o estado de inserción y eliminación en algunas literaturas, por ejemplo [2]) y al estado M, el estado de coincidencia (*match state*).

Asumiendo que los procesos de salida son idénticos, existen dos parámetros para definir las probabilidades de transición entre estos estados: μ como probabilidad de pasar del estado M a un estado de inserción (X o Y) y ϵ como la probabilidad de permanecer en un estado de inserción.

Puesto que estamos tratando con secuencias espaciales en lugar de temporales, tiene sentido definir un estado de inicio y un estado de fin. Ambos estados son estados silenciosos, estados que no producen ninguna salida. El hecho de añadir un estado de inicio facilita posteriormente en la modificación de los algoritmos. Este estado sustituye la función de la distribución inicial, de modo que el modelo siempre empezará por el estado de inicio. Para este problema, podemos considerar que el estado de inicio tenga las mismas probabilidades de transición que M.

Por otro lado, al añadir un estado de fin necesitamos introducir un nuevo parámetro, la probabilidad de pasar a dicho estado desde los otros estados. Asumiendo que es la misma desde cualquier de los otros estados, llamamos a dicho parámetro τ. Este valor influirá en la longitud media de los alineamientos generados por el modelo. Con estos elementos, tenemos el modelo que se muestra en la figura 5.

Una vez conocido el modelo, podemos aplicar el algoritmo de Viterbi adaptado para encontrar el alineamiento óptimo. En este caso, la entrada consiste en dos secuencias $x = (x_1, \ldots, x_n)$ e $y = (y_1, \ldots, y_m)$ y la salida es la secuencia de estados que conduce al alineamiento óptimo.

Utilizaremos los índices i y j para referirnos a los elementos de las secuencias x e y. Puesto que x_i e y_j son elementos del alfabeto $\mathcal{A}$, usaremos la notación $p_{x_i y_j}$, q_{x_i}, q_{y_j} para distinguir las coincidencias entre los símbolos y los posibles huecos de x e y.

Recordemos que en el algoritmo original:

$$\delta_t(i) = \max_{(q_0, q_1 \cdots, q_{t-1}) \in \mathbb{S}^t} P[\mathcal{X}_0^t = (q_0, q_1, \ldots, q_{t-1}, s_i), \mathcal{Y}_0^t = (O_0, \ldots, O_t)]$$

Y se calcula de la siguiente forma:

$$\begin{aligned} &\delta_0(i) = b_{s_i}(O_0) \cdot \pi_i, \\ &\delta_t(i) = \max_{j=1,\ldots,N} \left(a_{ji} \cdot \delta_{t-1}(j)\right) \cdot b_{s_i}(O_t), \qquad 1 \leq i \leq N, \quad 1 \leq t \leq r. \end{aligned} \tag{4.1}$$

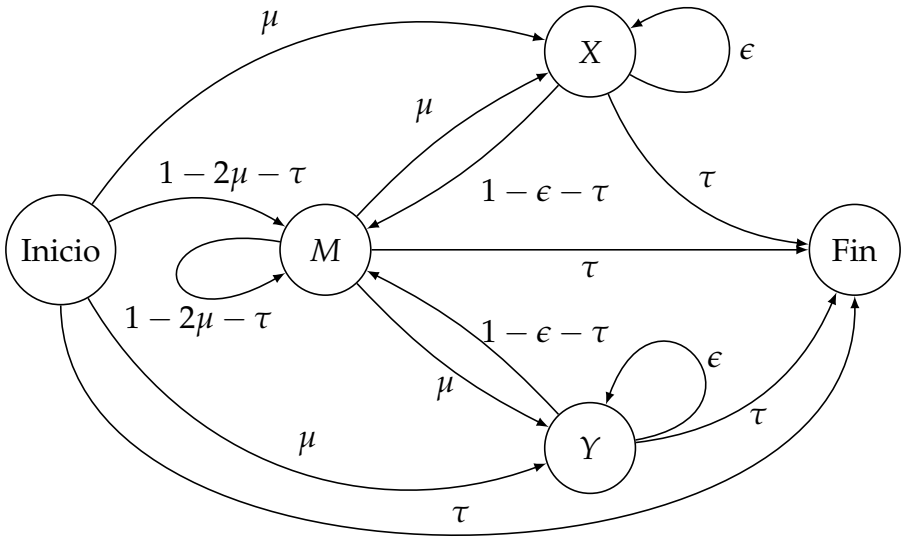


Figura 5: *Pair HMM* de Durbin

En este caso, sólo tenemos que calcular las variables relacionadas con los estados M, X, e Y. El estado de inicio se representará también por el estado M por simplicidad. Puesto que el estado fin es un estado silencioso y representa el final del alineamiento, tendrá un tratamiento especial.

Usando los índices i y j, la variable del algoritmo de Viterbi pasa a ser de forma $\delta_{i,j}(s)$ con $s \in \{M, X, Y\}$. Esta variable tiene el mismo significado que en el caso general: es la probabilidad de la secuencia de estado más probable terminado en el estado s, habiendo considerado hasta las posiciones i y j de las secuencias de entrada. Por lo tanto, el cálculo es idéntico al caso general, sustituyendo el índice temporal por los índices i y j.

Concretamente, inicializamos el algoritmo asignando los siguientes valores:

$$\delta_{0,0}(M) = 1$$

$$\delta_{0,j}(X) = \delta_{i,0}(Y) = 0, \qquad 0 \leq i \leq n,\ 0 \leq j \leq m$$
$$\delta_{0,j}(M) = \delta_{i,0}(M) = 0, \qquad 1 \leq i \leq n,\ 1 \leq j \leq m$$

El primer caso se debe a que siempre empezamos en el estados de inicio (que hemos representado en este caso por el estado M) y al ser silencioso podemos asumir que tiene la probabilidad de emisión 1. Los otros casos se deben a que son situaciones imposibles. Por ejemplo, $\delta_{0,1}(X)$ es la probabilidad de que se haya considerado y_1 y ningún símbolo de x habiendo tomado como último estado el estado X. Esto contradice con la propia definición de X, pues en dicho estado siempre se emite un símbolo de x con un hueco en y.

El resto de las variables se calculan de forma similar que en (4.1), iremos por casos:

- $\delta_{i,j}(M)$ indica que se ha alineado x_i e y_j tomando como último estado el estado M. Además, aplicando recursividad, tenemos que encontrar la probabilidad de la secuencia más probable habiendo considerado hasta las posiciones $i-1$ y $j-1$ terminado en un estado s y considerar la transición de s a M:

$$\delta_{i,j}(M) = p_{x_i y_j} \cdot \text{máx} \begin{cases} (1-2\mu-\tau)\cdot\delta_{i-1,j-1}(M) \\ (1-\epsilon-\tau)\cdot\delta_{i-1,j-1}(X) \\ (1-\epsilon-\tau)\cdot\delta_{i-1,j-1}(Y) \end{cases} \qquad 1 \leq i \leq n,\, 1 \leq j \leq m$$

- $\delta_{i,j}(X)$ indica que se ha emitido x_i y un hueco tomando como último estado el estado X. Puesto que en este caso no se emite ningún símbolo de y, quiere decir que y_j ya había sido emitido anteriormente. Como no es posible pasar del estado Y al X, existen dos posibilidades: el alineamiento entre x_{i-1} e y_j o el alineamiento entre x_k e y_j con $k < i-1$. Este último caso implicaría la emisión de x_{i-1} con un hueco, por lo tanto:

$$\delta_{i,j}(X) = q_{x_i} \cdot \text{máx} \begin{cases} \mu \cdot \delta_{i-1,j}(M) \\ \epsilon \cdot \delta_{i-1,j}(X) \end{cases} \qquad 1 \leq i \leq n,\, 0 \leq j \leq m$$

- $\delta_{i,j}(Y)$ se calcula de mismo modo que $\delta_{i,j}(X)$:

$$\delta_{i,j}(Y) = q_{y_j} \cdot \text{máx} \begin{cases} \mu \cdot \delta_{i,j-1}(M) \\ \epsilon \cdot \delta_{i,j-1}(Y) \end{cases} \qquad 0 \leq i \leq n,\, 1 \leq j \leq m$$

Finalmente, se calcula:

$$\delta(Fin) = \tau \cdot \text{máx}\{\delta_{n,m}(M),\, \delta_{n,m}(X),\, \delta_{n,m}(Y)\}$$

Para obtener la secuencia óptima mantenemos el estado correspondiente al argumento máximo de cada variable y lo recuperamos desde $\delta(Fin)$ del mismo modo que en el algoritmo original. Para evitar el problema de convergencia a 0, basta aplicar logaritmo a las variables tal como habíamos en la versión original.

Ejemplo 4.2. Introducir ejemplo programado aquí

Una de las ventajas que nos proporciona la utilización de *pair HMM* respecto a la programación dinámica, es que ahora podemos calcular la probabilidad de que dos secuencias estén relacionadas según el modelo independientemente del alineamiento. Lo hacemos considerando todos los posibles alineamientos:

$$P[x,y] = \sum_{Alineamiento\ Q} P[x,y,Q]$$

Para calcular esta suma adaptamos el algoritmo de avance a este modelo. La variable de avance se puede calcular de forma recursiva similar a la variable de Viterbi, pero sumando las variables previos en lugar de calcular el máximo.

Ejemplo 4.3. Introducir ejemplo programado aquí

BIBLIOGRAFÍA

[1] Acedo, L. (2019). A hidden markov model for the linguistic analysis of the voynich manuscript. *Mathematical and Computational Applications*, *24*(1). Disponible en: https://www.mdpi.com/2297-8747/24/1/14.

[2] Axelson-Fisk, M. (2015). *Comparative Gene Finding: Models, Algorithms and Implementation*. Computational Biology. Springer London. Disponible en: https://link.springer.com/book/10.1007/978-1-4471-6693-1.

[3] Barbosa Correa, R. (2016). *Procesos estocásticos con aplicaciones*. Universidad del Norte. Disponible en: https://elibro.net/es/lc/ugr/titulos/70066.

[4] Cave, R. L., & Neuwirth, L. P. (1980). Hidden markov models for english. *Hidden Markov Models for Speech*. Disponible en: https://www.cs.sjsu.edu/~stamp/RUA/CaveNeuwirth/index.html.

[5] Devijver, P. A. (1985). Baum's forward-backward algorithm revisited. *Pattern Recognition Letters*, *3*(6), 369–373. Disponible en: https://www.sciencedirect.com/science/article/pii/0167865585900236.

[6] Domínguez García, J. L., & García Planas, M. I. (2015). *Introducción a la teoría de matrices positivas: aplicaciones*. Universitat Politècnica de Catalunya. Disponible en: https://elibro.net/es/lc/ugr/titulos/52187.

[7] Durbin, R., Eddy, S., Krogh, A., & Mitchison, G. (1998). *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge: Cambridge University Press.

[8] Ewens, W. J., & Grant, G. R. (2013). *Statistical Methods in Bioinformatics: An Introduction*. Springer New York, NY. Disponible en: https://link.springer.com/book/10.1007/978-1-4757-3247-4.

[9] hmmlearn. hmmlearn documentation.
URL https://hmmlearn.readthedocs.io/en/latest/

[10] hmmlearn. hmmlearn github repository.
URL https://github.com/hmmlearn/hmmlearn

[11] Häggström, O. (2002). *Finite Markov Chains and Algorithmic Applications*. Cambridge: Cambridge University Press.

[12] J., Y. B. (2009). Hidden markov models and their applications in biological sequence analysis. *Current genomics*, *10*(6). Disponible en: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2766791/.

[13] Lou, X. Y. (2017). Hidden markov model approaches for biological studies. *Biometrics & Biostatistics International Journal*, *5*(4), 132–144.

[14] NumPy. Numpy documentation.
URL https://numpy.org/doc/stable/

[15] Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, *77*(2), 257–286.

[16] Ruiz Reina, J. L., Martín Mateos, F. J., & Graciani Díaz, C. Ampliación de Inteligencia Artificial: Modelos ocultos de Markov. https://www.cs.us.es/cursos/aia-2019/temas/tema-Markov.pdf. Accedido el 10-01-2023.

[17] Salinelli, E., & Tomarelli, F. (2014). *Discrete Dynamical Models*. Springer Cham.

[18] Stamp, M. (2017). *Introduction to Machine Learning with Applications in Information Security*. Chapman & Hall/CRC Press.

[19] Vidyasagar, M. (2014). *Hidden Markov Processes: Theory and Applications to Biology*. Princeton University Press. Disponible en: https://ebookcentral.proquest.com/lib/ugr/detail.action?docID=1680802.

[20] Vélez Ibarrola, R. (1991). *Procesos estocásticos*. Universidad Nacional de Educación a Distancia, 2 ed.