

Sesión 5

Cadenas de Markov. Distribuciones estacionarias. El algoritmo PageRank.

Recordemos que una secuencia $\{X_n\}_n$ de variables aleatorias con valores en un espacio finito Ω se denomina *cadena de Markov en tiempo discreto* cuando

$$P(X_{n+1} = x_{n+1} | X_1 = x_1, \dots, X_n = x_n) = P(X_{n+1} = x_{n+1} | X_n = x_n),$$

donde x_i se denomina estado del sistema en el instante i .

Para simplificar la notación, supondremos que el espacio finito de estados es $\Omega = \{1, \dots, n\}$.

Una cadena de Markov es *homogénea* cuando la probabilidad de pasar del estado i al j no depende del instante de tiempo en que nos encontramos:

$$P(X_{n+1} = j | X_n = i) = P(X_2 = j | X_1 = i).$$

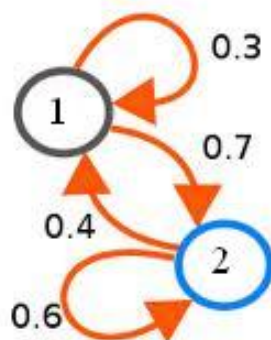
En ese caso, las probabilidades de pasar de un estado i a otro estado j pueden resumirse mediante la llamada *matriz de transición* T , donde $T_{ij} = P(X_{n+1} = j | X_n = i)$.

Una distribución de probabilidad p en el conjunto de posibles estados se dice *distribución estacionaria* cuando $pT = p$. Su interés reside en que, si existe la distribución estacionaria, entonces la cadena de Markov siempre converge a dicha distribución, independientemente de la distribución de probabilidad inicial que consideremos.

En esta práctica, veremos cómo realizar una estimación de la distribución estacionaria con RCommander, a partir de la propiedad de convergencia antes mencionada. También veremos una de las aplicaciones de las cadenas de Markov (y sus distribuciones estacionarias) dentro del análisis de datos: el algoritmo PageRank de Google.

5.1. Estimación de la distribución estacionaria

Consideremos de nuevo la cadena de Markov con la que trabajamos en la sesión 3:



Esto quiere decir que la matriz de transición asociada es:

$$T = \begin{pmatrix} 0,3 & 0,7 \\ 0,4 & 0,6 \end{pmatrix}$$

Existen resultados teóricos que permiten determinar la existencia de la distribución estacionaria y el valor de la misma. En esta práctica únicamente realizaremos una estimación determinando la distribución de probabilidad de los estados cuando n es lo suficientemente grande (entre 50 y 100):

```

> inic=matrix(c(0.5,0.5), 1, 2, byrow=T)

> trans=matrix(c(0.3,0.7,0.4,0.6),2,2,byrow=T)

> for (i in 1:75){
inic=inic %*% trans}

> inic
      [,1]      [,2]
[1,] 0.3636364 0.6363636

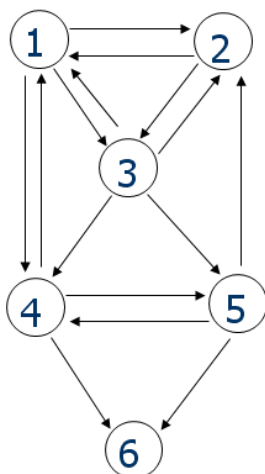
```

Estimaríamos así que la probabilidad de estar en el estado 1 converge hacia 0,3636.

5.2. El algoritmo PageRank

El algoritmo PageRank es utilizado por Google en sus motores de búsqueda. La idea del mismo consiste en clasificar las páginas web de acuerdo con un ranking, que es el que luego se usa para ordenarlas por orden de relevancia en función de la búsqueda. Dicho ranking tiene en cuenta el número de enlaces que apuntan a cada página, y la importancia de las páginas de salida.

Veamos una versión simplificada del mismo, suponiendo que existen sólo 6 páginas web, y que los enlaces entre las mismas vienen dados por el siguiente grafo:



PageRank define el ranking de la página i de manera recursiva, de acuerdo con la fórmula

$$r_i = \sum_{j \in I_i} \frac{r_j}{|O_j|},$$

donde:

- r_j es el ranking de la página j ;
- I_i es el conjunto de páginas que contienen un enlace a la página i ;
- O_j es el conjunto de páginas enlazadas desde la página j .

Esta definición es recursiva, por lo que inicialmente se asigna un ranking uniforme para todas las páginas: $r_i^0 := \frac{1}{n}$ para $i = 1, \dots, n$, y a continuación se itera de acuerdo con la fórmula anterior.

Dicho proceso puede verse como una cadena de Markov, en la que la matriz de transición viene dada por

$$T_{ij} = \begin{cases} \frac{1}{|O_i|} & \text{si } j \in O_i \\ 0 & \text{en otro caso.} \end{cases}$$

En este caso, se obtiene

$$\begin{pmatrix} 0 & 1/3 & 1/3 & 1/3 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 & 0 \\ 1/4 & 1/4 & 0 & 1/4 & 1/4 & 0 \\ 1/3 & 0 & 0 & 0 & 1/3 & 1/3 \\ 0 & 1/3 & 0 & 1/3 & 0 & 1/3 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Para que esta matriz sea *estocástica*, es decir, que la suma de los valores en cada fila sea igual a 1 (y por lo tanto represente a una distribución de probabilidad), se reemplazan las filas iguales a 0 por filas iguales a $1/n$.

$$\begin{pmatrix} 0 & 1/3 & 1/3 & 1/3 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 & 0 \\ 1/4 & 1/4 & 0 & 1/4 & 1/4 & 0 \\ 1/3 & 0 & 0 & 0 & 1/3 & 1/3 \\ 0 & 1/3 & 0 & 1/3 & 0 & 1/3 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \end{pmatrix}$$

Por otro lado, para asegurar que exista una distribución estacionaria, se necesita que todos los elementos de la matriz pertenezcan al intervalo $(0, 1)$. Para ello, se considera una matriz E constante en $1/n$, y se define la matriz de transición modificada como

$$\bar{T} = 0,85T + 0,15E$$

El factor $d = 0,15$ se interpreta como la probabilidad de que el usuario deje de presionar links y navegue directamente a otra web aleatoria. Así, en este caso la matriz de transición sería

$$\begin{pmatrix} 1/40 & 37/120 & 37/120 & 37/120 & 1/40 & 1/40 \\ 9/20 & 1/40 & 9/20 & 1/40 & 1/40 & 1/40 \\ 19/80 & 1/40 & 19/80 & 19/80 & 19/80 & 1/40 \\ 37/120 & 1/40 & 1/40 & 1/40 & 37/120 & 37/120 \\ 1/40 & 37/120 & 1/40 & 37/120 & 1/40 & 37/120 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \end{pmatrix}$$

Nótese que, si rellenásemos las distribuciones de probabilidad por columnas en lugar de por filas, el paso del estado en el momento n al estado en el momento siguiente se haría multiplicando por la transpuesta de esta matriz. Esto se consigue en RCommander sin más que usar la función $t(m)$, que devuelve la transpuesta de una matriz m .

Una vez obtenida la distribución estacionaria, ésta se utiliza para ordenar las páginas en cada búsqueda: cada página tiene unos atributos, y entre aquellas páginas con los atributos que introducimos en el buscador, nos devuelve por orden las que tienen un mayor PageRank.

En este caso, el vector estacionario es:

$$\begin{pmatrix} 0,2066 \\ 0,1770 \\ 0,1773 \\ 0,1770 \\ 0,1314 \\ 0,1309 \end{pmatrix}$$

Así, si introducimos dos términos de búsqueda, uno asociado a las páginas 2, 3 y 6 y el otro asociado a las páginas 1 y 3, el conjunto de relevancia sería $\{1, 2, 3, 6\}$, y se devolvería por orden de ranking: 1, 3, 2, 6. La página 1 sería la considerada más importante.

5.3. Ejercicios

- Consideremos el siguiente ejemplo de aplicación de las cadenas de Markov al marketing¹. Un comercial quiere captar a Jane como cliente. Si lo consigue, recibe un beneficio de 40 dólares. Por otro lado, en cada período de compra el marketing de la compañía hacia Jane tiene un coste de 4 dólares. Se supone que Jane puede comprar como mucho una vez al final de cada período de marketing considerado, y lo hace con probabilidad p_r que depende de su antigüedad como cliente de la compañía. Si alcanza $r = 5$ sin comprar nada la compañía interrumpe las actividades de marketing hacia Jane.

Sea $\Omega = \{1, 2, 3, 4, 5\}$, donde i es la antigüedad de Jane como cliente. La matriz de transición sería

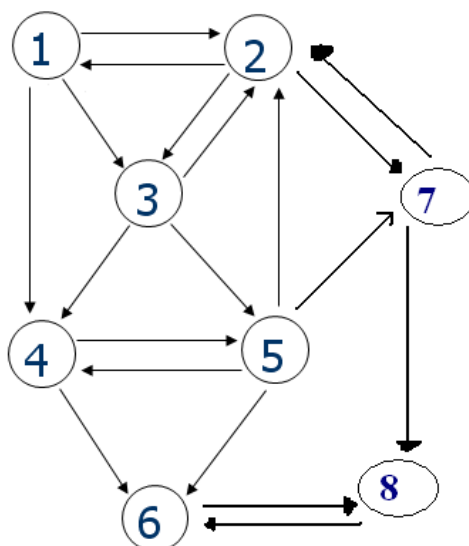
$$T = \begin{pmatrix} 0,3 & 0,7 & 0 & 0 & 0 \\ 0,2 & 0 & 0,8 & 0 & 0 \\ 0,15 & 0 & 0 & 0,85 & 0 \\ 0,05 & 0 & 0 & 0 & 0,95 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

- Estima la distribución estacionaria.
- De acuerdo con los costes y beneficios antes señalados, la ganancia del comercial en cada uno de los estados vendría dada por el vector

$$[36, -4, -4, -4, 0].$$

Da una estimación de la ganancia esperada al cabo de 5 años.

- Supongamos que la estructura de enlaces entre las páginas web viene dada por el siguiente grafo:



Determina la distribución estacionaria de acuerdo con el algoritmo PageRank.

¹Fuente: 'Modelling customer relationships as Markov chains'. P. Pfeifer, R. Carraway. Journal of Interactive Marketing, 14(2), 43–55 (2000).