Práctica 5. Cálculo de la Entropía de la Fuente de Markov

7 de noviembre del 2022 **Marcos Hidalgo Baños**

a) Cálculo del vector probabilidades de estado estacionario.

Esta función es empleada en el siguiente apartado para obtener el vector de probabilidades estacionarias mediante la resolución de un sistema de ecuaciones.

```r
p_inf <- function(p_i) {
  # Calcula el vector de probabilidades estacionario
  # @param p_i, matriz de probabilidades de transicion

  t_pi = t(p_i)
  # Calculamos la matriz transpuesta pi
  t_pi_i = t_pi - diag(x=1, nrow=nrow(t_pi))
  # Calculamos la matriz transpuesta pi menos I

  sistema = rbind(t_pi_i, c(1))
  # Creamos la matriz del sistema de ecuaciones para resolver

  y = c() # Vector de las soluciones del sistema

  for(i in 1:nrow(sistema)) {
    # Para cada ecuacion del sistema...
    y = c(y,0)
    # ..añadimos tantos ceros como ecuaciones hay
  }
  y = replace(y, length(y), 1)
  # Cambiamos el ultimo elemento por un uno...
  p_inf = qr.solve(sistema, y)
  # y resolvemos el sistema formado por la 'sistema' e 'y'

  return(p_inf)

}
```

b) Cálculo de la Entropía de una Fuente de Markov.

```r
calcEntropiaMarkov <- function(p_i, p_inf) {
  # @param p_i, matriz de probabilidades de transicion
  # @param p_inf, vector estacionario

  suma = 0 # inicializacion de la variable acumulativa suma
  h = 0 # inicializacion de la variable de la entropia

  for (i in 1:length(p_inf)) {
    # Recorremos el vector estacionario...
    if (p_inf[i] < 0) {
      # y si vemos que alguna componente es menor de 0, devolvemos error.
      return("No se puede obtener la entropia con los datos proporcionados.")
    }
  }

  for (j in 1:nrow(p_i)) {
    # Recorremos la matriz...
    suma = sum(p_i[j,] * log2(1/p_i[j,]))
    # ... para calcular H(X/E_i) y usarla a continuacion
    h = h + suma * p_inf[j]
    # Calculamos la entropia para esta iteracion
  }

  return(h)

}
```

Pruebas con el ejercicio de las transparencias.

```r
#Prueba ejercicio transparencias
p_i = matrix(c(c(1/4, 1/2), c(3/4, 1/2)), ncol=2)
p_inf(p_i)  # 0.4 , 0.6
calcEntropiaMarkov(p_i, p_inf(p_i)) # 0.92
```

```
> #Prueba ejercicio transparencias
> p_i = matrix(c(c(1/4, 1/2), c(3/4, 1/2)), ncol=2)
> p_inf(p_i)  # 0.4 , 0.6
[1] 0.4 0.6
> calcEntropiaMarkov(p_i, p_inf(p_i)) # 0.92
[1] 0.9245112
```

Funciones auxiliares.

➔ rbind (x , y)

Combina el contenido de la matriz x con el valor y.

En nuestro ejercicio, nos permite colocar el valor después del igual en la ecuación en la última posición del vector.

➔ replace (x , l , v)

Cambia el valor ‘x’ con índice ‘l’ por el dado en ‘v’

➔ log2 (x)

Operación matemática para el cálculo del logaritmo en base 2 de x.

➔ solve (x, y)

Resuelve el sistema x = y. Si x es un vector, y debe ser otro vector de igual longitud tal que x1 = y1; x2 = y2; etc.

- **Marcos Hidalgo Baños** -