

Práctica 2 - Tema 1 parte B. Entropías

5 de octubre del 2022

Marcos Hidalgo Baños

B1. Entropía compuesta

```
1 > entropiaCompuesta <- function(pAi, pBj_Ai) {  
2   # @param pAi, vector de probabilidades P(A)  
3   # @param pBj_Ai, matriz de probabilidades P(Bj/Ai)  
4  
5   res = 0 # Variable acumulativa 'resultado'  
6   pAiBj <- pAi*pBj_Ai # Matriz P(Ai, Bj)  
7  
8   for (i in 1:nrow(pAiBj)) {  
9     # Recorremos las filas de pAiBj para el sumatorio  
10    for (j in 1:ncol(pAiBj)) {  
11      # Recorremos las columnas de pAiBj para el sumatorio  
12      if (pAiBj[i,j] == 0 | sum(pAiBj) != 1) {  
13        NA  
14        # si los datos no son correctos, devolvemos NA  
15      } else {  
16        res = res + pAiBj[i,j]*log2(1/pAiBj[i,j])  
17        # expresion matematica de la entropia compuesta  
18      }  
19    }  
20  }  
21  
22  return(res)  
23 }
```

Pruebas realizadas con los ejemplos de las transparencias.

```
26 #Pruebas  
27 pAi <- c(0.5, 0.5)  
28 pBj_Ai <- matrix(c(c(1/4,1/4,1/4,1/4), c(1,0,0,0)), ncol=2)  
29 # Importante el ncol=2 del final, si no se pone falla la condicionada  
30  
31 entropiaCompuesta(pAi, pBj_Ai)
```

```
> #Pruebas  
> pAi <- c(0.5, 0.5)  
> pBj_Ai <- matrix(c(c(1/4,1/4,1/4,1/4), c(1,0,0,0)), ncol=2)  
> entropiaCompuesta(pAi, pBj_Ai)  
[1] 2
```

B2. Entropía condicionada

Para el cálculo de la entropía condicionada debemos saber que dependiendo de cuál sea la variable aleatoria (X ó Y) será necesario realizar algunos cálculos intermedios más.

Debido a la similitud de las expresiones matemáticas que las determinan, el código implementado resulta prácticamente igual en ambas versiones.

Entropía Condicionada

H(Y / X)

```
33 ▾ entropiaCondicionadaY_X <- function(pAi, pBj_Ai) {
34   # @param pAi, vector de probabilidades P(A)
35   # @param pBj_Ai, matriz de probabilidades P(Bj/Ai)
36
37   res = 0 # Variable acumulativa 'resultado'
38   pAiBj <- pAi*pBj_Ai # Matriz P(Ai, Bj)
39
40 ▾   for (i in 1:nrow(pBj_Ai)) {
41     # Recorremos las filas de pBj_Ai para el sumatorio
42 ▾     for (j in 1:ncol(pBj_Ai)) {
43       # Recorremos las columnas de pBj_Ai para el sumatorio
44 ▾       if (pBj_Ai[i,j] == 0 | sum(pAiBj) != 1) {
45         NA
46         # si los datos no son correctos, devolvemos NA
47 ▾       } else {
48         res = res + pAiBj[i,j]*log2(1/pBj_Ai[i,j])
49         # expresion matematica de la entropia compuesta
50 ▾       }
51 ▾     }
52 ▾   }
53
54   return(res)
55 ▾ }
```

Entropía Condicionada

$H(X / Y)$

```
57 ▾ entropiaCondicionadaX_Y <- function(pAi, pBj_Ai) {  
58   # @param pAi, vector de probabilidades P(A)  
59   # @param pBj_Ai, matriz de probabilidades P(Bj/Ai)  
60  
61   res = 0 # Variable acumulativa 'resultado'  
62   pAiBj <- pAi*pBj_Ai # Matriz P(Ai, Bj)  
63   pBj <- Bj(pAi, pBj_Ai) # Vector de prob. P(B)  
64   pAi_Bj <- ((pBj_Ai*pAi)/pBj) # Matriz de prob P(Ai/Bj)  
65  
66 ▾   for (i in 1:nrow(pAi_Bj)) {  
67     # Recorremos las filas de pAi_Bj para el sumatorio  
68 ▾     for (j in 1:ncol(pAi_Bj)) {  
69       # Recorremos las columnas de pAi_Bj para el sumatorio  
70 ▾       if (pAi_Bj[i,j] == 0 | sum(pAiBj) != 1) {  
71         NA  
72         # si los datos no son correctos, devolvemos NA  
73 ▾       } else {  
74         res = res + pAiBj[i,j]*log2(1/pAi_Bj[i,j])  
75         # expresion matematica de la entropia compuesta  
76 ▾       }  
77 ▾     }  
78 ▾   }  
79  
80   return(res)  
81 ▾ }
```

Observación

Para la implementación de esta función ha sido necesario el uso de otra auxiliar **Bj** descrita al final del documento.

Pruebas realizadas con los ejemplos de las transparencias.

```
pAi <- c(0.5, 0.5)  
pBj_Ai <- matrix(c(c(1/4,1/4,1/4,1/4), c(1,0,0,0)), ncol=2)
```

```
86 # Pruebas  
87 entropiaCondicionadaY_X(pAi, pBj_Ai)  
88 entropiaCondicionadaX_Y(pAi, pBj_Ai)
```

```
> entropiaCondicionadaY_X(pAi, pBj_Ai)  
[1] 1  
> entropiaCondicionadaX_Y(pAi, pBj_Ai)  
[1] 0.4512051
```

B3. Información mutua

Debido a que la expresión matemática de la Información Mutua tiene dos maneras de ser expresada, se ha tomado una decisión de diseño para poder devolver en una misma función ambos valores calculados a la vez a partir de los mismos datos de entrada.

Dichos resultados son almacenados en una lista que será la salida de la función.

Notar que para su cálculo se ha empleado tanto la fórmula de la entropía (descrita al final del documento) como la condicionada implementada en el apartado anterior.

```
92 ▾ informacionMutua <- function(pAi, pBj_Ai) {  
93   # @param pAi, vector de probabilidades P(A)  
94   # @param pBj_Ai, matriz de probabilidades P(Bj/Ai)  
95  
96   res <- list(entropia(pAi) - entropiaCondicionadaX_Y(pAi, pBj_Ai),  
97             entropia(Bj(pAi, pBj_Ai)) - entropiaCondicionadaY_X(pAi, pBj_Ai))  
98   # Almacenamos las dos formas de calcular la IM para corroborar que son iguales  
99  
100   return(res)  
101 ▴ }
```

Pruebas realizadas con los ejemplos de las transparencias.

```
pAi <- c(0.5, 0.5)  
pBj_Ai <- matrix(c(c(1/4,1/4,1/4,1/4), c(1,0,0,0)), ncol=2)
```

```
103 #Pruebas  
104 informacionMutua(pAi, pBj_Ai)
```

```
> informacionMutua(pAi, pBj_Ai)  
[[1]]  
[1] 0.5487949  
  
[[2]]  
[1] 0.5487949
```

Funciones auxiliares.

→ `Bj (pAi, pBj_Ai)`

Calcula la probabilidad $P(B)$ a partir de la condicionada $P(B/A)$ y $P(A)$.

→ `entropia (p_i)`

Función general de la entropía implementada en la práctica anterior.

→ `nrow (pBj_Ai) // ncol (pBj_Ai)`

Obtiene la longitud correspondiente a la longitud de una matriz $M \times N$.

```
107 #Funciones auxiliares
108
109 Bj <- function(pAi, pBj_Ai) {
110   # @param pAi, vector de probabilidades P(A)
111   # @param pBj_Ai, matriz de probabilidades P(Bj/Ai)
112
113   pBj <- c() # Matrix resultado P(B)
114
115   for(i in 1:nrow(pBj_Ai)) {
116     # Recorremos las filas de pBj_Ai
117     pBj <- c(pBj, sum(pAi*pBj_Ai[i,]))
118     # Expresion matematica de la prob. condicionada
119   }
120
121   return(pBj)
122 }
123
124 entropia <- function(p_i) {
125
126   if (sum(p_i) != 1) {NA}
127   # La sumatoria de los valores debe ser 1
128
129   else {sum(p_i*log2(1/p_i))}
130   # Expresión general de la entropía
131
132 }
```