EXTRA 2: COMPLEJIDAD DE COMPUTACIÓN

Autor *Martín Fernández de Diego*

1. INTRODUCCIÓN

De acuerdo con la teoría, existen diferentes maneras de medir la complejidad de una codificación. En la Práctica 2 hemos analizado la entropía del código de Huffman para la variable aleatoria $S_{English}$. En este ejercicio voluntario se pide:

Apartado i) **Programa un código en `Python` para estimar el índice de Gini y la diversidad ^{2}D de Hill para una muestra de una variable aleatoria y aplícalo al ejemplo {0, 1, 0, 0, 0, 2, 1, 1, 0, 2, 0, 1, 2, 0, 2, 0, 1, 1} para comprobar que está bien programado en comparación con el cálculo explícito para dicho ejemplo.**

Apartado ii) **Considera ahora la muestra de la variable aleatoria $S_{English}$. Calcula el índice de Gini y la diversidad ^{2}D de Hill.**

2. MATERIAL USADO

Un fichero inicial recoge el ejemplo {0, 1, 0, 0, 0, 2, 1, 1, 0, 2, 0, 1, 2, 0, 2, 0, 1, 1} y otro el lenguaje $S_{English}$.

Se han utilizado las funciones:

- `ind_gini(dataframe)`
- `div_2dhill(dataframe)`

2.1. Apartado i)

Para comprobar la corrección de las funciones implementadas, realizaremos el cálculo explícito del índice de Gini y de la diversidad ^{2}D de Hill para {0, 1, 0, 0, 0, 2, 1, 1, 0, 2, 0, 1, 2, 0, 2, 0, 1, 1}.

El **índice de Gini** se calcula como $GI := \frac{S}{S+A}$ donde S es el área comprendida entre la curva de equidistribución $y = x$ y la curva $y(x)$ y A es el área que hay debajo de la curva $y(x)$.

Para calcular efectivamente este coeficiente, se utilizará *la regla del trapecio*,

$$GI := 1 - \sum_{j=2}^{N} (y_j + y_{j-1})(x_j - x_{j-1})$$

En el caso concreto de la muestra de 18 elementos, el símbolo '0' aparece 8 veces, el '1' 6 veces y el '2' 4 veces. La frecuencia de aparición es $\frac{8}{18}$, $\frac{6}{18}$ y $\frac{4}{18}$ respectivamente.

Por un lado, se toman las frecuencias acumuladas y se ordenan de menor a mayor en $y := (\frac{4}{18}, \frac{10}{18}, \frac{18}{18})$. Por otro lado, se toma $\frac{i}{N}$ para cada $i \in (0, 1, 2)$ con $N = 3$ en la componente $x := (0, \frac{1}{3}, \frac{2}{3})$.

Basta, por tanto, con rellenar la fórmula.

$$\begin{aligned} GI :&= 1 - [(\frac{10}{18} - \frac{4}{18})(\frac{1}{3} - 0) + (\frac{18}{18} - \frac{10}{18})(\frac{2}{3} - \frac{1}{3})] \\ &= \frac{10}{18} \\ &= 0,222... \end{aligned}$$

La **diversidad ^{q}D de Hill** se calcula como

$${}^qD := \lim_{x \to q} (\sum_{j=1}^{N} P_j^x)^{\frac{1}{1-x}}$$

donde P_j^x es la proporción de símbolos del tipo j-ésimo.

En este caso, se pide la diversidad ^{2}D de Hill a partir de las frecuencias iniciales $P := (\frac{8}{18}, \frac{6}{18}, \frac{4}{18})$.

Rellenando la fórmula se obtiene:

$$\begin{aligned} {}^2D :&= ((\frac{8}{18})^2 + (\frac{6}{18})^2 + (\frac{4}{18})^2)^{-1} \\ &= 2,793... \end{aligned}$$

Efectivamente, estos resultados explícitos coinciden con los resultados arrojados por el código implementado que se encuentra en el anexo.

2.2. Apartado ii)

Aplica las funciones previamente verificadas a la variable aleatoria de $S_{English}$ utilizada en la Práctica 2.

3. RESULTADOS

3.1. Apartado i)

```
Índice de Gini de
    {0,1,0,0,0,2,1,1,0,2,0,1,2,0,2,0,1,1}:
    0.22222222222222232
Diversidad ^2D de Hill de
    {0,1,0,0,0,2,1,1,0,2,0,1,2,0,2,0,1,1}:
    2.793103448275862
```

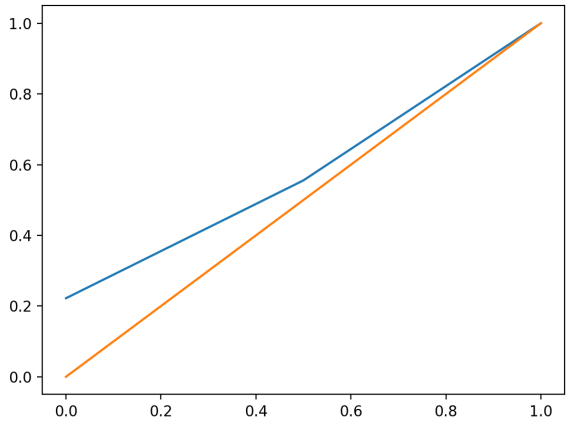

Fig. 1. Gráfica para el índice de Gini

3.2. Apartado ii)

```
Índice de Gini de S_english: 0.6547171500179019
Diversidad ^2D de Hill de S_english:
    11.6537683699609
```

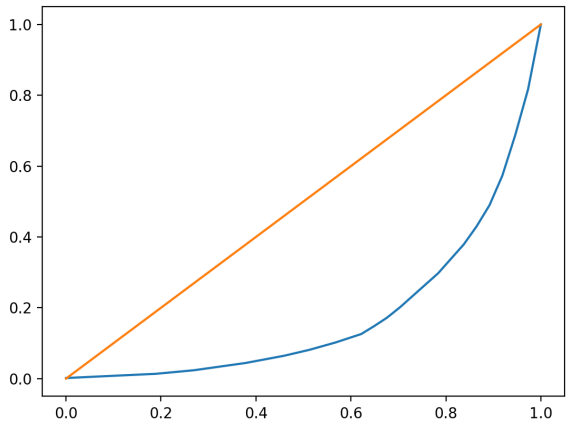

Fig. 2. Gráfica para el índice de Gini

4. CONCLUSIÓN

El **índice de Gini** trata de medir la desigualdad en la distribución de una variable.

En el primer apartado, nos arroja un valor bajo y basta con ver la gráfica para confirmar que la distribución es bastante equitativa. Pero el pequeño tamaño de la muestra sesga el resultado.

En el siguiente apartado, con una muestra significativamente mayor, observamos en el lenguaje $S_{English}$ una desproporción por encima del $0{,}6$ en el índice de Gini. Símbolos como el *espacio*, el más frecuente en el lenguaje, provocan que este coeficiente se dispare.

A partir de la inversa de la **diversidad ^{2}D de Hill** podemos obtener el *índice de Simpson* que representa la probabilidad de que dos elementos del conjunto tomados al azar sean del mismo tipo.

En el primer apartado, el índice de Simpson es $\frac{1}{2{,}793} = 0,358$. Es decir, hay aproximadamente un $36\,\%$ de posibilidades de, tras haber elegido un elemento de $\{0, 1, 0, 0, 0, 2, 1, 1, 0, 2, 0, 1, 2, 0, 2, 0, 1, 1\}$, tomar el mismo símbolo.

En el segundo apartado, este índice se reduce hasta el $\frac{1}{11{,}654} = 0,086$. Esto se debe a un conjunto de símbolos mucho mayor y a una distribución marcadamente asintótica.

5. ANEXO CON EL SCRIPT Y CÓDIGO UTILIZADO

5.1. Código

```
"""
EXTRA 2: COMPLEJIDAD DE COMPUTACIÓN
Martín Fernández de Diego
"""

import os
import numpy as np
import pandas as pd
import math
from collections import Counter
import matplotlib.pyplot as plt

"""
Dado un dataframe
devuelve su índice de Gini
"""
def ind_gini(distr):
    # Calculamos la frecuencia acumulada
    probab_acumulada = np.empty([len(distr['probab'])])
    probab_acumulada[0] = distr.at[0,'probab']
    for i in range(1,len(probab_acumulada)):
        probab_acumulada[i] = probab_acumulada[i-1]+distr.at[i,'probab']
    # Mostramos la gráfica de frecuencia acumulada
    plt.plot(np.linspace(0,1,len(probab_acumulada)),probab_acumulada)
    plt.plot(np.linspace(0,1,len(probab_acumulada)),np.linspace(0,1,len(probab_acumulada
    )))
    # Calculamos el índice de Gini
    sum = 0
    for i in range(1,len(probab_acumulada)):
        sum += (probab_acumulada[i]+probab_acumulada[i-1])/len(probab_acumulada)
    return 1-sum

"""
Dado un dataframe
devuelve su diversidad ^2D de Hill
"""
def div_2dhill(distr):
    h = 0
    for p in distr['probab']:
        h += p*p
    return 1/h


# FORMATO
class Formato:
    BOLD = "\033[1m"
    RESET = "\033[0m"

#### Vamos al directorio de trabajo####
os.getcwd()
#os.chdir(ubica)
#files = os.listdir(ruta)

with open('GCOM2022_pract2_auxiliar_num.txt', 'r',encoding="utf8") as file:
      num = file.read()

with open('GCOM2022_pract2_auxiliar_eng.txt', 'r',encoding="utf8") as file:
      en = file.read()

tab_num = Counter(num)
##### Transformamos en formato array de los carácteres (states) y su frecuencia
##### Finalmente realizamos un DataFrame con Pandas y ordenamos con 'sort'
tab_num_states = np.array(list(tab_num))
tab_num_weights = np.array(list(tab_num.values()))
tab_num_probab = tab_num_weights/float(np.sum(tab_num_weights))
distr_num = pd.DataFrame({'states': tab_num_states, 'probab': tab_num_probab})
```

```
66  distr_num = distr_num.sort_values(by='probab', ascending=True)
67  distr_num.index=np.arange(0,len(tab_num_states))
68
69  # APARTADO i)
70  print("\n" + Formato.BOLD + "Apartado i)" + Formato.RESET)
71  print("Índice de Gini de {0,1,0,0,0,2,1,1,0,2,0,1,2,0,2,0,1,1}:", ind_gini(distr_num))
72  print("Diversidad ^2D de Hill de {0,1,0,0,0,2,1,1,0,2,0,1,2,0,2,0,1,1}:", div_2dhill(
        distr_num))
73  plt.show()
74
75  tab_en = Counter(en)
76  ##### Transformamos en formato array de los carácteres (states) y su frecuencia
77  ##### Finalmente realizamos un DataFrame con Pandas y ordenamos con 'sort'
78  tab_en_states = np.array(list(tab_en))
79  tab_en_weights = np.array(list(tab_en.values()))
80  tab_en_probab = tab_en_weights/float(np.sum(tab_en_weights))
81  distr_en = pd.DataFrame({'states': tab_en_states, 'probab': tab_en_probab})
82  distr_en = distr_en.sort_values(by='probab', ascending=True)
83  distr_en.index=np.arange(0,len(tab_en_states))
84
85  # APARTADO ii)
86  print("\n" + Formato.BOLD + "Apartado ii)" + Formato.RESET)
87  print("Índice de Gini de S_english:", ind_gini(distr_en))
88  print("Diversidad ^2D de Hill de S_english:", div_2dhill(distr_en))
89  plt.show()
```