
Teoría de las telecomunicaciones

Laboratorio 1

MARTÍN NOBLÍA

PROFESORES:

FABIAN IAKINCHUK
MARTÍN CASTILLO



Universidad Nacional de Quilmes

1

Problema 1

Dada una fuente de datos en un archivo de texto nombrado: *texto.txt*. Escribir un algoritmo de manera tal que calcule la probabilidad de ocurrencia de cada símbolo existente en la fuente. Mostrar en un listado (sea en pantalla o en otro archivo *.txt* cada elemento de la fuente con su probabilidad calculada asociada. Calcular la entropía de la fuente.

1.1. resolución

Para resolver este problema utilizamos el lenguaje de programación Julia(<http://www.julialang.org>) el cual es *opensource*, dinámico y tiene velocidades de procesamiento relativas a C entre otras características.

El script que realiza la tarea es el siguiente:

```
#= Script para el problema 1
desde consola:
julia problema1.jl
=#
# incluimos el modulo
include("teleco.jl")
# leemos el archivo
T = open(readall, "../Archivos/texto.txt");
# calculamos las frecuencias y probabilidades
d,p = Teleco.probs(T[1:end-1]);
# Mostramos los resultados
Teleco.show_source(d)
# Calculamos la entropia de la fuente
h = Teleco.entropy(p);
print("La fuente discreta texto.txt tiene una entropia: $h [bits/simbolo]")
```

Las definiciones de funciones estan agrupadas en el modulo *Teleco* que esta definido en el archivo *teleco.jl*. El cual mostramos a continuación:

```
# Modulo para el laboratorio1:Teoria de las telecomunicaciones
module Teleco

function probs(iter)
    # Funcion para calcular las probabilidades de ocurrencia de cada simbolo
    # en una fuente de informacion discreta sin memoria.
    # Input:
    # -----
    # iter: Cualquier fuente iterable(texto,vectores...etc)
    # Output:
    # -----
    # d: Diccionario con los elementos de la fuente y sus frecuencias
    # p: Vector de probabilidades

    d = Dict{eltype(iter), Int}{}
    l = length(iter)

    for e in iter
        d[e] = get(d, e, 0) + 1
    end
end
```

```
p = collect((values(d))) / 1
return d,p
end

function freqs_pr(iter)
    # Funcion para calcular
    pr = Collections.PriorityQueue()

    for e in iter
        pr["$e"] = get(pr, "$e", 0) + 1
    end

    return pr
end

function entropy(p)
    # Funcion para calcular la entropia de una
    # fuente discreta sin memoria
    # Input:
    # -----
    # p: Vector de probabilidades de la fuente
    # Output:
    # -----
    # h: Entropia de la fuente
    i = log2(1./p)
    aux = p.*i
    h = sum(aux)

    return h
end

function show_source(d)
    # Funcion para mostrar los elementos de una fuente
    # y su frecuencia de aparicion
    for (v,k) in zip(values(d),keys(d))
        println("elemento: $k --> frecuencia: $v")
    end
end

export probs, probs_pr, entropy, show_source

end
```

Cuya salida es:

```
elsuizo@debian:~/Telecomunicaciones/Laboratorios/Lab1/Programas/Julia$ ~/julia/./julia problema1.jl
elemento:  --> frecuencia: 22
elemento: á --> frecuencia: 4
elemento: l --> frecuencia: 6
elemento: m --> frecuencia: 2
elemento: r --> frecuencia: 5
```

```
elemento: f --> frecuencia: 1
elemento: o --> frecuencia: 3
elemento: t --> frecuencia: 2
elemento: a --> frecuencia: 14
elemento: E --> frecuencia: 1
elemento: v --> frecuencia: 2
elemento: n --> frecuencia: 5
elemento: e --> frecuencia: 13
elemento: w --> frecuencia: 2
elemento: h --> frecuencia: 3
elemento: d --> frecuencia: 3
elemento: c --> frecuencia: 2
elemento: ; --> frecuencia: 1
elemento: p --> frecuencia: 2
elemento: i --> frecuencia: 4
elemento: q --> frecuencia: 5
elemento: s --> frecuencia: 4
elemento: u --> frecuencia: 8
```

La fuente discreta texto.txt tiene una entropía: 3.989736156368347 [bits/símbolo]

2

Problema 2

Dado un archivo de texto nombrado “texto2.txt” el cual posee una longitud fija de 30 caracteres ASCII:

Escribir un algoritmo que codifique cada elemento de la fuente usando codificación Huffman. (Dicha codificación puede ser mostrada en pantalla o en un archivo de salida *.txt). El software creado debe calcular la entropía y la longitud media del código generado. Calcular en forma manual o con el mismo software la eficiencia de la compresión. Verifique si el código generado es óptimo.

2.1. resolución

3**Problema 3**

Dado dos archivos de texto llamados Castellano.txt e ingles.txt, los cuales están en diferente lenguaje, comprimir los mismos en formato ZIP y:

- a) Realizar una tabla donde se pueda visualizar el cociente entre: $(\text{tamaño nuevo} / \text{tamaño original}) \times 100$.
- b) Realizar una tabla con la probabilidad de ocurrencia de cada uno de los caracteres para cada texto.
- c) Analizar los resultados de a) y b). Sacar conclusiones.

Referencias

- [1] Preparación y evaluación de Proyectos, cuarta edición, Nassir Sapag Chain, Reinaldo Sapag Chain
Enlaces:

www.google.com.ar/
<http://www.lanacion.com.ar/>
<http://www.espacioyconfort.com.ar/>