Teoría de la Información

Trabajo Integrador N° 1:

Unidad I a Unidad IV hasta Rendimiento y Redundancia de un código

Grupo 6

Integrantes:

Aguilera Marcos marcos.aguilera.7.13@gmail.com

Castorina Matías matiascastorina@gmail.com

Noseda Demian nosedademian@gmail.com

Índice

Resumen

En este informe vamos a tratar con fuentes de información de memoria nula y fuentes de Markov, mediante el uso de algoritmos desarrollados por nosotros e implementados en Java para realizar las operaciones correspondientes.

Introducción

Durante el desarrollo del trabajo se tratará con fuentes de memoria nula, para las mismas se requiere calcular la cantidad de información de cada símbolo y la entropía de la fuente. También se calculará un código instantáneo para ellas y se generarán secuencias simuladas de símbolos. Todo esto utilizando como dato la distribución de probabilidades de la fuente.

Trataremos también con fuentes de memoria de Markov, para dichas fuentes calcularemos su entropía, vector de estado estacionario y podremos generar una secuencia simulada de símbolos a partir de la distribución de probabilidades dada por su matriz de transición.

Fuentes de memoria nula

Se trata de una fuente de información donde la aparición de cada símbolo no depende del símbolo anterior obtenido, formalmente se dice que los símbolos son estadísticamente independientes.

Cantidad de información

Es la forma de cuantificar la información aportada en la observación de un suceso probabilístico, esta se calcula en base a su probabilidad y utilizamos un algoritmo basado en la lógica del pseudocódigo descripto a continuación.

Resultado = 0

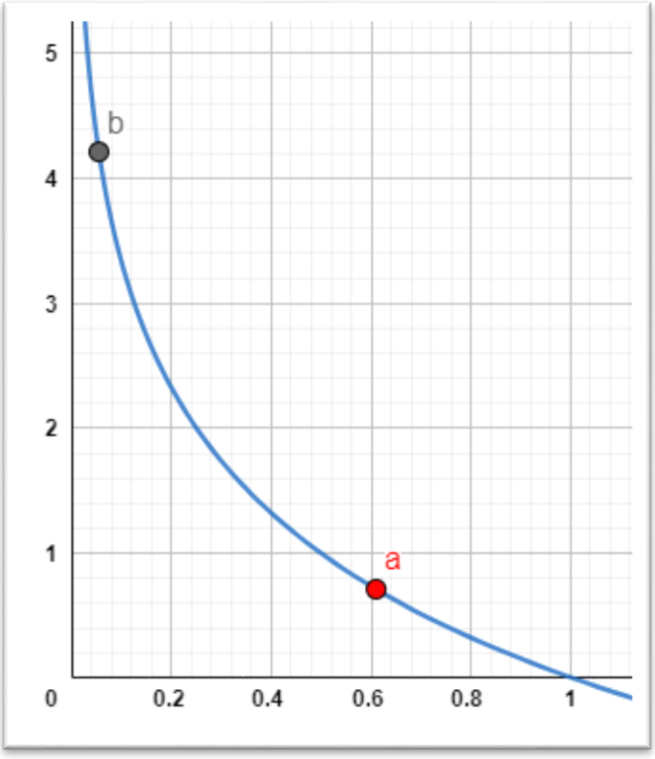
Si la probabilidad es mayor que cero entonces

*Resultado = (-1 *log (probabilidad) / log (base))*

Devolver resultado

FIN

Utilizaremos para ejemplificar una fuente de memoria nula binaria de cuatro símbolos. Tendremos en cuenta la probabilidad de un símbolo “a” es 0.61 al aplicar este algoritmo el programa da como resultado que la cantidad de información es 0.71312 bits. Luego si calculamos la cantidad de información del símbolo “d” con probabilidad 0.054 obtenemos que la cantidad de información es 4.2109 bits.



Podemos notar que el símbolo menos probable aporta más cantidad de información, si lo analizamos mediante el grafico de la cantidad de información en función de la probabilidad vemos el siguiente resultado:

Se puede ver que la curva es asintótica a cero por lo que a medida que la probabilidad se acerque a este número, la cantidad de información aumentará. De la misma forma a medida que la

probabilidad se acerca a 1 la cantidad de información se aproxima a cero siendo nula cuando la probabilidad alcanza su máximo valor en 1.

Entropía (fuente de memoria nula)

La entropía es el nombre que se le da a la cantidad media de información por símbolo de la fuente.

Utilizamos un algoritmo para calcularla que está representado en el siguiente pseudocódigo:

n= cantidad de símbolos

Vector= vector de probabilidades y cantidad de información por símbolos

Mientras posición actual sea menor a n hacer

> *Resultado= resultado + probabilidad en la posición actual del vector * cantidad de información en la posición actual del vector*
>
> *Posición actual = posición actual + 1*

Fin del mientras

Devuelve resultado

FIN

Para el cálculo de la entropía de las Fuentes de memoria nula A y C (ver en apéndice) se obtuvieron los siguientes resultados:

Entropía de la fuente A=1.51648

Entropía de la fuente C=2.56308

Teniendo en cuenta que la cantidad de símbolos de la fuente A es 4 y el de la C es 7, podemos apreciar que a mayor cantidad de símbolos es esperable una mayor entropía. Para contrastar este resultado si utilizamos la fuente B de 5 símbolos se obtiene una entropía de 2.08179.

Esto es lógico en general ya que la suma de las probabilidades asociadas a los símbolos deben sumar uno, entonces cuantos más símbolos posea una fuente, sus probabilidades asociadas tenderán a ser menores para poder cumplir con dicha suma, así aumentando la cantidad de información aportada por cada uno de ellos, y finalmente aumentando la también la entropía de la fuente. No obstante esto no puede ser considerado una regla ya que en ciertos casos esto no se cumplirá, por ejemplo

comparando una fuente de dos símbolos equiprobables contra una de cuatro símbolos donde la probabilidad de uno de ellos es mucho mayor que la del resto, anexamos en el apéndice una tabla de resultados comparativos a modo de contraejemplo.

Generador de código instantáneo

A la hora de generar un código instantáneo para una fuente de memoria nula de base 2 consideramos una solución basada en la siguiente secuencia de código:

0, 10 , 110 , 1…0

Elegimos esta secuencia dado que el primer símbolo es 0 y como el resto de los símbolos comienzan en 1 ningún símbolo será prefijo de otro (se usa 0 como separador). Se asigna al símbolo de mayor probabilidad el código más corto, continuando de manera descendente en probabilidad y ascendente en longitud de código.

Pseudocódigo de la función:

Vector= vector de símbolos

Ordena vector por probabilidad descendente

Código en la primera posición del vector=”0”

i=2

Mientras que i sea menor que la cantidad de símbolos hacer:

> *Código en la posición i del vector= i veces “1” y “0” (ejemplo i=2 “110”)*
>
> *i = i +1*

Fin del mientras

FIN

Cálculo de longitud media del código

Realizamos este cálculo para una fuente de memoria nula que posea una codificación. En nuestro caso la codificación es generada por nuestro mismo programa. Se trata de una media de longitudes de cada código ponderadas por la probabilidad.

Pseudocódigo de la función:

Vector= vector de símbolos (contiene probabilidad y códigos)

Sumatoria = 0

I=0

Mientras que i sea menor que la cantidad de símbolos hacer

Sumatoria= sumatoria+probabilidad del vector en i +longitud del código del vector en i

Fin del mientras

Devolver sumatoria

FIN

Ejemplo de uso:

Usamos una fuente de memoria nula "D" (ver en apéndice) cuya codificación es: S1=0, S2=1110, S3=10, S4=110

El resultado obtenido de la longitud media es 2. Teniendo en cuenta que la entropía de la fuente es 1.84644 bits podemos ver que cumple la condición que la entropía es menor igual a la longitud de cualquier código instantáneo para la fuente (Sabemos que se trata de un código instantáneo dado que lo da el mismo programa).

Cálculo de la inecuación de Kraft

El cumplimiento de esta inecuación es una condición necesaria para que una codificación con cierta longitud de código para cada símbolo sea instantánea.

Utilizaremos dicha ecuación para comprobar si los códigos generados por el programa la cumplen (todos deberían dado que estamos generando codificaciones instantáneas).

Pseudocódigo de la función:

Vector= vector de símbolos (contiene probabilidad y códigos)

Sumatoria = 0

Base= base de la fuente de memoria (en general usamos 2)

I=0

Mientras que i sea menor que la cantidad de símbolos hacer

Sumatoria= sumatoria+base elevado a la longitud del código en la posición i del vector

Fin del mientras

Devolver sumatoria

Este pseudocódigo devuelve el primer término de la inecuación de Kraft el cual debe cumplir con ser menor o igual a 1 para verificar dicha inecuación

Ejemplo de uso:

Volviendo a la fuente de memoria nula “D” (ver en apéndice) tenemos una longitud de código de 1 , 2 , 3 , 4 , para cada símbolo respectivamente. El resultado obtenido de la inecuación de Kraft es la siguiente: 0.9375<=1.

Podemos apreciar que cumple con la inecuación de Kraft de la forma esperada.

Verificación de código compacto

Un código compacto es aquel en el que las longitudes de las palabras cumplen con ser iguales al siguiente entero mayor a la cantidad de información de dicho símbolo

Techo= función que devuelve el siguiente entero mayor de un numero real

Vector= vector de símbolos (contiene probabilidad y códigos)

Condición =verdadero

I=0

Mientras que i sea menor que la cantidad de símbolos y condición sea verdadero hacer

> *Condición= Techo de (cantidad de información del vector en i) es igual a longitud del código del vector en i*

Fin del mientras

Devolver condición

Este pseudocódigo devuelve si para un vector que representa una fuente y posee una codificación dicha codificación es compacta.

Utilizando a la fuente de memoria nula “D” (ver en apéndice) con longitudes de código 1 , 2 , 3 , 4 , para cada símbolo respectivamente. El resultado obtenido es que el código no es compacto. Es lógico que la condición no se cumpla dado que la función utilizada para generar el código instantáneo no fue hecha teniendo en cuenta elegir longitudes de palabra que cumplan con la condición de compacto.

Fuentes de Markov

Trataremos con fuente de Markov con memoria de orden uno, esto significa que la probabilidad de aparición de un símbolo depende del símbolo anterior generado. Estas

fuentes de memorias estarán representadas por una matriz de transición de estados o de pasaje

Cálculo del vector estacionario

Si tenemos en cuenta que tratamos con fuentes ergódicas (todos los estados del proceso son alcanzables desde otro estado), el vector estacionario representa para estas fuentes las probabilidades a la que tienden los símbolos después de una gran cantidad de emisión de los mismos. Dicho en otras palabras, a medida que se emiten más símbolos la probabilidad de aparición de alguno en específico tiende a estabilizarse alcanzando el valor correspondiente en el vector estacionario.

Para calcular el vector estacionario a partir de la matriz de transición la multiplicamos por sí misma una elevada cantidad de veces de esta forma la probabilidad de cada símbolo a partir de otro símbolo tiende a la del vector estacionario.

Pseudocódigo de la multiplicación de matrices:

I = 0 (fila) ; J=0 (columna) ; K= 0

Mientras que I sea menor que la cantidad de símbolos hacer

Mientras que J sea menor que la cantidad de símbolos hacer

Matriz Auxiliar en [I][J] = 0

Mientras que K sea menor que la cantidad de símbolos hacer

*Matriz Auxiliar en [I][J] = Matriz Auxiliar en [I][J] +Matriz en [I][K] * matriz en [K][J];*

K=K+1

Fin Mientras

J=J+1

Fin Mientras

I=I+1

Fin Mientras

Devuelve Matriz Auxiliar

FIN

El pseudocódigo mostrado multiplica la matriz por sí misma y la devuelve. Elegimos ejecutar esta multiplicación arbitrariamente 20 veces consecutivas ya que pudimos apreciar con dicha cantidad la probabilidad de cada símbolo ya era estable en las fuentes de prueba.

Ejemplos de cálculo:

Utilizando una fuente de Markov "A" (ver en apéndice) a partir de su matriz de transición al realizar la ejecución del programa obtuvimos el vector estacionario es: a= 0.47534 b=0.10530 c=0.05331 d= 0.36604

Comprobaremos el vector estacionario a través de la generación de una secuencia de 100 símbolos y veremos la cantidad de apariciones de cada uno.

Secuencia 1:
CDACDADADAAADADAACBADADADDABBADAAACDADAABADADDDCDADADADABCDADADADADADABADDA
BBADADACDADAAADADABAADADADA

Cantidad de apariciones: A=47 , B=9, C=7, D=37

Secuencia 2:
ADADAAADCBBABADADADDADABABACDAAAADADABADDDAAAAADDADADADADADDADADADADADABB
BADDADACDADACDCBCDCDADADADA

Cantidad de apariciones: A=45 , B=10, C=7, D=38

Como podemos ver la cantidad de apariciones de cada símbolo concuerda con la probabilidad de aparición de los mismos en el vector estacionario.

Cálculo de la entropía (fuente de Markov)

Utilizamos un algoritmo para calcularla que está representado en el siguiente pseudocódigo:

Resultado = 0;

n = cantidad de símbolos;

i=0 (fila) , j=0 (columna)

Mientras i sea menor que n hacer

    Acumulador = 0;

Mientras j sea menor que n hacer

Probabilidad = matriz de Transición en [j][i]

Cantidad de Información = 0;

Si probabilidad es mayor que 0 hacer

*Cantidad de Información = -1 * log(probabilidad) / log(base)*

Fin Si

*Acumulador = acumulador+ probabilidad *Cantidad de Información*

Fin Mientras

*resultado =resultado+ vector Estacionario en [i] * acumulador*

Fin Mientras

Devolver resultado

FIN

Calculamos la entropía utilizando este pseudocódigo para la matriz de transición de la fuente de Markov "B" de 6 símbolos (ver en apéndice), el resultado fue: 1.8334.

Repitiendo la conclusión sacada a partir del cálculo de la entropía de la fuente de memoria nula podemos ver que al calcular la entropía de la fuente de Markov "A" de 4 símbolos nos dio: 1.36204, vuelve a cumplirse que la entropía aumenta con la cantidad de símbolos.

Generación de secuencia de símbolos

Para simular la generación de símbolos a partir de una fuente nuestra idea fue a partir de un número aleatorio entre 0 y 1 sumar las probabilidades de aparición de cada símbolo hasta alcanzar dicho número aleatorio. El último símbolo sumado al alcanzar el número aleatorio será el símbolo generado.

Pseudocódigo para memoria nula

Vector= Vector de símbolos

I= 0

Numero Random = Real entre 0 y 1

Suma de Probabilidades= la probabilidad del vector en posición I (en este caso es 0)

Mientras Suma de Probabilidades sea menor o igual a Numero Random hacer

I=I+1

Suma de Probabilidades = Suma de Probabilidades+ la probabilidad del vector en posición I

Fin Mientras

Devuelve símbolo correspondiente del vector en I

FIN

La lógica de este pseudocódigo muestra como obtuvimos un símbolo para una fuente de memoria nula pero también fue aplicada para fuentes de Markov de orden 1. Tuvimos en cuenta en el momento de calcular el símbolo para una fuente de Markov que la probabilidad a sumar es la proveniente del símbolo anterior y está en la matriz de transición de estados.

Los resultados de generación de secuencia de símbolos para todas las fuentes se pueden ver en el apéndice anexado.

Conclusión

Pudimos realizar un programa capaz de trabajar con fuentes de memoria nula y de Markov, permitiendo realizar los cálculos correspondientes a cada una así como simular la generación de una secuencia de símbolos de las mismas.

El uso de este programa y la prueba sobre diferentes fuentes nos permitió comprender ciertas características de su comportamiento a partir de los resultados obtenidos. Por ejemplo ver que el vector estacionario coincide con la cantidad de apariciones de los símbolos en una simulación o que la entropía en general aumenta al aumentar la cantidad de símbolos, entre otras conclusiones expuestas durante el desarrollo de este informe.

Apéndice

Fuente de memoria nula "A"

Símbolo	Probabilidad	Cantidad de Información	Código
a	0,61	0,71312	0
b	0,133	2,91050	110
c	0,203	2,30045	10
d	0,054	4,21090	1110

Entropía	1,51648
Inecuación de Kraft	0,9375
Código compacto	No es compacto
Longitud Media Código	1,631

Simulación	daaaababadaaacbaaacbabcbaaaaaababaaccaaaaaaaaaadaacacabaaaaccaaaaaacaaacabcdbacaaababbacaacacccabbaa

Fuente de memoria nula "B"

Símbolo	Probabilidad	Cantidad de Información	Código
a	0,257	1,96916	10
b	0,082	3,60823	1110
c	0,346	1,53116	0
d	0,252	1,98850	110
e	0,063	3,98850	11110

Entropía	2,08179
Inecuación de Kraft	0,96875
Código compacto	No es compacto
Longitud Media Código	2,25900

Simulación	caacdaeaeacccdccacddacddcaeacccaeadcccedcedaadcdeacadbcdccdbcadcacbbacdabdcdedbdccdcceaaccdaeacdbada

Fuente de memoria nula "C"

Símbolo	Probabilidad	Cantidad de Información	Código
a	0.066	3.92139	111110
b	0.132	2.92139	1110
c	0.193	2.37333	110
d	0.0312	5.00231	1111110
e	0.0958	3.38383	11110
f	0.28	1.83650	0
g	0.202	2.30757	10

Entropía	2.56308
Inecuación de Kraft	0.9921875
Código compacto	No es compacto
Longitud Media Código	2.88440

Simulación	gaaggcfffbbgefbgfggfefecgfcfcfcegggcbgfccffccbcfcbfafccacdfcggffgfbfegcbecfffacgeeffggcffb bebafgegba

Fuente de memoria nula "D"

Símbolo	Probabilidad	Cantidad de Información	Código
S1	0.4	1.32193	0
S2	0.1	3.32193	1110
S3	0.3	1.73697	10
S4	0.2	2.32193	110

Entropía	1.84644
Inecuación de Kraft	0.9375
Código compacto	No es compacto
Longitud Media Código	2.00

Simulación	S3S3S2S1S1S4S3S1S2S1S3S4S3S1S2S1S4S1S3S3S1S4S4S4S1S3S3S3S1S3S2S3S2S1S1S3S4S4S3 S3S1S1S3S3S2S2S1S4S1S1S3S3S1S4S1S1S1S2S4S4S1S3S1S3S1S4S1S3S1S1S4S1S3S3S1S3S2S4 S1S1S3S1S3S1S1S3S4S2S3S1S1S4S1S2S4S3S1S1S3S1

Fuente de Markov "A"

Matriz de transición				
	A	B	C	D
A	0,277	0,626	0,0673	0,749
B	0,135	0,247	0,246	0,0055
C	0,045	0,0655	0,0616	0,0594
D	0,543	0,0615	0,6251	0,1861

Vector estacionario	
Símbolo	Probabilidad estacionaria
A	0.47534
B	0.10530
C	0.05331
D	0.36604

Entropía	1.36204
Simulación	CDACDADADAAADADAACBADADADDABBADAAACDADAABADADDDCDADADADABCDADADADADADABADDABBADADACDADAAADADABAADADADA

Fuente de Markov "B"

Matriz de transición					
	A	B	C	D	E
A	0,285	0,367	0,0113	0,15	0,219
B	0,011	0,117	0,545	0,00885	0,405
C	0,438	0,0795	0,218	0,0936	0,0748
D	0,0477	0,297	0,0602	0,625	0,11
E	0,2183	0,1395	0,1655	0,12255	0,1912

Vector estacionario	
Símbolo	Probabilidad estacionaria
A	0.20577
B	0.19246
C	0.18120
D	0.25597
E	0.16460

Entropía	1.83334
Simulación	CBAACBACEBACEBDAACCBAACBEBDADEBEBAAAEACBBEDAAEACCBACBDDACBAACCBACEBDAEBAACBEEBBDDDACBEDDCBAAACEBDEDD