Trabajo Práctico N°2

Autores: Alejandro Olave, Mikel ; Giacri, Tobías ; Nievas, Nahuel Isaias

Carrera: Ingeniería Informática

Denominación de la materia: Teoría de la información

Profesores a cargo: Massa, Stella Maris; Spinelli, Adolfo Tomás

Emails:

- nievas.nahuel.1998@gmail.com
- mikelajandroolave@gmail.com
- tobiasgiacri@gmail.com

Repositorio: https://github.com/tobiasmdp/teoria-informacion

Resumen

En el siguiente informe se abarcaran temas teóricos sobre la compresión de datos, diferencia entre algoritmos de compresión y medios de transmisión. Se presentan casos experimentales con el fin de comprender y poner en práctica los conceptos teóricos.

Para la comprensión del informe se detallan a continuación los temas a tratar durante el mismo ordenados por orden de aparición:

- Algoritmo de Huffman
- Algoritmo de Shannon-Fano
- Tasa de compactación, rendimiento y redundancia
- Medios de transmisión
- Canales de información
- Probabilidades asociadas a un canal
- Entropías “a - priori” y “a- posteriori”
- Equivocación de un canal
- Información mutua

- Propiedades de la información mutua

Introducción

Para el desarrollo de la primera parte del informe se trabajará con los capítulos veintitrés, veinticuatro y veinticinco del libro “Don Quijote de la Mancha”. Se analizarán todas las palabras involucradas y sus frecuencias. Una vez hecho esto, se pasará a comprimir el mismo. Esta compresión se hará con los algoritmos de Huffman y de Shannon-Fano. Una vez comprimido, se analizará cual tuvo una mejor compresión que el otro. Por último se tendrán que descomprimir los archivos e interpretar los resultados.

Luego de resolver esto, pasaremos a la parte dos en donde a partir de tres canales de comunicación se tendrán que calcular para cada canal la equivocación, la información mutua, las propiedades de cada canal, las probabilidades condicionales de que los símbolos de entradas sean recibidos en la salida, las distintas entropías y las propiedades de la información mutua. Se analizaron los resultados obtenidos.

Para la resolución de esta problemática se han utilizado algoritmos desarrollados en el lenguaje de programación C. Se hizo el código de Shannon-Fano para la primera parte del informe, y el algoritmo de Huffman fue reutilizado del Trabajo Práctico N°1 (ver en la bibliografía). En la segunda parte, se realizó en una planilla de cálculo para que se puedan visualizar mejor los cálculos (ver en la bibliografía).

A medida que se avanza con el trabajo, se irán presentando las ecuaciones pertinentes y desarrollando los conceptos teóricos-prácticos.

Desarrollo

Primera Parte

Aclaraciones

Interpretación de casos particulares

- Al momento de determinar las palabras del diccionario para la compresión, dejando de lado letras y números, se pueden presentar espacios, puntuaciones y saltos de línea, los cuales también se tuvieron en cuenta.
- Espacios y puntuación: Son incluidos en la palabra posterior a su aparición.
- Saltos de línea: Se considera la palabra final del renglón, parte de la primera palabra de su siguiente renglón, concatenadas con un “\n” entre ellas.

Ejemplo:
“...
Quijote, dijo a su escudero:

–*Siempre, Sancho,*
…“
Las palabras del diccionario serán:
“Quijote, ” ; “dijo ”; “a ”; “su ”; “escudero:*n*Siempre, ” ; “Sancho,”

Encabezado

En el header se presenta la información necesaria para la decodificación del archivo. Su estructura es como sigue:

(32 bits)Cantidad de palabras del alfabeto=N			
(32 bits) Cantidad de caracteres de la palabra	(M * 8 bits, donde M será la longitud de palabra) Palabra 1	(32 bits) Longitud en bits de código	(32 bits) Código 1
Caracteres palabra 1	Palabra 1	Longitud del código 1	Código 1
Caracteres palabra 2	Palabra 2	Longitud del código 2	Código 2
Caracteres palabra 3	...	...	...
Caracteres palabra 4	Palabra N	Longitud del código 3	Código 3
Caracteres palabra 5	(32 bits) Tamaño en bits del body		
Body compuesto por el archivo codificado en binario.			

Es necesario destacar que el tamaño del header es muy grande cuando el alfabeto tiene muchas palabras. Entonces, si lo que queremos es compactar un archivo pequeño, puede que lo que ahorramos en compactación lo perdamos en la escritura del header. Esto justifica su uso solo para archivos extensos, y donde no haya equiprobabilidad entre las distintas palabras.

Pequeña muestra de los datos obtenidos:

Palabra	Frecuencia	Código de Huffman	Código de Shannon-Fano
“que “	796	1000	0000
“de “	649	1101	0001
“y “	549	00001	00100
…	…	…	…

“dejaremos ”	1	1111011100100	11111111111101
“breve. ”	1	0110000111011	1111111111111

Antes de continuar, se debe mencionar que en este trabajo no se explican los fundamentos de compactación, redundancia y rendimiento, ya que esto fue parte de una primera entrega, la cual se encuentra en la bibliografía para ser consultada.

Algoritmo de Huffman

El código de Huffman resultó no ser compacto. El rendimiento es de 99.68 % y la redundancia es de 0.32 %. Estos cálculos se hicieron gracias a la obtención de la longitud media de las palabras, que resultó ser de 9.35.
El pseudocódigo del algoritmo utilizado para la obtención de los códigos de Huffman:

```
while (no haya 1 solo elemento){
        buscoMinimos(mínimo 1,mínimo 2);
        nuevoElemento(árbol,Frecuencia mínimo 1+Frecuencia Mínimo 2,puntero mínimo
1,puntero mínimo 2);
        insertarNuevoElementoOrdenadaAscendente(NuevoElemento);
}
```

Algoritmo de Shannon-Fano

El código de Shannon-Fano en este caso no es compacto. El cálculo del rendimiento es de 99.45 %, y su redundancia de 0.55 %. La longitud media es de 9.37.
El pseudocódigo del algoritmo utilizado para la obtención de los códigos de Shannon-Fano:

```
if(!(InicioDelArreglo==FinalDelArrglo)){
        centroDelArreglo=frecuenciaTotal/2;
        while(iterador<= FinalDelArreglo && acumulador<centroDelArreglo){
                AcumuladorDeFrecuencia+=VectorCodigos[iterador].FrecCodigos;
                iterador+1;
        }
        recursionAlMismoMetodo(AcumuladorDeFrecuencia,inicioDelArreglo,iterador,cadenaAu
xiliar1+"0"));
        recursionAlMismoMetodo(FrecuenciaTotal-acumulador,iterador+1,finalDelArreglo,caden
aAuxiliar2,"1"));
}
else{
        Se guarda el código de Shannon-fano con su palabra correspondiente;
}
```

Comparación entre Huffman y Shannon-Fano

La entropía total dada por el archivo otorgado por la cátedra es de 9.316868 binits.

Algoritmo de codificación	Header	Body	Total	Archivo original sin comprimir
Huffman	77.66 KB	15.34 KB	93.0 KB	76.2 KB
Shannon-Fano	77.66 KB	15.44 KB	93.1 KB	76.2 KB

Tasa de compactación TC

La tasa de compactación es una manera de representar y medir lo que un código se ha compactado. Se denota:

$TC = N{:}\,1$

donde N se define como la relación entre el tamaño del código original y el tamaño del código comprimido:

$N = \frac{tamaño\ código\ original}{tamaño\ código\ comprimido}$

Una vez entendido esto, analicemos la tasa de compactación sin considerar el header.

Algoritmo de huffman: $N = \frac{76.2\ KB}{15.34\ KB} = 4.967$

Algoritmo de Shannon-Fano: $N = \frac{76.2\ KB}{15.44\ KB} = 4.935$

Notemos que la tasa de compactación entre el body y el archivo sin descomprimir es aproximadamente de **5:1** para ambas codificaciones. Esto quiere decir que al codificar el texto, se comprimió cerca de **5** veces su tamaño original.

Algoritmo de huffman: $N = \frac{76.2\ KB}{93.0\ KB} = 0.819$

Algoritmo de Shannon-Fano: $N = \frac{76.2\ KB}{93.1\ KB} = 0.818$

Sin embargo, al no ser suficiente el body para realizar una decodificación correcta, se debe agregar el header y esto aumenta considerablemente su tamaño, logrando que este ocupe el 83% del archivo en ambas codificaciones.

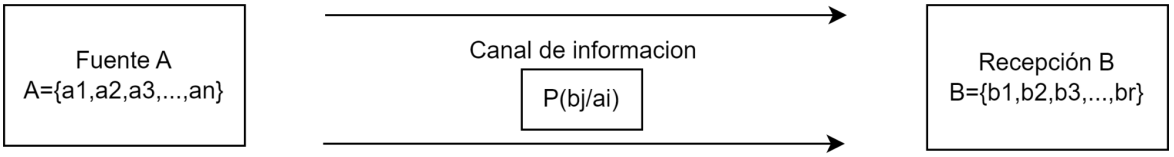
Esto genera una tasa de compactación final **0.82:1**, por lo que no se compacta, sino que aumenta su tamaño.

Este resultado es algo esperado, ya que la cantidad de palabras es muy grande, y al header no se lo comprime. La codificación para ambos casos fue muy similar, difiriendo en 0.1 KB.

Es necesario destacar, que aunque el resultado no produzca compactación alguna, ambos algoritmos cumplen muy bien su función. Para conseguir un mejor resultado, solo se necesitaria ser eficiente en la formulación del header optimizando bien el espacio, e incluso codificando las palabras del diccionario con Huffman o Shannon-Fano.

Segunda Parte

En esta segunda parte se trabajará centrado en el medio de transmisión de un mensaje.
Durante la primera parte del trabajo (Ver en bibliografía), se consideraba que el envío de un mensaje desde un emisor a un receptor no se alteraba durante su transporte. Este mensaje se transportaba a través de un canal de información ideal.
En la realidad esto no ocurre ya que hay muchos factores no controlables que pueden alterar el mensaje, por lo que es importante poder analizar el canal por dónde viaja nuestra información.
Veremos que cuando uno envía un mensaje a través de un canal de información y se desconoce lo que sucede en el medio, pero si se conoce cómo llegan esos símbolos al final de la transmisión, es información suficiente para describir y analizar un canal.



El modelo de transmisión utilizado en este trabajo consta de una fuente A que envía un mensaje compuesto por un alfabeto de símbolos $A = \{a1, a2, a3,... an\}$ con probabilidades $P(a_i) = \{p(a1), p(a2),... p(an)\}$, a través de un canal de información C. El alfabeto recibido $B = \{b1, b2, b3,... br\}$ consta de probabilidades condicionales(probabilidad de que suceda b habiendo sucedido a) .

En esta segunda parte tenemos como objetivo analizar tres canales de información distintos, cada uno con sus probabilidades de entrada y sus matrices del canal. Estas mismas se encuentran enunciadas en el anexo del informe. Se procederá a analizar cada una y por último compararlas entre sí.

	Canal 1	Canal 2	Canal 3
Entradas	5	4	6
Salidas	3	4	4

Cálculos previos

Matriz del canal **P**: En cada posición de la matriz podemos ver representada la probabilidad de la salida de un símbolo conociendo qué símbolo se tiene de entrada $P(b_j/a_i)$.

Matriz de probabilidad mutua **Q**:
En cada posición de la matriz podemos ver representada la probabilidad de entrada de un símbolo "a_i" y la salida de un símbolo "b_i", sin tener información previa.
La fórmula para calcular la probabilidad mutua de cada celda se presenta tal que:

$$P(a_i, b_j) = P(b_j/a_i).\ P(a_i)$$

Matriz de probabilidad condicional hacia atrás: $P(a_i/b_j)$

En cada posición de la matriz podemos ver representada la probabilidad de la entrada de un símbolo conociendo qué símbolo se tiene de salida.
La fórmula para calcular la probabilidad condicional hacia atrás de cada celda se presenta tal que:

$$P(a_i/b_j) = \frac{P(b_j/a_i).\ P(a_i)}{\sum_{k=1}^{r} P(b_j/a_k)P(a_k)} = \frac{P(b_j/a_i).\ P(a_i)}{P(b_j)}$$

Matriz de probabilidad de entropía "a-posteriori": $H(A/b_j)$

En cada posición de la matriz podemos ver representada la cantidad media de información necesaria para representar un símbolo de una fuente con una probabilidad a posteriori P(ai/bj).
La fórmula para calcular la probabilidad mutua de cada celda se presenta tal que:

$$H(A/b_j) = \sum_{B} P(a/b_j) log\ \frac{1}{P(a/b_j)}$$

Vector de probabilidad de salida:

Hasta ahora las probabilidades que mencionamos en el alfabeto de salida son condicionales del símbolo de entrada, pero los símbolos de salida deben tener una probabilidad de aparición general.
La probabilidad de salida entonces se define como la probabilidad de que un símbolo b_j aparezca al final de un canal de información, independientemente del símbolo de entrada.
Esta probabilidad se calcula como:

$$P(b_j) = \sum_{i=1}^{r} P(a_i)P(b_j/a_i)$$

En el anexo B se pueden ver los resultados de las probabilidades de salida de cada canal.

Entropía de entrada H(A)

Para calcular la entropía de entrada utilizaremos la siguiente ecuación:

$$H(A) = \sum_{i=1}^{n} P(a_i) * log(\frac{1}{P(a_i)})$$

Esta entropía representa el número medio (en este caso medido en binits) que necesitamos para poder escribir un símbolo de la fuente A con un probabilidad $P(a_i)$.

La siguiente tabla muestra los resultados obtenidos al aplicar la fórmula a los canales dato.

Canal	H(A)

1	2,1710
2	1,9484
3	2,5273

Podemos ver cómo a mayor cantidad de sucesos de entrada posibles, conocer un suceso de entrada aporta en promedio más información. Esto se debe a que cada uno de los hechos en el canal 3 es menos probable que en el canal 1 y 2

Entropía de salida H(B)

Para calcular la entropía de salida utilizaremos la siguiente ecuación:

$$H(B) = \sum_{j=1}^{r} P(b_j) * log(\frac{1}{P(b_j)})$$

El resultado de esta entropía es el resultado mínimo de preguntas binarias para poder determinar cuál será la salida. Para su obtención se utiliza $P(b_j)$ que es la probabilidad de que un símbolo salga por la salida b_j.

La siguiente tabla muestra las entropías resultantes de cada canal.

Canal	H(B)
1	1,4999
2	1,7777
3	1,9303

Se puede observar que en el canal 1 la entropía es menor ya que al haber solo 3 posibles salidas, la información de las fuentes de entrada tienen menos posibilidad de dispersar a diferencia de los canales 2 y 3.

Entropía media “a-posteriori” / Equivocación / Ruido H(A/B)

Para calcular entropía media “a-posteriori” utilizaremos la siguiente ecuación:

$$H(A/B) = \sum_{j=1}^{r} P(b_j)H(A/b_j)$$

La entropía media “a-posteriori”, también conocida como “Equivocación” de A respecto a B. Representa la pérdida de información de la entrada, a causa del canal. Mide la información que queda en la entrada, luego de observar la salida. Lo que el canal no deja pasar de la entrada. También se le puede llamar ruido, y se puede definir como la mínima cantidad de preguntas binarias en promedio para determinar la entrada conocida la salida. Si H(A/B)=0 quiere decir que el canal no tiene ruido, en consecuencia la transmisión del mensaje es 100% confiable.

La siguiente tabla muestra los resultados obtenidos.

Canal	H(A/B)
1	2,1122
2	1,9100
3	2,4849

H(A/B) en los 3 canales tiene valores cercanos a la entropía de entrada, podemos decir que el ruido de cada canal es muy alto, y hay mucha información proveniente de la entrada, que no deja pasar.

Perdida H(B/A)

Para calcular la pérdida utilizaremos la siguiente ecuación:

$$H(B/A) = \sum_{i=0}^{n} P(a_i, b_j) * log(\frac{1}{P(b_j/a_i)})$$

Está asociada al número de preguntas binarias promedio que se requieren para determinar la salida conociendo la entrada.
La siguiente muestra los resultados obtenidos.

Canal	H(B/A)
1	1,4411
2	1,7393
3	1,8879

Podemos ver como en los tres canales existe pérdida.

Información mutua I(A,B)

Denominamos información mutua a la cantidad de información de A menos la cantidad de información de A una vez observada la salida.

Para calcular la información mutua utilizaremos la siguiente ecuación:
$I(A,B) = H(B) - H(B/A) = H(A) - H(A/B)$

La información mutua cobra relevancia cuando los sucesos no son independientes, debido a que si lo fueran $I(A,B)$=0. Si esto sucede, se pierde completamente la información, o sea que el ruido es total.
Observando las matrices en el anexo A, podemos ver que los sucesos son dependientes.

Notemos que si nos situamos en la fuente A conociendo sus probabilidades de entrada, la información mutua representa la cantidad de información que tenemos del lado opuesto del canal gracias a la información que tenemos de nuestro lado. Esto quiere decir, que hay información de un lado que “cruza” el canal y se puede observar del otro lado. Esto tiene su

sentido práctico, ya que si estamos situados en el canal A con información de la salida de B, no necesitamos H(A) binits para definir nuestros símbolos de entrada, sino que B me aporta algo de información, necesitando solamente (A/B) binits de entrada

Si bien planteamos la información mutua I(A,B), se cumple que es simétrica, por lo que I(B,A)= I(A,B)

La siguiente tabla muestra los resultados obtenidos para los canales dados:

Canal	I(A,B)
1	0,0588
2	0,0384
3	0,0424

Podemos observar que el canal 1 es de los 3, el que mayor información transmite desde las fuentes de entradas a las salidas, esto se debe a que su índice de información mutua es el mayor de los tres.

La información mutua permite obtener un índice que indica lo bien o mal que se está usando un canal (obviamente, lo que cambia es la fuente que genera los símbolos de entrada al canal).

A pesar de que el canal uno es el que presenta mayor valor de información mutua,no se puede alegar que un canal es más adecuado que otro para la transmisión de símbolos, ya que no estamos hablando de una fuente fija, en cada caso la cantidad de símbolos de entrada varía.

Entropía afín H(A,B)

Para calcular la entropía afín utilizaremos la siguiente ecuación:

$$H(A,B) = \sum_{A,B} P(a,b) log \frac{1}{P(a,b)}$$

La entropía afín está relacionada con el suceso simultáneo $P(a_i, b_j)$, y mide su incertidumbre. Es la información media que aporta conocer una determinada entrada a_i y una determinada salida b_j.

La siguiente tabla muestra los resultados obtenidos para los canales dados:

Canal	H(A,B)
1	3,6120
2	3,6877
3	4,4152

Conclusiones

Para concluir con el informe, durante el mismo se abordaron todos los temas vistos en la segunda parte de la materia y se incluyeron algunos de la primera parte.
Durante el transcurso de la primera parte se analizaron los algoritmos de compresión de Huffman y de Shannon-fano. Se observó que su compresión es muy buena, pero su mayor defecto es que al necesitar una tabla diccionario para saber cómo decodificar, esta ocupa mucho más espacio que el archivo original, lo cual es un factor importante a la hora de decidir si utilizar o no estos algoritmos de codificación.

En la segunda parte, a partir de los tres canales dados, se calcularon todas sus probabilidades y sus entropías. No se puede definir cuál de los tres canales es mejor que el otro, ya que las fuentes de entrada, y de salida son diferentes para cada uno, no estamos hablando de una fuente fija. Lo que sí podemos afirmar es que en ninguno de los tres canales son buenos transmitiendo información porque sus valores de información mutua se encuentran muy distantes a sus valores de entropía de entrada.

Bibliografía

Alejandro, M; Giacri, T; Nievas N, (2022).Trabajo Práctico N°1. Recuperado el 21 de Noviembre de 2022, de https://drive.google.com/drive/folders/1wjOULpJZEjyMXZdcSXv_bmvqhBCah2Jl

Abramson N., Teoría de la Información y Codificación, Ed. Paraninfo, 1981.

Anexo:

A. Probabilidades de entrada y matriz del canal

Canal 1

Símbolo	P(ai)
S1	0,2
S2	0,1
S3	0,3
S4	0,3
S5	0,1

Matriz del canal			
P(bj/ai)	B1	B2	B3
S1	0,3	0,15	0,55

S2	0,2	0,4	0,4
S3	0,3	0,15	0,55
S4	0,15	0,4	0,45
S5	0,3	0,2	0,5

Canal 2

Símbolo	P(ai)
S1	0,25
S2	0,33
S3	0,27
S4	0,15

Matriz del canal				
P(bj/ai)	B1	B2	B3	B4
S1	0,2	0,15	0,1	0,55
S2	0,1	0,3	0,1	0,5
S3	0,15	0,15	0,2	0,5
S4	0,15	0,3	0,15	0,4

Canal 3

Símbolo	P(ai)
S1	0,15
S2	0,1
S3	0,2
S4	0,25
S5	0,14
S6	0,16

Matriz del canal				
P(bj/ai)	B1	B2	B3	B4
S1	0,3	0,15	0,2	0,35
S2	0,06	0,15	0,3	0,49
S3	0,2	0,2	0,18	0,42
S4	0,15	0,3	0,2	0,35
S5	0,2	0,18	0,15	0,47
S6	0,2	0,18	0,3	0,32

B. Probabilidades de salida

Canal 1

Símbolo	B1	B2	B3	
P(bj)	0,2450	0,2550	0,5000	1,0000

Canal 2

Símbolo	B1	B2	B3	B4	
P(bj)	0,1460	0,2220	0,1345	0,4975	1,0000

Canal 3

Símbolo	B1	B2	B3	B4	
P(bj)	0,1885	0,2065	0,2150	0,3900	1,0000