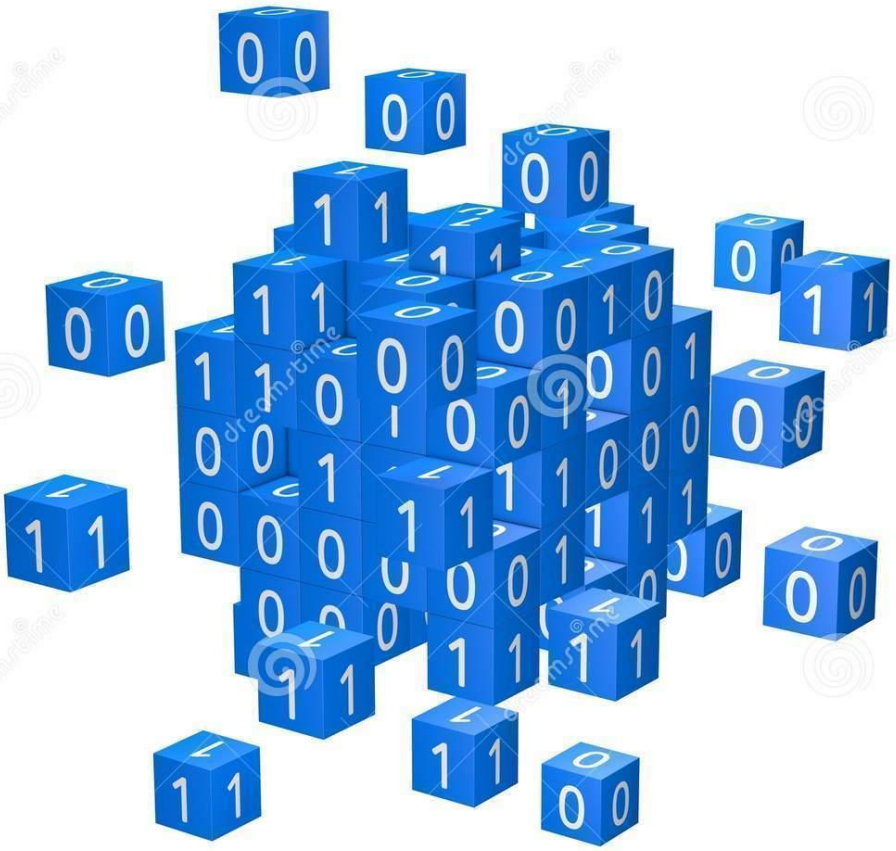
TRABAJO INTEGRADOR №2



Integrantes del grupo:

- Alan Gutiérrez, legajo: 13172, e-mail: alan.gutierrez@alumnos.fi.mdp.edu.ar
- Francisco Stimmler, legajo: 15409, e-mail: franciscostimmler@yahoo.com.ar
- María Josefina Oller, legajo: 11609, e-mail: josefinaoller19@gmail.com

Link del repositorio:
https://github.com/JosefinaOller/TeoriaDeLaInformacion

Fecha de entrega: 22 de noviembre de 2022

Índice

Resumen

En este informe se hablará sobre el análisis y resolución de múltiples problemas que surgen al comprimir información y al transmitirla a través de canales de comunicación. Se desarrollarán las técnicas y algoritmos utilizados para su resolución, y se harán conclusiones respecto a los resultados obtenidos. Se abordarán las temáticas de:

- ➔ Codificación, compresión y descompresión de información;
- ➔ Canales de información.

Introducción

Antes de abordar las temáticas mencionadas previamente, procederemos a explicar los conceptos necesarios para entender el análisis que se va a realizar en el informe.

- Compresión de información: Se refiere a la reducción de la cantidad de información de una fuente de datos, para así emplear una menor cantidad de espacio al almacenarla y transmitirla.
- Canal de comunicación: Un canal es el medio por el cual se transmite información desde una fuente hasta un destino.

Desarrollo

Codificación, compresión y descompresión de información

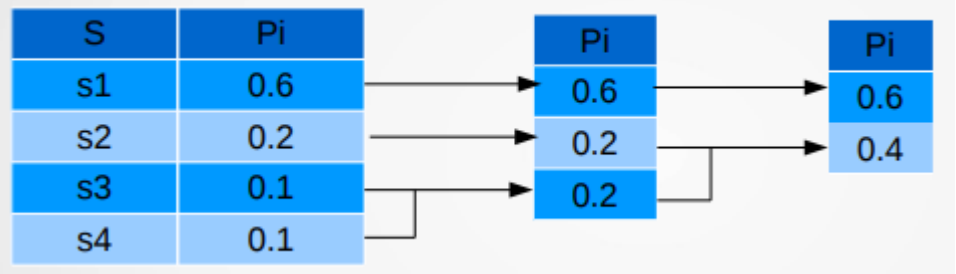
En esta sección, mostraremos el análisis sobre la codificación, compresión y descompresión de información, y canales de comunicación, a partir de datos proporcionados por la cátedra. Se analizó una fuente de símbolos, que es un texto de tamaño grande.

Codificación de Huffman

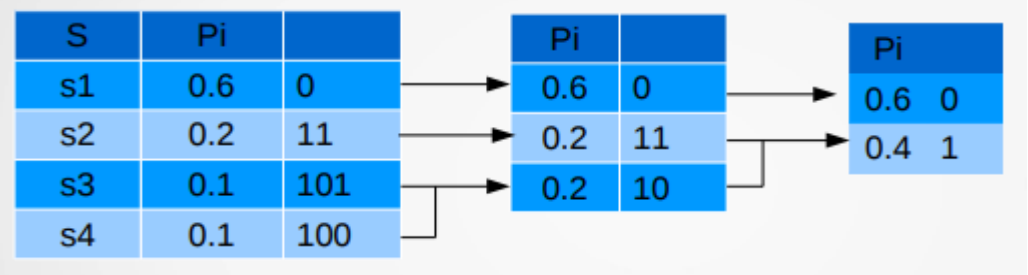
Es un método utilizado para la generación de un código compacto usando el alfabeto binario, mediante la creación de un árbol binario que tiene como hojas a cada símbolo donde el objetivo es la compresión sin pérdida. Dicho método genera secuencias de bits de longitud variable llamados códigos de tal manera que la palabra que aparece con mayor frecuencia tiene la longitud de código más corta, por lo que garantiza una compresión de datos sin pérdidas y evita la ambigüedad.

El algoritmo utilizado es el siguiente:

1. Se ordenan las probabilidades en forma descendente.
2. Se crea una lista de árboles, uno por cada uno de los símbolos del alfabeto, consistiendo cada uno de los árboles en un nodo sin hijos, y etiquetado cada uno con su símbolo asociado y su probabilidad de aparición.



3. Se toman los dos árboles de menor frecuencia, y se unen creando un nuevo árbol. La etiqueta de la raíz será la suma de las probabilidades de las raíces de los dos árboles que se unen, y cada uno de estos árboles será un hijo del nuevo árbol. Esos dos árboles se eliminan de la lista. También se etiquetan las dos ramas del nuevo árbol: con un 0 la de la izquierda, y con un 1 la de la derecha.



4. Se repite el paso 2 hasta que solo quede un árbol.

Una vez generados los códigos hechos por el método de Huffman, se genera un diccionario de palabras origen con su código asignado de la siguiente forma, itera la lista de palabras código de Huffman y por cada palabra se le agrega un carácter especial “>”, en vez de utilizar “:” como una forma de separación con el código. Una vez finalizada la iteración de dicha lista, el diccionario finaliza con una cadena igual a *“|-------FIN DICCIONARIO-------”*, la cual nos será útil en el momento de realizar la descompresión. Todo eso se imprime en un archivo con extensión .huf.

Como la codificación Huffman produce códigos prefijos que siempre consiguen la menor longitud esperada de palabra de código, bajo la restricción de que cada símbolo es representado por un código formado por un número integral de bits, se esperaría que tenga resultados óptimos, es decir, que tenga un alto rendimiento y una menor pérdida al realizar la compresión de información. Realizamos los cálculos de rendimiento, redundancia y la tasa de compresión. El rendimiento fue calculado dividiendo la entropía por la longitud media de la fuente, mientras que la redundancia es 1 menos el valor de rendimiento. Y la tasa de compresión fue calculada como la longitud del archivo original sobre la longitud del archivo comprimido.

Los resultados obtenidos por la codificación de Huffman son los siguientes:

Tasa de compresión: 0,0972 → Es menor a la unidad, eso es debido a que generamos un diccionario de palabras origen con sus códigos, por lo que el tamaño comprimido es mayor al tamaño original del archivo.

Rendimiento: 0,9967 → 99,67% → Se aprovecha al máximo toda la información posible.

Redundancia: 0,0031804 → 0,32% → Casi no hay redundancia de información.

Codificación de Shannon-Fano

Es un procedimiento subóptimo para construir un código, que alcanza una cota L <= H(S) + 2. El objetivo es compresión sin pérdida. Utiliza las probabilidades de ocurrencia de una palabra y asigna un código único de longitud variable a cada uno de ellos.

El algoritmo es el siguiente:

1. Ordenar las probabilidades de forma ascendente.
2. Elegir un símbolo S_k tal que $\left|\sum_{I=I}^{k} P_i - \sum_{i=k+1}^{m} P_i\right|$ sea mínima.
3. Asignar un símbolo diferente a cada uno de los subconjuntos que divide la fuente.
4. Repetir el procedimiento para todos los subconjuntos.

S	Pi					
s1	0.4	1	1			11
s2	0.2		0			10
s3	0.15	0	1	1		011
s4	0.1			0		010
s5	0.06		0	1	1	0011
s6	0.04				0	0010
s7	0.03			0	1	0001
s8	0.02				0	0000

Una vez generados los códigos hechos por el método de Shannon-Fano, se genera un diccionario de palabras origen con su código asignado de la misma forma explicada anteriormente con el método de Huffman, pero en vez de imprimir el diccionario en el archivo con extensión .huf, se imprime en un archivo con extensión .fan.

Como el método utiliza la función de distribución acumulativa y no es óptimo en el sentido de que no consigue la menor longitud de palabra código esperada posible, se esperaría que no tenga buenos resultados en el sentido del rendimiento y la tasa de compresión.

Los resultados obtenidos por la codificación de Shannon-Fano son los siguientes:

Tasa de compresión: 0,000892 → Es un valor muy pequeño, eso significa que el tamaño del archivo comprimido es mucho mayor al tamaño del archivo original, esto es debido a la generación del diccionario de palabras origen con sus códigos.

Rendimiento: 0,0083 → 0,83% → No se aprovecha al máximo toda la información posible.

Redundancia: 0,9917 → 99,17% → Mucha redundancia de información.

Es importante aclarar que consideramos al salto en línea como una palabra más para su mejor compresión (en ambos métodos de codificación).

Descompresión

Para realizar la descompresión de un archivo comprimido hecho por el método Huffman o Shannon-Fano, hay que seguir los pasos siguientes:

Se lee el diccionario con sus códigos de los archivos anteriormente generados, ya sea en el archivo .huf como en el .fan. Se lee de carácter a carácter.

Si encuentra un '>' quiere decir que termina la palabra y lo que sigue hasta el salto de línea ('\n') es el código generado.

Esto se repite hasta que encuentre el carácter '|', lo que quiere decir que se llegó al fin del diccionario. A partir de ahí, y hasta el fin de archivo, se leen los bits para su decodificación de acuerdo a los códigos del diccionario generado.

Los archivos descomprimidos de ambos métodos se encuentran en el repositorio.

Comparación de los resultados

Al realizar la comparación de los resultados dados de los métodos Huffman y Shannon-Fano:

Resultados	**Huffman**	**Shannon-Fano**
Tasa de compresión	**0,0972**	**0,000892**
Rendimiento	**99,67%**	**0,83%**
Redundancia	**0,32%**	**99,17%**

Se analiza que en el método Huffman se obtiene mejores resultados, es decir, tiene un buen rendimiento (rendimiento máximo y redundancia mínima) y una tasa de compresión más alta que la de Shannon-Fano, eso quiere decir que la longitud del archivo Huffman es menor

que la de Shannon-Fano. Era esperable que el método Shannon-Fano no haya generado buenos resultados en el momento de comprimir un archivo con varias palabras diferentes debido a su función de distribución acumulativa, y además no siempre puede generar palabras código de menor longitud posible, eso influye en la longitud del archivo comprimido (teniendo una tasa de compresión inferior).

Canales de comunicación

Como se mencionó anteriormente, un canal es el medio por el que se transmite información desde una fuente a un destino. Los canales tienen las siguientes características:

1. La información se codifica a la entrada y se decodifica a la salida.
2. Puede haber ruido, por lo que se genera la perturbación en la información transmitida.

Para el desarrollo de esta etapa del trabajo, se obtuvieron las matrices para cada canal y se analizaron la equivocación de los canales, y su información mutua junto a sus propiedades. Para su resolución, se utilizó una planilla de cálculo que está en el repositorio.

Veremos más sobre el concepto de un canal de información. Un canal de información viene determinado por un alfabeto de entrada A = $\{a_i\}$, i = 1, 2, ..., r; un alfabeto de salida B = $\{b_j\}$, j = 1, 2, ..., s ; y un conjunto de probabilidades condicionales P (b_j/a_i). P (b_j/a_i) es la probabilidad de recibir a la salida el símbolo b_j cuando se envía el símbolo de entrada a_i.

Probabilidades a priori de entrada P(ai)

Las probabilidades a priori ‘ai’ representan la probabilidad de entrar al canal a través de la entrada ‘ai’, sin tener en cuenta la salida a la cual se dirigirá el mensaje a través del canal. Estas probabilidades asignadas por la cátedra se encuentran en el apéndice del informe.

Matriz del canal

Esta matriz, de MxN, representa las probabilidades de observar que la salida del canal resultó en ‘bj’, sabiendo que se ingresó al canal por el símbolo ‘ai’. Las matrices se muestran en el apéndice del informe.

Probabilidades a priori de salida P(bj)

Representan la probabilidad de salir de un canal por el símbolo ‘bj’ deseado, sin importar el símbolo por el cual se ingrese al canal. Para calcularlas, se utilizó la sumatoria del producto de cada probabilidad a priori de entrada i con el elemento i,j de la matriz del canal. Se obtuvieron los siguientes resultados:

Canal 1: P(b1) = 0,31 P(b2) = 0,35 P(b3) = 0,34

Canal 2: P(b1) = 0,28 P(b2) = 0,27 P(b3) = 0,22 P(b4) = 0,24

Canal 3: P(b1) = 0,24 P(b2) = 0,28 P(b3) = 0,26 P(b4) = 0,23

Al analizar los valores obtenidos, se puede observar que la suma de las probabilidades siempre es 1, ya que siempre se transmite información.

Probabilidades a posteriori P(ai/bj)

Representan la probabilidad de entrada al canal por el símbolo ai sabiendo el símbolo bj de salida del canal. Dichas probabilidades se muestran en la planilla del cálculo. Se puede observar que ninguna de las probabilidades es 0, significa que entrando por el canal 'ai', siempre se saldrá por el canal 'bj'. Por lo que en este caso, al saber de antemano que se salió por el símbolo 'bj', se deduce que se ingresó por el símbolo 'ai'. También se observa que la suma de todas las entradas es igual a 1, tal como se esperaba. Esto es así debido a que un mensaje siempre entra en su totalidad a un canal.

Probabilidades de sucesos simultáneos P(a,b)

Representan el numerador del cálculo de las probabilidades a posteriori P(ai/bj). Dichas probabilidades se muestran en la planilla del cálculo.

Entropía 'a priori' de entrada de H(A) y de salida H(B)

Representan el número medio de binits que se necesitan para representar un símbolo de un canal con probabilidades a priori de entrada y salida. En otras palabras, es el valor medio de información al ingresar o salir de un canal.

Se calculó la entropía 'a priori' del alfabeto de entrada 'A' y del alfabeto de salida 'B', utilizando la siguiente fórmula:

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

Y de esa forma, se obtuvieron los resultados para cada canal.

	Canal 1 (binits)	**Canal 2 (binits)**	**Canal 3 (binits)**
H(A)	2,170950594	1,948388868	2,527250435
H(B)	1,583068912	1,992563825	1,994687158

Podemos observar que el canal 2 de entrada necesita menos cantidad de binits para representar un símbolo. Eso es debido a que tiene menor cantidad de símbolos. El canal 3 de entrada tiene una entropía máxima que el resto de los canales.

Lo mismo para las entropías de salida, se observa que la dispersión es menor al salir del canal 1 que de los canales 2 y 3.

Entropías a posteriori H(A/bj)

Representan el número medio de binits necesarios para representar un símbolo de una fuente con una probabilidad a posteriori P(ai/bj), i= 1, 2, …n. Y también la incertidumbre

sobre la entrada enviada, al haber recibido el mensaje por el símbolo ‘bj’. Se calcularon usando la siguiente fórmula:

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

Los resultados son:

Canal 1:

	B1	**B2**	**B3**
H(A/bj)	2,20183139	2,184750159	2,078528649

Canal 2:

	B1	**B2**	**B3**	**B4**
H(A/bj)	1,924522883	1,933910908	1,985308497	1,820293364

Canal 3:

	B1	**B2**	**B3**	**B4**
H(A/bj)	2,476270567	2,525946715	2,539386226	2,42904851

En todos los canales se puede observar que la incertidumbre varía de símbolo en símbolo, y esto significa que nos conlleva a usar ciertas cantidades de binits para representarla.

Equivocación, información mutua y propiedades

Vamos a definir los conceptos de equivocación con ruido y con pérdida, y de información mutua.

Equivocación del canal de entrada H(A/B) (RUIDO): Representa el ruido de un canal, es decir, lo que va a generar pérdida de información durante la transmisión en un canal. Se calcula con la siguiente fórmula:

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

Equivocación del canal de salida H(B/A) (PÉRDIDA): Representa la pérdida de información que hubo durante la transmisión a través del canal. Se calcula con la misma fórmula de ruido mostrada anteriormente pero se invierten los alfabetos.

Información mutua I(A,B): Es la cantidad de información que se obtiene de A gracias al conocimiento de B. También es la cantidad de información sobre A que atraviesa el canal. Se calcula con la siguiente fórmula: I(A,B) = H(A) - H(A/B).

Entropía afin H(A,B): Es la cantidad de información media que se transmite por el canal cuando a partir de una entrada ocurre una salida. Se calcula con la siguiente fórmula: H(A,B) = H(B) + H(A/B) ó también H(A,B) = H(A) + H(B/A).

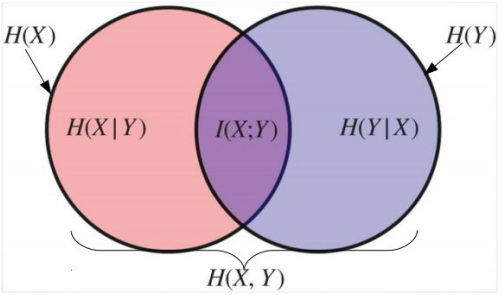
	H(A/B)	**H(B/A)**	**I(A,B) = I(B,A)**	**H(A,B) = H(B,A)**
Canal 1	2,153930027	1,566048344	0,017020567	3,736998939
Canal 2	1,915452325	1,959627282	0,032936543	3,90801615
Canal 3	2,495964997	1,963401719	0,031285439	4,490652155

Al analizar estos resultados, se puede observar que el canal 3 es el que más ruido tiene, y por lo tanto, tiene sentido que también tenga más pérdida de información, esto se debe a la cantidad de símbolos, generando dispersión al atravesar el canal.

El canal 2 es el que más cantidad de información posee, ya que la cantidad de información que se transmite es mayor a la de los otros dos canales. Sin embargo, también podemos decir que el canal 3 es el que más cantidad de información posee después del canal 2, ya que sus valores son bastante próximos relativamente.

Con respecto al ruido, podemos observar que en los tres canales, todos los valores de ruido son menores que sus entropías a priori de entrada, esto quiere decir que en promedio, nunca se pierde información al conocer la salida de un canal.

La información mutua suele indicar qué tanta información se transmite de manera correcta a través de un canal, lo que deja en claro que esta cantidad siempre será mayor o igual a 0, y se podría considerar como un indicador de eficiencia del canal. Dicho esto se puede interpretar que si I(A,B) = H(A) = H(B), el canal será ideal, ya que se transmitirá toda la información que ingresa sin ruido ni errores, ya que H(A/B) y H(B/A) serán iguales a 0. En este caso podemos observar que en los tres canales, todos los valores de I(A,B) son mayores a cero, por lo que no se pierde en absoluto información por observar la salida del canal. Y además, los valores obtenidos no son nulos ya que, los símbolos de entrada y salida no son estadísticamente independientes. Aunque se transmite mayor cantidad de información por el canal 2, los canales 1 y 3 son bastante eficientes en comparación.



Finalmente la entropía afín más grande es la del tercer canal. Este suceso se da gracias a que es el canal que más ruido tiene y su valor es mayor a la entropía a priori de B, o

también se da gracias a que el canal que más pérdida de información y su valor es menor a la entropía a priori de A .

Conclusiones

- El código Huffman es mucho más eficiente y óptimo que el de Shannon-Fano, esto se puede observar en los valores obtenidos de rendimiento, por ejemplo 99,67% vs 0,83%. Esto se debe a que está basado en las probabilidades de los símbolos y no en una función de distribución acumulativa. Sin embargo, el método de Shannon-Fano es más simple de implementar.
- Era esperable que los tamaños de los archivos comprimidos no fueran menores al tamaño del archivo sin comprimir. Esto es debido a la generación del diccionario de palabras origen junto a sus códigos, y dicha tabla no está comprimida, por lo tanto, las tasas de compresión se ven afectadas por esta razón.
- La tasa de compresión de Shannon-Fano es mucho menor que la de Huffman, ya que la longitud del archivo comprimido por Shannon-Fano es un valor muy grande, esto es porque el mismo método no siempre puede generar palabras código de menor longitud posible y además se basa en una función de distribución acumulativa, por lo que no es óptimo para comprimir un archivo de muchas palabras distintas.
- En todos los canales, la equivocación es menor a la entropía de A, por lo que en promedio nunca se pierde información al conocer la salida del canal.
- El ruido que se genera en promedio durante la transmisión por el canal es grande.
- Los símbolos de entrada y salida no son estadísticamente independientes, por lo que, además del ruido que se genera, podemos concluir que no son canales de comunicaciones ÓPTIMOS.
- Es imposible evitar que un canal tenga ruido y pérdida, por lo que siempre se tratará de minimizar el impacto producido por el mismo.

Apéndice

Canal 1:

Símbolo entrada p(a)	P(i)
S1	0,2
S2	0,1
S3	0,3
S4	0,3
S5	0,1

Matriz del canal 1:

	B1	B2	B3
S1	0,3	0,3	0,4
S2	0,4	0,4	0,2
S3	0,3	0,3	0,4

S4	0,3	0,4	0,3
S5	0,3	0,4	0,3

Canal 2:

Símbolo entrada p(a)	**P(i)**
S1	0,25
S2	0,33
S3	0,27
S4	0,15

Matriz del canal 2:

	B1	**B2**	**B3**	**B4**
S1	0,2	0,3	0,2	0,3
S2	0,3	0,3	0,2	0,2
S3	0,3	0,2	0,2	0,3

S4	0,3	0,3	0,3	0,1

Canal 3:

Símbolo entrada p(a)	**P(i)**
S1	0,15
S2	0,1
S3	0,2
S4	0,25
S5	0,14
S6	0,16

Matriz del canal 3:

	B1	**B2**	**B3**	**B4**
S1	0,2	0,3	0,2	0,3
S2	0,3	0,3	0,3	0,1

S3	0,2	0,2	0,3	0,3
S4	0,3	0,3	0,2	0,2
S5	0,2	0,3	0,3	0,2
S6	0,2	0,3	0,3	0,2