Teoría de la información
Trabajo Integrador N°2

- Integrantes: Gutierrez Paredes José María
- Email: GutierrezJoseMaria01@hotmail.com

- Fecha de entrega: 22/11
- Repositorio: https://github.com/guteeeeeeeee/teoriaInformacion.git

Índice

Resumen

La realización de este trabajo busca asimilar los conceptos vistos durante la teoría y poder aplicarlos de manera práctica para así poder sacar nuestras propias conclusiones.

Los temas que se tratan son la codificación y compresión de archivos, y los canales de comunicación.

La codificación se estudia porque resulta fundamental ya que permite, mediante el uso de algún algoritmo, que un mensaje transmita la información que contiene en las menores longitudes posibles. De esta manera logrando reducir su tamaño lo que resulta en una compresión de su tamaño original.

Además se estudian los canales de comunicación porque cumplen un papel fundamental en el esquema de la comunicación ya que es el medio por el que se transmite el mensaje enviado desde la fuente a su destino.

Introducción

El problema tratado es poder codificar el mensaje enviado mediante los algoritmos ya estudiados. Lo que se busca es lograr que el mensaje que se quiera transmitir pueda ser enviado de la manera más rápida y ocupando la menor cantidad de espacio posible (compresión). Esto se logra mediante un código compacto. Se estudiaron dos métodos para lograr estos códigos: el algoritmo de Huffman y el de Shannon-Fano.

El otro problema tratado es el de analizar las propiedades de ciertos canales de comunicación para así poder compararlos entre sí y obtener conclusiones acerca de sus propiedades. En ellos lo más común es que haya ruido y ocurran pérdidas de la información enviada ya que los símbolos de entrada X pueden convertirse en símbolos de salida Y con ciertas probabilidades propias de cada canal. Provocando de esta manera que el receptor conociendo el símbolo de salida no pueda saber con certeza cuál símbolo fue el que se le envió provocando así problemas en la comunicación del mensaje.

Desarrollo

Parte 1

La realización de esta parte se puede dividir en 3: procesamiento, codificación y compresión.

En el procesamiento lo que se hizo fue leer el archivo de texto dado e ir almacenando cada palabra con sus datos importantes como su frecuencia de aparición, largo de la palabra. Luego de procesar todo el archivo se obtuvieron propiedades importantes como la entropía de la fuente y el largo medio de las palabras leídas sin codificar.

En la etapa de codificación lo que se hizo fue aplicar los algoritmos de Huffman y Shannon-Fano para así poder obtener la codificación binaria de cada palabra.

El primero consiste en ir agrupando en un nodo padre los dos símbolos (también son nodos) con menor probabilidad de aparición logrando de esta manera ir formando un árbol binario. Una vez construido, mediante su recorrido se podrá saber la codificación correspondiente de cada símbolo.

En cambio el algoritmo de Shannon-Fano lo que hace es, a partir de ordenar todas las probabilidades de emisión, ir subdividiendo el vector que las contiene en dos partes iguales buscando que tengan las probabilidades de cada parte sean las más cercanas posibles. De esta manera a cada subdivisión se le otorga un bit distinto de codificación.

Una vez obtenidas las codificaciones de cada palabra leída luego de haber aplicado ambos métodos sigue la etapa de compresión.

Primero se escribe la tabla con las palabras y sus correspondientes codificaciones. Lo que primero se indica es en 1 byte la cantidad de palabras codificadas, luego para cada palabra se sigue el siguiente formato: 1 byte para indicar cantidad de caracteres que conforman la palabra, 1 byte por cada carácter, 1 byte para indicar la cantidad de bits de la codificación y 2 bytes para la codificación. Esto debido a que en C no se puede emplear unidades más pequeñas de escritura que el byte.

Luego se vuelve a leer el archivo de texto original y se reemplaza cada palabra por su codificación.

De esta forma se obtiene el archivo binario con el archivo con el texto original comprimido pero también contiene la tabla de codificaciones que impacta en su tamaño. Esto es debido a que para poder informar al receptor la información necesaria para que pueda descomprimirlo debe utilizar la tabla que debe contener cada palabra con su correspondiente codificación. La codificación no es problema ya que ocupa 2 bytes, el problema ocurre al momento de almacenar la palabra (hay 4155) ya que se ocupará 1 byte por cada carácter y en promedio está formada por 7.44 caracteres por lo que la tabla solo con las palabras ocupará 30 Kb, con el resto 47 Kb.

Se puede concluir que se debe dar otro manejo a la tabla de codificaciones, la cual pesa en total 47 Kb, ya que no puede pesar más que todo el texto comprimido (16 Kb para Shannon-Fano). Pero no todo es malo ya que también se puede concluir que la codificación fue exitosa debido a que se redujo considerablemente el tamaño del texto de 79 Kb a 16 Kb (sin contar la tabla). Esto ocurrió debido a que se aprovechó que los algoritmos codificaron los códigos en bits y se los fue escribiendo en el archivo uno tras otro aprovechando el tamaño de los integers en C logrando de manera que no se desperdicie espacio y se escriba un bit tras otro.

Esta conclusión está dada con el ejemplo del algoritmo de Huffman pero también aplica para Shannon-Fano.

Se obtiene para el texto sin codificar un rendimiento de 0.04 y una redundancia de 0.96. Es un resultado lógico ya que las palabras utilizadas no tienen longitudes que aprovechen las probabilidades de emisión de cada una, generando así que se transmitan palabras importantes en muchos más caracteres de los necesarios (redundancia).

Mientras que para Shannon-Fano y Huffman se obtiene un rendimiento de 0.99 y 0.01 de redundancia. Es lo esperable ya que aunque tienen diferentes métodos

su objetivo es el mismo: lograr códigos compactos. Por lo que cada palabra codificada tendrá la menor longitud posible de acuerdo a su probabilidad.

Para el texto sin codificar cada palabra tiene un largo medio de 4.75 caracteres donde cada uno ocupa 1 byte (8 bits) mientras que para la codificación de Huffman el largo medio de cada una es de 9.44 bits. De esta manera se hace evidente el nivel de compresión de tamaño que se logra a partir de la codificación de palabras.

Segunda parte

Para caracterizar los canales utilizados en la transmisión de información se utiliza para cada uno la matriz de canal la cual nos indica las probabilidades de que salga la salida Y si se emite la entrada X. Éstas se encuentran en el apéndice de este informe.

Canal 1

P(B1)	P(B2)	P(B3)
0.31	0.35	0.34

H(A/B1)	H(A/B2)	H(A/B3)
2.201831	2.184750	2.078529

H(A/B)	I(A,▫
2.153930	0.017021

Canal 2

P(B1)	P(B2)	P(B3)	P(B4)
0.275	0.273	0.215	0.237

H(A/B1)	H(A/B2)	H(A/B3)	H(A/B4)
1.924523	1.933911	1.985308	1.820293

H(A/B)	I(A,▫
1.915452	0.032937

Canal 3

P(B1)	P(B2)	P(B3)	P(B4)
0.235	0.28	0.26	0.205

H(A/B1)	H(A/B2)	H(A/B3)	H(A/B4)
2.476271	2.525947	2.539386	2.459829

H(A/B)	I(A,B)
2.453694	0.073556

En el análisis de los canales dados se encuentra que todos contienen ruido.

Las probabilidades de los símbolos de salida son cercanas a 1/k siendo k cantidad de símbolos de salida lo que demuestra que son estadísticamente independientes.

Se puede verificar en cada canal estudiado que las entropías a posteriori son valores cercanos entre sí. Recordando que estos valores son la cantidad media de incertidumbre de conocer el valor de entrada sabiendo el símbolo de salida. Esto indica que el receptor no sabría cual símbolo puede ser el de entrada aun

conociendo el de salida. Por lo que los canales estudiados no serían los ideales para transmitir la información dada ya que no habría una correlación entre la entrada y la salida lo que dificultaría la interpretación por parte del receptor.

Otro variable a analizar es la equivocación del canal la cual indica la cantidad de información sobre la entrada que no deja pasar el canal. Mientras mayor sea este número menor fiabilidad tendrá el canal. En el caso ideal de un canal sin ruido este valor sería nulo ya que esto significa que si conozco la salida, conozco perfectamente la entrada por lo que no habría problemas de interpretar el mensaje enviado ya que durante su transmisión por el canal no hubo ruido que perturbara la información transmitida.

Esto es malo ya que la información obtenida al conocer el símbolo de salida no servirá para identificar al símbolo de entrada por lo que se tiene una pérdida de la información.

La información mutua se puede definir como la cantidad de información que se obtiene de la entrada conociendo la salida. Para un correcto uso del canal este valor tiene que ser máximo.

En los canales estudiados este valor es cercano a cero lo que significa que no son los óptimos para ser utilizados en la transmisión de las fuentes dadas.

Dado que los canales ya están dados y no se pueden modificar lo que se podría hacer es modificar las probabilidades de emisión buscando minimizar la equivocación del canal y maximizar la información mutua.

Si se desea enviar un mensaje el canal elegido sería el segundo ya que este tiene el menor ruido aunque tengo un bajo valor en la cantidad de información.

Conclusión

Como conclusión se puede decir que fue un trabajo interesante ya que se aplicaron conceptos vistos en la teoría que luego de ser implementados quedaron mucho más claros.

Se evidenció la importancia de la codificación ya que los textos que utilizamos generalmente tienen poco rendimiento y mucha redundancia. Lo que significa que para transmitir la información que contienen ocupan mucho más espacio del necesario en el almacenamiento y en su transmisión. Además las palabras utilizadas resultan ser más largas que la información que transmiten. Pero a partir de otorgar longitudes de acuerdo a la probabilidad de aparición se logra una lectura más rápida de las palabras más utilizadas y por lo tanto a la hora de almacenarlas una reducción importante de su tamaño respecto al texto original.

Lo que también se dio cuenta es que a la hora de comprimir se le debe dar otro tratamiento a la tabla de codificaciones ya que al no estar comprimida, para indicar su codificación, cada palabra ocupará mucho más espacio del necesario por lo que lo recomendable sería reducir su tamaño de alguna manera porque sino toda la compresión del texto original se ve opacada por ella.

También que hay que tener cuidado con los canales de comunicación utilizados ya que pueden provocar problemas a la hora de transmitir el mensaje que deseamos que llegue y entienda el receptor. En los casos estudiados los canales tenían ruido y pérdida de la información por lo que el receptor al recibir el mensaje luego de pasar por el canal no podrá lo que le fue enviado ya que las salidas que le llegan no tienen correspondencia con los símbolos enviados. Resulta fundamental el análisis para detectar canales que se adecuen a la fuente que queremos utilizar ya que aunque se hagan correctamente los pasos para armar y codificar un mensaje finalmente al receptor le llegará algo que no podrá entender.

Anexo

Canal 1:

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

Matriz del canal			
	B1	B2	B3
S1	0,3	(a) 0,3	1-B1-B2 0,4
S2	(b) 0,4	0,4	1-B1-B2 0,2
S3	0,3	(a) 0,3	1-B1-B2 0,4
S4	(a) 0,3	0,4	1-B1-B2 0,3
S5	0,3	(b) 0,4	1-B1-B2 0,3

Canal 2:

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

Matriz del canal					
	B1	B2	B3	B4	
S1	0,2	(a) 0,3	(b) 0,2	1-B1-B2-B3	0,3
S2	(a) 0,3	0,3	(b) 0,2	1-B1-B2-B3	0,2
S3	(c) 0,3	(b) 0,2	0,2	1-B1-B2-B3	0,3
S4	(c) 0,3	0,3	(a) 0,3	1-B1-B2-B3	0,1

Canal 3:

Símbolo	P(i)
S1	0,15
S2	0,1
S3	0,20
S4	0,25
S5	0,14
S6	0,16

Matriz del canal					
	B1	B2	B3	B4	
S1	0,2	(a) 0,3	(b) 0,2	1-B1-B2-B3	0,3
S2	(c) 0,3	(a) 0,3	0,3	1-B1-B2-B3	0,1
S3	(b) 0,2	0,2	(c) 0,3	1-B1-B2-B3	0,2
S4	(a) 0,3	0,3	(b) 0,2	1-B1-B2-B3	0,2
S5	0,2	(c) 0,3	(a) 0,3	1-B1-B2-B3	0,2
S6	(b) 0,2	(c) 0,3	0,3	1-B1-B2-B3	0,2