Guia 3 ITI : Resolucion

Di Maria, Franco Martín,
Padrón : 100498
2do. Cuatrimestre de 2019
75.06 - Organización de Datos
Facultad de Ingeniería
Universidad de Buenos Aires

1. Ejercicio 1

1.1. Enunciado

Se tiene un archivo con 10 caracteres en total formado por tres caracteres distintos (ej: ABC). De todos los archivos posibles con estas características mostrar el archivo de máxima entropía que se pueda comprimir mejor usando LZ77. No es necesario comprimir el archivo.

1.2. Solución

Sea X, un archivo con 10 caracteres en total formado por tres caracteres distintos (ej: ABC).

Se desea que entropía del archivo X (H(X)) sea máxima, pero compresible.

$$H(X) = \sum_{i=0} P_i \log_2(P_i) = P_a \log_2(P_a) + P_b \log_2(P_b) + P_c \log_2(P_c)$$

H(X) es máximo cuando:

$$P(a) = P(b) = P(c) = \frac{1}{3}$$

Pues entonces los 3 caracteres son equiprobables.

Pero la longitud del archivo es de 10 caracteres, por lo que uno de ellos deberá tener una probabilidad:

$$P(x_0) = \frac{4}{10}$$

$$x_0 \epsilon \{a, b, c\}$$

mientras que para los otros dos será:

$$P(x_1) = P(x_2) = \frac{3}{10}$$

$$x_1 \epsilon \{a, b, c\} - \{x_0\}$$

$$x_2 \epsilon \{a, b, c\} - \{x_0, x_1\}$$

LZ77 (o LZSS, formalmente hablando), comprime a base de repeticiones de patrones, de un largo mínimo igual a 2 caracteres.

Un archivo de máxima entropía, pero fácilmente compresible por LZ77 podría ser:

$$A\ A\ A\ B\ B\ B\ C\ C\ C$$

1.2.1. Compresión por LZ77 (no pedido por el enunciado)

$$0A \quad 1(0,2) \quad 0B \quad 1(0,2) \quad 0C\ 1(0,2)$$

1.3. EXTRA : ¿Y si fuese el menos compresible, pero aun así compresible por LZ77?

Un archivo incompresible por LZ77, puede ser el siguiente:

$$A\ A\ B\ A\ C\ C\ B\ B\ C\ A$$

En este caso todos los patrones son : AA, AB, BA, AC, CC, CB, BB, BC, CA, AAB, ABA, BAC, etc ...

Como se podrá comprobar ninguno de estos patrones se repite por lo que la complejidad será máxima, y no se podrá comprimir mediante el LZ77

Para hallar un archivo que se pueda comprimir, y mantener tal grado de complejidad, se puede reemplazar al ultimo carácter del archivo anterior, por otro que provoque la repetición de un patrón. Por ejemplo, podría reemplazar la ultima A por una C, de forma de obtener el siguiente archivo:

$$A\ A\ B\ A\ C\ C\ B\ B\ C\ C$$

En este caso, se repite el patrón CC, por lo ahora el archivo es comprimible por LZ77.

1.3.1. Compresión por LZ77

La compresión LZ77 del ultimo archivo podría ser l siguiente:

$$0A \quad 0A \quad 0B \quad 0A \quad 0C \quad 0C \quad 0B \quad 0B \quad 1(3,2)$$

2. Ejercicio 2

2.1. Enunciado

Explique en que casos sería una buena idea usar un compresor aritmético estático de orden 3.

2.2. Solución

Del apunte de la cátedra [1], para cualquier algoritmo de compresión estadístico:

> En un modelo de orden "n" la frecuencia de cada carácter se calcula en base a los "n" caracteres anteriores. Cada combinación de "n" caracteres del archivo nos proporciona un contexto y en base a dicho contexto calculamos la probabilidad del carácter siguiente.

Por lo tanto, un compresor aritméticos de orden 3, conviene usarlo, cuando en el archivo hay varios patrones de 3 caracteres, repetidos.

Además, al ser estático, estaríamos guardando tablas de frecuencias para cada contexto de 3 caracteres. Si solo consideramos los 256 caracteres posibles de la tabla ASCII, estas tablas estarán ocupando 4GB en el archivo comprimido.

Para que la compresión tenga sentido, y estas tablas no terminen aumentando el tamaño del archivo en lugar de reducirlo, necesitaríamos que el archivo sea bastante grande, es decir, que originalmente ocupe un tamaño mucho mayor a estos 4GB mencionados.

Esto vale para cualquier compresor estadístico estático de orden 3.

Específicamente para compresión aritmética, habría que agregar que destaca sobre otros algoritmos (como Huffman), si la longitud óptima de los códigos (de la codificación de los símbolos) es una cantidad no entera de bits.

3. Ejercicio 3

3.1. Enunciado

Tenemos un compresor aritmético dinámico de orden 0 que trabaja procesando bit por bit. Si comprimimos un archivo que está formado por una serie de 1000 bits en 1 y luego dos bits en 0. ¿cuántos bits ocupará el archivo comprimido

3.2. Solución

$Intervalo\ inicial = [0,\ 1)$

$bit\ 0 \quad \| \quad P(1) = \frac{1}{2} \quad P(0) = \frac{1}{2} \quad \| \quad Nuevo\ intervalo = [0,\ 1 * \frac{1}{2}) = [0,\ \frac{1}{2})$

$bit\ 1 \quad \| \quad P(1) = \frac{2}{3} \quad P(0) = \frac{1}{3} \quad \| \quad Nuevo\ intervalo = [0,\ \frac{1}{2} * \frac{2}{3}) = [0,\ \frac{1}{3})$

$bit\ 2 \quad \| \quad P(1) = \frac{3}{4} \quad P(0) = \frac{1}{4} \quad \| \quad Nuevo\ intervalo = [0,\ \frac{1}{3} * \frac{3}{4}) = [0,\ \frac{1}{4})$

$bit\ n \quad \| \quad P(1) = \frac{n+1}{n+2} \quad P(0) = \frac{1}{n+2} \quad \| \quad Nuevo\ intervalo = [0,\ \frac{1}{n+2})$

$(n \quad \epsilon \quad [0, 999])$

$bit\ 999 \quad \| \quad P(1) = \frac{1000}{1001} \quad P(0) = \frac{1}{1001} \quad \| \quad Nuevo\ intervalo = [0,\ \frac{1}{1002})$

Siguen los ultimos dos bits que son 0 :

$bit\ 1000 \quad \| \quad P(1) = \frac{1000}{1002} \quad P(0) = \frac{2}{1002}$

$\| \quad Nuevo\ intervalo = [\frac{1000}{(1002)^2},\ \frac{1}{1002})$

$bit\ 1001 \quad \| \quad P(1) = \frac{1000}{1003} \quad P(0) = \frac{3}{1003}$

$\| \quad Nuevo\ intervalo = [\frac{(1000)^2}{1003 * (1002)^2},\ \frac{1}{1002}) =$

$\| \quad [0{,}000993032,\ 0{,}0000998003)$

i	2^{-i}	bit	acumulado
0	1	0	0
1	0.5	0	0
2	0.25	0	0
3	0.125	0	0
4	0.0625	0	0
5	0.03125	0	0
6	0.015625	0	0
7	0.0078125	0	0
8	0.00390625	0	0
9	0.001953125	0	0
10	0.0009765625	1	0.0009765625
11	0.00048828125	0	0.0009765625
12	0.00024414062	0	0.0009765625
13	0.0001220703125	0	0.0009765625
14	0.00006103515625	0	0.0009765625
15	0.00003051757813	0	0.0009765625
16	0.00001525878906	1	0.0009918212891
17	0.000007629394531	0	0.0009918212891
18	0.000001907348633	1	0.0009956359863

Cuadro 1:

Por lo tanto la codificación será:
0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 1
Por ende, la respuesta a esta pregunta es: se necesitan 19 bits para comprimir, por compresión aritmética dinámica, al archivo descrito.

4. Ejercicio 4

4.1. Enunciado

Una planta industrial decidió instalar un sistema monitor de temperatura, a fin de obtener registro de las variaciones que existen, y poder tomar acciones de ser necesario. Dicho monitor cuenta con un sensor que emite cada 5 segundos un registro (fecha: AAAAMMDD , hora: HH:MM:SS , Temperatura: XX.XX , Variación: Numérico, puede ser positivo o negativo). Más allá de las acciones inmediatas que se puedan tomar, esta información se quiere almacenar para realizar consultas o análisis a futuro. Se pide proponer una solución que permita almacenar estos datos comprimidos. Se pueden utilizar uno o más algoritmos de los vistos en clase, o proponer variantes adaptadas a la estructura específica de los datos con los que se cuentan. Se debe explicar cómo queda la estructura final del archivo, y el análisis en el que se basó la solución.

4.2. Solución

Falta terminar

5. Ejercicio 5

5.1. Enunciado

Determinar si las siguientes afirmaciones son V / F justificando la respuesta:

5.1.1. Item A

La entropía es una aproximación de cuanto se puede comprimir, dado que no podemos calcular cuanto se puede comprimir un string

5.1.2. Solución

VERDADERO.
La entropía nos da una noción de la longitud media de compresión de cada símbolo en el archivo, por lo que multiplicarla por la cantidad de símbolos en el archivo, obtenemos una aproximación de a cuantos bits es posible comprimir el archivo.

5.1.3. Item B

Una forma posible de comprimir un stream de datos es utilizar un huffman estático.

5.1.4. Solución

FALSO.

Un stream de datos, es un flujo constante de datos, donde generalmente, el tiempo entre que llega un dato y el siguiente es muy corto. Por esta razón, para comprimir un stream de datos, es necesario (en el caso de compresores estadísticos) que el algoritmo sea dinámico

5.1.5. Item C

La entropía puede utilizarse para construir clasificadores de texto.

5.1.6. Solución

FALSO.

Lo que puede usarse para construir clasificadores de texto es el mejor compresor construido hasta la fecha (en este caso PAQ). Comparando la compresión junto a conjuntos archivos de cierta clasificación cada conjunto, puede clasificarse el texto en determinado conjunto

5.1.7. Item D

Un compresor estadistico estático comprime siempre mejor que un compresor estadístico dinámico.

5.1.8. Solución

VERDADERO.

Un compresor estadístico estático, lee una primera vez el archivo a comprimir, y obtiene las frecuencias de cada carácter en el archivo. Con lo que, luego, en un segunda pasada, comprime cada símbolo basándose en su probabilidad de ser el siguiente. Por otro lado, un compresor estadístico dinámico, parte de un de la suposición de que todos los símbolos son equiprobables, y a medida que recorre el archivo, balancea estas probabilidades. La única excepción ocurre cuando, en el caso dinámico, la suposición de que todos los símbolos son equiprobable es cierta, por lo que el nivel de compresión sera el mismo que en el caso estático.

5.1.9. Item E

Podemos determinar la longitud final del archivo comprimido utilizando huffman estático de orden 1, calculando la entropía y multiplicándola por la cantidad de caracteres del archivo.

5.1.10. Solución

FALSO.

Esto solo es cierto, cuando la entropía resulta en una cantidad entera de bits.

5.1.11. Item F

Tenemos 2 archivos, uno con longitud pequeña y el otro muy grande que se comprimen utilizando huffman estático de orden 5. Si observamos que tienen la mismas tablas de frecuencias podemos afirmar que el cociente entre el tamaño del archivo sin comprimir y el tamaño del archivo comprimido será similar.

5.1.12. Solución

FALSO
En huffman estático de orden 5, tendremos tablas que, si se tienen en cuenta los 256 símbolos de la tabla ASCII, ocuparan aproximadamente 1 TB

$$\frac{s1}{c1 + 1TB}? = \frac{s2}{c2 + 1TB}$$

$$1TB >> s1 > c1$$

Entonces el primer termino tiende a cero.

$$s2 >> 1TB > c2$$

Entonces el segundo termino tendera a S2
Por ende es FALSO.

5.1.13. Item G

Todo archivo con complejidad de kolmogorov baja tendrá una entropía de Shannon baja.

5.1.14. Solución

VERDADERO.
Pues la entropía de Shannon es un medida para aproximar la complejidad de Kolmogorov.

5.1.15. Item H

Es imposible que un archivo comprimido mediante huffman estático de orden 1 iguale la máxima compresión dada por la entropía del mismo

5.1.16. Solución

FALSO,
Si la entropía es una cantidad no entera de bits, entonces, huffman estático ocupara mas bits por cada símbolo, que la longitud media establecida por la entropía.

6. Bibliografia

Referencias

[1] 75.06, 95.58 Organización de Datos - Apunte del Curso - October 19, 2018 - v2.0 ; Luis Argerich, Natalia Golmar, Damián Martinelli, Martín Ramos Mejía, Juan Andrés Laura - Universidad de Buenos Aires - Facultad de Ingenieria