Codificador de fuente para una fuente discreta sin memoria

Autor: Gerez Jimenez, Juan Jose Armando

Resumen

Este informe muestra los conceptos de la codificación para fuentes discretas sin memoria en sistemas de comunicaciones digitales. Se investigan temas como la entropía, codificación óptima, el teorema de codificación de fuente, y el uso del algoritmo de Huffman para generar códigos eficientes. Además, se implementan simulaciones en Python para la generación de tablas de frecuencias de caracteres en archivos de texto, la creación de fuentes discretas sin memoria y la construcción de codificadores Huffman para finalmente evaluar los resultados obtenidos en base a la teoría.

Introducción

En el campo de la teoría de la información, la codificación de fuentes es un proceso fundamental para la compresión de datos. La codificación de Huffman es un algoritmo utilizado para minimizar la longitud promedio de los códigos (optimización),el cual se basa en un árbol binario de búsqueda para asignar códigos binarios a los símbolos de acuerdo con sus probabilidades de ocurrencia. Este método es particularmente eficiente para fuentes discretas sin memoria, donde cada símbolo es independiente de los demás y tiene una probabilidad de ocurrencia definida.

Objetivo 1: Investigación Bibliográfica:

Conceptos de fuente de información discreta y fuente discreta sin memoria (DMS):

Una fuente de información discreta es un sistema que produce una secuencia de símbolos discretos finitos (el número de miembros generados puede ser finito o contable infinito). Cada símbolo tiene una probabilidad asociada, lo que permite modelar el proceso de generación de datos. Una fuente discreta sin memoria (DMS, por sus siglas en inglés) es un tipo específico de fuente discreta en la que cada símbolo generado es independiente de los símbolos anteriores, lo que implica que la probabilidad de un símbolo no depende de los valores previos generados por la fuente.

Concepto de entropía y entropía de una DMS:

La entropía de una fuente de información discreta es una medida de la información. Para una fuente discreta sin memoria, la entropía H(X) se calcula como:

$$H[X] = -\sum_j p_j \log p_j.$$

Gallager, R. (2008). *Principles of Digital Communication - cap. 2.5.1*

donde p(j) es la probabilidad del símbolo j. La entropía proporciona un límite inferior teórico para la cantidad de bits necesarios para codificar los símbolos de la fuente sin pérdida de información.

Concepto de Codificación de Fuente: Unicode y UTF-8:

La codificación de fuente tiene como objetivo reducir la redundancia en los datos generados por una fuente para minimizar la cantidad de bits necesarios para representar la información. El estándar Unicode define un sistema universal para representar caracteres de todos los lenguajes. UTF-8 es una codificación de longitud variable donde cada símbolo puede ser representado por 1 a 4 bytes, lo que permite una representación eficiente de caracteres más frecuentes.

Codificación óptima para una DMS, Teorema de codificación de fuente:

El Teorema de codificación de fuente establece que es posible comprimir los datos generados por una fuente de información hasta su entropía (aunque en realidad será lo más próximo posible) sin pérdida de información. En otras palabras, ningún esquema de codificación puede, en promedio, utilizar menos bits por símbolo que la entropía de la fuente.

Codificación Huffman:

La codificación de Huffman es un algoritmo óptimo para codificar una fuente discreta sin memoria. Genera un código de longitud variable donde los símbolos más frecuentes se representan con códigos (binarios) más cortos, lo que reduce el promedio de bits por símbolo.

Ley de los grandes números:

La ley de los grandes números afirma que, a medida que aumenta el tamaño de la muestra, la frecuencia de los símbolos tiende a aproximarse a su probabilidad real. Esto es fundamental para estimar la probabilidad de ocurrencia de símbolos en una DMS en base a una muestra.

Sub-Objetivos del Objetivo 2:

1) Desarrollo de un generador de cadenas de símbolos basado en una fuente discreta sin memoria.

2) Implementación de un analizador de frecuencias para calcular las probabilidades de ocurrencia de los símbolos.

3) Construcción de un árbol de Huffman a partir de las frecuencias de los símbolos.

4) A partir de un libro.txt en formato utf-8, generar códigos binarios, árbol de Huffman y evaluación de su eficiencia.

Desarrollo

1. Generador de Cadenas de Símbolos:
Se desarrolló un generador de cadenas (ver Objetivo2_1.ipynb) que utiliza las probabilidades de ocurrencia de los símbolos para producir secuencias aleatorias de longitud definida. Esto simula el comportamiento de una fuente discreta sin memoria y posteriormente servirá como entrada para el codificador de Huffman.

2. Análisis de Frecuencias:
El analizador de frecuencias (ver Objetivo2_2.ipynb) se utiliza para calcular la frecuencia de cada símbolo en la cadena generada. A partir de estas frecuencias, se puede determinar la probabilidad de ocurrencia de cada símbolo, lo cual servirá para la construcción del árbol de Huffman más adelante.

3. Construcción del Árbol de Huffman:
El árbol de Huffman es una estructura de datos que se construye de manera jerárquica, fusionando los símbolos con las menores probabilidades de ocurrencia en nodos intermedios (ver Objetivo2_3.ipynb). Este proceso continúa hasta que se obtiene un solo nodo raíz, que representa el árbol completo. Cada camino desde la raíz hasta un símbolo representa el código binario del símbolo correspondiente.

4. Generación de Códigos Binarios:
Una vez que se ha construido el árbol de Huffman, los códigos binarios para cada símbolo se generan siguiendo los caminos desde la raíz hasta las hojas del árbol (ver Objetivo2_3.ipynb). Los códigos más cortos se asignan a los símbolos con mayor probabilidad, garantizando una longitud promedio mínima del código.

5. Implementación del Codificador de Huffman:
El código desarrollado permite la codificación y decodificación de cadenas de símbolos utilizando los códigos binarios generados por el árbol de Huffman. Esta implementación se evaluó para verificar su eficiencia y precisión en la compresión de datos. Se usa como base de prueba un archivo .txt ubicado dentro de la carpeta de trabajo (ver Objetivo2_4.ipynb), el cual previamente se sacó del Proyecto Gutemberg, en formato utf-8.

Programas:

Objtivo2_1.ipynb: implementa una función que analiza las frecuencias de caracteres en un texto. La función analizaFrecuencias toma una cadena de caracteres como entrada y devuelve un diccionario donde las claves son los caracteres y los valores son la probabilidad de que cada carácter aparezca en el texto. También se incluyen pruebas (test_analizaFrecuencias) para verificar que la función devuelve resultados correctos en diferentes casos de prueba.

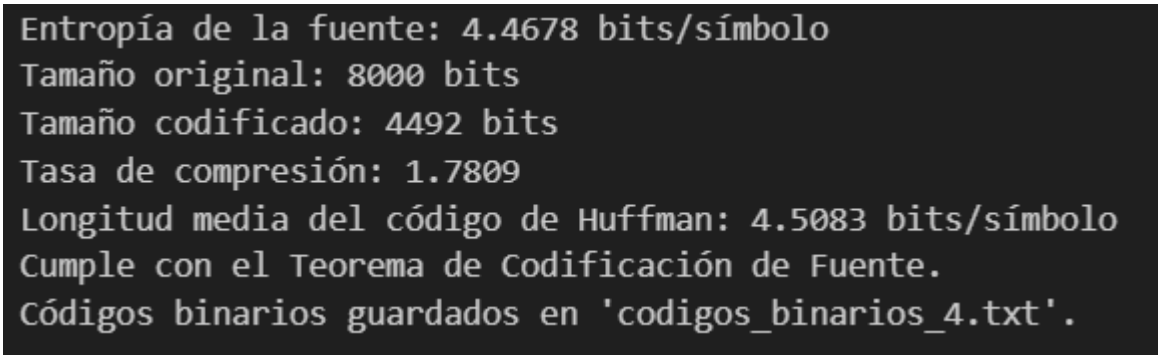
Objetivo2_2.ipynb: implementa una clase llamada FuenteDiscretaSinMemoria que simula una fuente discreta sin memoria. La clase toma como entrada un diccionario donde las claves son los símbolos y los valores son sus probabilidades asociadas. La clase incluye un método generar_cadena que permite generar una secuencia de símbolos con base en las probabilidades dadas y una longitud especificada. Cada símbolo se genera de manera independiente, lo que cumple con la definición de una fuente discreta sin memoria.

Objetivo2_3.ipynb: Contiene métodos pertenecientes a los programas anteriores, y se le suman métodos para la generación del árbol de Huffman, tales como Comb_nodos_min, Hacer_arbol_huff y Hacer_codigo, que respectivamente se encargaron de unir los nodos de menor frecuencia, darle estructura al arbol y generar el código para cada símbolo. Además se incluyen métodos para la creación de archivos .txt para facilitar la visualización del árbol.

Objetivo2_4.ipynb: Es muy similar al programa Objetivo2_3.ipynb, con la diferencia de que ahora se utiliza a La_Divina_Comedia.txt para hacer el árbol de Huffman, y se suman métodos para el cálculo de su ¨eficiencia¨, como por ejemplo calcular_entropia y evaluar_rendimiento (que son operaciones matemáticas y comprobaciones).

Resultados y Discusión

Los resultados obtenidos muestran que la codificación de Huffman es eficiente en la reducción de la redundancia de los datos. La longitud promedio de los códigos generados es cercana a la entropía de la fuente, lo que valida la teoría de la información.



(Objetivo2_4.ipynb)

Interfaz de Programación de Aplicaciones (API):
Se desarrolló en Python una aplicación que permite a los usuarios generar cadenas de símbolos, construir el árbol de Huffman, generar códigos binarios y realizar la codificación y decodificación de datos. Esta aplicación es fácil de usar y puede integrarse en otros sistemas de compresión de datos. Simplemente modificando algunos parámetros dentro de los métodos de prueba el usuario puede obtener distintos resultados según lo que el mismo necesite.

Contraste con la Teoría:
Los resultados obtenidos se contrastan con la teoría presentada en el libro de Gallager, confirmando que la codificación de Huffman es óptima para fuentes discretas sin memoria en términos de la longitud promedio de los códigos generados (en nuestro caso se pudo apreciar que la longitud promedio estuvo muy próxima a la entropía).

$$\mathsf{H}[X] \leq \overline{L}_{\min} < \mathsf{H}[X] + 1 \quad \text{bit/symbol.}$$

Gallager, R. (2008). *Principles of Digital Communication - cap. 2.5.2*

Nota: También se observó en el programa Objetivo2_2.ipynb como al colocar un k demasiado bajo (cantidad de símbolos), los porcentajes de aparición de los símbolos en el archivo txt estaba muy lejana a la probabilidad ingresada en el diccionario, pero al colocar un k mucho más grande las probabilidades se acercaban (en otras palabras se apreció el ¨efecto¨ de la ley de los grandes números).

Conclusión

La implementación del codificador de Huffman para una fuente discreta sin memoria (el cual no habría podido lograrse sin la investigación y comprensión previa de los conceptos descritos anteriormente dentro del Objetivo 1) demuestra ser una solución eficiente y efectiva para la compresión de datos (como vimos en ¨Resultados¨). En conclusión, la investigación y desarrollo práctico realizados en este proyecto proporcionan una mejor comprensión de los conceptos teóricos y su aplicación práctica en sistemas de comunicación digital.

Referencias

Gallager, R. G. (2008). Principles of Digital Communication. Cambridge University Press.