

Compresión de datos sin pérdida

Tarea 2

Año 2021

1. Objetivos

Esta tarea práctica tiene los siguientes objetivos específicos:

- Consolidar a través de la práctica la teoría de codificación universal vista en el curso.
- Ensayar técnicas de programación apropiadas para la implementación de algoritmos de compresión.
- Desarrollar la capacidad de análisis y comunicación escrita de resultados experimentales.

2. Programación

Implementar un codificador/decodificador universal para la familia de fuentes de Markov de orden k sobre el alfabeto binario \mathcal{B} , donde k es un parámetro, $0 \leq k \leq 64$. El codificador lee una secuencia binaria de un archivo (interpretando cada byte como un octeto de símbolos) y la codifica utilizando un codificador aritmético y una distribución universal asintóticamente óptima para esta familia (en el sentido mín-máx). El codificador recibe por parámetro el nombre del archivo a codificar, el orden de Markov k y el nombre del archivo comprimido a crear. El decodificador recibe por parámetro el nombre del archivo a descomprimir (generado por el codificador), el orden de Markov k y el nombre del archivo descomprimido a crear.

3. Experimentación

Para el trabajo de experimentación usaremos archivos de imágenes blanco y negro en formato PBM¹, en las cuales cada píxel se representa con un bit (0 para blanco, 1 para negro). Este formato de archivos es muy sencillo: contiene un pequeño cabezal que incluye un identificador del formato de archivo, el ancho (W) y la altura (H) de la imagen, seguido de la secuencia de valores de los píxeles. En primer lugar se representa la fila de píxeles de más arriba usando $\lceil W/8 \rceil$ bytes; el primer byte representa los 8 píxeles de más a la izquierda de la fila, con el bit más significativo representando el píxel de más a la izquierda. Los bytes subsiguientes representan los píxeles de la fila siguiendo el mismo criterio, ordenados de izquierda a derecha. El último de los $\lceil W/8 \rceil$ bytes de la fila representa los píxeles de más a la derecha; los bits menos significativos de este byte se rellanan arbitrariamente si W no es múltiplo de 8. Las siguientes filas se representan una a continuación de la otra, siguiendo este mismo formato, ordenadas desde arriba hacia abajo.

Para simplificar el trabajo, comprimiremos cada archivo de imagen como si fuera una secuencia homogénea de bits, ignorando el hecho de que cada archivo incluye una pequeña cantidad de bits que no representan valores de píxeles estrictamente. Para los experimentos están disponibles en EVA un conjunto de imágenes, sin perjuicio de que puedan usarse también otras. La tarea de experimentación consiste en comprimir cada imagen usando diferentes valores de k , calculando en cada caso la tasa de compresión (bits/bit), calculada como el cociente entre el tamaño del archivo comprimido y el tamaño del archivo original. Repetir el experimento para varias imágenes y analizar los resultados.

¹<https://en.wikipedia.org/wiki/Netpbm>

4. Consideraciones generales

- Para resolver esta tarea puede utilizarse un codificador aritmético disponible públicamente. El codificador descrito en [1] está disponible en varios sitios².
- Esta tarea es parte de la evaluación del curso y se resuelve individualmente.
- Los lenguajes de programación que se aceptan son C y C++.
- Las decisiones de diseño e implementación que influyen sobre el rendimiento de los programas son importantes y serán evaluadas.
- Las entregas se aceptan en la página del curso hasta el 5 de julio inclusive. Debe incluirse:
 - Código fuente acompañado de un makefile, script de compilación o similar.
 - Instrucciones de cómo compilar y ejecutar los experimentos.
 - Informe donde se describa la solución y se analicen los resultados de los experimentos.

Referencias

- [1] WILLEN, I. H., NEAL, R. M., AND CLEARY, J. G. Arithmetic coding for data compression. *Communications of the ACM* 30, 6 (june 1987).

²Ej. <https://marknelson.us/posts/1991/02/01/arithmetic-coding-statistical-modeling-data-compression.html>