Compresión de datos sin pérdida
Tarea 2

Año 2017

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

Se debe implementar un codificador y decodificador de acuerdo al siguiente esquema general:

1. La codificación se realiza utilizando un codificador aritmético.
2. El codificador se diseña para la familia de modelos de Markov de orden k, donde k es un parámetro en el rango $0 \leq k \leq 3$. El parámetro k se debe poder configurar en tiempo de ejecución o compilación a criterio de cada estudiante; tener en cuenta sin embargo que la configuración en tiempo de ejecución puede facilitar la tarea de experimentación.
3. El alfabeto de entrada, $\mathcal{A}$, es el subconjunto $\{0, 1, 2, \ldots, M-1\}$, donde $M \leq 256$. El parámetro M se debe poder configurar en tiempo de ejecución o compilación a criterio de cada estudiante. El codificador lee siempre un símbolo por byte del archivo de entrada, independientemente del valor de M; puede asumirse que todos los bytes del archivo de entrada, interpretados como entero sin signo, tienen un valor menor que M.
4. Puede asumirse que un archivo comprimido con ciertos valores de k y M va a ser decodificado con esos mismos valores de los parámetros, es decir, no es necesario incluir una codificación de los valores de k y M como parte del archivo comprimido.
5. La distribución de probabilidad que se le asigna al símbolo en posición $i+1$ de una secuencia X se define de la siguiente manera

$$P\{X_{i+1} = b | X_{i-k+1}^{i} = a_1 \ldots a_k\} = \frac{n_{b|a_1 \ldots a_k} + 1}{n_{a_1 \ldots a_k} + |\mathcal{A}|}, \quad i \geq k, \tag{1}$$

donde $n_{b|a_1 \ldots a_k}$ es la cantidad de veces que el símbolo b ocurre a continuación de los símbolos $a_1 \ldots a_k$ en la secuencia X^i, y

$$n_{a_1 \ldots a_k} = \sum_{b \in \mathcal{A}} n_{b|a_1 \ldots a_k}.$$

Se debe establecer algún criterio para asignar una probabilidad a los primeros k símbolos; por ejemplo podría aplicarse (1) asumiendo que la secuencia X está precedida por una cadena X_{-k+1}^{0} fija preestablecida.

2.1. Opcional

Adaptar el esquema anterior para trabajar con archivos de secuenciación de ADN en el mismo formato que la tarea 1. Para esto agregamos al modelo estadístico una máquina de estados con un estado asociado a cada uno de los tres tipos de sección del formato FASTQ. Denotamos a estos estados:

- S_T, para las líneas de texto con la identificación y descripción (líneas de tipo 1 y 3),
- S_G, para las líneas con secuencias de genoma (líneas de tipo 2),
- S_Q, para las líneas con indicadores de calidad (líneas de tipo 4).

Cada estado determina un modelo estadístico y un alfabeto que se aplica para codificar los símbolos de la sección correspondiente a ese estado. En otras palabras, cada estado s mantiene su propio juego de contadores, $n_{.|a_1 \ldots a_k}$, $a_i \in \mathcal{A}_s$, para un alfabeto $\mathcal{A}_s$ específico para cada estado (por ejemplo, $\mathcal{A}_{S_G}$ podría ser el conjunto $\{A, C, G, T, \backslash n\}$). La asignación de probabilidad (1) para el símbolo X_{i+1} se calcula utilizando los contadores asociados al estado en el cual ocurre ese símbolo.

3. Experimentación

Realizar experimentos con diferentes órdenes de Markov; comparar y analizar los resultados de compresión que se obtienen con cada uno. Para analizar los resultados es conveniente graficar la evolución de la tasa de compresión a medida que aumenta la cantidad de símbolos comprimidos. Si implementó la parte opcional, conviene analizar cada sección por separado.

4. Consideraciones generales

- Para resolver esta tarea puede utilizarse un codificador aritmético disponible públicamente. El codificador descrito en [1] está disponible en varios sitios,[1] pero puede usarse otro.
- 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 30 de noviembre inclusive. Debe incluirse:
  - Código fuente acompañado de un makefile, sript de compilación o similar.
  - Referencias en línea a los datos utilizados en los experimentos.
  - Instrucciones de cómo compilar y ejecutar los experimentos.
  - Informe donde se describe la solución y se analizan los resultados de los experimentos.

Referencias

[1] I. H. Willen, R. M. Neal, and J. G. Cleary, "Arithmetic coding for data compression," *Communications of the ACM*, vol. 30, no. 6, june 1987.

[1] Por ejemplo en http://marknelson.us/1991/02/01/arithmetic-coding-statistical-modeling-data-compression/