Compresión de Datos

Introducción
Compresión estadística
Compresión de Huffman
Half Coding

Introducción

Compresión de datos

- Es la representación de información utilizando menos bits que en el original
- Permite reducir las necesidades de almacenamiento y el uso de capacidad de red
- Existen algoritmos de compresión con y sin pérdida de información

Compresión lossy y lossless

- Con pérdida / lossy / compactación: se utilizan cuando es aceptable alguna pérdida de fidelidad (imágenes, audio, video) con tal de mejorar la tasa de compresión. No los veremos en este curso
- Sin pérdida / lossless: permiten recrear exactamente el archivo original.

Tipos de compresores lossless

- Compresores estadísticos
  - Se basan exclusivamente en la probabilidad de un símbolo de aparecer
- Compresores no estadísticos
  - Run-length
    - Codifican secuencias repetidas
  - Predictores
  - Por sustitución
- Compresores híbridos

Compresión estadística

Teoría de la información

- Estudia y cuantifica los procesos que se realizan sobre la información
- Provee una *medida* de la información
- Hay más información en un suceso cuando su probabilidad de ocurrencia es baja

Teoría de la información

- ¿Cuánta información hay en las siguientes frases?
  - En Londres el tiempo está…
  - El “caballo blanco de San Martín” era …
  - Maradona fue un destacado…

Teoría de la información

- ¿Cuánta información hay en las siguientes frases?
  - En Londres el tiempo está … lluvioso
  - El “caballo blanco de San Martín” era … en realidad un burro
  - Maradona fue un destacado…jugador de fútbol

Entropía

- Propuesta por Claude Shannon en 1948
- Dentro de una fuente F, recibimos el símbolo F_i, que tenía probabilidad p_i

$$I(F = F_i) = -\log(p_i)$$

$$H(F) = \sum_{i=1}^{n} p_i * I(F = F_i) = -\sum_{i=1}^{n} p_i * \log(p_i)$$

- H(F) es la entropía de toda la fuente F

Códigos prefijos

- La codificación de un valor posible nunca puede empezar con la codificación de otro
  - Esta codificación no es prefija:
    - 1 = “0” 2 = “01” 3 = “10” 4 = “11”
  - porque el archivo “010110”, se podría interpretar tanto como “1341” o como “223”
  - Esta codificación sí es prefija
    - 1 = “0” 2 = “10” 3 = “110” 4 = “111”

Compresión estática y dinámica

- Compresión estática
  - Una pasada para obtener estadísticas
  - Otra pasada para comprimir
  - Nivel de compresión óptimo
- Compresión dinámica
  - Se obtienen estadísticas al mismo tiempo que se comprime
  - Más rápido, pero comprime menos

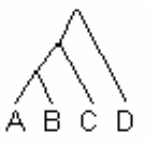
Compresión de Huffman

Compresión de Huffman

- Es un compresor estadístico
- Se construyen árboles de símbolos
- A cada símbolo se le asigna una codificación en bits
- La codificación de un símbolo se determina recorriendo el árbol
- La asignación es óptima: ninguna otra asignación que utilice códigos de cantidad entera de bits comprime más

Árboles de Huffman

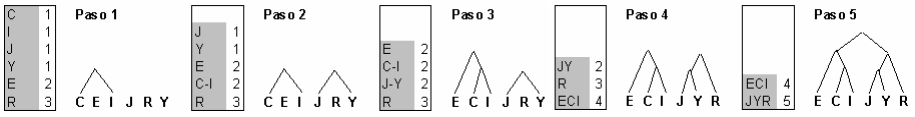
- Si se tiene el árbol:



- Las codificaciones de los símbolos son:
  - A: 000 B: 001 C: 01 D: 1
- El código dado por el árbol es prefijo

Huffman estático

Generación del árbol para la fuente JERRYRICE



Codificación: C=010, E=00, I=011, J=100, R=11, Y=101

Representación final: 100-00-11-11-101-11-011-010-00

- 22 bits de longitud contra 21.74 dados por la entropía

Huffman estático

- La emisión debe incluir la tabla de frecuencias
- El descompresor lee la tabla de frecuencias, arma el árbol y descomprime

Emisión en bytes

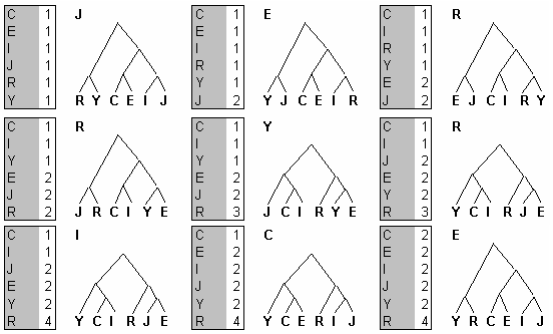
- Luego de comprimir, puede quedar una cantidad de bits que no es múltiplo de 8
- Para señalar el fin de archivo, se puede:
  - Agregar un 1 y completar con ceros hasta el fin del byte

  - Indicar la longitud del archivo comprimido al principio
  - Utilizar un carácter extra como EOF

Huffman dinámico

- Todas las frecuencias se asumen inicialmente iguales a 1
- Se construye el árbol completo en cada paso para la tabla de frecuencias actual
- En la siguiente diapositiva se muestra el árbol final antes de cada emisión, no la construcción paso a paso

Huffman dinámico



Emisión: 111-101-110-01-110-10-011-010-101

25 bits (3 más que el estático)

Huffman dinámico

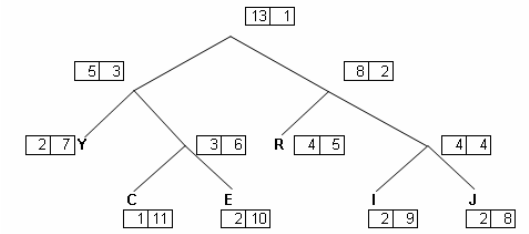
- La emisión no incluye la tabla de frecuencias
- El compresor y el descompresor conocen implícitamente el alfabeto (por ejemplo, los bytes del 0 a 255), no hace falta transmitirlo
- Para emitir bytes completos, se usan las mismas técnicas que las vistas en “Emisión en bytes”

Huffman dinámico: manejo eficiente

- Logra una actualización rápida del árbol de un paso al otro
- No cambia el nivel de compresión
- Para tener un árbol óptimo alcanza con que:
  - Si $P(A) > P(B)$ entonces $L(A) \leq L(B)$
  - Si $P(A) = P(B)$ entonces $|L(A)-L(B)| \leq 1$

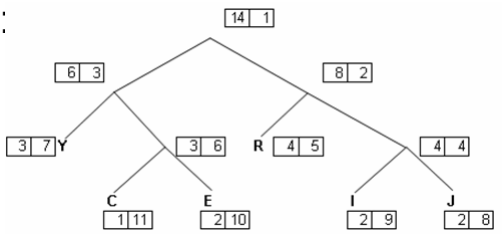
Huffman dinámico: manejo eficiente

- Se numeran los nodos en forma ascendiente, de la raíz hacia abajo y de derecha a izquierda, y se escribe la frecuencia total del nodo



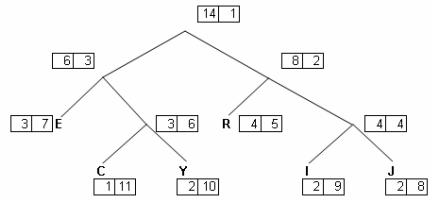
Huffman dinámico: manejo eficiente

- Si aumenta en 1 la frecuencia de la Y, la estructura del árbol no necesita cambiar porque la frecuencia aumenta de izq a der, de abajo a arriba:



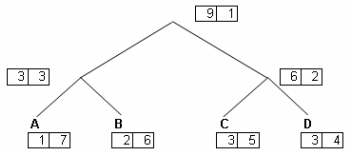
Huffman dinámico: manejo eficiente

- Si en vez de eso aumenta en 1 la frecuencia de la E, la estructura del árbol cambia: se intercambia el nodo de la E (nodo 10) y el nodo de menor numeración con frecuencia superior (nodo 7):



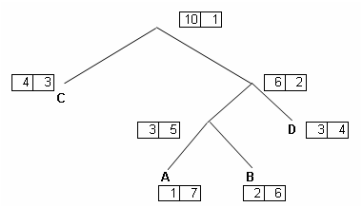
Huffman dinámico: manejo eficiente

- Caso particular: al aumentar la frecuencia de C en este árbol



el nodo candidato para el intercambio es el 3. Se realiza el intercambio y el árbol queda..

Huffman dinámico: manejo eficiente



- Este árbol sigue siendo un árbol de Huffman ya que cumple con todas las propiedades

Half Coding

Half Coding

- Combina compresión run-length con estadística (Huffman)
- Sólo es eficiente cuando un carácter es muy repetido, más que el 50% de ocurrencias
- Para caracteres tan frecuentes, Huffman no es efectivo porque no puede codificar con menos de 1 bit
- Puede ser estático o dinámico

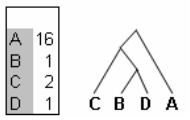
Half Coding

- Se codifica la longitud del carácter más probable
- Por cada ocurrencia, se escribe su longitud en binario, se le suma 1, se le quita el bit más significativo (un 1) y se reemplaza 0=α y 1=β:
  - 1=α, 2=β, 3=αα, 4=αβ, 5=βα, 6=ββ…
- Se trata a α y β como dos caracteres más y se los codifica con Huffman

Half Coding: ejemplo estático

- Ejemplo: AAAABAACAAAAAAACDAAA
- Con Huffman estático se comprimiría a 26 bits:

  1-1-1-1-010-1-1-00-1-1-1-
  1-1-1-1-00-011-1-1-1



- Con Half Coding se convierte a:

  α β B β C α α α C D α α

y se comprime a 24 bits:

0-111-100-111-110-0-0-0-110-101-0-0