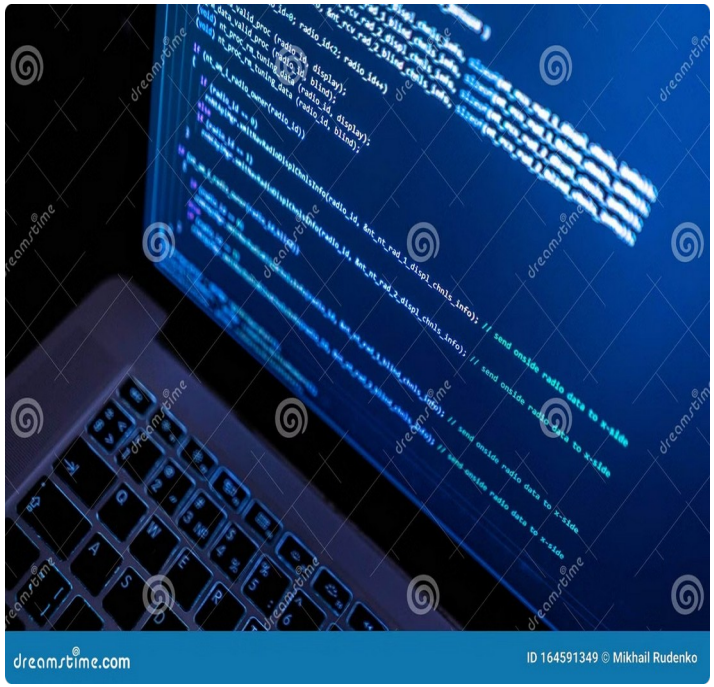
Teoría de Información y la Comunicación
Clase 7

Unidad IV: Codificación y Compresión

Temas a tratar:

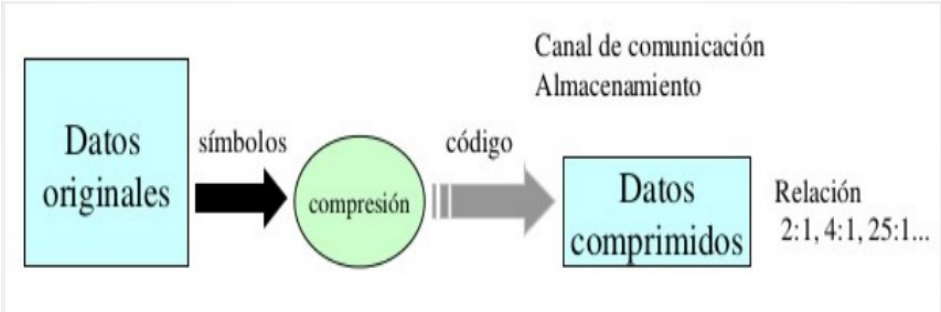
Compresión de datos
Métodos de compresión
Métodos de compresión sin pérdida
Algoritmo de Huffman
Run Lenght Coding (RLC)



La compresión reduce el tamaño de un archivo:

– Para ahorrar espacio al guardarlo.
– Para ahorrar tiempo al transmitirlo.
– La mayoría de los archivos tienen mucha redundancia.

Unidad IV: Codificación y Compresión



Texto → AAAAAAAAAAABBBBBBBBCCCCC

(A,11)(B,8)(C,5) Run Length Code (RLC)

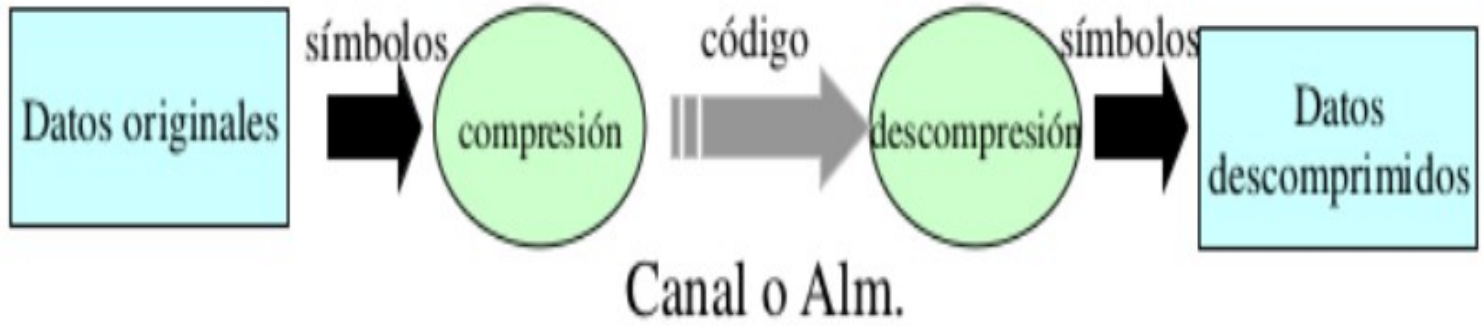
La compresión permite:

Aumentar la capacidad De almacenamiento de los dispositivos.

Transmitir en menos tiempo información por el canal

Ejemplos de compresión:

1. Texto
2. Códigos Fuente y Objeto
3. Datos Numéricos
4. Imágenes
5. Gráficos
6. Sonidos
7. Datos Binarios (Fax, etc.)
8. Imágenes



1

2

Compresor

El compresor codifica un mensaje, transformándolo en una representación más compacta.

Descompresor

El descompresor decodifica el mensaje comprimido, recuperando el mensaje original o una aproximación del mismo.

Unidad IV: Codificación y Compresión

La compresión de datos ha estado omnipresente desde la antigüedad:

- **Sistemas numéricos,**
- **lenguajes naturales,**
- **notación matemática**

Ha desempeñado un papel central en la tecnología de las comunicaciones:

- **Braille**
- **Código Morse**
- **Sistema telefónico**

Y es parte de la vida moderna:

- **ZIP**
- **MP3**
- **MPEG**

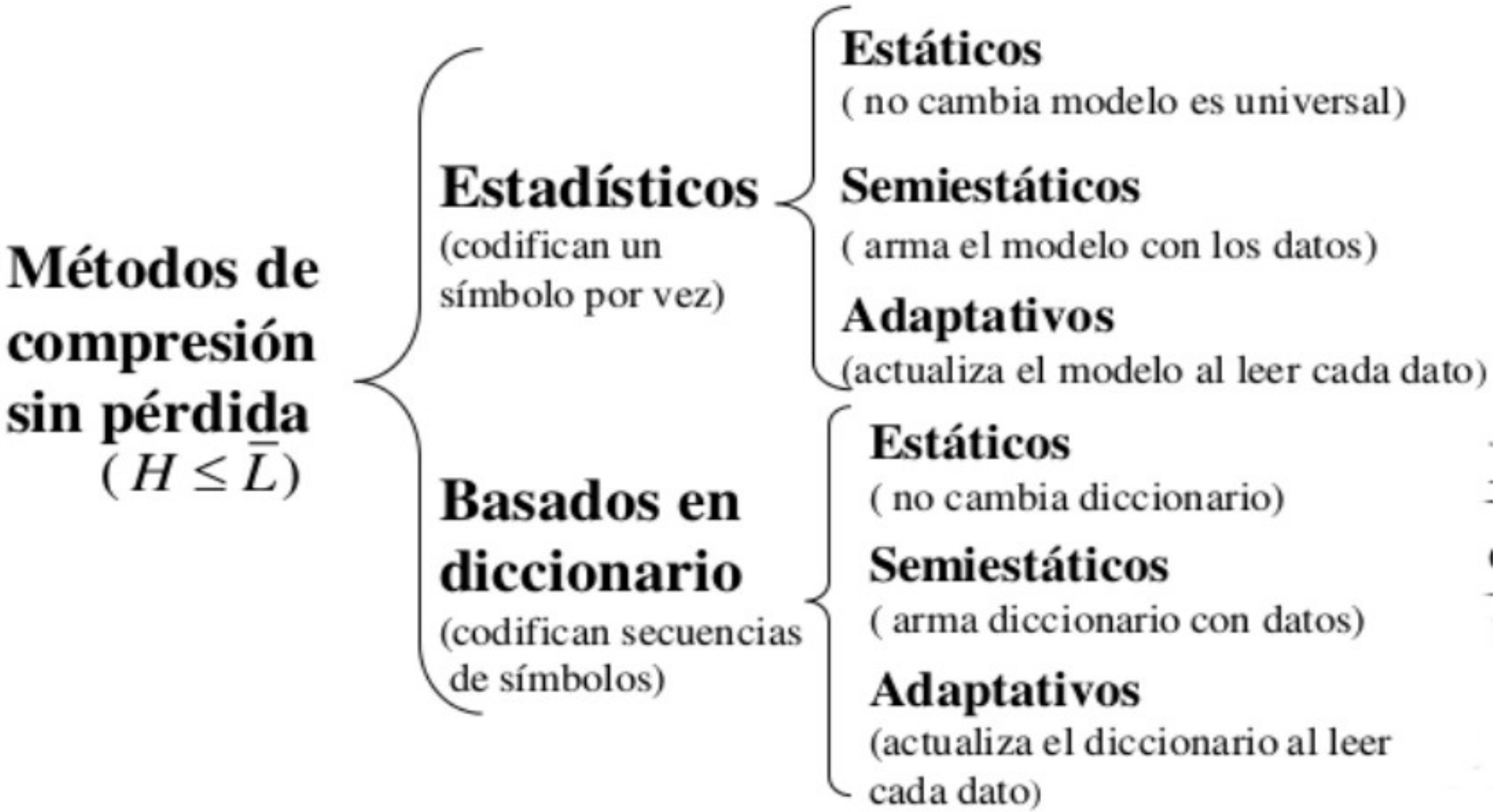
Métodos de Compresión

Sin Pérdida

- Mantienen la integridad de la información (datos descomprimidos iguales a los datos originales).
- texto, base de datos, código fuente, imágenes críticas, etc.
- Tasas de compresión 2:1 (texto) 4:1 imágenes (bajas).
- La long. media de código es mayor o igual a la entropía : $H \leqslant L$

Con Pérdida

- No mantienen la integridad de la información (datos descomprimidos son aproximados al original).
- Imágenes, video, sonido (jpeg, mpeg, mp3, entre otros).
- Tasas de compresión 30:1 (imágenes) 200:1(video) (altas).
- El proceso es no reversible
- No tienen como límite la entropía: $L<H$


Métodos de compresión sin pérdida
$(H \leq \bar{L})$
Estadísticos
(codifican un símbolo por vez)
Estáticos
(no cambia modelo es universal)
Semiestáticos
(arma el modelo con los datos)
Adaptativos
(actualiza el modelo al leer cada dato)
Basados en diccionario
(codifican secuencias de símbolos)
Estáticos
(no cambia diccionario)
Semiestáticos
(arma diccionario con datos)
Adaptativos
(actualiza el diccionario al leer cada dato)

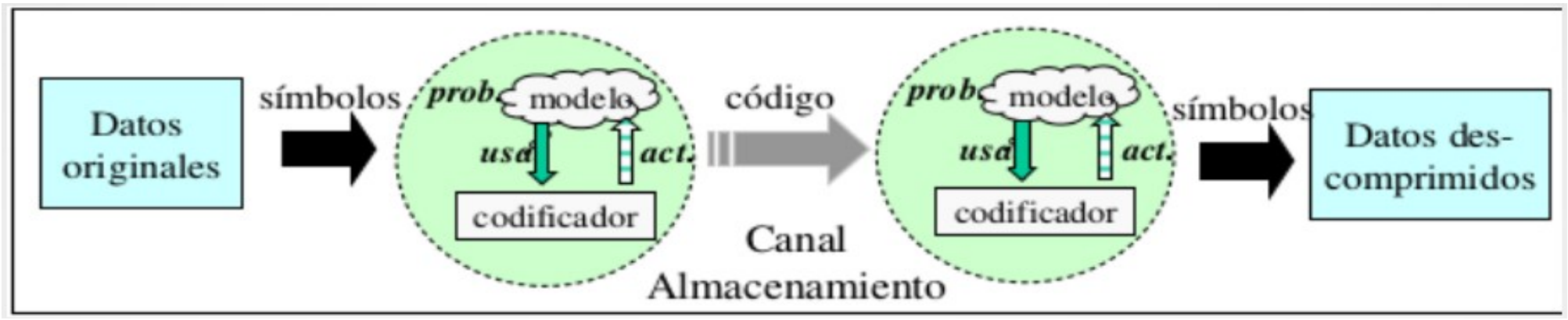
Unidad IV: Codificación y Compresión

Semi estático;
(-) Requiere dos pasadas por los datos para armar la tabla, y la otra codificar
(+) La distribución de probabilidades se ajusta a los datos
(-) Se debe transmitir/almacenar la distribución de probabilidades al decodificador

Estático:
(+) Requiere una sola pasada por los datos para codificar
(-) La distribución de probabilidades es fija puede diferir de los datos a codificar
(+) No se debe transmitir/almacenar la distribución de probabilidades al decodificador

Adaptativo o Dinámico:
(+) Requiere una pasada por los datos para codificar
(+) La distribución de probabilidades se va ajustando a los datos al procesar
(+) No se debe transmitir/almacenar la distribución de probabilidades al decodificador
(-) Algoritmos más complejos, de mayor costo computacional

Unidad IV: Codificación y Compresión



Los modelos estáticos y semi estáticos usan una única tabla de probabilidades y son del tipo Huffman. Serán estáticos si la distribución de probabilidades es impuesta desde el exterior y semi estática si dependen de los datos a comprimir.

Los modelos Adaptativos Dinámicos usan y actualizan la codificación durante el proceso (Huffman dinámico)

El codificador y el decodificador deben basarse en el mismo modelo

Métodos de Compresión

Sin pérdida H≦ L

- Estadísticos
- Shannon-Fano
- Huffman estático
- Huffman semi-estático
- Huffman dinámico
- Aritmético
- Otros

Con pérdida L<H

- Jpeg
- Mpeg
- MP3
- Run Leght
- Compresión fractal
- Cuantificación vectorial
- Otros

- Tasa de compresión N:1 donde N=tamaño_original/ tamaño_comprimido
- Tiempo de compresión/descompresión
- Calidad: Error entre datos originales y datos descomprimidos (Ej. Error cuadrático medio)

Depende del interés del usuario qué considerar. Si se mejora algún parámetro seguramente será en perjuicio de otro (Aumenta la compresión pero baja la calidad).

Si se considera Huffman:

- La longitud media L m del código Huffman está imitado: H(S) ≤ Ln /n< H(S)+1/n
- Huffman no alcanza generalmente el límite H(S) porque no se permiten´bits fraccionales´, necesita al menos un bit por símbolo.

Por ejemplo:

Dado un alfabeto fuente (s1, s2) con probabilidades p1 = 0.99 y p2 = 0.01. Y una codificación mediante un alfabeto código binario (0,1). Tenemos un H(S)= 0.0808 y la longitudes optimas serían: l_1: -log2(0.99) = 0.0145 y l_2: -log(0.01) = 6.644. Este ejemplo es solo teórico y no tiene ningún fin práctico. Resulta obvio que en Huffman asignara 1 bit a cada símbolo entonces la redundancia será = 1.84, cosa que es absurda. **La performance del código mejora a expensas de un aumento exponencial del tamaño de la tabla de códigos.**

Los métodos de compresión sin perdida se aprovechan de la existencia de una cierta distribución de probabilidad aplicable a el alfabeto fuente. Esto se da en general en los alfabetos fuentes destinados a representar textos. Sin embargo existen muchos tipos de archivos que no responden a esa singularidad, por ejemplo las imágenes, video o sonidos. Todas ellas pueden tener alguna distribución asociada a una instancia específica, pero no en general consideremos el código Run Length Coding (RLC).

Este código agrupa los símbolos consecutivos colocando la cantidad de repeticiones seguida del símbolos. Un códigos es este tipo puede contar con L<H, si estamos comprimiendo una imagen, pero salvo casos especiales con texto sería siempre L>H:

Run Lenght Coding (RLC) :

Se codifican secuencias de símbolos iguales con el par (símbolo, repeticiones), por ejemplo:

A la secuencia AAAAAAAAABBBAAAAAAAA le corresponde A 9 B 3 A 8

Suponiendo Imágenes B/N donde ‘0’ es blanco y ‘1’ es negro, tendremos una secuencia de ‘0’ y ‘1’ alternados. En este caso asumiendo que el primero en la imagen es 0 se puede codificar solo con la cantidad de símbolos. Este es un caso muy especial, dependerá del largo de la longitudes su performance.

Los métodos de compresión con pérdida buscan comprimir pensando en la función del archivo a comprimir. Por ejemplo si pretendemos comprimir imágenes o videos, tratan de reducir la imágenes a zonas de color similar tomando una tolerancia y juegan con la capacidad de diferenciación del ojo humano. Cuando se trata de sonido se refieren a la sensibilidad del oído humano.

Cuando se transmite un video en televisión se juega con la capacidad del ojo humano para percibir movimiento, eso establece la cantidad de imágenes fijas a transmitir por segundo. Por otro lado está la calidad de la imagen, esta necesariamente ha de transmitirse comprimida y el factor de compresión dependerá de la calidad que se desea transmitir. La compresión aprovechará las zonas de color similar aprovechando la sensibilidad del ojo humano.

Por ejemplo:

En un partido de futbol la cámara detecta infinidad de tonalidades de verde en el campo de juego, sin embargo, al comprimir solo se transmiten unos pocos tonos.