**Teoría de Información**

Carrera: Licenciatura en Ciencias de la Computación

Año: 2025

**Curiosidades de la información que pueden extraer las personas.......**

****

**ÁREA TEMÁTICA No1**

**PRÁCTICO 1 TEORÍA DE LA INFORMACIÓN**

**CANTIDAD DE INFORMACIÓN - ENTROPÍA**

1. Supón una imagen de resolución 2560 x 1440 píxeles, codificada en 30 bits por píxel. Calcula la cantidad de información que proporciona un cuadro de imagen. Luego, considera una imagen de 7680 x 4320 píxeles (resolución 8K) codificada en HDR con 12 bits por canal de color, ¿cuánta información se genera?

### Análisis desde la Teoría de la Información

## Considerando una imagen de 2560 × 1440 píxeles

**Datos:**

* Resolución: 2560 × 1440 = 3,686,400 píxeles
* Profundidad: 30 bits por píxel
* Alfabeto de símbolos: 2³⁰ = 1,073,741,824 colores posibles

### Cálculo de Información Teórica Máxima

**Entropía máxima por píxel:**

H\_max = log₂(2³⁰) = 30 bits por píxel

**Información total (caso equiprobable):**

I\_total = Número de píxeles × H\_max

I\_total = 3,686,400 × 30 = 110,592,000 bits

**Conversiones:**

* En bytes: 110,592,000 ÷ 8 = **13,824,000 bytes**
* En MB: 13,824,000 ÷ 1,024² ≈ **13.18 MB**

### Interpretación con Entropía

Este cálculo asume **entropía máxima**, es decir, que cada color tiene la misma probabilidad de aparecer (distribución uniforme). En la realidad:

* **Imágenes naturales** tienen entropía menor debido a correlaciones espaciales
* **Píxeles vecinos** tienden a ser similares (reduciendo incertidumbre)
* **Redundancia espacial** permite compresión efectiva

## Considerando una imagen de 7680 x 4320 píxeles

### Datos del Problema

* Resolución: 7680 × 4320 = 33,177,600 píxeles
* HDR con 12 bits por canal RGB
* Alfabeto por canal: 2¹² = 4,096 niveles
* Alfabeto total por píxel: (2¹²)³ = 2³⁶ colores posibles

### Cálculo de Información Teórica

**Entropía máxima por píxel:**

H\_max = log₂(2³⁶) = 36 bits por píxel

**Información total (RGB):**

I\_total = 33,177,600 × 36 = 1,194,393,600 bits

**Conversiones:**

* En bytes: 1,194,393,600 ÷ 8 = **149,299,200 bytes**
* En MB: 149,299,200 ÷ 1,024² ≈ **142.38 MB**

2. Un narrador utiliza 1000 palabras tomadas al azar de un vocabulario de 20.000 palabras. Calcula la información generada. Luego, compárala con la información contenida en una única imagen de resolución 640x480 píxeles codificada en 24 bits por píxel (color verdadero). Analiza cuál contiene más información y verifica si se cumple el dicho popular 'una imagen vale más que mil palabras' incluso con baja resolución..

## Considerando la información generada del narrador

### Datos del Problema

* **Número de palabras**: 1000 palabras
* **Vocabulario disponible**: 20,000 palabras
* **Selección**: Al azar (distribución uniforme)

### Cálculo de Información usando Entropía

**Entropía por palabra (distribución uniforme):**

* H\_palabra = log₂(20,000) = log₂(2 × 10⁴)
* H\_palabra = log₂(2) + log₂(10⁴) = 1 + 4×log₂(10)
* H\_palabra = 1 + 4×3.322 = 1 + 13.288 = 14.288 bits/palabra

**Cálculo más preciso:**

H\_palabra = log₂(20,000) ≈ 14.288 bits por palabra

**Información total en 1000 palabras:**

I\_texto = 1000 × 14.288 = 14,288 bits

**Conversiones:**

* En bytes: 14,288 ÷ 8 = **1,786 bytes**
* En KB: 1,786 ÷ 1024 ≈ **1.74 KB**

### Interpretación desde Entropía

Este cálculo asume **entropía máxima** (cada palabra equiprobable). En textos reales:

* **Entropía real** del español ≈ 1.5-2 bits/carácter ≈ 7-10 bits/palabra

## Considerando una imagen 640×480

### Datos del Problema

* **Resolución**: 640 × 480 = 307,200 píxeles
* **Profundidad**: 24 bits por píxel (color verdadero)
* **Canales**: RGB (8 bits cada uno: 256³ = 16,777,216 colores)

### Cálculo de Información Teórica

**Entropía máxima por píxel:**

H\_pixel = log₂(2²⁴) = 24 bits por píxel

**Información total:**

I\_imagen = 307,200 × 24 = 7,372,800 bits

**Conversiones:**

* En bytes: 7,372,800 ÷ 8 = **921,600 bytes**
* En KB: 921,600 ÷ 1024 = **900 KB**
* En MB: 900 ÷ 1024 ≈ **0.88 MB**

### Entropía Real en Imágenes

En imágenes naturales típicas:

* **Entropía efectiva**: 4-8 bits/píxel (después de compresión)
* **Redundancia**: permite compresión JPEG 10:1 o superior

## Comparación y Análisis

### Tabla Comparativa de Información

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Contenido** | **Elementos** | **Bits/elemento** | **Información Teórica** | **Información Práctica** |
| **1000 palabras** | 1000 | 14.288 | 14,288 bits (1.74 KB) | ~7,000-10,000 bits |
| **Imagen 640×480** | 307,200 | 24 | 7,372,800 bits (900 KB) | ~1-3 MB comprimida |

### Factor de Diferencia

**Relación de información teórica:**

Ratio = 7,372,800 / 14,288 ≈ 516:1

**La imagen contiene ~516 veces más información que el texto**

## Conclusión

Una imagen contiene mucha más información en términos de datos que mil palabras, así que el dicho “una imagen vale más que mil palabras” es cierto desde lo cuantitativo. Igual, las palabras sirven para expresar ideas y significados más complejos que una imagen sola no logra.

3. Calcula la información generada por un mensaje de 200 caracteres tomados al azar de un alfabeto de 32 símbolos. ¿Y por un mensaje de 200 caracteres con un alfabeto de 64 símbolos? Ahora repite el cálculo para un mensaje de 400 caracteres con alfabetos de 32 y 64 símbolos. Analiza cómo cambia la cantidad de información y verifica si duplicar el tamaño del alfabeto duplica la información total..

## Caso 1: Mensaje de 200 caracteres

### Alfabeto de 32 símbolos

**Entropía por carácter:**

H₃₂ = log₂(32) = log₂(2⁵) = 5 bits/carácter

**Información total:**

I₁ = 200 × 5 = 1,000 bits

**Conversiones:**

* En bytes: 1,000 ÷ 8 = **125 bytes**
* En KB: 125 ÷ 1,024 ≈ **0.122 KB**

### Alfabeto de 64 símbolos

**Entropía por carácter:**

H₆₄ = log₂(64) = log₂(2⁶) = 6 bits/carácter

**Información total:**

I₂ = 200 × 6 = 1,200 bits

**Conversiones:**

* En bytes: 1,200 ÷ 8 = **150 bytes**
* En KB: 150 ÷ 1,024 ≈ **0.146 KB**

### Análisis del incremento (200 caracteres):

Ratio = I₂/I₁ = 1,200/1,000 = 1.2

**El alfabeto de 64 símbolos genera 1.2× más información (20% adicional)**

## Caso 2: Mensaje de 400 caracteres

### Alfabeto de 32 símbolos

**Información total:**

I₃ = 400 × 5 = 2,000 bits

**Conversiones:**

* En bytes: 2,000 ÷ 8 = **250 bytes**
* En KB: 250 ÷ 1,024 ≈ **0.244 KB**

### Alfabeto de 64 símbolos

**Información total:**

I₄ = 400 × 6 = 2,400 bits

**Conversiones:**

* En bytes: 2,400 ÷ 8 = **300 bytes**
* En KB: 300 ÷ 1,024 ≈ **0.293 KB**

### Análisis del incremento (400 caracteres):

Ratio = I₄/I₃ = 2,400/2,000 = 1.2

**Nuevamente, el alfabeto de 64 símbolos genera 1.2× más información**

## Tabla Comparativa Completa

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Configuración** | **Caracteres** | **Alfabeto** | **H (bits/char)** | **Información Total** | **Tamaño** |
| **Caso 1A** | 200 | 32 símbolos | 5 | 1,000 bits | 125 bytes |
| **Caso 1B** | 200 | 64 símbolos | 6 | 1,200 bits | 150 bytes |
| **Caso 2A** | 400 | 32 símbolos | 5 | 2,000 bits | 250 bytes |
| **Caso 2B** | 400 | 64 símbolos | 6 | 2,400 bits | 300 bytes |

## 

## Análisis de Patrones

### 1. Efecto de Duplicar el Tamaño del Alfabeto

**Relación matemática:**

H₆₄/H₃₂ = log₂(64)/log₂(32) = 6/5 = 1.2

**Conclusión:** Duplicar el alfabeto (32→64) **NO duplica la información**, sino que la **incrementa en 20%**.

### 2. Efecto de Duplicar la Longitud del Mensaje

**200 → 400 caracteres (alfabeto 32):**

I₃/I₁ = 2,000/1,000 = 2.0

**200 → 400 caracteres (alfabeto 64):**

I₄/I₂ = 2,400/1,200 = 2.0

Duplicar la longitud del mensaje **SÍ duplica.**

### Conclusión:

### "Duplicar el alfabeto duplica la información"

### Incremento real al duplicar alfabeto (32→64):

Incremento = (log₂(64) - log₂(32))/log₂(32) = (6-5)/5 = 20%

Para duplicar realmente la información con cambio de alfabeto:

2 × log₂(32) = 10 bits/carácter

|A|\_necesario = 2¹⁰ = 1,024 símbolos

Se necesitarían **1,024 símbolos** (no 64) para duplicar la información.

4. Demostrar las siguientes igualdades:



Por propiedad de logaritmo:

Sea , entonces:

Aplicando a ambos lados:

Luego:

Como , entonces:

Quedando demostrada la ecuación **.** De la expresión anterior:

Pero:

Entonces:

Quedando demostrada la ecuación

Por lo tanto, hemos demostrado:

5. Una fuente F tiene 6 símbolos con probabilidades: p1=0.4, p2=0.2, p3=0.15, p4=0.1, p5=0.1, p6=0.05. Calcula la información individual y la entropía de la fuente.

## Información individual de cada símbolo

Fórmula: I(xᵢ) = -log₂(pᵢ)

Para p₁ = 0.4: I₁ = -log₂(0.4) ≈ 1.322 bits

Para p₂ = 0.2: I₂ = -log₂(0.2) ≈ 2.322 bits

Para p₃ = 0.15: I₃ = -log₂(0.15) ≈ 2.737 bits

Para p₄ = 0.1: I₄ = -log₂(0.1) ≈ 3.322 bits

Para p₅ = 0.1: I₅ = -log₂(0.1) ≈ 3.322 bits

Para p₆ = 0.05: I₆ = -log₂(0.05) ≈ 4.322 bits

## Entropía de la fuente

Fórmula: H(F) = Σᵢ₌₁⁶ pᵢ · I(xᵢ)

Cálculo término a término: 0.4 × 1.322 ≈ 0.529 0.2 × 2.322 ≈ 0.464 0.15 × 2.737 ≈ 0.411 0.1 × 3.322 ≈ 0.332 0.1 × 3.322 ≈ 0.332 0.05 × 4.322 ≈ 0.216

**Suma total:** H(F) ≈ 0.529 + 0.464 + 0.411 + 0.332 + 0.332 + 0.216 = 2.284 bits

## Resultado final:

Información individual: I₁ ≈ 1.322, I₂ ≈ 2.322, I₃ ≈ 2.737, I₄ ≈ 3.322, I₅ ≈ 3.322, I₆ ≈ 4.322 bits Entropía de la fuente: H(F) ≈ 2.284 bits

6. Justifica por qué una distribución uniforme maximiza la entropía. Compara un dado justo (6 caras, p=1/6) con un dado sesgado: p1=0.3, p2=0.25, p3=0.15, p4=0.15, p5=0.1, p6=0.05..

Fórmula: H(X) = -Σᵢ₌₁ⁿ pᵢ log₂(pᵢ)

Para el dado justo, cada cara tiene probabilidad p = 1/6:

H\_justo = -Σᵢ₌₁⁶ (1/6) log₂(1/6)

H\_justo = -6 × (1/6) log₂(1/6)

H\_justo = -log₂(1/6)

H\_justo = log₂(6)

H\_justo ≈ 2.585 bits

## Dado sesgado

Probabilidades: p₁=0.3, p₂=0.25, p₃=0.15, p₄=0.15, p₅=0.1, p₆=0.05

H\_sesgado = -(0.3×log₂(0.3) + 0.25×log₂(0.25) + 0.15×log₂(0.15) +

0.15×log₂(0.15) + 0.1×log₂(0.1) + 0.05×log₂(0.05))

Cálculos término a término:

-0.3 × log₂(0.3) ≈ 0.521

-0.25 × log₂(0.25) = 0.500

-0.15 × log₂(0.15) ≈ 0.411

-0.15 × log₂(0.15) ≈ 0.411

-0.1 × log₂(0.1) ≈ 0.332

-0.05 × log₂(0.05) ≈ 0.216

Suma total: H\_sesgado ≈ 0.521 + 0.500 + 0.411 + 0.411 + 0.332 + 0.216 = 2.391 bits

## Comparación de resultados:

* **Dado justo**: H ≈ 2.585 bits (máxima entropía)
* **Dado sesgado**: H ≈ 2.391 bits (menor entropía)

**Diferencia**: 2.585 - 2.391 = 0.194 bits

## Conclusión:

El dado justo tiene mayor entropía porque cada resultado es equiprobable, maximizando la incertidumbre. El dado sesgado tiene menor entropía porque algunos resultados son más probables que otros, reduciendo la imprevisibilidad total.

7. Sean 12 monedas una de las cuales tiene peso diferente, indicar cuántas pesadas son necesarias para encontrarla, especifique la metodología de peso

**Fórmula de posibilidades totales:**

Posibilidades = Número de monedas × Estados posibles

Posibilidades = 12 × 2 = 24

(12 monedas pueden ser pesadas o livianas)

**Fórmula de información por pesada:**

Resultados por pesada = 3 (izquierda > derecha, equilibrio, izquierda < derecha)

Información máxima con n pesadas = 3ⁿ combinaciones

**Condición de suficiencia:**

3ⁿ ≥ 24 posibilidades

3ⁿ ≥ 24

n ≥ log₃(24) ≈ 2.89

Por tanto: n = 3 pesadas mínimas

**Verificación de necesidad:**

3² = 9 < 24 → 2 pesadas insuficientes

3³ = 27 > 24 → 3 pesadas suficientes

## Metodología de Resolución: Algoritmo de División Ternaria

### Estrategia General

**Principio:** División en grupos de igual tamaño para maximizar la información obtenida por pesada.

**Regla de división:** 12 monedas → 3 grupos de 4 monedas cada uno.

## PRIMERA PESADA: Identificación del Grupo Sospechoso

### Configuración

Grupo A = {monedas 1, 2, 3, 4}

Grupo B = {monedas 5, 6, 7, 8}

Grupo C = {monedas 9, 10, 11, 12}

### Pesada

Balanza: A vs B (Grupo C queda fuera)

### Análisis de Resultados

* **CASO 1: A = B (equilibrio)**

Conclusión: La moneda diferente está en C

Monedas normales confirmadas: A y B (8 monedas)

Candidatos restantes: 4 monedas en C

* **CASO 2: A > B (A pesa más)**

Conclusión: La moneda diferente está en A o B

- Si está en A → es más pesada

- Si está en B → es más liviana

Monedas normales confirmadas: C (4 monedas)

Candidatos restantes: 8 monedas en A∪B

* **CASO 3: A < B (B pesa más)**

Conclusión: La moneda diferente está en A o B

- Si está en A → es más liviana

- Si está en B → es más pesada

Monedas normales confirmadas: C (4 monedas)

Candidatos restantes: 8 monedas en A∪B

## SEGUNDA PESADA: Reducción del Espacio de Búsqueda

### Camino 1: Para CASO 1 (A = B)

**Configuración:**

Pesada: (9, 10, 11) vs (1, 2, 3)

Monedas fuera: {12}

**Metodología:** Usar 3 monedas sospechosas vs 3 monedas normales conocidas.

**Análisis:**

* **Si (9,10,11) = (1,2,3)** → La moneda 12 es la diferente
* **Si (9,10,11) > (1,2,3)** → Una de {9,10,11} es más pesada
* **Si (9,10,11) < (1,2,3)** → Una de {9,10,11} es más liviana

### Camino 2: Para CASO 2 (A > B)

**Configuración:**

Pesada: (1, 2, 5) vs (3, 4, 6)

Monedas fuera: {7, 8}

**Metodología:** Mezclar monedas de grupos sospechosos para crear balance de información.

**Lógica:**

* Lado izquierdo: 2 de A + 1 de B
* Lado derecho: 2 de A + 1 de B
* Si hay desequilibrio, podemos determinar el origen

### Camino 3: Para CASO 3 (A < B)

**Configuración:**

Pesada: (1, 2, 6) vs (3, 4, 5)

Monedas fuera: {7, 8}

**Metodología:** Similar al Camino 2, pero ajustado para el caso A < B.

## TERCERA PESADA: Identificación Final

### Para cada sub-caso de la segunda pesada

**Cuando quedan 2 candidatos:**

Pesada: Candidato₁ vs Candidato₂

- Si Candidato₁ > Candidato₂ y esperamos pesada → Candidato₁ es la respuesta

- Si Candidato₁ < Candidato₂ y esperamos liviana → Candidato₁ es la respuesta

- Y viceversa

**Cuando queda 1 candidato claro:**

Pesada: Candidato vs Moneda\_normal

Confirmación del tipo (pesada o liviana)

8. Calcula la información de una letra al azar de un alfabeto de 28 símbolos (incluyendo ñ). Luego, de pares y ternas de letras.

## Fórmula general:

I = log₂(1/p) = log₂(n), donde n es el número de símbolos posibles.

### Una letra:

Cada letra tiene probabilidad p = 1/28

I = log₂(28) ≈ 4.807 bits

### Un par de letras:

La cantidad de pares posibles es 28² = 784

I = log₂(28²) = 2 log₂(28) ≈ 9.614 bits

### Una terna de letras:

La cantidad de ternas posibles es 28³ = 21,952

I = log₂(28³) = 3 log₂(28) ≈ 14.421 bits

9. ¿Cómo serán las cifras anteriores respecto de la forma castellana de escritura cotidiana? Si a su criterio existiera diferencia ¿a qué se debería?

En la escritura castellana cotidiana estas cifras son menores porque las letras no tienen la misma probabilidad de aparición y, además, existen reglas ortográficas y gramaticales que generan dependencia entre los símbolos.

10. Explica las propiedades de la cantidad de información. Da un ejemplo para cada una de estas propiedades.

## 1. La cantidad de información es siempre positiva (mayor o igual que cero)

**Fórmula:** I(x) = -log₂(p(x)) ≥ 0

**Ejemplo:** si tiro un dado justo, la probabilidad de que salga un "3" es p = 1/6.

I = -log₂(1/6) = 2,58 bits

## 2. La cantidad de información de un evento es inversamente proporcional a la probabilidad de ocurrencia

Es decir, mientras más raro sea el evento, más información aporta.

**Ejemplo:** al lanzar dos dados:

* La probabilidad de que salga un "7" es p = 6/36 = 0,167. I = -log₂(0,167) = 2,58 bits
* La probabilidad de que salga un "2" es p = 1/36 ≈ 0,0278. I = -log₂(0,0278) = 5,17 bits

El resultado "2" entrega más información porque es más raro.

## 3. La cantidad de información aumenta con la cantidad de posibles eventos o mensajes

**Ejemplo:**

* Un dado (6 resultados): I = log₂(6) = 2,58 bits
* Un mazo de naipes españoles (40 cartas): I = log₂(40) = 5,32 bits
* Una baraja inglesa (54 cartas): I = log₂(54) = 5,75 bits

## 4. La cantidad de información de eventos independientes es aditiva

**Fórmula:** I(A,B,…,C) = I(A) + I(B) + ⋯ + I(C)

**Ejemplo:** si lanzo una moneda (1 bit) y un dado (2,58 bits) de manera independiente:

I\_total = 1 + 2,58 = 3,58 bits

## 5. La cantidad de información promedio (entropía) se maximiza en sucesos equiprobables

**Ejemplo:**

* Un dado justo de 4 caras (p = 1/4 cada una): H = 4 · (1/4) · log₂(4) = log₂(4) = 2 bits
* El mismo dado con probabilidades desiguales (p₁ = 0,5; p₂ = 0,25; p₃ = 0,15; p₄ = 0,10): H = 0,5log₂(1/0,5) + 0,25log₂(1/0,25) + 0,15log₂(1/0,15) + 0,10log₂(1/0,10) ≈ 1,74 bits

En este caso, la entropía es menor porque los sucesos no son equiprobables.

11. Dada una variedad V = 2000 sucesos encontrar la base *b* óptima de representación que hace mínima la siguiente relación número más corto, es decir que minimice I . b (cantidad de información por base).

# Derivación General

L\_b = b · (ln V)/(ln b) con V = 2000

Para minimizar, basta estudiar la función

g(b) = b/(ln b), b > 1.

La derivada:

g'(b) = (ln b - 1)/(ln b)².

Se anula en ln b = 1 ⟹ b = e ≈ 2,718.

Cálculo en bases concretas

* **Base 2:** I₂ = log₂ 2000 = (ln 2000)/(ln 2) ≈ 10,9658 bits L₂ = 2 · 10,9658 = 21,932
* **Base 3:** I₃ = log₃ 2000 = (ln 2000)/(ln 3) ≈ 6,9180 trits L₃ = 3 · 6,9180 = 20,756
* **Base 4:** I₄ = log₄ 2000 = (ln 2000)/(ln 4) ≈ 5,4829 L₄ = 4 · 5,4829 = 21,932

## Conclusión

* **Óptimo continuo:** la base que minimiza L\_b es b = e ≈ 2,718.
* **Óptimo entero:** como solo se pueden usar bases enteras, la mejor elección es **b = 3**, ya que da el menor valor: L₃ ≈ 20,756 < L₂ = L₄ ≈ 21,932

12. Para el texto contenido en el código QR de la parte superior de la página encuentre la entropía de orden cero o independiente tanto de lo relevado en el QR como en el texto correcto. Que conclusiones saca?

### Análisis del texto con errores ortográficos (extraído del QR):

**Texto analizado:** "Sgeun un etsduio de una uivenrsdiad ignlsea, no ipmotra el odren en el que las ltears etsan ersciats, la uicna csoa ipormtnate es que la pmrirea y la utlima ltera esten ecsritas en la psiocion cocrrtea. El rsteo peuden estar ttaolmntee mal y aun pordas lerelo sin pobrleams. Etso es pquore no lemeos cada ltera por si msima preo la paalbra es un tdoo. Pesornamelnte me preace icrneilbe..."

### Tabla de frecuencias y probabilidades:

|  |  |  |
| --- | --- | --- |
| **Símbolo** | **Ocurrencia** | **Probabilidad** |
| ' ' (espacio) | 67 | 0.1727 |
| , | 2 | 0.0052 |
| . | 6 | 0.0155 |
| a | 36 | 0.0928 |
| b | 3 | 0.0077 |
| c | 10 | 0.0258 |
| d | 9 | 0.0232 |
| e | 49 | 0.1263 |
| g | 2 | 0.0052 |
| i | 18 | 0.0464 |
| l | 23 | 0.0593 |
| m | 12 | 0.0309 |
| n | 23 | 0.0593 |
| o | 23 | 0.0593 |
| p | 13 | 0.0335 |
| q | 3 | 0.0077 |
| r | 25 | 0.0644 |
| s | 27 | 0.0696 |
| t | 21 | 0.0541 |
| u | 13 | 0.0335 |
| v | 1 | 0.0026 |
| y | 2 | 0.0052 |
| **Total** | **388** | **1.0000** |

### 

### Cálculo de la entropía:

Aplicando la fórmula de Shannon para entropía de orden cero:

**H₀ = -Σ pᵢ · log₂(pᵢ)**

Resultado: **H₀(texto incorrecto) = 3.90 bits**

### Análisis del texto corregido:

**Texto corregido:** "Según un estudio de una universidad inglesa, no importa el orden en el que las letras estén escritas, la única cosa importante es que la primera y la última letra estén en la posición correcta. El resto pueden estar totalmente mal y aun podrás leerlo sin problemas. Esto es porque no leemos cada letra por sí misma, sino que la palabra es un todo. Personalmente me parece increíble..."

### Conclusión:

Dado que ambos textos contienen exactamente la misma cantidad y variedad de símbolos (solo cambia el orden de las letras dentro de las palabras, pero no se agregan ni eliminan caracteres), y considerando que estamos calculando una entropía de orden cero (independiente), obtenemos:

**H₀(texto correcto) = 3.90 bits = H₀(texto incorrecto)**

**PRÁCTICO 2 CANAL DE INFORMACIÓN**

1. Sea el siguiente canal:

|  |  |  |  |
| --- | --- | --- | --- |
|  | b1 | b2 | b3 |
| a1 | 0,6 | 0,3 | 0,1 |
| a2 | 0,2 | 0,5 | 0,3 |
| a3 | 0,4 | 0,2 | 0,4 |

Calcular los valores de p(ai/bj) y las probabilidades de salida para el caso particular de p(a1)=0.5, p(a2)=0.25, p(a3)=0.25.

**Para resolver el ejercicio empezamos calculando las Probabilidades de salida p()**

Comprobación:

**Luego seguimos con las Probabilidades conjuntas p(ai|bj)**

Usando el teorema de Bayes:  
Para

Para :

Para :

2. Considera un Canal Binario Asimétrico con las siguientes probabilidades:

● p(a1)=0,4

● p(b1/a1)=4/5

● p(b1/a2)=1/4

a) Calcula las probabilidades condicionales hacia atrás y las probabilidades conjuntas.

b) Obtén la entropía del emisor, y las entropías condicionales de la fuente para

cualquier símbolo de salida.

c) Calcula la información mutua y la capacidad del canal.

**a)** De los datos podemos deducir que ya que por ser un canal binario. Lo mismo ocurre para las probabilidades condicionales. Es decir:

Luego calculamos las probabilidades conjuntas donde:

Como siguiente paso obtenemos las probabilidades de salida:

Por último calculamos las probabilidades conjuntas **:**

:

:

**b)**

Propiedad de Logaritmos a tener en cuenta:

Primero calculamos la Entropía de entrada

Luego las entropías condicionales de la fuente para cualquier símbolo de salida:

Para resolverlo de manera mas ordenada vamos proseguir por partes:

Ahora si obtenemos:

**c)**

Calculamos la Información mutua:

Luego para calcular la capacidad del canal debemos maximizar Para ello maximizamos sobre la distribución de entrada y donde numéricamente se obtiene:

\* , \*

Finalmente, el máximo de capacidad obtenido usando las entradas óptimas es:

3. Considera un canal determinista con cuatro símbolos de entrada (a₁ ,a₂ ,a₃ ,a₄ ) y cuatro símbolos de salida (b₁ ,b₂ ,b₃ ,b₄ ), donde cada símbolo de entrada se corresponde exclusivamente con un símbolo de salida. Define las probabilidades para cada símbolo de entrada y calcula la información mutua.

Las probabilidades de entrada las vamos a definir como:

Luego la Información mutua va ser calculada como:

Donde ya que al ser un canal determinista tenemos:

Entonces el cálculo de la Información mutua nos queda:

4. Dada la siguiente matriz de canal, obtén las respectivas codificaciones de la fuente.

p(a)=1/3, p(b)=1/6, p(c)=1/2

|  |  |  |  |
| --- | --- | --- | --- |
|  | a | b | c |
| a | 2/5 | 2/5 | 1/5 |
| b | 2/5 | 1/5 | 2/5 |
| c | 1/4 | 1/2 | 1/4 |

Primero calculamos el Código de Huffman para la fuente con las probabilidades de entrada:

Probabilidades Ordenadas:

Si combinamos

Ahora tenemos dos nodos (c y ab) que van a recibir valor 0 y 1 respectivamente.

Dentro del nodo ab asignamos .

Por lo que el código de Huffman nos queda como:

De longitudes: 1,2,2.

Media:

Luego calculamos el Código de Huffman para la fuente utilizando el canal:

**Empezamos calculando las Probabilidades de salida p(B)**

**Luego seguimos con las Probabilidades conjuntas p(ai|bj)**

Usando el teorema de Bayes:

Para :

Seguimos con la Codificación de las probabilidades para este caso:

Probabilidades Ordenadas:

Si combinamos

Ahora tenemos dos nodos (a y bc) que van a recibir valor 0 y 1 respectivamente.

Dentro del nodo bc asignamos .

Por lo que el código de Huffman nos queda como:

De longitudes: 1,2,2.

Media:

Para :

Seguimos con la codificación de las probabilidades para este caso:

Probabilidades Ordenadas:

Si combinamos

Ahora tenemos dos nodos (c y ab) que van a recibir valor 0 y 1 respectivamente.

Dentro del nodo ab asignamos .

Por lo que el código de Huffman nos queda como:

De longitudes: 1,2,2.

Media:

Por último para :

Seguimos con la codificación de las probabilidades para este caso:

Probabilidades Ordenadas:

Si combinamos

Ahora tenemos dos nodos (c y ab) que van a recibir valor 0 y 1 respectivamente.

Dentro del nodo ab asignamos .

Por lo que el código de Huffman nos queda como:

De longitudes: 1,2,2.

Media: