EL CANAL DE ELIMINACIÓN: RESULTADOS, ALGORITMOS Y APROXIMACIONES

Diego Ernesto Hernández Jiménez

UNIVERSIDAD NACIONAL DE COLOMBIA
FACULTAD DE CIENCIAS
DEPARTAMENTO DE MATEMÁTICAS
BOGOTÁ, D.C
2013

EL CANAL DE ELIMINACIÓN: RESULTADOS, ALGORITMOS Y APROXIMACIONES

DIEGO ERNESTO HERNÁNDEZ JIMÉNEZ

Trabajo de grado presentado como requisito parcial para optar al título de Magíster en Matemáticas Aplicadas

Director
RICARDO RESTREPO LÓPEZ
GEORGIA INSTITUTE OF TECHNOLOGY
Codirector
EDWARD SAMUEL BECERRA
UNIVERSIDAD NACIONAL DE COLOMBIA

UNIVERSIDAD NACIONAL DE COLOMBIA
FACULTAD DE CIENCIAS
DEPARTAMENTO DE MATEMÁTICAS
BOGOTÁ, D.C
2013

A mi Padre Jose Antonio, Una guıa,

una inspiracion, un amigo...

A mi Madre Esperanza,

simplemente me lo dio todo...

Y a mi esposa Karo,

quien me motivo a terminar este proyecto.

Agradecimientos

El autor agradece a la Universidad Nacional de Colombia, a la Facultad de Ciencias y al departamento de Matematicas por los espacios concedidos para la realizacion del presente proyecto.
Al Profesor Ricardo Restrepo Lopez del departamento de matematicas de la Universidad de Antioquia por su enorme aporte y colaboracion en el desarrollo del presente trabajo en su calidad de director de tesis.
Y al Profesor Edward Samuel Becerra del departamento de matematicas de la Universidad Nacional de Colombia, que ademas de ser un gran amigo, tambien colaboro bastante en la culminacion de este proyecto en su calidad de codirector.

A todos muchas gracias.

Resumen

Las fallas en la transmisión de datos pueden afectar a cualquier persona, desde el que envía un mensaje de texto en un chat, hasta el que guarda datos en una memoria USB. ¿Si se pierden datos, se podrán recuperar? ¿Fue culpa del canal utilizado? El *canal de eliminación* es el canal de comunicación punto a punto más simple que modela la falta de sincronización, (aparición o pérdida de datos). A pesar de importantes esfuerzos, poco se sabe sobre su capacidad, y menos aún sobre los esquemas de codificación y algoritmos óptimos.

Para este trabajo se realizó un estudio sistemático de este problema, se efectuó una revisión de la teoría actual presentando algunos avances recientes y se propuso un esquema de codificación por repetición, basado en las cotas de Chernoff, que incluye las funciones de codificación y decodificación apropiadas y eficientes junto con su implementación en *Matlab,* todo en un *canal artificial* llamado canal de **Ω**-eliminación, así se implementó un algoritmo que simula el canal y se estableció la relación de capacidad entre el canal de eliminación y el canal artificial y como al desarrollar la teoría para este nuevo canal, se pueden conseguir avances significativos en el problema de calcular la capacidad del canal de eliminación, problema aun abierto al día de hoy.

Palabras clave: Información, Entropía, Capacidad, Canal, Transmisión, Sincronización, Codificación.

Abstract

The communication failures can affect anyone, from sending a text message on a chat, to which stores data on a USB stick. If data is lost, they can be recovered? Was it the fault of the channel used? The *binary deletion channel* is the communication channel simpler point to point modeling synchronization errors (appearance or loss of data). Despite significant efforts, little is known about their capacity, and even less about the optimal coding schemes and algorithms.

For this work, a systematic study of this problem was performed, a review of current theory was made by presenting some recent developments and a coding scheme by repetition, based on the Chernoff bounds, including the functions of encoding and decoding appropriate and efficient proposed with their implementation in *Matlab*, all in an artificial channel called Ω-deletion channel and an algorithm that simulates the channel was implemented and the relationship between the channel capacity and the deletion artificial channel was established to develop and as theory for this new channel, can make significant progress on the problem of calculating the deletion channel capacity problem still open today.

Keywords and phrases: Information, Entropy, Capacity Bounds, Communication, Binary Deletion Channel, Coding, Encoding.

Índice general

Convenciones y Notación III

Introducción V

1. Marco Teórico **1**
1.1. Generalidades en teoría de la información 1
1.2. La capacidad de información y ejemplos de canales 12

2. Los Teoremas de Shannon y Características de los Canales. **21**
2.1. Preliminares a los teoremas de Shannon 21
2.2. Los teoremas de Shannon y la codificación 29

3. El Canal de Eliminación: Definiciones **37**
3.1. Ideas principales 37
3.2. El canal de eliminación 41
3.3. El canal de eliminación y sus variantes 46

4. El Canal de Eliminación: Resultados y Propuestas **49**
4.1. Idea central 49
4.2. La propuesta central, el canal de Ω-eliminación 54
4.3. Simulación del canal de Ω-eliminación 66
4.4. Conclusiones 79

Bibliografía **81**

Convenciones y Notación

En este trabajo $\mathbb{Z}$ denota el conjunto de los números enteros, $\mathbb{N}$ el conjunto de los números enteros positivos y $\mathbb{N}_0$ el conjunto de los números enteros no negativos. El conjunto de los números racionales y el conjunto de los números reales se denotan por $\mathbb{Q}$ y $\mathbb{R}$ respectivamente.

La cantidad de elementos de un conjunto A, o cardinal del conjunto A, se denota A, se denota como $[[X]]$ a la parte entera del número real X, y es el menor entero mayor o igual a X.

Las matrices y funciones aquí mencionadas son definidas en el conjunto de los números reales $\mathbb{R}$, a menos que se indique lo contrario. Una matriz es estocástica por filas si la suma de los elementos de cualquiera de sus filas es 1.

Dado un espacio de probabilidad $(\Omega, \mathcal{F}, \mathbb{P})$ y X una variable aleatoria, su función de distribución de probabilidad $F_X(x) = \mathbb{P}(\omega \in \Omega \,|\, X(\omega) \leq x)$ se denota como $P(X \leq x)$ y para varias variables aleatorias X_n, su función de distribución conjunta $F_{X_1,X_2,\ldots,X_n}(x_1, x_2, \ldots, x_n)$ se denota como $P(X_1, X_2, \ldots, X_n)$. En el presente trabajo se suponen todas las variables aleatorias discretas a menos que se indique lo contrario.

Introducción

El canal de eliminación es el canal de comunicación punto a punto más simple que modela la falta de sincronización, (aparición o pérdida de datos). A pesar de importantes esfuerzos, poco se sabe sobre su capacidad, y menos aún sobre los esquemas de codificación y algoritmos óptimos.

Para este trabajo se realizara un estudio sistemático de este problema, así como de los algoritmos involucrados en el cálculo de la capacidad de este canal. Las fallas en la transmisión de datos pueden afectar a cualquier persona, desde el que envía un mensaje de texto en un chat, hasta el que guarda datos en una memoria USB. ¿Si se pierden datos, se podrán recuperar? ¿Fue culpa del canal utilizado? de ahí la importancia de estudiar la capacidad del canal de eliminación y como tratar de superar dichas fallas.

En la teoría de telecomunicaciones e informática, el término canal de comunicación, o simplemente canal, se refiere tanto a un medio físico de transmisión de datos, (como un cable o fibra óptica), como a una conexión lógica a través de un medio multiplexado como un canal de radio. Un canal se utiliza para transmitir una señal de información, por ejemplo, un flujo de bits digitales, de uno o varios transmisores a uno o varios receptores. Un canal tiene una cierta capacidad para la transmisión de información, a menudo se mide por su ancho de banda en Hz o en la velocidad de transmisión en bits por segundo. Por ejemplo, un dispositivo de almacenamiento es también una especie de canal, que puede ser enviado (escrito) y recibido (leído) a cierta tasa de transferencia de datos.

Un canal puede ser modelado físicamente al tratar de calcular los procesos físicos que modifican la señal transmitida. Por ejemplo, en las comunicaciones inalámbricas el canal se puede modelar mediante el cálculo de la reflexión de todos los objetos del entorno. Una secuencia de números al azar también puede ser añadido para simular las interferencias externas o el ruido electrónico en el receptor.

Estadísticamente un canal de comunicación suele ser modelado como una tripla que está formada por: un alfabeto de entrada, un alfabeto de salida, y para cada par de elementos de entrada y salida, se asigna una probabilidad de transición, esta es la probabilidad de que la letra transmitida por el canal sea recibida. Modelos estadísticos y físicos se pueden combinar, por ejemplo, en las comunicaciones inalámbricas el canal se modela a menudo por una atenuación aleatoria (conocido como oscurecimiento) de la señal transmitida, seguido por el ruido. La atenuación es una medida de los procesos físicos subyacentes y captura el cambio en la potencia de la señal a lo largo de la transmisión así como el tiempo que una señal tarda en llegar a través del canal.

Los canales pueden ser digitales o análogos, entre los canales análogos tenemos a las ondas de radio y entre los canales digitales tenemos: el canal binario simétrico, en donde cada bit recibido posee un error de recepción independiente o el canal de borrado binario, en donde cada bit es borrado con una probabilidad p. Shannon en [1] desarrolló los resultados fundamentales sobre la capacidad de estos canales (en la década de 1940), y en los últimos años, a través del desarrollo y el análisis de la capacidad, el rendimiento de dichos canales se ha vuelto extremadamente eficiente.

Consideremos ahora el siguiente canal: Una secuencia de n bits se envía, pero cada bit es eliminado independientemente con una probabilidad fija. Este es el *canal de eliminación binario de forma independiente e idénticamente distribuido, i.i.d* o simplemente, el *canal de eliminación*. Un canal de eliminación no debe ser confundido con un canal de borrado binario, pues en un canal de borrado, cuando cierta cantidad de bits se envían, algunos de estos no son recibidos; en ese caso, un tercer símbolo denotado a menudo por ? se obtiene en el receptor para indicar un borrado.

Por el contrario, con un canal de eliminación, no hay ninguna señal de que los bits se han eliminado. Por ejemplo, si se envía el mensaje 10101010 y los bits que están de tercero, sexto y octavo se eliminan, el receptor obtendría 10011, en lugar de obtener 10?01?1? que es lo que se obtiene cuando los bits se borran. Esta sería la principal diferencia entre los canales de eliminación y borrado.

Ahora, a diferencia del canal de borrado, que está plenamente estudiado, por ejemplo ver [16] y [17], del canal de eliminación se desconocen muchas cosas como su capacidad, o la forma de recuperar los datos. Actualmente, no se tiene ninguna expresión de forma cerrada que describa la capacidad, ni se tiene un algoritmo eficiente para calcular numéricamente dicha capacidad.

En términos más generales, los canales con errores de sincronización, incluyendo tanto inserciones como supresiones de datos, así como errores de tiempo, simplemente no están adecuadamente entendidos por la teoría actual. Dado el conocimiento casi completo que se tiene sobre los canales con borrones y errores, tanto en términos de la capacidad del canal como de los algoritmos que pueden casi alcanzar dicha capacidad, la falta de comprensión acerca de canales con errores de sincronización es realmente notable.

En la actualidad se realizan diferentes estudios que están encaminados a describir y profundizar el conocimiento que se tiene sobre el canal de eliminación y su capacidad. De estos estudios destacamos un resultado reciente, debido a Mitzenmacher y Drinea en [7] que muestra que la capacidad del canal de eliminación es por lo menos $\frac{1-p}{9}$, cuando cada bit se borra independientemente con probabilidad fija p, el desarrollo de Kanoria y Montanari en [16] donde se estima la capacidad del canal usando aproximaciones por procesos estocásticos con procesos ergódicos y estacionarios y el trabajo de Diggavvi, Pfister y Mitzenmacher en [10] donde abordan el problema de una cota superior para la capacidad del canal por medio del estudio de los runs de las secuencias de datos a transmitir.

El presente trabajo tiene como objetivo principal estudiar un poco más el canal de eliminación, proponer algunas variantes a los estudios actuales para modelar el problema usando los runs de las palabras binarias a transmitir y unas funciones de codificación y decodificación propuestas que brindan nuevas formas de abordar el problema por medio del canal de $\Omega-$eliminación y principalmente ayudar a difundir la teoría de la información como una rama relativamente joven de las matemáticas y la ingeniería con mucho camino a desarrollar por delante.

Capítulo 1

Marco Teórico

1.1. Generalidades en teoría de la información

Definición 1.1. A continuación se brindan algunas definiciones fundamentales para el presente trabajo.

i) Un *espacio de probabilidad* $(\Omega, \mathcal{F}, \mathbb{P})$ está formado por una tripla donde Ω es un conjunto no vacío llamado el *espacio muestral* y esta formado por *eventos* o *estados* ω. Una $\sigma-$álgebra $\mathcal{F}$ la cuál es un sistema de subconjuntos de Ω que cumple: *1)* $\emptyset, \Omega \in \mathcal{F}$, *2)* si $A \in \mathcal{F}$ entonces $A^c \in \mathcal{F}$, *3)* si $A, B \in \mathcal{F}$ entonces $A \cap B \in \mathcal{F}$ y *4)* si $A_n \in \mathcal{F}$ entonces $\bigcup_{n=1}^{\infty} A_n \in \mathcal{F}$, donde dado $\omega \in \Omega$ y algún evento $A \in \mathcal{F}$ no es posible decir cuando ocurrirá ω, pero si es posible decidir cuando $\omega \in A$ o $\omega \notin A$. Y una aplicación $\mathbb{P} : \mathcal{F} \to [0, 1]$ llamada *medida de probabilidad*, que cumple $\mathbb{P}(\emptyset) = 0$, $\mathbb{P}(\Omega) = 1$ y si $A, B \in \mathcal{F}$ disjuntos, entonces $\mathbb{P}(A \cup B) = \mathbb{P}(A) + \mathbb{P}(B)$.

ii) Una función $X : \Omega \to \mathbb{R}$ es llamada *variable aleatoria* si para todo $-\infty < a < b < \infty$ se tiene que:

$$X^{-1}((a, b)) = \{\omega \in \Omega \,|\, a < X(\omega) < b\} \in \mathcal{F}$$

iii) Dada una variable aleatoria $X : \Omega \to \mathbb{R}$ sobre un espacio de probabilidad $(\Omega, \mathcal{F}, \mathbb{P})$, a la función $F_X(x) = \mathbb{P}(\omega \in \Omega \,|\, X(\omega) \leq x)$ se le denomina *función de distribución de probabilidad*. Para facilitar la notación de ahora en adelante se denota:

$$F_X(x) = \mathbb{P}(\omega \in \Omega \,|\, X(\omega) \leq x) = P(X \leq x)$$

iv) Una variable aleatoria es *discreta* si su rango contiene un número finito o contable de puntos y la *función de probabilidad de masa* $p_X(x_i)$ está dada por:

$$p_X(x_i) = P(X = x_i) = P(X \leq x_i) - P(X \leq x_{i-1}) = F_X(x_i) - F_X(x_{i-1})$$

v) Un *símbolo* i es una representación discreta de una serie de eventos o sucesos, puede incluso ser un solo evento, se denotara con $X = i$ a la variable aleatoria cuyo símbolo representa al evento i, luego si $F_X(i) = P(X(\omega) = i) = P(X = x_i) = P(X = i)$ representa la probabilidad de ocurrencia de un solo evento, denotado por el símbolo i, se tiene que:

$$\sum_i P(X(\omega) = i) = \sum_i P(X = i) = 1$$

$$F_X(x) = P(X \leq x) = \sum_{i \leq x} P(X = i)$$

vi) Un *alfabeto* $\mathbb{X}$ es el conjunto de todos los símbolos que representan un evento. Este contiene a todos los símbolos probables o eventos que puedan representar una secuencia de información. Una *palabra* es una sucesión o secuencia finita de elementos del alfabeto. La *longitud de la palabra* es la cantidad de símbolos en la secuencia. Un *alfabeto binario* $\mathbb{X}$ esta compuesto por los símbolos $\{0, 1\}$, estos son llamados *bits*.

vii) Una *fuente de información*, es una distribución de probabilidad, es decir, un conjunto de probabilidades asignadas a un conjunto de eventos, luego la información contenida en un evento no solo esta determinada por el evento mismo, también influye que tan incierta es su ocurrencia. Un evento *casi cierto* aporta poca información, es decir, hay poca incertidumbre en el resultado, por otro lado, un evento *casi improbable*, aporta mucha información, luego se tiene que la *medida de la información* recibida por la ocurrencia de un evento, es inversamente proporcional a la probabilidad de que ocurra.

Podemos decir que:

$$\textit{evento cierto} \longrightarrow \textit{probabilidad de ocurrencia} = 1 \longrightarrow \textit{información aportada nula}$$

$$\textit{evento falso} \longrightarrow \textit{probabilidad de ocurrencia} = 0 \longrightarrow \textit{información aportada infinita}$$

Definición 1.2. La *medida de la información* contenida en un evento se introdujo por Hartley en 1928 en [3] de la siguiente forma: Sea $(\Omega, \mathcal{F}, \mathbb{P})$ un espacio de probabilidad, una variable aleatoria $X : \Omega \to \mathbb{R}$ y $F_X(x) = \mathbb{P}(\omega \in \Omega \,|\, X(\omega) \leq x)$ una función de distribución de probabilidad. La *información propia*, o simplemente, *información* contenida en el evento $X = i$ es:

$$I(X = i) = log_d\left(\frac{1}{F_X(i)}\right) = -log_d(F_X(i)) = -log_d(P(X = i)) = -log_d P(X = i)$$

En general, si $A \in \mathcal{F}$ la información de A es $I(A) = -log_d \mathbb{P}(A)$

Ejemplo 1.3. El experimento a estudiar es lanzar una moneda que contiene dos posibles eventos: cara (C) o sello (S). Se tienen varios escenarios:

i) Con una moneda legal, los eventos $\{C, S\}$ son equiprobables cada uno con probabilidad de ocurrencia de $\frac{1}{2}$.

ii) Con una moneda con solo caras, los eventos $\{C, S\}$ tiene probabilidad de ocurrencia de 1 y 0 respectivamente.

iii) Con una moneda cargada, los eventos $\{C, S\}$ tiene probabilidad de ocurrencia de $\frac{1}{4}$ y $\frac{3}{4}$ respectivamente.

El experimento *ii*) es el que tiene más información y menor incertidumbre, mientras que el experimento *i*) es el que tiene menos información y por lo tanto mayor incertidumbre.

La medida de la información asociada al evento $X = i$, que ocurre con probabilidad $P(X = i)$ es definida y notadas sus unidades así:

$$\begin{aligned} I(X = i) &= -log_2 P(X = i) && bits \\ I(X = i) &= -log_{10} P(X = i) && hartleys \\ I(X = i) &= -ln P(X = i) && nants \end{aligned}$$

Por lo general, la unidad de medida más utilizada en teoría de la información son los bits.

Ejemplo 1.4. Si la probabilidad de que aparezca un valor al lanzar un dado limpio es $\frac{1}{6}$, la cantidad de información necesaria para predecir el valor de uno de esos lados es:

$$I(X = i) = -log_2 \frac{1}{6} = log_2 6 = 2{,}5857\ldots \quad bits, \;\; i = 1 \ldots 6.$$

Con esta medida vemos que se cumple el hecho de que la información es proporcional a la incertidumbre, pues si $P(X = i) = 1$, entonces $I(X = i) = 0$, es decir un evento cierto no contiene información y si $P(X = i) = 0$, entonces $I(X = i) = \infty$, o sea, un evento falso o improbable contiene infinita información.

Definición 1.5. Sean $(\Omega, \mathcal{F}, \mathbb{P})$ un espacio de probabilidad, $X_n : \Omega \to \mathbb{R}$ una sucesión de variables aleatorias, $F_{X_n}(x_n) = \mathbb{P}(\omega \in \Omega \,|\, X_n(\omega) \leq x_n)$ funciones de distribución de probabilidad. Las variables aleatorias $\{X_n\}$ son *independientes* si:

$$P(X_1 \leq x_1, X_2 \leq x_2, \ldots, X_n \leq x_n,) = \prod_{k=1}^{n} P(X_k \leq x_k)$$

para todo $x_k \in \mathbb{R}$. Las variables aleatorias son *idénticamente distribuidas* si:

$$F_{X_n}(x) = F_{X_m}(x)$$

para todo x y todo $n, m \in \mathbb{N}_0$

Nótese que la información total de eventos independientes se puede sumar.

Ejemplo 1.6. La cantidad de información en un mensaje enviado por una fuente de información, cuyo tiempo de envío es de por lo menos i segundos y que aparecen en promedio cada $\frac{1}{n}$ segundos, sigue una función de densidad de probabilidad exponencial así:

$$P(X) = P(X \leqslant i) = 1 - e^{-in}$$

luego:

$$I(X) = I(X \leqslant i) = -log_2 P(X = i) = -log_2(1 - e^{-in})$$

Si el tiempo de envío es menor a 2.5 segundos y los mensajes aparecen en promedio cada 6.5 segundos se tiene que:

$$I(X \leqslant 2{,}5) = -log_2(1 - e^{\frac{-2,5}{6,5}}) = 1{,}6471 \ bits$$

Ejemplo 1.7. Un caso importante ocurre cuando un alfabeto $\mathbb{X}$ tiene M símbolos equiprobables, en cuyo caso si X es una variable aleatoria que toma valores en $\mathbb{X}$:

$$P(X = m) = \frac{1}{M}$$

y así:

$$I(X = m) = -log_2 P(X = m) = -log_2 \frac{1}{M} = log_2 M, \quad m = 1, 2, \ldots, M$$

Para calcular la cantidad de información necesaria para predecir un evento B dado que ya ocurrió un evento A usamos el siguiente teorema:

Teorema 1.8. (Regla de Bayes). *Si $B_1, B_2, \ldots, B_n$ son n eventos disjuntos de los cuales uno de ellos debe ocurrir, es decir, $\sum_{i=1}^{n} P(B_i) = 1$ entonces:*

$$P(B_j \mid A) = \frac{P(B_j)P(A \mid B_j)}{\sum_{i=1}^{n} P(B_i)P(A \mid B_i)} \quad j = 1, 2 \ldots, n$$

Una prueba de este conocido resultado se puede ver en [18], Pag 30.

Ejemplo 1.9. Con el teorema anterior podemos calcular la *cantidad de información condicional* que aporta un evento B dado que ya ocurrió un evento A, así:

$$\begin{aligned} I(X = B \mid X = A) &= log_2 \frac{1}{P(X = B \mid X = A)} = log_2 \frac{P(A)}{P(B)P(A \mid B)} \\ &= log_2 P(A) - log_2 P(B) - log_2 P(A \mid B) \\ &= I(X = A \mid X = B) + I(X = B) - I(X = A) \end{aligned}$$

Al aplicar un determinado experimento, se tiene una fuente que emite varios eventos o resultados y se recibe información de cada uno de ellos, dependiendo de su probabilidad de ocurrencia. Si todos los eventos fueran equiprobables se tendría mayor incertidumbre sobre los resultados, luego se desea una medida para dicha incertidumbre.

Definición 1.10. (*Entropía, Hartley, 1928* [3]) Sea $(\Omega, \mathcal{F}, \mathbb{P})$ un espacio de probabilidad, una variable aleatoria discreta $X : \Omega \to \mathbb{R}$ que es una fuente de información que contiene varios eventos y $p_X(x) = P(X = x)$ una función de probabilidad de masa. La *medida de entropía*, o, *entropía* simplemente, define el contenido de información de la fuente como si esta fuera un evento completo, esta es la *información promedio* proporcionada por cada evento de la fuente o símbolo y es el valor esperado de la información $I(X)$ así si:

$$H(X) = E\{I(X)\} = \sum_i P(X = i)I(X = i) = -\sum_i P(X = i)log_2 P(X = i)$$

Acá la entropía es definida para variables aleatorias discretas (pero es posible definirla para el caso continuo) al igual que la información, está medida en bits y representa la cantidad promedio de bits en un código que son necesarios para transmitir la información. Se suponen todas las variables aleatorias discretas, a menos que se diga lo contrario.

Ejemplo 1.11. Si X tiene M eventos finitos, independientes y equiprobables, su entropía es:

$$H(X) = E\{I(X)\} = \sum_{m=1}^{M} P(X = m)I(X = m) = -\sum_{m=1}^{M} P(X = m)log_2 P(X = m)$$
$$= \sum_{m=1}^{M} \frac{1}{M} log_2 M = log_2 M$$

Si los eventos son equiprobables, se tiene mayor incertidumbre o entropía en la ocurrencia de un evento, luego se necesitaran más bits para transmitir la información o resultados de dicho experimento. Si X tiene M eventos finitos, es decir $\mid X \mid = M$, entonces: $H(X) \leq log_2 \mid X \mid$ llegando a la igualdad en el caso de que X sea una variable aleatoria uniforme.

Definición 1.12. Un *canal de comunicación*, o, *canal* simplemente, es una tripla (X, Y, T) formada por un alfabeto de entrada $\mathbb{X}$, o *fuente*, un alfabeto de salida $\mathbb{Y}$, o *receptor* y una matriz T, de tamaño m x n, que es una matriz de transición de probabilidades (estocástica por filas), que indica las probabilidades de que al enviar la palabras X, se reciban la palabras Y.

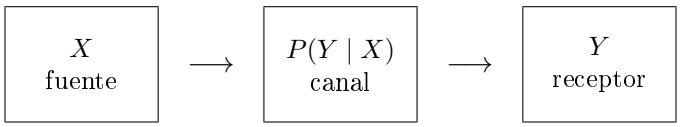
La matriz T también se denota como $P(Y \mid X)$ y queda expuesta la relación entre X y Y en el siguiente lema:

Lema 1.13. *Si A es una matriz de tamaño* m x n*, estocástica por filas y X una distribución de probabilidad dada como un m-vector fila, entonces:*

$$Y = XA$$

es una distribución de probabilidad dada por el n-vector fila Y.

Ahora estamos interesados en encontrar la entropía del alfabeto $\mathbb{Y}$, el cual está correlacionado con el alfabeto $\mathbb{X}$, según el diagrama inicial:



Como X y Y están correlacionadas, debe existir una función de distribución de probabilidad conjunta $F_{X,Y}(x,y) = P(X \leq x, Y \leq y) = P(X,Y)$ y con esta se brindaran las definiciones de entropía e información conjunta.

Definición 1.14. Sean $\mathbb{X}$ y $\mathbb{Y}$ alfabetos con M y N símbolos respectivamente, X y Y variables aleatorias que toman valores en $\mathbb{X}$ y $\mathbb{Y}$ respectivamente, la *entropía conjunta* $H(X,Y)$ es:

$$H(X,Y) = -\sum_{m=1}^{M}\sum_{n=1}^{N} P(X=m, Y=n) log_2 P(X=m, Y=n)$$

También es posible definir la entropía de forma condicional para los alfabetos $\mathbb{X}$ y $\mathbb{Y}$, así:

Definición 1.15. Sean $\mathbb{X}$ y $\mathbb{Y}$ alfabetos con M y N símbolos respectivamente, X y Y variables aleatorias que toman valores en $\mathbb{X}$ y $\mathbb{Y}$ respectivamente, la *entropía condicional* es:

$$H(Y \mid X=m) = -\sum_{n=1}^{N} P(Y=n \mid X=m) log_2 P(Y=n \mid X=m)$$

de forma más general se tiene:

$$H(Y \mid X) = \sum_{m=1}^{M} P(X=m) H(Y \mid X=m)$$

Ahora para las palabras X y Y se tiene que $P(X,Y) = P(Y)P(X \mid Y) = P(X)P(Y \mid X)$ luego se observa que el nivel de independencia entre las palabras enviadas y recibidas depende del factor:

$$\frac{P(X,Y)}{P(X)P(Y)}$$

Con esto se brinda la siguiente:

Definición 1.16. Sean $\mathbb{X}$ y $\mathbb{Y}$ alfabetos con M y N símbolos respectivamente, X y Y variables aleatorias que toman valores en $\mathbb{X}$ y $\mathbb{Y}$ respectivamente, la *información mutua promedio*, o simplemente la *información mutua* es:

$$I(X,Y) = \sum_{m=1}^{M}\sum_{n=1}^{N} P(X=m, Y=n) log_2 \frac{P(X=m, Y=n)}{P(X=m)P(Y=n)}$$

Se usara con frecuencia el siguiente resultado:

Lema 1.17. *Sean $\mathbb{X}$ y $\mathbb{Y}$ alfabetos con M y N símbolos respectivamente, X y Y variables aleatorias que toman valores en $\mathbb{X}$ y $\mathbb{Y}$ respectivamente,:*

i) $P(X = m) = \sum_{n=1}^{N} P(X = m, Y = n)$

ii) $P(Y = n) = \sum_{m=1}^{M} P(X = m, Y = n)$

Este resultado es inmediato a partir de tomar la probabilidad sobre todo el alfabeto completo $\mathbb{X}$ o $\mathbb{Y}$. Se usara el lema anterior en el siguiente:

Lema 1.18. *Sean $\mathbb{X}$ y $\mathbb{Y}$ alfabetos con M y N símbolos respectivamente, X y Y variables aleatorias que toman valores en $\mathbb{X}$ y $\mathbb{Y}$ respectivamente,:*

i) $H(X, Y) = H(X) + H(Y \mid X)$

ii) $H(X, Y) = H(Y) + H(X \mid Y)$

Demostración.

$$
\begin{aligned}
H(X,Y) &= -\sum_{m=1}^{M}\sum_{n=1}^{N} P(X=m,Y=n) log_2 P(X=m,Y=n) \\
&= -\sum_{m=1}^{M}\sum_{n=1}^{N} P(X=m,Y=n) log_2 P(X=m) P(Y=n \mid X=m) \\
&= -\left(\sum_{m=1}^{M}\sum_{n=1}^{N} P(X=m,Y=n) log_2 P(X=m) + \sum_{m=1}^{M}\sum_{n=1}^{N} P(X=m,Y=n) log_2 P(Y=n \mid X=m)\right) \\
&= -\sum_{m=1}^{M}\left(\sum_{n=1}^{N} P(X=m,Y=n)\right) log_2 P(X=m) \\
&- \sum_{m=1}^{M}\sum_{n=1}^{N} P(X=m) P(Y=n \mid X=m) log_2 P(Y=n \mid X=m) \\
&= -\sum_{m=1}^{M} P(X=m) log_2 P(X=m) - \sum_{m=1}^{M} P(X=m) \sum_{n=1}^{N} P(Y=n \mid X=m) log_2 P(Y=n \mid X=m) \\
&= -\sum_{m=1}^{M} P(X=m) log_2 P(X=m) - \sum_{m=1}^{M} P(X=m) H(Y \mid X=m) \\
&= H(X) + H(Y \mid X).
\end{aligned}
$$

La parte *ii*) se sigue de forma análoga a la parte *i*). □

Usando los lemas anteriores y la definición de información mutua tenemos el siguiente:

Teorema 1.19. *Sean $\mathbb{X}$ y $\mathbb{Y}$ alfabetos con M y N símbolos respectivamente, X y Y variables aleatorias que toman valores en $\mathbb{X}$ y $\mathbb{Y}$ respectivamente, entonces:*

$$I(X,Y) = H(X) + H(Y) - H(X,Y)$$

Demostración.

$$\begin{aligned}
I(X,Y) &= \sum_{m=1}^{M}\sum_{n=1}^{N} P(X=m,Y=n) log_2 \frac{P(X=m,Y=n)}{P(X=m)P(Y=n)} \\
&= \sum_{m=1}^{M}\sum_{n=1}^{N} P(X=m,Y=n)\left(log_2 P(X=m,Y=n) - log_2 P(X=m)P(Y=n)\right) \\
&= \sum_{m=1}^{M}\sum_{n=1}^{N} P(X=m,Y=n)\left(log_2 P(X=m,Y=n) - log_2 P(X=m) - log_2 P(Y=n)\right) \\
&= \sum_{m=1}^{M}\sum_{n=1}^{N} P(X=m,Y=n) log_2 P(X=m,Y=n) \\
&- \sum_{m=1}^{M}\sum_{n=1}^{N} P(X=m,Y=n) log_2 P(X=m) - \sum_{m=1}^{M}\sum_{n=1}^{N} P(X=m,Y=n) log_2 P(Y=n) \\
&= -H(X,Y) - \sum_{m=1}^{M} P(X=m) log_2 P(X=m) - \sum_{n=1}^{N}\sum_{m=1}^{M} P(X=m,Y=n) log_2 P(Y=n) \\
&= -H(X,Y) + H(X) - \sum_{n=1}^{N} P(Y=n) log_2 P(Y=n) \\
&= H(X) + H(Y) - H(X,Y)
\end{aligned}$$

□

Con base en la definición de información mutua, tenemos que si X y Y son independientes entonces:

$$\begin{aligned}
I(X,Y) &= \sum_{m=1}^{M}\sum_{n=1}^{N} P(X=m,Y=n) log_2 \frac{P(X=m,Y=n)}{P(X=m)P(Y=n)} \\
&= \sum_{m=1}^{M}\sum_{n=1}^{N} P(X=m,Y=n) log_2 \frac{P(X=m)P(Y=n)}{P(X=m)P(Y=n)} \\
&= \sum_{m=1}^{M}\sum_{n=1}^{N} P(X=m,Y=n) log_2 1 = 0
\end{aligned}$$

Lo cual es coherente dado que si X no tiene nada que ver con Y, pues son independientes, no hay información mutua, teniendo así el siguiente:

Corolario 1.20. *Sean $\mathbb{X}$ y $\mathbb{Y}$ alfabetos independientes con M y N símbolos respectivamente, X y Y variables aleatorias que toman valores en $\mathbb{X}$ y $\mathbb{Y}$ respectivamente, entonces:*

$$H(X,Y) = H(X) + H(Y)$$

Veamos un ejemplo en donde se calculan algunas entropías:

Ejemplo 1.21. Sean $\mathbb{X} = \{0,1\}$ un alfabeto fuente y $\mathbb{Y} = \{0,1\}$ un alfabeto receptor.
Las probabilidades de ocurrencia de los eventos conjuntos en los alfabetos $\mathbb{X}$ y $\mathbb{Y}$ están dadas en la siguiente tabla:

$P(X,Y)$	$P(Y=0)$	$P(Y=1)$	$P(X=m)$
$P(X=0)$	$\frac{1}{3}$	$\frac{1}{3}$	$P(X=0)=\frac{2}{3}$
$P(X=1)$	0	$\frac{1}{3}$	$P(X=1)=\frac{1}{3}$
$P(Y=n)$	$P(Y=0)=\frac{1}{3}$	$P(Y=1)=\frac{2}{3}$	

Algunas entropías son:

$$H(X) = -\sum_{m=0}^{1} P(X=m) log_2 P(X=m) = -\left(\frac{2}{3}\right) log_2\left(\frac{2}{3}\right) - \left(\frac{1}{3}\right) log_2\left(\frac{1}{3}\right) = 0,918 = H(Y)$$

$$\begin{aligned} H(X \mid Y) &= \sum_{n=0}^{1} P(Y=n) H(X \mid Y=n) = P(Y=0)H(X \mid Y=0) + P(Y=1)H(X \mid Y=1) \\ &= \frac{1}{3}H(1) + \frac{2}{3}H\left(\frac{1}{2},\frac{1}{2}\right) = 0 + \frac{2}{3}\left(-\left(\frac{1}{2}\right) log_2\left(\frac{1}{2}\right) - \left(\frac{1}{2}\right) log_2\left(\frac{1}{2}\right)\right) = \frac{2}{3}\left(\frac{1}{2}+\frac{1}{2}\right) = \frac{2}{3} \end{aligned}$$

$$\begin{aligned} H(Y \mid X) &= \sum_{m=0}^{1} P(X=m) H(Y \mid X=m) = P(X=0)H(Y \mid X=0) + P(X=1)H(Y \mid X=1) \\ &= \frac{1}{3}H\left(\frac{1}{2},\frac{1}{2}\right) + \frac{2}{3}H(1) = \frac{2}{3}\left(-\left(\frac{1}{2}\right) log_2\left(\frac{1}{2}\right) - \left(\frac{1}{2}\right) log_2\left(\frac{1}{2}\right)\right) + 0 = \frac{2}{3}\left(\frac{1}{2}+\frac{1}{2}\right) = \frac{2}{3} \end{aligned}$$

$$H(X,Y) = H(X) + H(Y \mid X) = H(Y) + H(X \mid Y) = 0{,}918 + \frac{2}{3} = 1{,}585$$

$$I(X,Y) = H(X) + H(Y) - H(X,Y) = 0{,}251$$

Aunque en el ejemplo anterior $H(X \mid Y) = H(Y \mid X)$, en muchos casos estos valores son diferentes, luego lo habitual es que $H(X \mid Y) \neq H(Y \mid X)$.

Una función $f(x)$ se llama *cóncava* sobre un intervalo (a, b) si para todo $x_1, x_2 \in (a, b)$ y $0 \leq \lambda \leq 1$, se tiene: $f(\lambda x_1 + (1-\lambda)x_2) \geq \lambda f(x_1) + (1-\lambda)f(x_2)$ usando esta definición se presenta el siguiente:

Lema 1.22. (Desigualdad de Jensen). *Si f es una función cóncava, entonces:*

$$f\left(\sum_{i=1}^{n} a_i X_i\right) \geqslant \sum_{i=1}^{n} a_i f(X_i)$$

con $0 \leqslant a_i$ para todo i y $\sum_{i=1}^{n} a_i = 1$.

Demostración. Se puede proceder por inducción, supongamos la hipótesis de concavidad para $n = 2$ y que es valida para n, se verificara para $n+1$. Al menos uno de los a_i es estrictamente positivo, digamos a_1 luego por la concavidad de f se tiene:

$$f\left(\sum_{i=1}^{n} a_i X_i\right) = f\left(a_1 x_1 + (1-a_1)\sum_{i=2}^{n+1} \frac{a_i}{1-a_1} X_i\right) \geq a_1 f(x_1) + (1-a_1) f\left(\sum_{i=2}^{n+1}\left(\frac{a_i}{1-a_1} X_i\right)\right)$$

Dado que $\sum_{i=2}^{n+1} \frac{a_i}{1-a_1} = 1$, se usa la hipótesis de inducción al ultimo termino de la desigualdad anterior obteniendo así el resultado. □

Usaremos el anterior resultado para probar el siguiente:

Teorema 1.23. *Sean X y Y alfabetos con M y N símbolos respectivamente, X y Y variables aleatorias que toman valores en $\mathbb{X}$ y $\mathbb{Y}$ respectivamente, entonces:*

$$H(X \mid Y) \leqslant H(X)$$

Demostración. Es equivalente comprobar que $H(X) - H(X \mid Y) \geqslant 0$, que junto con el lema 1.18 equivale a mostrar que $H(X) + H(Y) - H(X, Y) \geqslant 0$
Ahora:

$$\begin{aligned} H(X) + H(Y) - H(X,Y) = &- \sum_{m=1}^{M} P(X=m) log_2 P(X=m) - \sum_{n=1}^{N} P(Y=n) log_2 P(Y=n) \\ &+ \sum_{m=1}^{M}\sum_{n=1}^{N} P(X=m, Y=n) log_2 P(X=m, Y=n) \end{aligned}$$

Por el lema 1.17 se tiene:

$$
\begin{aligned}
&= -\sum_{m=1}^{M}\sum_{n=1}^{N} P(X=m,Y=n)log_2P(X=m) - \sum_{n=1}^{N}\sum_{m=1}^{M} P(X=m,Y=n)log_2P(Y=n) \\
&\quad + \sum_{m=1}^{M}\sum_{n=1}^{N} P(X=m,Y=n)log_2P(X=m,Y=n) \\
&= -\sum_{m=1}^{M}\sum_{n=1}^{N}(P(X=m,Y=n)log_2P(X=m) + P(X=m,Y=n)log_2P(Y=n) \\
&\quad - P(X=m,Y=n)log_2P(X=m,Y=n)) \\
&= -\sum_{m=1}^{M}\sum_{n=1}^{N} P(X=m,Y=n)\left(log_2P(X=m) + log_2P(Y=n) - log_2P(X=m,Y=n)\right) \\
&= -\sum_{m=1}^{M}\sum_{n=1}^{N} P(X=m,Y=n)\left(log_2\left(\frac{P(X=m)P(Y=n)}{P(X=m,Y=n)}\right)\right)
\end{aligned}
$$

Tomando a $P(X=m,Y=n)$ como a_i en el lema 1.22 y usado la concavidad del logaritmo se tiene:

$$
\begin{aligned}
&\geqslant -log_2\left(\sum_{m=1}^{M}\sum_{n=1}^{N} P(X=m,Y=n)\frac{P(X=m)P(Y=n)}{P(X=m,Y=n)}\right) \\
&= -log_2\left(\sum_{m=1}^{M}\sum_{n=1}^{N} P(X=m)P(Y=n)\right) = -log_21 = 0
\end{aligned}
$$

Lo cual prueba la afirmación. □

Un resultado directo del lema 1.18 y el teorema 1.19 es el siguiente:

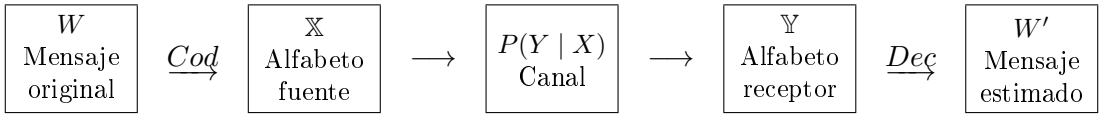
Corolario 1.24. *Sean $\mathbb{X}$ y $\mathbb{Y}$ alfabetos con M y N símbolos respectivamente, X y Y variables aleatorias que toman valores en $\mathbb{X}$ y $\mathbb{Y}$ respectivamente, entonces:*

$$I(X,Y) = H(Y) - H(Y \mid X) = H(X) - H(X \mid Y)$$

1.2. La capacidad de información y ejemplos de canales

Ahora que definimos el concepto de canal y tenemos algunas nociones sobre el envío de información, surge una pregunta natural: ¿Como se hace en la practica?.

La idea principal es tomar un mensaje W que se desea transmitir y codificarlo o reescribirlo en otro lenguaje (Cod), más apropiado para la transmisión, (por ejemplo, el lenguaje binario), este lenguaje escogido está formado por el alfabeto fuente $\mathbb{X}$ y es enviado a través del canal según lo indique la matriz de transición $P(Y \mid X)$ hasta obtener un alfabeto receptor $\mathbb{Y}$, luego este mensaje es decodificado o reescrito al lenguaje original (Dec), obteniendo un mensaje estimado de llegada W'. Se ilustra la idea a continuación, en este nuevo diagrama:



Surgen algunas preguntas importantes:

¿Que tanta información es posible enviar por el canal?

¿Es confiable la transmisión, en la medida que llegue el mensaje esperado?

¿Codificar un mensaje de diversas formas mejora la transmisión de datos?

Para estudiar más a fondo estas cuestiones, es necesario introducir algunas definiciones adicionales:

Definición 1.25. Un canal es un *canal sin memoria* si la distribución de probabilidad del alfabeto receptor $\mathbb{Y}$ depende unicamente del alfabeto fuente $\mathbb{X}$, y no es afectada por mensajes previamente enviados, es decir:
Si $X = \{x_1, x_2, \ldots, x_m\}$ y $Y = \{y_1, y_2, \ldots, y_n\}$ se tiene:

$$P(y_n \mid x_1, x_2, \ldots, x_m, y_1, y_2, \ldots, y_n) = P(y_n \mid x_n)$$

Definición 1.26. Una *fuente binaria simétrica* (FBS) es una fuente X con dos eventos $\{0, 1\}$ cuyas probabilidades son p y $1 - p$ respectivamente. La entropía de esta fuente es:

$$H(X) = -p\, log_2 p \; - \; (1-p)\, log_2(1-p)$$

Cuando uno u otro evento son ciertos, la entropía es cero. Cuando p aumenta, la entropía también, hasta que alcanza un máximo con $p = 1 - p = 0{,}5$, acá $H(X) = 1$. Cuando p es mayor que 0.5 , la curva tiende a cero simétricamente hasta que llega a cero con $p = 1$.

En teoría de la información tradicionalmente se usa la palabra *bit* para describir tanto al símbolo como a su contenido. Una FBS cuyos eventos son 1 o 0 tiene un valor que se describe como bit, y la entropía también se mide en bits.

La entropía se maximiza cuando ambos eventos son equiprobables, así para una fuente cuyo alfabeto tiene M símbolos equiprobables se tiene que:

$$0 \leqslant H(X) \leqslant log_2 M$$

Si se tiene un alfabeto binario, del cual se tienen dos símbolos y se envía un mensaje que esta representado por una sucesión K de símbolos independientes y equiprobables, cada uno de ellos tiene una probabilidad de ocurrencia de 0.5, y:

$$I(X = m) = -log_2 \frac{1}{2} = log_2 2 = 1$$

La información del mensaje es:

$$I(X) = \sum_{m=1}^{M} I(X = m) = K \quad bits$$

Si hay M mensajes, representados por K símbolos binarios por mensaje y estos son equiprobables, se tiene que:

$$I(X = m) = K = log_2 M$$

Así, por ejemplo, si un símbolo puede representarse por 15 bits, el alfabeto puede tener:

$$M = 2^K = 2^{15} = 32768 \text{ símbolos posibles.}$$

Definición 1.27. Sea X una variable aleatoria continua, la *tasa de información*, o, simplemente *tasa*, es una medida de la velocidad a la que la información es transmitida y esta dada por:

$$R = \frac{H(X)}{t}$$

Donde $H(X)$ es la entropía de la fuente y t es el intervalo de tiempo en el que la fuente transmite el evento o serie de eventos, en el caso de transmisiones continuas.

Si se tienen M símbolos equiprobables, estadísticamentes independientes y de igual duración t, la tasa de transmisión máxima es:

$$R = \frac{H(X)}{t} = \frac{1}{t}\sum_{m=1}^{M} I(X=m)P(X=m) = \frac{1}{Mt}\sum_{m=1}^{M} log_2 M = \frac{1}{t} log_2 M \quad bits/segundo$$

La tasa de información para una FBS es:

$$R = -\frac{1}{t_0} p\, log_2 p \; - \; \frac{1}{t_1}(1-p)\, log_2(1-p)$$

Donde t_0 y t_1 determinan la duración de cada uno de los eventos. Si la duración de estos eventos esta determinada por una unidad de tiempo t, entonces:

$$R = -\frac{1}{kt} p\, log_2 p \; - \; \frac{1}{t(1-k)}(1-p)\, log_2(1-p)$$

Donde $0 \leqslant k \leqslant 1$, este valor se denomina *ciclo de trabajo de la fuente*.

Tal vez, la pregunta más relevante asociada con un canal de comunicación es la cantidad máxima de información presente en un canal y cuál es la tasa máxima a la que se puede transmitir dicha información, minimizando los errores que se puedan generar, pues por un canal no se puede enviar más de cierta cantidad de datos.

Por otro lado, si en un canal fijamos la matriz de transición $P(Y \mid X)$, podemos considerar a la información mutua $I(X;Y)$ como una función en las probabilidades $p_i = P(X = i)$, es decir $I(X;Y) = f(p_1, p_2, \ldots, p_m)$, así tiene sentido calcular el máximo de la información mutua $I(X,Y)$ con respecto a las probabilidades p_i y con esto llegamos a la siguiente:

Definición 1.28. (*Capacidad de información, Shannon, 1948,* [1]) Sean $\mathbb{X}$ y $\mathbb{Y}$ alfabetos con M y N símbolos respectivamente, X y Y variables aleatorias que toman valores en $\mathbb{X}$ y $\mathbb{Y}$ respectivamente, y $(\mathbb{X}, \mathbb{Y}, P(Y \mid X))$ un canal de información sin memoria.

La capacidad de información del canal, denotada por $\mathbb{C}$ es:

$$\mathbb{C} = \max_{P(X)} I(X,Y) = \max_{P(X)} I(X=m, Y=n)$$

Donde $I(X,Y)$ es la información mutua obtenida por X y Y y el máximo se toma sobre todas las posibles distribuciones de probabilidad de entrada $P(X)$.

Para ilustrar este concepto veamos algunos ejemplos de canales conocidos:

Ejemplo 1.29. ***Canal binario sin ruido***

Sean $\mathbb{X} = \{0, 1\}$ y $\mathbb{Y} = \{0, 1\}$ los alfabetos binarios fuente y receptor respectivamente.
En este caso las entradas en la fuente son transmitidas exactamente en la salida del receptor.
No hay errores en la transmisión, según se ve en el diagrama:

$$\begin{array}{ccccc} & 0\bullet & \longrightarrow & \bullet 0 & \\ X & & & & Y \\ & 1\bullet & \longrightarrow & \bullet 1 & \end{array}$$

Cada bit se transmite al usar el canal.

La matriz de transición del canal es:

$$P(Y \mid X) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

En este caso la capacidad de información es:

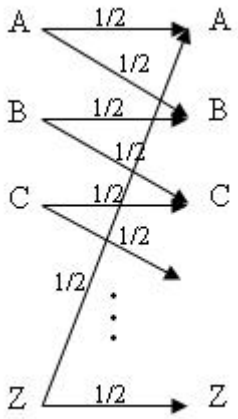
$$\mathbb{C} = \max_{P(X)} I(X, Y) = -log_2 \left(\frac{1}{2}\right) = 1 \quad bit$$

La cual se obtiene cuando $P(X) = (\frac{1}{2}, \frac{1}{2})$.

Ejemplo 1.30. ***Canal del alfabeto ruidoso***

Sean $\mathbb{X} = \{A, B, \ldots, Z\}$ y $\mathbb{Y} = \{A, B, \ldots, Z\}$ los alfabetos fuente y receptor respectivamente, formados por las 26 letras del alfabeto ingles (sin la ñ). En este caso el símbolo de la fuente es recibido correctamente con probabilidad $\frac{1}{2}$, o, es transformado en la siguiente letra del alfabeto con probabilidad $\frac{1}{2}$.

Luego se puede transmitir uno de 13 símbolos sin error en cada transmisión, según el diagrama:



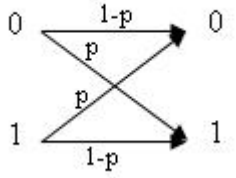
La capacidad de información de este canal es:

$$\begin{aligned}\mathbb{C} &= \max_{P(X)} I(X,Y) = \max_{P(X)}\{H(Y) - H(Y \mid X)\} = \max_{P(X)}\{H(Y)\} - \max_{P(X)}\{H(Y \mid X)\} \\ &= \max_{P(X)}\{log_2 26\} - \max_{P(X)}\{log_2 2\} = log_2 26 - 1 = log_2 13\end{aligned}$$

La cual es alcanzada si $P(X)$ se distribuye uniformemente sobre todas las entradas.

Ejemplo 1.31. ***Canal binario simétrico***

Sean $\mathbb{X} = \{0,1\}$ y $\mathbb{Y} = \{0,1\}$ los alfabetos binarios fuente y receptor respectivamente. En este caso las probabilidades son: llegada correcta $1-p$ y llegada errónea p según el diagrama:



La matriz de transición del canal es:

$$P(Y \mid X) = \begin{pmatrix} 1-p & p \\ p & 1-p \end{pmatrix}$$

Dado que:

$$I(X,Y) = H(Y) - H(Y \mid X) = H(Y) - \sum_{m=1}^{M} P(X=m)H(Y \mid X=m)$$
$$= H(Y) - \sum_{m=1}^{2} P(X=m)H(Y \mid X=m) = H(Y) - (-p\,log_2 p\, -\, (1-p)\,log_2(1-p))$$
$$= H(Y) + p\,log_2 p\, +\, (1-p)\,log_2(1-p)$$

Si las entradas se distribuyen de manera uniforme tenemos que:

$$H(Y) = -log_2 \frac{1}{2} = log_2 2 = 1$$

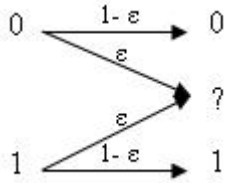
que es el valor máximo de la entropía, con esto la capacidad de este canal es:

$$\mathbb{C} = \max_{P(X)} I(X,Y) = \max_{P(X)} \{H(Y) + p\,log_2 p\, +\, (1-p)\,log_2(1-p)\}$$
$$\leqslant 1 + p\,log_2 p\, +\, (1-p)\,log_2(1-p)$$

Ejemplo 1.32. ***Canal de borrado binario***

Sean $\mathbb{X} = \{0,1\}$ y $\mathbb{Y} = \{0,1,?\}$ los alfabetos binarios fuente y receptor respectivamente. En este caso las probabilidades son: llegada correcta $1-p$ y llegada errónea p, pero a diferencia del canal anterior una porción ϵ de los bits son corrompidos, cuando faltan, el receptor *conoce cuales bits se perdieron*, en cuyo caso se asigna el símbolo ? para indicar donde se presento el borrón.

Este es el primer ejemplo de canal con *errores de sincronización* o perdida de datos, según el diagrama:



La matriz de transición del canal es:

$$P(Y \mid X) = \begin{pmatrix} 1-\epsilon & \epsilon & 0 \\ 0 & \epsilon & 1-\epsilon \end{pmatrix}$$

La capacidad de información de este canal es:

$$\begin{aligned}\mathbb{C} &= \max_{P(X)} I(X,Y) = \max_{P(X)}\{H(X) - H(X \mid Y)\} = \max_{P(X)}\{H(X) - \sum_{n=1}^{3} P(Y=n)H(X \mid Y=n)\} \\ &= \max_{P(X)}\{H(X) - (\, p(1-\epsilon)H(X \mid Y=0) + (\, p\epsilon + (1-p)\epsilon)H(X \mid Y =?) \\ &+ (1-p)(1-\epsilon)H(X \mid Y=1))\} \\ &= \max_{P(X)}\{H(X)\} - \max_{P(X)}\{(\, p(1-\epsilon)0 + (\, p\epsilon + (1-p)\epsilon)H(X) + (1-p)(1-\epsilon)0\} \\ &= \max_{P(X)}\{H(X)\} - \max_{P(X)}\{\epsilon H(X)\} = \max_{P(X)}\{(1-\epsilon)H(X)\} \leqslant 1-\epsilon\end{aligned}$$

El máximo se alcanza cuando $p = 0{,}5$, pues la entropía $H(X)$ es 1.

Ejemplo 1.33. ***Canales simétricos***

Sean $\mathbb{X}$ y $\mathbb{Y}$ alfabetos con M y N símbolos respectivamente, un canal $(\mathbb{X}, \mathbb{Y}, P(Y \mid X))$ es un canal *simétrico respecto a la entrada* si todas las filas de la matriz de transición $P(Y \mid X)$ tienen los mismos valores, salvo permutación.

Si se denota por υ a una fila cualquiera de la matriz de transición, se tiene que todas las filas de $P(Y \mid X)$ tienen la misma entropía, luego $H(Y \mid X) = H(\upsilon)$, y por lo tanto la capacidad de este canal es:

$$\begin{aligned}\mathbb{C} &= \max_{P(X)} I(X,Y) = \max_{P(X)}\{H(X) - H(X \mid Y)\} = \max_{P(X)}\{H(X)\} - H(\upsilon) \\ &\leq log_2 \mid Y \mid - H(\upsilon).\end{aligned}$$

La igualdad en este caso se obtiene si existe un alfabeto $\mathbb{X}$ para el cual $\mathbb{Y}$ es uniforme, pues si X es uniforme se tiene:

$$p(y) = \sum_{x \in X} p(y \mid x)p(x) = \frac{1}{\mid X \mid} \sum_{x \in X} p(y \mid x) = c\frac{1}{\mid X \mid} = \frac{1}{\mid Y \mid}$$

Donde c es la suma de las entradas en una columna de la matriz de transición.

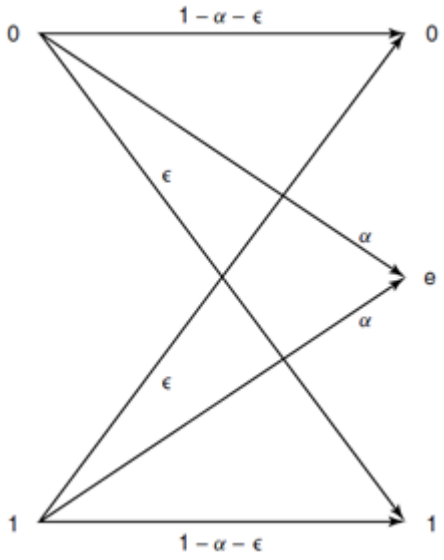
Un canal $(\mathbb{X}, \mathbb{Y}, P(Y \mid X))$ es un canal *simétrico respecto a la salida* si todas las columnas de la matriz de transición $P(Y \mid X)$ tienen los mismos valores, salvo permutación. Con esto decimos que un canal es *simétrico* si es simétrico respecto a la entrada y la salida.

El canal binario simétrico (CBS), es un ejemplo de canal simétrico.

Ejemplo 1.34. ***Canal de borrado binario con errores***

El canal de borrado binario con errores (CBBE), es un canal simétrico respecto a la entrada, pero no respecto a la salida, generaliza los casos ocurridos en el canal de borrado binario (CBB) y el canal binario simétrico (CBS).

Sean $\mathbb{X} = \{0, 1\}$ y $\mathbb{Y} = \{0, 1, ?\}$ los alfabetos binarios fuente y receptor respectivamente y sean ϵ la porción de datos erróneos y α la porción de datos borrados, según el diagrama:



La matriz de transición del canal es:

$$P(Y \mid X) = \begin{pmatrix} 1-\epsilon-\alpha & \alpha & \epsilon \\ \epsilon & \alpha & 1-\epsilon-\alpha \end{pmatrix}$$

La capacidad de información de este canal es máxima cuando hay equiprobabilidad, es decir cuando $P(X = 0) = P(X = 1) = \frac{1}{2}$ así, por simetría:

$$\begin{aligned} H(Y \mid X) &= P(X=0)H(Y \mid X=0) + P(X=1)H(Y \mid X=1) \\ &= \frac{1}{2}H(Y \mid X=0) + \frac{1}{2}H(Y \mid X=1) \\ &= H(Y \mid X=0) = -(1-\alpha-\epsilon)log_2(1-\alpha-\epsilon) - \epsilon log_2\epsilon - \alpha log_2\alpha. \end{aligned}$$

Por otro lado se tiene:

$$\begin{aligned} H(Y) &= -P(Y=0)log_2P(Y=0) - P(Y=1)log_2P(Y=1) - P(Y=?)log_2P(Y=?) \\ &= -(1-\alpha)[log_2(1-\alpha)-1] - \alpha log_2\alpha. \end{aligned}$$

Con lo anterior, la capacidad del CBBE es:

$$\begin{aligned} \mathbb{C} &= \max_{P(X)} I(X,Y) = \max_{P(X)}\{H(Y) - H(Y \mid X)\} \\ &= -(1-\alpha)[log_2(1-\alpha)-1] - \alpha log_2\alpha + (1-\alpha-\epsilon)log_2(1-\alpha-\epsilon) + \epsilon log_2\epsilon + \alpha log_2\alpha \\ &= -(1-\alpha)[log_2(1-\alpha)-1] + (1-\alpha-\epsilon)log_2(1-\alpha-\epsilon) + \epsilon log_2\epsilon \end{aligned}$$

Se tiene entonces que:

Si $\alpha = 0$, no hay borrones, se obtiene la capacidad del CBS.

Si $\epsilon = 0$, no hay errores, se obtiene la capacidad del CBB.

Por ejemplo, si la probabilidad de error es $\epsilon = \frac{1}{10}$ y la probabilidad de borrón es $\alpha = \frac{1}{3}$, se tiene:

$$(P(X=0), P(X=1)) = \begin{pmatrix} 1-\epsilon-\alpha & \alpha & \epsilon \\ \epsilon & \alpha & 1-\epsilon-\alpha \end{pmatrix} = \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)$$

Luego el máximo se alcanza cuando $P(X=0) = P(X=1) = \frac{1}{2}$ y la capacidad del CBBE es:

$$\mathbb{C} = \max_{P(X)} I(X,Y) = log_2 3 - H\left(\frac{17}{30}, \frac{1}{3}, \frac{1}{10}\right) = 0,26 bits.$$

Capítulo 2

Los Teoremas de Shannon y Características de los Canales.

2.1. Preliminares a los teoremas de Shannon

Con los anteriores ejemplos, calculamos la capacidad de información de algunos canales, sin embargo existe otro concepto de capacidad para un canal, llamado *la capacidad operativa* del canal, esta es la mayor tasa de transmisión (en bits/ segundo) de datos para un canal, a la que se puede enviar información arbitraria con baja probabilidad de error, antes de concluir el capitulo con los relevantes teoremas de Shannon sobre la capacidad de un canal y la existencia de códigos, debemos ilustrar unas definiciones adicionales:

Sean $(\Omega, \mathcal{F}, \mathbb{P})$ un espacio de probabilidad, $X_n : \Omega \to \mathbb{R}$ una sucesión de variables aleatorias, $F_{X_n}(x_n) = \mathbb{P}(\omega \in \Omega \,|\, X_n(\omega) \leq x_n)$ funciones de distribución de probabilidad.

Definición 2.1. Una sucesión de variables aleatorias $(X_n)_{n\in\mathbb{N}} : \Omega \to \mathbb{R}$ *converge en probabilidad* a X, denotado $X_n \xrightarrow{p} X$ si:

$$\forall \epsilon > 0, \ \lim_{n\to\infty} \mathbb{P}(\{\omega \in \Omega : |X_n(\omega) - X(\omega)| > \epsilon\}) = \lim_{n\to\infty} P\left(\mid X_n - X \mid > \epsilon\right) = 0$$

Definición 2.2. Una sucesión de variables aleatorias $(X_n)_{n\in\mathbb{N}} : \Omega \to \mathbb{R}$ *converge con probabilidad 1* a X, denotado $X_n \xrightarrow{1} X$ si y solo si:

$$\mathbb{P}(\{\omega \in \Omega : X_n(\omega) \to X(\omega) \text{ si } n \to \infty\}) = P(X_n \to X) = 1 \text{ si } n \to \infty$$

Esto es, todas las X_n siguen el *modelo* de X.

Teorema 2.3. (Ley débil de los grandes números, forma 1) *Sea* $(X_n)_{n\in\mathbb{N}}$ *una sucesión de variables aleatorias independientes e idénticamente distribuidas, tal que* $X_n \xrightarrow{1} X$ *entonces:*

$$\frac{1}{n}\sum_{i=1}^{n} X_i \xrightarrow{p} E(X)$$

Si μ *es la media común de las variables, entonces:*

$$\frac{1}{n}\sum_{i=1}^{n} X_i \xrightarrow{p} \mu$$

Teorema 2.4. (Ley débil de los grandes números, forma 2) *Sea* $(X_n)_{n\in\mathbb{N}}$ *una sucesión de variables aleatorias independientes e idénticamente distribuidas, tal que* $X_n \xrightarrow{1} X$ *entonces:* $\forall\epsilon > 0\,,\, \delta > 0\ \exists\, n_{\epsilon,\delta} \in \mathbb{N}$ *tal que si* $n > n_{\epsilon,\delta}$ *se tiene:*

$$P\left(\left|\frac{1}{n}\sum_{i=1}^{n} X_i - E(X)\right| > \delta\right) < \epsilon$$

$$P\left(\left|\frac{1}{n}\sum_{i=1}^{n} X_i - E(X)\right| \leq \delta\right) > 1 - \epsilon$$

Esto es, a valores grandes de n *el promedio de las variables aleatorias* X_i *se acerca al valor esperado de la variable a la que convergen* $E(X)$ *y si* μ *es la media común de las variables aleatorias* X_n *es posible, en las expresiones anteriores, sustituir* $E(X)$ *con* μ*.*

La demostración de estos hechos de puede consultarse en [18], Pag 211.

En teoría de la información existe un teorema que es el análogo al teorema anterior y es el *teorema de la propiedad de equipartición asintótica*, (AEP) enunciado por Shannon en 1948 en [1], usando la notación propuesta al inicio del trabajo, si se tienen $(X_n)_{n\in\mathbb{N}}$ variables aleatorias discretas, su función de probabilidad de masa conjunta es:

$$p_{X_1,X_2,\ldots,X_n}(x_1, x_2, \ldots, x_n) = P(X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n) = P(X_1, X_2, \ldots, X_n)$$

Con esta notación se tiene el siguiente:

Teorema 2.5. (AEP) *Sean* $(\Omega, \mathcal{F}, \mathbb{P})$ *un espacio de probabilidad,* $(X_n)_{n\in\mathbb{N}}$ *una sucesión de variables aleatorias* i.i.d*, tal que* $X_n \xrightarrow{1} X$ *y sea* $H(X)$ *la entropía de* X*, entonces:*

$$-\frac{1}{n} log_2 P(X_1, X_2, \ldots, X_n) \xrightarrow{p} H(X)$$

Esto es equivalente a tener:

$\forall\epsilon > 0\ \exists\, n_\epsilon \in \mathbb{N}$ *tal que si* $n > n_\epsilon$ *se tiene:*

$$\mathbb{P}\left(\left|-\frac{1}{n} log_2 P(X_1, X_2, \ldots, X_n) - H(X)\right| \leq \epsilon\right) > 1 - \epsilon$$

Demostración. Dado que las variables aleatorias X_i son independientes e idénticamente distribuidas, también los son las variables $log_2P(X_i)$ y si $X_n \xrightarrow{1} X$ entonces por la continuidad del logaritmo, $log_2P(X_n) \xrightarrow{1} log_2P(X)$ y por el teorema *2.3* se tiene:

$$\begin{aligned} -\frac{1}{n}log_2P(X_1, X_2, \ldots, X_n) &= -\frac{1}{n}log_2P(X_1)P(X_2), \ldots, P(X_n) \\ &= -\frac{1}{n}\sum_{i=1}^{n} log_2P(X_i) \xrightarrow{p} -E(log_2P(X)) = H(X) \end{aligned}$$

□

Se deduce del teorema anterior que si:

$$\begin{aligned} -\frac{1}{n}log_2P(X_1, X_2, \ldots, X_n) &\approx H(X) \\ log_2P(X_1, X_2, \ldots, X_n) &\approx -nH(X) \\ P(X_1, X_2, \ldots, X_n) &\approx 2^{-nH(X)} \end{aligned}$$

O dicho de otra forma:

Si $(X_n)_{n\in\mathbb{N}}$ es una sucesión de variables aleatorias independientes e idénticamente distribuidas tal que $X_n \xrightarrow{1} X$ se tiene que:

$$\mathbb{P}\left(\omega \in \Omega :\ P(X_1(\omega) = x_1, X_2(\omega) = x_2, \ldots, X_n(\omega) = x_n) = P(X_1, X_2, \ldots, X_n) = 2^{-n(H(X)\pm\epsilon)}\right) \approx 1$$

Definición 2.6. *(Conjunto de Secuencias Típicas, CST)* Sean $(\Omega, \mathcal{F}, \mathbb{P})$ un espacio de probabilidad, $(X_n)_{n\in\mathbb{N}}$ una sucesión de variables aleatorias *i.i.d* definidas en un conjunto Ω, tal que $X_n \xrightarrow{1} X$, y $\epsilon > 0$. El *conjunto de secuencias típicas, CST* denotado $A_\epsilon^{(n)}$, es el conjunto de secuencias o sucesiones $W^{(n)} = (\omega_1, \omega_2, \ldots, \omega_n) \in \Omega^n$ que cumplen el teorema *2.5* (AEP) para ϵ y n. Es decir, para todo $i \in \{1, 2, \ldots, n\}$:

$$\begin{aligned} A_\epsilon^{(n)} &= \{W^{(n)} :\ \left|-\frac{1}{n}log_2P(X_1(\omega_1), X_2(\omega_2), \ldots, X_n(\omega_n)) - H(X)\right| \leq \epsilon\} \\ &= \{W^{(n)} :\ H(X) - \epsilon \leq -\frac{1}{n}log_2P(X_1(\omega_1), X_2(\omega_2), \ldots, X_n(\omega_n)) \leq H(X) + \epsilon\} \\ &= \{W^{(n)} :\ 2^{-n(H(X)+\epsilon)} \leq P(X_1(\omega_1), X_2(\omega_2), \ldots, X_n(\omega_n)) \leq 2^{-n(H(X)-\epsilon)}\} \end{aligned}$$

Un resultado del teorema AEP es que el conjunto de todas las posibles sucesiones o palabras $W^{(n)}$ se divide en dos conjuntos, el CST donde la entropía de la muestra se acerca a la entropía real y el conjunto de secuencias no típicas que contiene al resto de secuencias.

Unas consecuencias directas del teorema AEP son:

Corolario 2.7. *Sean $(\Omega, \mathcal{F}, \mathbb{P})$ un espacio de probabilidad, $(X_n)_{n\in\mathbb{N}}$ una sucesión de variables aleatorias independientes e idénticamente distribuidas tal que $X_n \xrightarrow{1} X$ entonces:*

1. *Si $W^{(n)} \in A_\epsilon^{(n)}$ entonces $2^{-n(H(X)+\epsilon)} \leq P(W^{(n)}) \leq 2^{-n(H(X)-\epsilon)}$.*

2. *$P(A_\epsilon^{(n)}) > 1-\epsilon$ para n suficientemente grande.*

3. *$(1-\epsilon)2^{n(H(X)-\epsilon)} \leq \left|A_\epsilon^{(n)}\right| \leq 2^{n(H(X)+\epsilon)}$*

Las partes 1) y 2) se deducen directamente al reescribir el AEP en términos del CST, para la parte 3), basta con reescribir:

$$1 = \sum_{W^{(n)}\in\Omega^n} P(W^{(n)}) \geq \sum_{W^{(n)}\in A_\epsilon^{(n)}} P(W^{(n)}) \geq \sum_{W^{(n)}\in A_\epsilon^{(n)}} 2^{-n(H(X)+\epsilon)} = 2^{-n(H(X)+\epsilon)} \left|A_\epsilon^{(n)}\right|$$

Luego

$$\left|A_\epsilon^{(n)}\right| \leq 2^{n(H(X)+\epsilon)}$$

Ahora como:

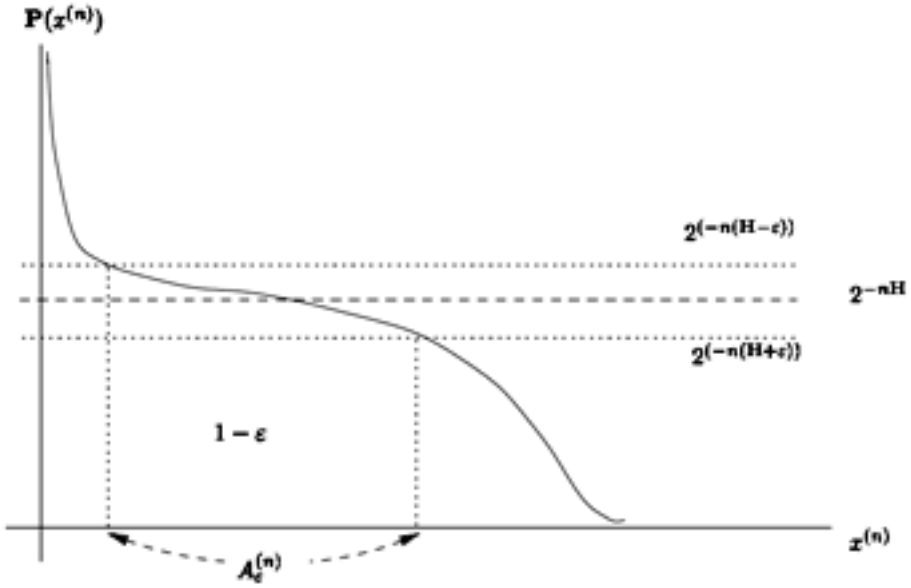
$$2^{-n(H(X)-\epsilon)} \left|A_\epsilon^{(n)}\right| = \sum_{W^{(n)}\in A_\epsilon^{(n)}} 2^{-n(H(X)-\epsilon)} \geq P(A_\epsilon^{(n)}) > 1-\epsilon$$

Se concluye:

$$(1-\epsilon)2^{n(H(X)-\epsilon)} \leq \left|A_\epsilon^{(n)}\right| \leq 2^{n(H(X)+\epsilon)}$$

Con este corolario se concluye que las secuencias típicas tienen probabilidad de ocurrencia casi 1, todos sus elementos son equiprobables y el número de elementos en $A_\epsilon^{(n)}$ es aproximadamente $2^{nH(X)}$.

En base a lo anterior, cada propiedad de las secuencias del CST será verdadera con alta probabilidad esto se ilustra con el siguiente diagrama:



Entonces la probabilidad de que una secuencia de Ω^n sea una secuencia típica es igual a $\frac{1}{2nH(X)}$, si n es suficientemente grande, cabe esperar que prácticamente todas las salidas de $\mathbb{X}^n$ sean secuencias típicas. Dicho de otro modo, es de esperar que casi todas las secuencias emitidas o generadas sean típicas y que el número de secuencias no típicas sea despreciable.

Definición 2.8. Dado un alfabeto $\mathbb{A}$, el conjunto de todas las palabras que se pueden formar con elementos de $\mathbb{A}$ se denota $\mathbb{A}^*$ y es independiente de la longitud de las palabras. Esto es:

$$\mathbb{A}^* = \bigcup_{n\in\mathbb{N}} \mathbb{A}^n$$

Definición 2.9. Un *código* sobre un alfabeto $\mathbb{A}$ es un subconjunto C de $\mathbb{A}^*$, (está formado por palabras de alfabeto). A los elementos del código C se les llama palabras de código. *(codeword, en la literatura inglesa)* y al alfabeto $\mathbb{A}$ se le llama *alfabeto código* . El número de elementos del código C, que usualmente es finito, se denota por $|C|$ y es el *tamaño del código*. Si el alfabeto $\mathbb{A}$ tiene q elementos, es decir $|\mathbb{A}| = q$, se dice que C es un *código q-ario sobre* $\mathbb{A}$.

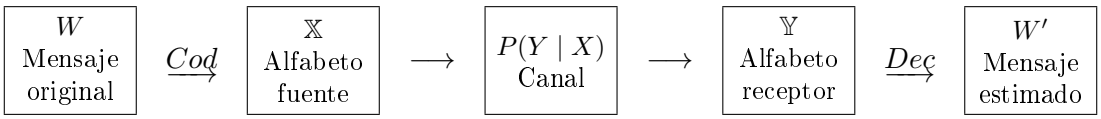
Por ejemplo, si $\mathbb{A} = \{0, 1\}$, tenemos códigos binarios, así $C = \{0100, 0010, 0011\}$ es un código binario.

Definición 2.10. Si C es un código cuyas palabras tienen todas la misma longitud n, se dice que C es un *código de longitud fija* o un *código de bloques* y n es la *longitud del código* C.

En el ejemplo anterior C es un código de bloques de longitud 4.
Si $C = \{001, 1011, 10\}$, C no es un código de bloques y no hay longitud del código.

Si C es un código de longitud n y tamaño m, se dice que C es un (n, m)-*código*.
En este caso, $C = \{0100, 0010, 0111\}$ es un $(4, 3)$-código.

Retomando el esquema inicial del trabajo:



Nos falta brindar algunas definiciones adicionales, para mostrar todo el proceso.

La codificación es el proceso a través del cual se ponen en correspondencia dos alfabetos finitos, el alfabeto fuente $\mathbb{X}$ y el alfabeto código $\mathbb{A}$, de tal forma que a los símbolos del alfabeto fuente se le hacen corresponder símbolos del alfabeto código. Este paso es previo a la transmisión o al almacenamiento, la *eficiencia de la codificación* consiste, en reducir la *redundancia* o términos redundantes, de los mensajes expresados mediante el alfabeto fuente. Dicho proceso está a cargo de una función, llamada *función de codificación*, definida a continuación.

Definición 2.11. *(Función de codificación, Cod.)* Dado el alfabeto fuente $\mathbb{X}$ y un código C sobre el alfabeto $\mathbb{A}$, la *función de codificación Cod* es una aplicación f tal que:

$$\begin{array}{ccc} f : \mathbb{X} & \longmapsto & C \\ X & \longmapsto & f(X) = Y \\ (X_1 X_2 \dots X_n) & \mapsto & (f(X_1) f(X_2) \dots f(X_n)) \end{array}$$

Donde $\mathbb{X}$ es el alfabeto que contiene la información o palabras que se quieren codificar.

En el caso ideal, sin errores de transmisión se tendría que f es biyectiva y el código es *descifrable*.

Ejemplo 2.12. Dado el alfabeto $\mathbb{X} = \{a, b, c\}$ una función de codificación sería:

$$\begin{array}{ccc} a & \longmapsto & f(a) = 0010 \\ b & \longmapsto & f(b) = 0100 \\ c & \longmapsto & f(c) = 0111 \end{array}$$

Luego:

$$(babc) \longmapsto (0100001001000111)$$

El código ASCII estándar usa palabras de 7 bits, (longitud 7), luego contempla $2^7 = 128$ palabras, luego es un (7,128)-código.

Un canal que introduce errores en la transmisión se llama *canal ruidoso*, en este caso, es posible que la palabra recibida no sea la palabra enviada, los errores comunes son el intercambio de símbolos, (1 por 0), los borrones, (aparece ? en lugar de algún símbolo) y las eliminaciones, (caso en el que no llega el símbolo, ni hay rastro de su perdida). Acá el ruido se representa por la matriz de transición del canal T.

Luego es posible que una palabra enviada X sea recibida como una palabra $\hat{X}$, con $X \neq \hat{X}$ lo que genera el error en la transmisión.

Se requiere de una función $g(x)$ que invierta o anule el proceso realizado por la función codificadora $f(x)$, es decir, que decodifique las palabras al llegar del canal antes de ser recibidas por el receptor.

Definición 2.13. *(Función decodificadora, Dec.)* Dado el alfabeto receptor $\mathbb{Y}$ y un código C sobre el alfabeto $\mathbb{A}$, la *función decodificadora Dec* es una aplicación g tal que:

$$\begin{array}{ccc} g : C & \longmapsto & \mathbb{Y} \\ Z & \longmapsto & g(Z) = \hat{X} \\ (Z_1 Z_2 \dots Z_n) & \mapsto & (g(Z_1)g(Z_2)\dots g(Z_n)) \end{array}$$

Donde $Z_i = f(X_i)$, note que no es Y_i por la presencia del ruido al pasar por el canal, de no tener ruido $Z_i = Y_i$.

Con lo anterior, lo que se desea es que $X = \hat{X}$, y las funciones f (Cod) y g $Dec)$ se escojan de tal forma que se minimice el *error de decodificación*, es decir:

$$P(X \neq \hat{X}) < \epsilon$$

En la practica Cod y Dec deberían ser inversas la una de la otra el problema es cuando, en el medio del proceso, surgen los errores de transmisión y $g(f(X)) = \hat{X}$ con $X \neq \hat{X}$.
Para detectar y corregir los errores de transmisión, a los códigos se les añaden datos adicionales o *redundancia*, lo que ocasiona que se pierda *eficiencia*, pues se gastan bits de más en el envío del mensaje.

Estos son los códigos con *detección de errores* que se usan para recuperar la información que llego incorrectamente y prevenir las fallas del canal. Por ejemplo, se usan en los discos compactos para recuperar información a pesar de que el disco este rayado y se produzcan fallas de lectura.

Luego para crear un código con detección de errores se debe:

1. Para detectar errores: El código debe estar formado por palabras de tal forma que si a una se le cambia un solo símbolo, la palabra resultante no sea del código, para así saber que se ha producido un error.

2. Para corregir errores: Como no se conoce la palabra enviada, pues el canal es el único medio de comunicación entre las partes, se toma la palabra recibida y se compara con todas las palabras del código asignándole la palabra con la que difiera en menos símbolos.

Lo anterior nos lleva a introducir una idea que nos permita diferenciar entre secuencias o palabras, así se tiene la siguiente:

Definición 2.14. *La distancia de Hamming* entre dos palabras X y Y de igual longitud, es la cantidad de bits o símbolos distintos que tengan entre si, se denota $d_H(X,Y)$.

Por ejemplo, si $X = 1011101$ y $Y = 1001001$, $d_H(X,Y) = 2$.

Entre las propiedades de la distancia de Hamming tenemos:

1. $d_H(X,Y) = d_H(Y,X)$.

2. $d_H(X,Y) = 0$ si y solo si $X = Y$.

3. $d_H(X,Z) + d_H(Z,Y) \leq d_H(X,Y)$

Es posible usar el teorema *2.3* (AEP) para poder tomar de todo el conjunto de posibles palabras o secuencias en Ω^n, solo aquellas a la que se les da más uso, (o sean más lógicas en el contexto) y formar así el CST, sobre este conjunto se hace la codificación.

Este proceso se conoce como *Compresión AEP* y nos da la idea de cuantos bits se usan para lograr una codificación adecuada.

Lo que se hace es:

1. Ordenar las secuencias o palabras de $\Omega^n = \mathbb{A}^*$.

2. A cada palabra se le asigna un indice n, notado $W^{(n)}$.

3. Dado que $\left|A_\epsilon^{(n)}\right| \leq 2^{n(H(X)+\epsilon)}$, solo se necesitan $n(H(X)+\epsilon) \leq n(H(X)+\epsilon)+1$ bits en la codificación del CST.

4. Para el conjunto $\bar{A}_\epsilon^{(n)} = \Omega^n - A_\epsilon^{(n)}$, hay libertad en el uso de bits. pues $P(\bar{A}_\epsilon^{(n)}) \leq \epsilon$, para codificarlo, se usan tantos bits como si se deseara codificar Ω^n, es decir, $nlog_2|\Omega| \leq nlog_2|\Omega| + 1$ bits. Se suma 1 bits por si el resultado es un decimal.

Para diferenciar si las palabras están o no en el CST y de ahí su forma de codificar, en ocasiones se antepone un bit inicial que puede indicar si la palabra está o no en el CST. Estos bits ubicados en la parte inicial reciben el nombre de *prefijos*.

2.2. Los teoremas de Shannon y la codificación

Con las definiciones previas, estamos en capacidad de enunciar uno de los primeros resultados clásicos de Shannon en teoría de la información:

Teorema 2.15. (Primer Teorema de Shannon, 1948, [1]) *Sean* $(\Omega, \mathcal{F}, \mathbb{P})$ *un espacio de probabilidad,* $(X_n)_{n\in\mathbb{N}}$ *una sucesión de variables aleatorias* i.i.d *tal que* $X_n \xrightarrow{1} X$ *sobre un alfabeto finito* $\mathbb{A}$ *y* $\epsilon > 0$. *Entonces para* n *suficientemente grande, existe un código tal que si* $l(W^{(n)})$ *es la longitud de las palabras código, se tiene que:*

$$E\left(\frac{1}{n}l(W^{(n)})\right) \leq H(X) + \epsilon$$

Es decir, existe un código biunivoco en donde el valor esperado de la longitud de las palabras del código, por caracter de $W^{(n)}$*, es menor que la entropía de la fuente.*

Demostración. Sea n suficientemente grande como para que $P(A_\epsilon^{(n)}) > 1 - \epsilon$, entonces con $A^{(n)} = A_\epsilon^{(n)} \cup \bar{A}_\epsilon^{(n)}$ se tiene:

$$\begin{aligned}
E\left(l(W^{(n)})\right) &= \sum_{W^{(n)}\in A^{(n)}} P(W^{(n)})l(W^{(n)}) \\
&= \sum_{W^{(n)}\in A_\epsilon^{(n)}} P(W^{(n)})l(W^{(n)}) + \sum_{W^{(n)}\in \bar{A}_\epsilon^{(n)}} P(W^{(n)})l(W^{(n)}) \\
&\leq \sum_{W^{(n)}\in A_\epsilon^{(n)}} P(W^{(n)})(n(H(X)+\epsilon)+2) + \sum_{W^{(n)}\in \bar{A}_\epsilon^{(n)}} P(W^{(n)})(nlog_2|\mathbb{A}|+2) \\
&= (n(H(X)+\epsilon)+2)\underbrace{P(A_\epsilon^{(n)})}_{\leq 1} + (nlog_2|\mathbb{A}|+2)\underbrace{P(\bar{A}_\epsilon^{(n)})}_{<\epsilon} \\
&\leq nH(X) + \underbrace{n\epsilon + 2 + n\epsilon log_2|\mathbb{A}| + 2\epsilon}_{n\acute{\epsilon}} = n(H(X)+\acute{\epsilon})
\end{aligned}$$

Luego:

$$\frac{E\left(l(W^{(n)})\right)}{n} \leq H(X) + \acute{\epsilon}$$

Con lo anterior se concluye que la longitud media de un código puede ser llevada cerca de la entropía de la fuente pero nunca por debajo de ella, pues:

$$\frac{E\left(l(W^{(n)})\right)}{n} = H(X) + \acute{\epsilon}$$

Entonces:

$$H(X) \leq \frac{E\left(l(W^{(n)})\right)}{n}$$

□

Este es un teorema de tipo existencial, pues nos dice que el código que cumple esa correspondencia biunivoca existe siempre, pero no nos dice como construirlo.

Ejemplo 2.16. Sean $(X_n)_{n\in\mathbb{N}}$ una sucesión de variables aleatorias *i.i.d* tal que $X_n \xrightarrow{1} X$ sobre un alfabeto finito $\mathbb{A} = \{0,1\}$, donde $P_1 = P(X_i = 1) = 0{,}6$ y $P_2 = P(X_i = 0) = 0{,}4$. Entonces:

$$H(X) = H(0{,}6\,,\,0{,}4) = (-0{,}6)log_2(0{,}6) - (0{,}4)log_2(0{,}4) = 0{,}971$$

Con secuencias de $n = 25$ elementos y $\epsilon = 0{,}1$, el CST según la definición *2.6*, cumple:

$$\begin{array}{rcccc} H(X) - \epsilon & \leq & -\frac{1}{n}log_2P(X_1, X_2, \ldots, X_n) & \leq & H(X) + \epsilon \\ 0{,}971 - 0{,}1 & \leq & -\frac{1}{n}log_2P(X_1, X_2, \ldots, X_n) & \leq & 0{,}971 + 0{,}1 \\ 0{,}871 & \leq & -\frac{1}{n}log_2P(X_1, X_2, \ldots, X_n) & \leq & 1{,}071 \end{array}$$

Según la tabla [1] del anexo, se tiene que las secuencias con $11 \leq k \leq 19$ unos, forman el conjunto $A_\epsilon^{(n)}$, y la probabilidad del CST es:

$$P(A_\epsilon^{(n)}) = P(11 \leq k \leq 19) = P(k \leq 19) - P(k \leq 10) = 0{,}97063 - 0{,}03439 = 0{,}93624$$

Tomando:

$$n\acute{\epsilon} = n\epsilon + 2 + n\epsilon log_2|\mathbb{A}| + 2\epsilon = 0{,}288$$

Según el teorema *2.15 (primer teorema de Shannon)*, se tiene:

$$P(A_\epsilon^{(n)}) = 0{,}93624 > 1 - 0{,}288 = 0{,}712$$

Así, según la tabla [1], existe un código donde $|A_\epsilon^{(n)}| = 26366510$ y si la entropía de la fuente binaria está cerca al máximo de 1 bit, (en este caso 0,971), el algoritmo de compresión no es muy efectivo, pues se necesitan:

$$n(H(X) + \epsilon) + 1 = 25(0{,}971 + 0{,}1) + 1 = 28\,bits$$

Para codificar cada secuencia del CST, y:

$$nlog_2|X| + 1 = 25 + 1 = 26\,bits$$

Para codificar la secuencias no típicas.

Los resultados propuestos por Shannon, acerca de la existencia de códigos cuya longitud promedio se acerca a la entropía, nos lleva a revisar a continuación la probabilidad del error al decodificar y como evitar dichos errores.

El código utilizado para evitar los errores de transmisión, es el *código de repetición.*

Definición 2.17. Sea K un canal binario simétrico con probabilidad de error $0 < p < \frac{1}{2}$, con $\mathbb{X} = \mathbb{Y} = \{0, 1\}$ los alfabetos fuente y receptor respectivamente. Un *código de repetición*, es una función f_n que cumple:

$$\begin{array}{rcl} f_n : \mathbb{X} & \longmapsto & \mathbb{Y}^n \\ b & \longmapsto & f_n(b) = (b, b, \ldots b) = (bb \ldots b) \end{array}$$

Es decir, repite b veces el bit b.

Definición 2.18. Una función decodificadora g_n cumple la *regla de decodificación por lógica mayoritaria*, si cumple:

$$\begin{array}{rcl} g_n : \mathbb{Y}^n & \longmapsto & \mathbb{Y}^n \\ Z = (b_1, b_2, \ldots, b_n) & \longmapsto & g_n(Z) = \begin{cases} (0, 0, \ldots 0) & si \quad mas\ de\ [[\frac{n+1}{2}]]\ de\ los\ b_i\ son\ ceros. \\ (1, 1, \ldots 1) & si \quad mas\ de\ [[\frac{n+1}{2}]]\ de\ los\ b_i\ son\ unos. \end{cases} \end{array}$$

Acá es posible ajustar la cantidad de términos repetidos requeridos para estimar el valor enviado.

Ejemplo 2.19. Sea f_3 un código de repetición 3:

$$\begin{array}{rcl} f_3 : \mathbb{X} & \longmapsto & \mathbb{Y}^3 \\ b & \longmapsto & f_3(b) = (b, b, b) = (bbb) \end{array}$$

En este caso la regla de decodificación por lógica mayoritaria es:

$$\begin{array}{rcl} g_3 : \mathbb{Y}^3 & \longmapsto & \mathbb{Y}^3 \\ Z = (b_1, b_2, b_3) & \longmapsto & g_3(Z) = \begin{cases} (0, 0, 0) = (000) & si \quad mas\ de\ 2\ de\ los\ b_i\ son\ ceros. \\ (1, 1, 1) = (111) & si \quad mas\ de\ 2\ de\ los\ b_i\ son\ unos. \end{cases} \end{array}$$

De esta forma el decodificador le entrega al receptor el bit $\acute{X} = 0$, o, $\acute{X} = 1$ según sea $\acute{Y} = (000)$, o $\acute{Y} = (111)$.

Denotemos $P(E_g)$ a la probabilidad de error en la decodificación, en este caso:

$$P(E_{g_3}) = P(\textit{2, o 3 errores}) = \left(\begin{array}{c} 3 \\ 2 \end{array} \right) P^2(1 - P) + \left(\begin{array}{c} 3 \\ 3 \end{array} \right) P^3 = 3P^2 - 2P^3$$

Dado que $0 < P < \frac{1}{2}$, se tiene que:

$$P(E_{g_3}) = 3P^2 - 2P^3 < P \longleftrightarrow 2P^2 - 3P + 1 > 0$$

Pues:

$$2P^2 - 3P + 1 = 2(P - \frac{1}{2})(P - 1) > 0$$

Luego la regla de decodificación usada ha disminuido el error en la recepción, con un costo de multiplicar por 3 la longitud del mensaje.

A continuación se define el conjunto de errores y la probabilidad de error en la decodificación.

Definición 2.20. Dada una regla de decodificación:

$$\begin{aligned} g_n : C &\longmapsto \mathbb{Y}^n \\ Z &\longmapsto g_n(Z) = \acute{Y} \end{aligned}$$

Donde Y es la palabra del código enviada por el canal y $\acute{Y}$ es la palabra recibida, el *conjunto de errores* E_{g_n} es:

$$E_{g_n} = \{(Y, Z) \in \mathbb{Y}^n \times C | g_n(Z) = \acute{Y} \neq Y\}$$

Donde C es (n, m)-código y $\mathbb{Y}$ es el alfabeto del canal.

Al transmitir la palabra del código Y, la *probabilidad de error en la decodificación* es:

$$P(E_{g_n}|Y) = \sum_{Z \notin g_n^{-1}(Y)} P(Z|Y)$$

Lema 2.21. *Con las condiciones anteriores, la* probabilidad media de error $P(E_{g_n})$ *es:*

$$P(E_{g_n}) = 1 - \sum_{Z \in \mathbb{Y}^n} P(g_n(Z), Z)$$

Demostración. Dado que:

$$\begin{aligned} P(E_{g_n}) &= \sum_{Y \in C} P(E_{g_n}|Y)P(Y) = \sum_{Y \in C} \sum_{Z \notin g_n^{-1}(Y)} P(Z|Y)P(Y) \\ &= \sum_{(Y,Z) \in E_{g_n}} P(Y, Z) = 1 - \sum_{Z \in \mathbb{Y}^n} P(g_n(Z), Z) \end{aligned}$$

Donde esta probabilidad depende de la distribución de la fuente. □

Definición 2.22. Una regla de decodificación g_n es un *decodificador optimo*, si cumple:

$$P(g_n(Z), Z) = \max_{Y \in C} P(Y, Z) \quad \forall Z \in \mathbb{Y}^n$$

Una consecuencia directa del lema 2.21 es el siguiente:

Corolario 2.23. *Dado un canal $K = (\mathbb{X}, \mathbb{Y}, T)$ y un código C, el mínimo de la probabilidad media de error, $P(E_{g_n})$ para toda regla de decodificación g_n se tiene en un decodificador optimo.*

Definición 2.24. La probabilidad máxima de error asociada a una regla de decodificación g_n es:

$$P^{max}(E_{g_n}) = \max_{Y \in C} P(E_{g_n}|Y)$$

Corolario 2.25. *La probabilidad máxima de error asociada a una regla de decodificación g_n, $P^{max}(E_{g_n})$, es una cota superior de la probabilidad de error $P(E_{g_n})$, para todas las variables aleatorias X de la fuente $\mathbb{X}$.*

Definición 2.26. Una regla de decodificación g_n es de *máxima verosimilitud*, si cumple:

$$P(Z|g_n(Z)) = \max_{Y \in C} P(Z|Y) \quad \forall Z \in \mathbb{Y}^n$$

Si la distribución X de la fuente es uniforme, entonces en una regla de decodificación g_n coinciden las propiedades de decodificador optimo y de máxima verosimilitud, es decir:

$$P(g_n(Z), Z) = \max_{Y \in C} P(Y, Z) = \max_{Y \in C} P(Z|Y) = P(Z|g_n(Z))$$

Se ilustra el manejo de las reglas de decodificación y sus propiedades, con el siguiente:

Ejemplo 2.27. Sea $\mathbb{X} = \{0, 1\}$ una fuente binaria que emite bits según una variable aleatoria X, donde $P(X = 1) = 0{,}3$.
El alfabeto fuente es $\mathbb{A}$ es:

$$\mathbb{A} = \{X_0, X_1, X_2, X_3\} = \{(0,0), (0,1), (1,0), (1,1)\}$$

Este se transmite a través de un canal binario simétrico CBS, K con probabilidad de salida errónea $p = 0{,}2$.

La función codificadora f_2 opera añadiendo un bit a cada secuencia según la formula:

$$\begin{aligned} f_2 : \mathbb{X}^2 &\longmapsto C \subset \mathbb{Y}^3 \\ X = (b_1, b_2) &\longmapsto f_2(X) = Y = (b_1, b_2, b_1 + b_2) = (b_1 b_2 b_1 + b_2) \end{aligned}$$

De forma matricial:

$$Y = (b_1, b_2) = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}$$

Así el código resultante $C \subset \mathbb{Y}^3$ es un $(3, 4)$-código binario pues:

$$C = \{(000), (011), (101), (110)\}$$

La matriz de transición $T_{Z|Y} = P(Z|Y)$ es:

Y \ Z	(000)	(001)	(010)	(011)	(100)	(101)	(110)	(111)
(000)	0,512	0,128	0,128	0,032	0,128	0,032	0,032	0,008
(011)	0,032	0,128	0,128	0,512	0,008	0,032	0,032	0,128
(101)	0,032	0,128	0,008	0,032	0,128	0,512	0,032	0,128
(110)	0,032	0,008	0,128	0,032	0,128	0,032	0,512	0,128

La función de decodificación por máxima verosimilitud g_{mv}, decodifica cada $Z \in \mathbb{Y}^3$ como la $Y \in C$ con *mayor probabilidad* en la columna correspondiente de la matriz $T_{Z|Y}$:

$$\begin{array}{llll} g_{mv}(000) = (000) & & g_{mv}(100) = (000) \\ g_{mv}(001) = (000) & & g_{mv}(101) = (101) \\ g_{mv}(010) = (000) & & g_{mv}(110) = (110) \\ g_{mv}(011) = (011) & & g_{mv}(111) = (011) \end{array}$$

Dado que $(p_0, p_1, p_2, p_3) = (0,44,\ 0,21,\ 0,21,\ 0,09)$, la matriz de probabilidad conjunta $P(Y, Z)$ es:

Y \ Z	(000)	(001)	(010)	(011)	(100)	(101)	(110)	(111)
(000)	0,251	0,063	0,063	0,016	0,063	0,016	0,016	0,004
(011)	0,007	0,027	0,027	0,107	0,002	0,007	0,007	0,027
(101)	0,007	0,027	0,002	0,007	0,027	0,107	0,007	0,027
(110)	0,003	0,001	0,012	0,003	0,012	0,003	0,046	0,012

Aquí la función o regla de decodificación que es un decodificador optimo g_{do}, decodifica cada $Z \in \mathbb{Y}^3$ como la $Y \in C$ con *mayor probabilidad* en la columna correspondiente de la matriz $P(Y, Z)$:

$$\begin{array}{lcllcl} g_{do}(000) & = & (000) & g_{do}(100) & = & (000) \\ g_{do}(001) & = & (000) & g_{do}(101) & = & (101) \\ g_{do}(010) & = & (000) & g_{do}(110) & = & (110) \\ g_{do}(011) & = & (011) & g_{do}(111) & = & (011) \end{array}$$

A continuación se enuncian unas definiciones adicionales, para poder enunciar el *segundo teorema de Shannon.*

Definición 2.28. Un conjunto C es un (n, m)-código q-ario, si esta formado por m palabras de código de longitud n, sobre un alfabeto de q elementos.

Retomando la definición *1.27*, sobre la tasa de información, la adaptamos al caso discreto:

Definición 2.29. Dado un (n, m)-código q-ario C, se llama *tasa de información o ratio* del código, denotado $R(C)$ a la expresión:

$$R(C) = \frac{log_q m}{n}$$

El valor $1 - R(C)$ se llama *tasa de redundancia.*

Teorema 2.30. (Segundo Teorema de Shannon, 1948 [1])
Sea K un canal con ruido, con capacidad $\mathbb{C}(\mathbb{K})$ y sea $R \in \mathbb{R}$, tal que $0 < R < \mathbb{C}(\mathbb{K})$. Existe una sucesión $(C_n, g_n)_{n\in\mathbb{N}}$ de (n, m_n)-códigos q-arios y de funciones decodificadoras tales que:

$$1)\ \ R \leq R(C_n) = \frac{log_q m_n}{n}$$

$$2)\ \ \lim_{n\to\infty} P^{max}(E_{g_n}) = 0$$

La prueba de este resultado fue presentada por Shannon de forma no tan rigurosa en [1], años después fue mejorada y reconstruida en [2] y en [17]. El principal problema que presentan las pruebas, tanto la de Shannon como las posteriores, es que se prueba la existencia del código que satisface las condiciones dadas, más no se construye, ni se brindan ideas claras de como crearlo, luego se dan cotas para la capacidad, pero no se dan formulas para encontrarla, así que esta debe ser calculada de forma independiente según el canal, siendo este el principal problema que se tiene en el manejo y modelado de nuevos canales.

Capítulo 3

El Canal de Eliminación: Definiciones

3.1. Ideas principales

Hasta el momento, en virtud a las afirmaciones que brindan los teoremas de Shannon, se han comentado dos problemas. El primero es relativo a las fuentes sin memoria y a formas eficientes de convertir las palabras de la fuente en palabras de código, de tal manera que la longitud promedio de las palabras del código sea cercana, por arriba, a la entropía de la fuente.

El segundo se refiere al problema del transporte de las palabras del código a través de los canales y la capacidad de estos para transmitirlas. En los canales con errores de transmisión cada palabra del código, al ser codificada como una tira de bloques binarios, tiene una probabilidad de modificarse, haciendo que el bloque sea transformado en otro, modificándose la información transmitida.

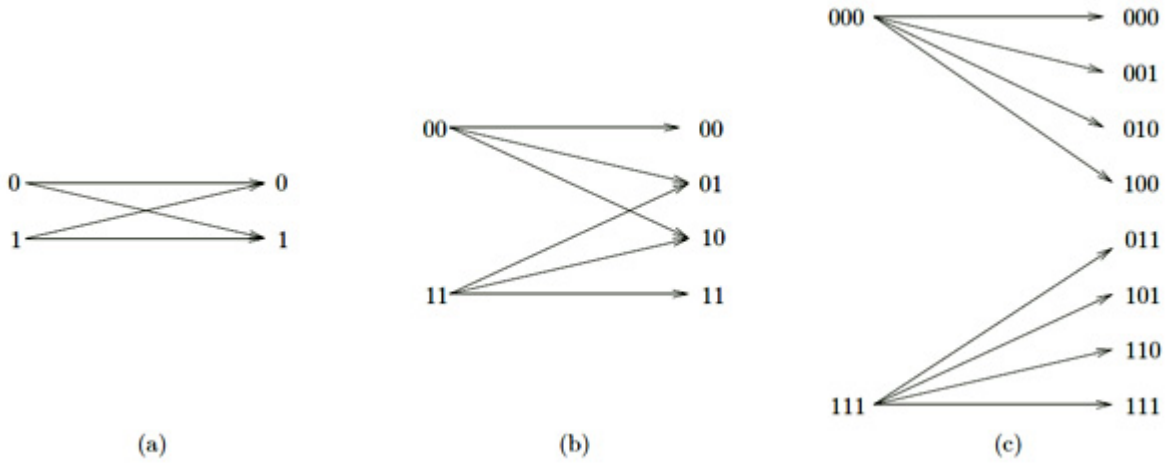
Para prevenir el error en la recepción, se acude a codificar la palabra a transmitir usando una función codificadora por repetición, sin embargo, la idea central será *no repetir bit a bit en igual cantidad*, sino considerar la palabra por *bloques o tiras de bits* y repetir los bloques en ciertas proporciones, con el objetivo de minimizar el ingreso de esa información redundante, que a la larga afecta la eficiencia en la transmisión.

Se ilustra esta idea en el siguiente ejemplo:

Ejemplo 3.1. *Repetición de bits en un CBS*

Sean $\mathbb{X} = \{0, 1\}$ y $\mathbb{Y} = \{0, 1\}$ los alfabetos binarios fuente y receptor respectivamente. En este caso las probabilidades son: llegada correcta $1 - p$ y llegada errónea p, supongamos que se tienen dos bits a transmitir 0 y 1, a través de un CBS.

Estos se envían a través del CBS directamente, o se repiten 2 o 3 veces, según el diagrama:



Se envía información $I(X)$ de tres formas:

(a) Se envía un bit y p es la probabilidad de transformarse en el bit opuesto.

(b) Cada bit se transmite 2 veces (2 repeticiones), luego cada tira tiene 3 opciones reales de llegada, suponiendo muy baja la probabilidad de que ambos valores se cambien. Aquí el problema radica en que hay 2 salidas 01 y 10 que podrían corresponder a cualquiera de las 2 tiras, luego la recepción podría ser ambigua y la repetición no es muy efectiva.

(c) Cada bit se repite 3 veces (3 repeticiones), si se supone muy baja la probabilidad de que 2 o más bits en una misma tira se modifiquen, se observa que los resultados o posibles salidas del canal, no se sobreponen y con una regla de decodificación por lógica mayoritaria es posible determinar si el bit enviado fue 0 o 1. En este caso la repetición fue efectiva.

Esta forma de codificar tiene un costo, que en este ejemplo es reducir en $\frac{1}{3}$ la tasa de transmisión, frente a la tasa de generación inicial.

Definición 3.2. Dada una palabra de código o secuencia binaria X, un *run* r_i, es una tira de términos binarios iguales consecutivos, luego una palabra codificada está formada por varios runs. La *longitud* de un run r_i, denotada $l(r_i)$, es la cantidad de términos repetidos en el i-ésimo run r_i. El *vector de runs*, denotado $V_R(X)$, es un vector en forma de potencias, donde cada base es el termino que caracteriza cada run y cada exponente es la longitud del respectivo run.

Ejemplo 3.3. Si $X = (000111101000)$ está palabra esta formada por 5 runs donde:

$$r_1 = 000 \quad r_2 = 1111 \quad r_3 = 0 \quad r_4 = 1 \quad r_5 = 000$$

$$l(r_1) = 3 \quad l(r_2) = 4 \quad l(r_3) = 1 \quad l(r_4) = 1 \quad l(r_5) = 3$$

$$V_R(X) = (0^{l(r_1)}, 1^{l(r_2)}, 0^{l(r_3)}, 1^{l(r_4)}, 0^{l(r_5)}) = (0^3, 1^4, 0^1, 1^1, 0^3)$$

Queremos entonces diseñar una función codificadora por repetición que no repita bit a bit sino al tomar una palabra, identifique los runs en ella y los repita a cada uno de ellos en bloque, de tal forma que la información sea protegida ante errores de sincronización.

En el ejemplo anterior, se asumía que solo un bit se modificaba y podía incluso conocerse la posición de dicho bit en la palabra, luego en una palabra de longitud n, podíamos suponer que una proporción np de los bits son modificados.

Pensando en la decodificación, ¿en cuantas palabras de longitud n podría convertirse una palabra de longitud n si se modifican np bits si estas atraviesan un CBS?. Este resultado se puede analizar usando el siguiente:

Lema 3.4. *(Aproximación de Stirling) Si $n \in \mathbb{N}$ y es lo suficientemente grande, entonces:*

$$ln(n!) \approx nln(n) - n$$

es decir:

$$n! \approx n^n e^{-n}$$

Con este lema es posible reformular la estimación inicial de las posibles palabras de longitud n que saldrían al modificar np bits, según el siguiente:

Teorema 3.5. *Sea K un canal binario simétrico, (CBS), $\mathbb{X} = \{0,1\}$ y $\mathbb{Y} = \{0,1\}$ los alfabetos binarios fuente y receptor respectivamente. Las probabilidades son: llegada correcta $1-p$ y llegada errónea p. La cantidad de posibles palabras de longitud n que pueden salir al atravesar el CBS K es aproximadamente $2^{nH(Y|X)}$.*

Demostración. Una secuencia de longitud n puede originar la combinación de n tomadas de np secuencias, esto es $\binom{n}{np}$, usando el lema anterior tenemos:

$$\begin{aligned}
\binom{n}{np} &= \frac{n!}{(n-np)!(np)!} \approx \frac{n^n e^{-n}}{(n-np)^{n-np}\, e^{-n+np}\, (np)^{np}\, e^{-np}} \\
&\approx \frac{n^n e^{-n}}{n^n\, n^{-np}\, (1-p)^{n-np}\, e^{-n} e^{np}\, n^{np}\, p^{np}\, e^{-np}} \approx (1-p)^{-n(1-p)}\, p^{-np} \\
&\approx 2^{log_2(1-p)^{-n(1-p)}\, p^{-np}} \approx 2^{log_2(1-p)^{-n(1-p)} + log_2 p^{-np}} \\
&\approx 2^{-n(1-p)log_2(1-p) - nplog_2 p} \approx 2^{n(-plog_2 p - (1-p)log_2(1-p))} \\
&\approx 2^{nH(Y|X)}
\end{aligned}$$

Donde $H(Y \mid X)$ es la entropía del CBS. □

Con este resultado podemos estimar que cada palabra de n bits se podría eventualmente convertir en una de $2^{nH(Y|X)}$ palabras a la salida del CBS, por otro lado, en virtud al corolario *2.7* se tendrían aproximadamente $2^{nH(X)}$ palabras o secuencias típicas, luego si hay M conjuntos disjuntos con $2^{nH(Y|X)}$ palabras, siendo todas estas secuencias típicas se tendría:

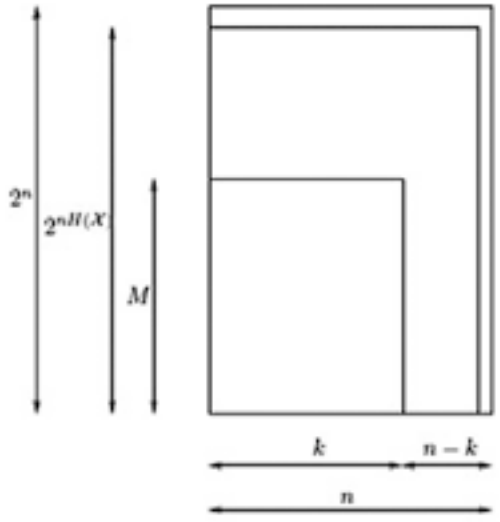
$$\left|A_\epsilon^{(n)}\right| \approx 2^{nH(X)} = M\, 2^{nH(Y|X)}$$

Es decir:

$$M = \frac{2^{nH(X)}}{2^{nH(Y|X)}} = 2^{nH(X)-nH(Y|X)} = 2^{n(H(X)-H(Y|X))}$$

Dado que M representa los conjuntos disjuntos en la salida, cada uno en correspondencia con una de las posibles entradas, se observa que máximo M entradas o palabras podrán ser reconocidas sin error en la salida, o mejor dicho, con un error que es más pequeño a medida que la longitud n de la palabra es más grande.

En el caso de los canales con errores de sincronización, es posible que no se puedan transmitir sin errores las 2^n palabras de longitud n que pudiera generar el alfabeto, es más ni siquiera las $2^{nH(X)}$ palabras más factibles, sino solo M palabras libres de errores con $M < 2^{nH(X)} < 2^n$, así es posible construir esas M palabras con un numero $k < n$ de bits, según se ilustra en el siguiente diagrama:



Así si $M = 2^k$, se tiene que $k = log_2 M < n$ y con esto se tendría la posibilidad de armar las palabras con no más de k bits para que cada una de ellas degenere, producto de los errores, en secuencias pertenecientes a un mismo conjunto, teniendo en cuenta que deberán transmitirse efectivamente por el canal los n bits, para garantizar la existencia de los M conjuntos disjuntos a la salida.

Luego, para proteger contra los errores del canal CBS la información que se puede enviar en tiras de k bits, debería agregarse a cada tira una *redundancia* de $n-k$ bits y transmitir, en definitiva los n bits resultantes.

En cuanto al CBS, para determinar k basta con tomar:

$$k = log_2 M = log_2 2^{n(H(X)-H(Y|X))} = n(H(X) - H(Y \mid X))$$

y con esto, usando el corolario *1.24* se tiene:

$$\frac{k}{n} = H(X) - H(Y \mid X) = I(X,Y)$$

Donde $\frac{k}{n}$ es la tasa que indica el porcentaje de bits que directamente están asociados a la información que transmite la fuente por el canal sobre el número total de bits transmitidos, resultado que ilustra la definición *2.29*, para el caso $q = 2$.
Así para aprovechar al máximo el canal, se debe llevar la tasa $\frac{k}{n}$ al máximo valor posible, siendo esta precisamente la capacidad $\mathbb{C}$ del canal, es decir:

$$\text{máx}\,\frac{k}{n} = max\frac{log_2 M}{n} = \underset{P(X)}{\text{máx}}\, I(X,Y) = \mathbb{C}$$

Se concluye que de las palabras de longitud n que se envían a través de un canal, sólo k bits deben ser encargados de portar información y los $n-k$ bits restantes, son los responsables del mecanismo de protección de dicha información contra los errores de sincronización producidos por el canal, esto se debe tener muy encuenta a la hora de construir una función decodificadora, pues ésta debe poder diferenciar dichos bits. La medida k fue claramente determinada en un CBS, sin embargo puede ser muy difícil o incluso indeterminada, al día de hoy, en otros canales con errores de sincronización.

3.2. El canal de eliminación

Luego de comentar los conceptos básicos en teoría de la información, y los teoremas de Shannon sobre codificación y capacidad de los canales, estamos en capacidad de presentar algunas definiciones puntuales y nuestro problema principal, *el canal de eliminación.*

Definición 3.6. *(Canal de Eliminación Binario, CEB)* Sean $\mathbb{X} = \{0,1\}$ y $\mathbb{Y} = \{0,1\}$ los alfabetos binarios fuente y receptor respectivamente. El alfabeto $\mathbb{A}$ contiene todas las posibles palabras a ser enviadas, es decir, $\mathbb{A} = \{0,1\}^n$. En este caso, el canal recibe palabras de longitud n y elimina cada bit aleatoriamente y de forma independiente e idénticamente distribuida, con probabilidad p y la probabilidad de recibir el bit es $(1-p)$, pero a diferencia del canal de borrado binario (CBB) no queda el símbolo ? para indicar el borrón, simplemente no queda rastro de la eliminación, así la palabra en la salida puede tener menos de n bits. Este es el *canal de eliminación binario independiente e idénticamente distribuido,* (CEB), o simplemente *canal de eliminación.*

En este caso el alfabeto de salida no es $\mathbb{A} = \{0,1\}^n$, sino $\acute{\mathbb{A}} = \bigcup_{m=0}^{n}\{0,1\}^m$ con $m \in \{0,1,2,\ldots,n\}$, por lo tanto el codebook o diccionario de palabras del receptor no se limita a aquellas de longitud n, sino que debe contener $\sum_{m=0}^{n} 2^{c_m}$ palabras, donde c_m es la longitud de las posibles palabras de salida. Además si n bits son enviados y m bits son recibidos, toda palabra o subconjunto de $n-m$ bits podría ser igualmente eliminado.

Definición 3.7. Sea X una palabra de longitud n. Una *subpalabra* de X, denotada $S_m(X)$, es una palabra con longitud $m \leq n$ donde $S_m(X)$ coincide en casi todos los bits de la palabra X, salvo posición y longitud, note que se admite que algunos bits falten y eso provoque el cambio en longitud y posición.

Ejemplo 3.8. Si $X = (101110000100000)$, su longitud es $l(X) = 15$. Dos subpalabras de X son:

$S_{13}(X) = (1011000100000)$ con $l(S_{13}(X)) = 13$.

$S_{10}(X) = (1011100000)$ con $l(S_{10}(X)) = 10$.

Con esto se observa que al pasar una palabra X de longitud n por el canal de eliminación, el resultado es una subpalabra $S_m(X)$ con longitud $m \leq n$

Uno de los problemas relacionados con el canal de eliminación es el desconocer que subpalabra $S_m(X)$ puede salir al pasar por el canal, pues los bits son eliminados de forma aleatoria y con probabilidad p, en el ejemplo anterior $S_{13}(X)$ puede ser el resultado de pasar X por el canal, sin embargo no se sabe con certeza que bits se eliminaron pues $S_{13}(X)$ coincide con X hasta el tercer run, puesto que para X, $r_3(X) = 111$ y para $S_{13}(X)$ se tiene que $r_3(S_{13}(X)) = 11$, pero no se sabe cuál de los 3 unos se eliminó.

Es aún más complejo lo que ocurrió con $S_{10}(X)$, ya que es una posible salida de X al pasar por el canal, en donde se eliminaron dos runs por completo, pues X tiene 6 runs mientras que $S_{10}(X)$ solo tiene 4 runs, esto ofrece la posibilidad de observar algunos bits perdidos (los de los runs desaparecidos), pero volvemos a desconocer los restantes bits eliminados y sus posiciones. Esta idea ilustra el hecho de que es más conveniente tomar la palabra X y estudiarla por sus runs o tiras internas y su longitud, que tomarla bit a bit.

Definición 3.9. Sea $X = (x_1x_2\ldots x_n)$ una palabra enviada por el canal de eliminación y $Y = (y_1y_2\ldots y_m)$ la palabra recibida, es decir una subpalabra de X. El número de veces que Y podría ser subpalabra de X se denota por $\#S(X,Y)$.

En el ejemplo anterior, del run 3 de X hay 3 opciones para el run 3 de $S_{13}(X)$ y del run 4 de X hay 4 opciones del run 4 de $S_{13}(X)$, luego $\#S(X,Y) = 3\dot{4} = 12$.

Por lo tanto, si p es la probabilidad de eliminación por bit, se tiene:

$$P(X \mid Y) = P(Y \mid X)\frac{P(X)}{P(Y)} = \#S(X,Y)p^{n-m}(1-p)^m$$

Nótese que si todas las palabras del código en principio tienen el mismo chance de ser enviadas, entonces al recibir la palabra Y se tiene que $P(X \mid Y)$ es proporcional a $\#S(X,Y)$. Además se desea que si X y W son 2 palabras distintas del diccionario, entonces si se envía X por el canal de eliminación:

$$P(\#S(X,Y) > \#S(W,Y) \mid X) \longrightarrow 1$$

Se ilustra las anteriores ideas en el siguiente ejemplo:

Ejemplo 3.10. Sean $X_1 = (000)$, $X_2 = (010)$ y $X_3 = (100)$ tres palabras de longitud 3 del diccionario que se envían a través del canal de eliminación con probabilidad de eliminación de un bit p y de llegada $(1-p)$. La siguiente tabla ilustra las posibles palabras de salida Y y sus probabilidades de ocurrencia:

X \ Y	$\emptyset$	(0)	(1)	(00)	(01)	(10)	(000)	(010)	(100)
(000)	p^3	$3p^2(1-p)$	0	$3p(1-p)^2$	0	0	$(1-p)^3$	0	0
(010)	p^3	$2p^2(1-p)$	$p^2(1-p)$	$p(1-p)^2$	$p(1-p)^2$	$p(1-p)^2$	0	$(1-p)^3$	0
(100)	p^3	$2p^2(1-p)$	$p^2(1-p)$	$p(1-p)^2$	0	$2p(1-p)^2$	0	0	$(1-p)^3$

Acá es posible evidenciar como en el diccionario del receptor se deben incluir palabras de longitud 3, 2, 1 y es incluso posible una eliminación de todos los bits, (aunque en la práctica es muy poco probable).

El canal de eliminación presenta inconvenientes con respecto al canal de borrado binario CBB y al canal binario simétrico CBS que complican a tal punto su estudio que muchas de sus propiedades son desconocidas o poseen unas ligeras aproximaciones. En particular podemos suponer que se procede de forma semejante al ejemplo *3.1* y sus conclusiones posteriores en un CBS, buscando de ésta manera encontrar una fórmula para la capacidad del canal de eliminación, este tipo de razonamiento está basado en los teoremas de Shannon y la forma en la que él comprobó la existencia de la capacidad y los códigos óptimos para un canal, sin embargo, en el razonamiento de Shannon se usa fuertemente el hecho de que la longitud de las palabras recibidas sea n y luego poder establecer los k bits que llevan la información y controlar los $n-k$ bits de redundancia para establecer la decodificación. Además se requiere que las palabras a la salida tengan la misma longitud y poder compararlas pensando en la decodificación, por medio de la distancia de Hamming, (definición *2.14*).

En el canal de eliminación, para una palabra de longitud n y una probabilidad de eliminación p de cada bit, podríamos esperar np eliminaciones, luego esto reduce en la misma proporción la longitud de la palabra, así que para tratar de descifrar la palabra enviada a partir de la palabra recibida se desearía tener una función decodificadora que fuera, en lo posible, una regla de decodificación por máxima verosimilitud (definición *2.26*) que pudiera *sobreponerse* a la perdida de dichos bits, es decir, si se toma la palabra de salida Y, esta tiene longitud m (que es un número aleatorio menor que la longitud de X), luego la función decodificadora no solo tendría que compararla con todas las posibles entradas de longitud n sino con todas aquellas de longitud w con $m \leq w \leq n$.

Ahora consideremos el hecho de que la probabilidad de eliminación por bit sea $p > \frac{1}{2}$, como $(1-p)$ es la probabilidad de transmisión correcta, se tendría $(1-p) < \frac{1}{2}$ y por lo tanto para una palabra con n bits que son enviados por el canal, la palabra de salida tendría $n(1-p) < \frac{n}{2}$ bits, es decir, *su llegada sería con menos de la mitad de los bits enviados*, así, si tomamos otra palabra W como posible palabra enviada, para determinar si Y es subpalabra de W necesitaríamos en promedio 2 bits de W por cada bit de Y para comprobarlo, suponiendo que esos bits son uniformemente distribuidos, entonces si el número de bits de W necesarios para compararse con un bit de Y están geométricamente distribuidos con media $\frac{1}{2}$, entonces es muy probable que únicamente $2n(1-p) < n$ bits de W sean necesarios para determinar si Y es subpalabra de W, luego es altamente probable que Y sea subpalabra de W y de X cuando $p > \frac{1}{2}$ así la decodificación sería poco factible, y el canal prácticamente inútil teniendo éste una capacidad cercana a cero.

En este canal, en general, se afecta el argumento de Shannon en su segundo teorema (Teorema *2.30*) pues la tasa de transmisión $\frac{k}{n}$ definida en la observación posterior al ejemplo *3.1*, referente al CBS, no sería factible de encontrar ya que tampoco se garantiza que W sea la única palabra del diccionario, junto con X, que cumpla esto. Así este teorema no brinda una cota efectiva para estimar la capacidad del canal de eliminación, y resulta necesario suponer que la eliminación por bit se de con una probabilidad $p < \frac{1}{2}$, para que aún sea viable un estudio.

A pesar de no poder aplicar los teoremas de Shannon al canal de eliminación, por los problemas antes mencionados, Dobrushin en [2] estableció una versión del segundo teorema de Shannon aplicado a los canales con errores de sincronización, en el siguiente teorema:

Teorema 3.11. (Dobrushin, 1967 [2]) *Sea K el canal de eliminación, $R \in \mathbb{R}$ y X una palabra de longitud n enviada a través del canal. Una constante C es definida para que una transmisión confiable sea posible si y solo si $R < C$, con:*

$$C = \lim_{n\to\infty} \frac{1}{n} C_n, \ \ con \ \ C_n = \sup_{P(X)} I(X,Y)$$

Donde Y es la palabra recibida y su longitud es un valor aleatorio $m < n$. Esto es, la máxima tasa de transmisión posible coincide con el máximo de la información mutua por bit.

Dobrushin probo este resultado de forma más general, pues lo verifico para el canal de eliminación y el *canal de inserción*, siendo este último, un canal donde se añaden bits de forma aleatoria y en este caso la longitud de Y, m es mayor que n. Este objetivo debería buscarse al tratar de encontrar una palabra código X en el diccionario, en donde Y ocurra la mayor cantidad de veces como subpalabra, sin embargo el problema radica en no tener una forma exacta o cerrada para la capacidad C. Dobrushin muestra la existencia de dicha C, pero al igual que Shannon, solo afirma que sería posible llegar a ese valor C por medio de una función decodificadora que sea óptima.

Por lo tanto y concluyendo algunas ideas centrales tenemos:

1. Dobrushin, en [2] prueba que la capacidad existe para canales con errores de sincronización, como eliminaciones e inserciones, sin embargo no brinda una formula cerrada para esta.

2. Es más factible encontrar errores al estudiar los cambios en los runs de cada palabra que estudiar los cambios bit a bit.

3. Es posible aplicar un código de repetición a un bit y convertirlo en un run o tira de bits, con el fin de evitar errores de sincronización.

4. Si la palabra es más larga, es más fácil observar los cambios al pasar por el canal de eliminación que con palabras más cortas y así estudiar mejor la tasa de transmisión o capacidad del canal, aunque este valor disminuya.

Recientemente, Mitzenmacher, Diggavi, Drinea, Fertonani y Montanari entre otros, en [4], [6], [7], [10], [9] y [16] respectivamente abordan el problema del canal de eliminación siguiendo principalmente las ideas anteriores, con esto han encontrado diversas cotas para la capacidad $\mathbb{C}$ del canal de eliminación de estos resultados destacamos:

1. *Mitzenmacher y Drinea*, [4], *(cota inferior, 2006):* Para cualquier probabilidad de eliminación p,
$$\mathbb{C} \geq \frac{1-p}{9}$$

2. *Mitzenmacher,* [7]*, (cota inferior, 2007):* Si p es una probabilidad de eliminación, preferiblemente $p \leq 0{,}5$, entonces:
$$\mathbb{C} \geq \sup_{0<r<1\ \delta>0} \left\{ -(1-p)log_2((1-q)A + qB) - \frac{\delta}{ln2} \right\}$$
con:
$$A = \frac{(1-r)e^{-\delta}}{1-re^{-\delta}},\ \ B = \frac{(1-r)e^{-2\delta}}{1-re^{-\delta}} + re^{-\delta},\ \ q = 1 - \frac{1-r}{1+p(1-2r)}$$

3. *Diggavi, Mitzenmacher y Pfister,* [10]*, (cota superior, 2007):* Computacionalmente y usando una aplicación del algoritmo de Blahut-Arimoto [13] y [14] se estimó una cota que en síntesis concluye que:

$$\mathbb{C} \leq 0{,}7918(1-p), \quad si\ p \to 1$$

4. *Montanari y Kanoria* [16]*, (aproximación, 2011):* Si $\mathbb{C}_p$ es la capacidad del canal de eliminación con probabilidad p, entonces para pequeños valores p y cualquier $\epsilon > 0$:

$$\mathbb{C}_p = 1 + plog_2 p - A_1 p + A_2 p^2 + O(p^{3-\epsilon}),$$

donde:

$$A_1 \approx 1{,}15416377 \quad A_2 \approx 1{,}67814594$$

Acá mostramos 2 cotas inferiores de varias que aparecen en los artículos mencionados y comentamos una estimación de una cota superior, omitiendo otra cota que es más natural, esta sería la capacidad del canal de borrado binario CBB, cuyo valor sería $1-p$, entendiendo que este canal tiene más capacidad que el canal de eliminación, pues el bit borrado deja la marca de borrado y es posible encontrar o estimar dicha información ya que la longitud de la palabra es la misma, esto es se reducen los problemas de longitud y posición del bit borrado.

3.3. El canal de eliminación y sus variantes

Las observaciones anteriores sugieren que es necesario un pre proceso de las palabras antes de ser enviadas por el canal de eliminación, dicho proceso consiste en repetir los bits de la palabra X una cierta cantidad de veces con el fin de hacerla más *resistente* a la eliminación, pero no repetir bit a bit sino runs de bits, ya que estos ayudan a detectar bits perdidos si el número de runs disminuye, este canal que combina repeticiones con eliminaciones se denomina *canal de *-eliminación*, en donde se reduce la posibilidad de perdida de bits *importantes*, por eliminación, a cambio de eliminar bits *redundantes* aunque esto aumenta la longitud de la palabra y proporcionalmente disminuyendo la tasa de transmisión $\frac{k}{n}$, esto es disminuyendo su capacidad. Mitzenmacher propone en [4] una variante del canal de *-eliminación, llamada *canal poisson-eliminación.*

Definición 3.12. Sea f una función de distribución de poisson, con parámetro λ definida sobre palabras X de longitud n sobre un alfabeto binario, en donde cada bit que pasa por el canal es reemplazado por un número poisson de copias formando un run cuya longitud, es decir el número de copias, tiene media λ y es independiente para cada bit.

Surge una pregunta natural y es la relación entre el canal de eliminación con parámetro p y el canal poisson-eliminación con parámetro λ. En este caso *antes* de enviar el mensaje se reemplaza cada bit por un *número aleatorio* de copias de acuerdo a la variable aleatoria de poisson, con media $\frac{\lambda}{1-p}$, esto hace que el número de copias para cada bit que llegan al receptor tengan exactamente una distribución de poisson con media λ y que este número de copias sea independiente para cada bit.

En este caso el canal de eliminación se comportaría como un canal poisson-eliminación con media λ sobre las palabras originales o sin extender, de esta idea se puede concluir que si una función codificadora y decodificadora funcionan para el canal poisson-eliminación entonces tendríamos un par de funciones aplicables al canal de eliminación, en donde se debe incluir un paso de codificación por repetición, o regla de codificación con longitud aleatoria según la distribución poisson, pues cada bit debe ser expandido en un factor de $\frac{\lambda}{1-p}$ veces.

Así la capacidad del canal poisson-eliminación con parámetro λ es un valor L_λ que cumple para $\mathbb{C}_p$ la capacidad del canal de eliminación con probabilidad de eliminación p la siguiente desigualdad:

$$\frac{\lambda}{1-p}\mathbb{C}_p \geq L_\lambda$$

es decir:

$$C_p \geq \mathbb{L}_\lambda \frac{1-p}{\lambda}$$

sería una cota inferior para la capacidad del canal de eliminación.

En síntesis se tiene el siguiente esquema propuesto por Mitzenmacher:

Se desea enviar la palabra X por el canal de eliminación, con probabilidad de eliminación p por bit, luego:

Se comienza con enviar X por el canal poisson eliminación

$\Downarrow$

Se expande o codifica X a $\acute{X}$ usando un número poisson de veces con media $\frac{1}{1-p}$ para cada bit

$\Downarrow$

Se envía $\acute{X}$ por el canal de eliminación

$\Downarrow$

Se decodifica la palabra recibida Y usando una función decodificadora adecuada al canal

Esta idea central sirvió de inspiración en el presente trabajo para buscar alternativas a la codificación y aplicaciones del canal de eliminación, que serán expuestas en el próximo capítulo, pues para este canal no se tiene una forma cerrada y directa para calcular su capacidad y mucho menos códigos eficientes.

Capítulo 4

El Canal de Eliminación: Resultados y Propuestas

4.1. Idea central

Basándonos en las ideas propuestas por Mitzenmacher en [4], comenzaremos a estudiar el problema de encontrar las funciones codificadoras y decodificadoras explícitamente y como su escogencia puede ser modelada y puesta en práctica en el canal de eliminación, comenzamos proponiendo una variante a la idea central de Mitzenmacher y el canal poisson-eliminación.

Se inicia con una palabra X en un diccionario o codebook C de un alfabeto binario $\mathbb{A} = \{0,1\}^n$, cuya longitud es n, luego se consideran todos los runs r_i de la palabra X y su longitud $l(r_i)$. Supongamos que dichos runs se distribuyen geométricamente, luego la probabilidad de que un run tenga longitud 1 es $\frac{1}{2}$, de que tenga longitud 2 es $\frac{1}{2^2} = \frac{1}{4}$ y así sucesivamente hasta concluir que:

$$P(l(r_i) = m) = \frac{1}{2^m}$$

El valor esperado en general, de la longitud de los runs sería:

$$E(P(l(r_i) = m)) = \sum_{m=1}^{\infty} \frac{m}{2^m}$$

Para calcular dicho valor usamos el siguiente:

Lema 4.1. *Sea $\sum_{n=1}^{\infty} a_n$ una serie donde $a_n = (an + b)r^n$, esto es, a_n es el producto de 2 términos uno en progresión aritmética y el otro en progresión geométrica, entonces si $|r| < 1$ se tiene:*

$$S = \sum_{n=1}^{\infty} a_n = \sum_{n=1}^{\infty} (an + b)r^n = \frac{(a+b)r - br^2}{(1-r)^2}$$

Demostración. Note que la n-ésima suma parcial:

$$\begin{aligned}
S_n &= (a+b)r + (2a+b)r^2 + (3a+b)r^3 + \cdots + (an+b)r^n \\
-rS_n &= -(a+b)r^2 - (2a+b)r^3 - (3a+b)r^4 - \cdots - (an+b)r^{n+1} \\
(1-r)S_n &= (a+b)r + ar^2 + ar^3 + \cdots + ar^n - (an+b)r^{n+1} \\
(1-r)S_n &= (a+b)r + a(r^2 + r^3 + \cdots + r^n) - (an+b)r^{n+1} \\
(1-r)S_n &= (a+b)r + ar(r + r^2 + \cdots + r^{n-1}) - (an+b)r^{n+1}
\end{aligned}$$

Tomando limites:

$$\begin{aligned}
(1-r)S &= (a+b)r + \frac{ar^2}{1-r} - 0 = \frac{(a+b)r(1-r) + ar^2}{1-r} \\
&= \frac{(a+b)r - (a+b)r^2 + ar^2}{1-r} = \frac{(a+b)r - br^2}{1-r}
\end{aligned}$$

Luego:

$$S = \frac{(a+b)r - br^2}{(1-r)^2}$$

□

Con este lema se tiene con: $a = 1$, $b = 0$ y $r = \frac{1}{2}$

$$E(P(l(r_i) = m)) = \sum_{m=1}^{\infty} \frac{m}{2^m} = \sum_{m=1}^{\infty} m \left(\frac{1}{2}\right)^m = \frac{(1+0)\frac{1}{2} - 0\left(\frac{1}{2}\right)^2}{(1-\frac{1}{2})^2} = 2$$

Esto es, el número de runs en promedio para la palabra X es $E(\text{número de runs}) = \frac{n}{2}$, así:

$$\text{número de runs por palabra } < (1+\epsilon)\frac{n}{2} \text{ con } \epsilon > 0$$

Para estimar cuantas de las palabras del diccionario cumplen esta afirmación usamos los siguientes resultados:

Lema 4.2. (Desigualdad de Markov). *Sean $(\Omega, \mathcal{F}, \mathbb{P})$ un espacio de probabilidad y X una variable aleatoria, si $f(X) = 0$, para $X < 0$, entonces para todo $\epsilon > 0$:*

$$P(X \geq \epsilon) \leq \frac{E(X)}{\epsilon}$$

Demostración. Esto pues:

$$E(X) = \int_0^{\infty} Xf(X)dX \geq \int_{\epsilon}^{\infty} Xf(X)dX \geq \epsilon \int_{\epsilon}^{\infty} f(X)dX = \epsilon P(X \geq \epsilon)$$

□

De forma más general si se sustituye X por una función positiva $h(X)$ se tiene:

$$P(h(X) \geq h(\epsilon)) \leq \frac{E(h(X))}{h(\epsilon)}$$

Y si $h(X)$ es creciente, entonces:

$$P(X \geq \epsilon) = P(h(X) \geq h(\epsilon)) \leq \frac{E(h(X))}{h(\epsilon)}$$

Corolario 4.3. *Con el lema anterior y sus consecuencias, si $f(X) = X^2$, se obtiene:*

$$P(|X - E(X)| \geq \epsilon) = P(|X - E(X)|^2 \geq \epsilon^2) \leq \frac{E(|X - E(X)|^2)}{\epsilon^2} = \frac{Var(X)}{\epsilon^2}$$

tomando $\epsilon = k\sigma$ se puede reescribir como:

$$P(|X - E(X)| \geq k\sigma) \leq \frac{Var(X)}{\epsilon^2} = \frac{1}{k^2}$$

Y es llamada *desigualdad de Chebyshev* o *cotas de Chebyshev.*

Teorema 4.4. *(Cotas de Chernoff) Sean $(\Omega, \mathcal{F}, \mathbb{P})$ un espacio de probabilidad y $X = \sum_{i=1}^{n} X_i$ donde X_i son variables aleatorias independientes e idénticamente distribuidas, $X_i \in \{0, 1\}$, $E(X_i) = p_i$ y $\mu = E(X) = E(\sum_{i=1}^{n} X_i)$. Entonces:*

$$P(X \geq (1 + \epsilon)\mu) \leq e^{\frac{-\epsilon^2 \mu}{2+\epsilon}} = e^{-\theta\mu}$$

y

$$P(X \leq (1 - \epsilon)\mu) \leq e^{\frac{-\epsilon^2 \mu}{2+\epsilon}} = e^{-\theta\mu} \ \ con \ \ \theta = \frac{-\epsilon^2}{2 + \epsilon}$$

Demostración. Usando el lema *4.2* con $f(X) = e^{tX}$ se tiene:

$$P(X \geq (1 + \epsilon)\mu) = P(e^{tX} \geq e^{(1+\epsilon)t\mu}) \leq \frac{E(e^{tX})}{e^{(1+\epsilon)t\mu}}$$

Ahora, usando la independencia de las variables:

$$E(e^{tX}) = E(e^{t\sum_{i=1}^{n} X_i}) = E(\prod_{i=1}^{n} e^{tX_i}) = \prod_{i=1}^{n} E(e^{tX_i})$$

$$= \prod_{i=1}^{n} (p_i e^t + (1 - p_i)1) = \prod_{i=1}^{n} (1 + p_i(e^t - 1))$$

Usando el hecho de que $e^x \geq 1 + x$ para todo x real, se tiene:

$$E(e^{tX}) = \prod_{i=1}^{n}(1 + p_i(e^t - 1)) \leq \prod_{i=1}^{n} e^{p_i(e^t-1)}$$
$$= e^{\sum_{i=1}^{n} p_i(e^t-1)} = e^{(e^t-1)\mu}$$

Sustituyendo $E(e^{tX}) \leq e^{(e^t-1)\mu}$ en la ecuación inicial se tiene:

$$P(X \geq (1+\epsilon)\mu) \leq \frac{E(e^{tX})}{e^{(1+\epsilon)t\mu}} \leq \frac{e^{(e^t-1)\mu}}{e^{(1+\epsilon)t\mu}}$$
$$= e^{(e^t-1)\mu-(1+\epsilon)t\mu} = e^{\mu(e^t-1-t-t\epsilon)}$$

El valor mínimo en la expresión de la derecha se obtiene cuando $t = ln(1+\epsilon)$, al sustituir esto en la ecuación tenemos:

$$P(X \geq (1+\epsilon)\mu) \leq e^{\mu(\epsilon-(1+\epsilon)ln(1+\epsilon))}$$

Usando la serie de Taylor de $ln(1+\epsilon) = \sum_{i=1}^{\infty}(-1)^{i+1}\frac{\epsilon^i}{i}$ se tiene:

$$(1+\epsilon)ln(1+\epsilon) = \epsilon + \sum_{i=2}^{\infty}(-1)^i\epsilon^i\left(\frac{1}{i-1} - \frac{1}{i}\right)$$

Si tomamos $0 < \epsilon < 1$, podemos ignorar términos altos de la serie y obtener:

$$(1+\epsilon)ln(1+\epsilon) > \epsilon + \frac{\epsilon^2}{2} - \frac{\epsilon^3}{6} \geq \epsilon + \frac{\epsilon^2}{3}$$

Así, al reemplazar a $(1+\epsilon)ln(1+\epsilon)$ en la expresión final se tiene:

$$P(X \geq (1+\epsilon)\mu) \leq e^{\mu(\epsilon-\epsilon-\frac{\epsilon^2}{3})} = e^{\mu(-\frac{\epsilon^2}{3})} \quad \text{con } 0 < \epsilon < 1$$

Con un cálculo similar se obtiene:

$$P(X \geq (1+\epsilon)\mu) \leq e^{\mu(-\frac{\epsilon^2}{2})} \quad \text{con } 0 < \epsilon < 1$$

Ahora $ln(1+\epsilon) > \frac{2\epsilon}{2+\epsilon}$ para todo $\epsilon > 0$, luego se tiene:

$$\epsilon - (1+\epsilon)ln(1+\epsilon) \leq \frac{-\epsilon^2}{2+\epsilon}$$

Así se concluye que:

$$P(X \geq (1+\epsilon)\mu) \leq e^{\frac{-\epsilon^2}{2+\epsilon}\mu} = e^{-\theta\mu}$$

y

$$P(X \leq (1-\epsilon)\mu) \leq e^{\frac{-\epsilon^2}{2+\epsilon}\mu} = e^{-\theta\mu}$$

□

Usando el teorema anterior y sumando los dos últimos resultados se obtiene:

$$P(X \geq (1+\epsilon)\mu) + P(X \leq (1-\epsilon)\mu) \leq 2e^{-\theta\mu}$$

es decir:

$$P(X \geq (1+\epsilon)\mu, \quad \text{o} \quad , X \leq (1-\epsilon)\mu) \leq 2e^{-\theta\mu}$$

simplificando:

$$P(X - \mu \geq \epsilon\mu, \quad \text{o} \quad , X - \mu \leq -\epsilon\mu) \leq 2e^{-\theta\mu}$$

se tiene:

$$P(|X - \mu| \geq \epsilon\mu) \leq 2e^{-\theta\mu}$$

tomando complementos, se concluye el siguiente:

Corolario 4.5. *Sean $(\Omega, \mathcal{F}, \mathbb{P})$ un espacio de probabilidad y $X = \sum_{i=1}^{n} X_i$ donde X_i son variables aleatorias independientes e idénticamente distribuidas, $X_i \in \{0, 1\}$, $E(X_i) = p_i$ y $\mu = E(X) = E(\sum_{i=1}^{n} X_i)$.Entonces:*

$$P(|X - \mu| \leq \epsilon\mu) \leq 1 - 2e^{-\theta\mu}$$

Este resultado, consecuencia directa del teorema *4.4*, puede ser utilizado para estimar el cálculo de cuantas palabras binarias de longitud n, tienen un número de runs promedio de $\frac{n}{2}$. Para esto, sea N el número de runs promedio por palabra, entonces $\mu = E(\text{N}) = \frac{n}{2}$, luego por el corolario *4.5* se tiene:

$$\begin{aligned} P(|N - \mu| \leq \epsilon\mu) &= P(|N - \frac{n}{2}| \leq \epsilon\frac{n}{2}) = P(-\epsilon\frac{n}{2} < N - \frac{n}{2} < \epsilon\frac{n}{2}) \\ &= P(\frac{n}{2} - \epsilon\frac{n}{2} < N < \epsilon\frac{n}{2} + \frac{n}{2}) \\ &= P((1-\epsilon)\frac{n}{2} < N < (1+\epsilon)\frac{n}{2}) \leq 1 - 2e^{-\frac{\theta n}{2}} \end{aligned}$$

Luego para valores cada vez mayores de n, prácticamente todas las palabras X que escojamos cumplirán que su número de runs estén entre $(1-\epsilon)\frac{n}{2}$ y $(1+\epsilon)\frac{n}{2}$.

4.2. La propuesta central, el canal de Ω-eliminación

La modificación hecha por Mitzenmacher al canal de eliminación consiste en ampliar las palabras antes de enviarlas a través del canal, con el fin de protegerlas de la eliminación, y crear así el canal poisson-eliminación, en donde cada run en la palabra es expandido un número aleatorio poisson de veces.

En el presente trabajo se busca modelar de forma alternativa dicho número y conseguir un muy buen código de repetición que se adapte al problema del canal de eliminación, es decir, una función codificadora por repetición $f(t_i)$ distinta, donde $t_i = l(r_i)$ es la longitud de cada run y con ella estudiar y modelar un nuevo canal que denotaremos el *canal de* Ω*-eliminación*, que es una variante al canal poisson-eliminación que podría ser útil en el estudio de la capacidad de dicho canal.

Sea X la palabra del diccionario binario a enviar por el canal de eliminación, p la probabilidad de eliminar un bit al pasar por el canal de forma independiente e idénticamente distribuida y $V_R(X)$ el vector de runs de X, (definición *3.2*), la función de codificación por repetición deseada es de la forma $f(t)$, donde $t = l(r)$ es la longitud de cada run.
Si t indica cuantos bits tiene cada run de X, $f(t)$ indica cuantos bits contiene cada run expandido por repetición de $\acute{X}$, creando una *redundancia* en los bits, en este caso al pasar $f(t)$ bits por el canal de eliminación, el resultado luego de la transmisión, es esperado de la forma $f(t)-\epsilon$ donde ϵ representa a los bits eliminados por el canal, es decir, usando el corolario *4.3*, se podría estimar que:

$$\epsilon = (\text{porción de datos eliminados}) \pm \sigma\sqrt{f(t)}$$

es decir:

$$\epsilon = pf(t) \pm M\sqrt{f(t)}$$

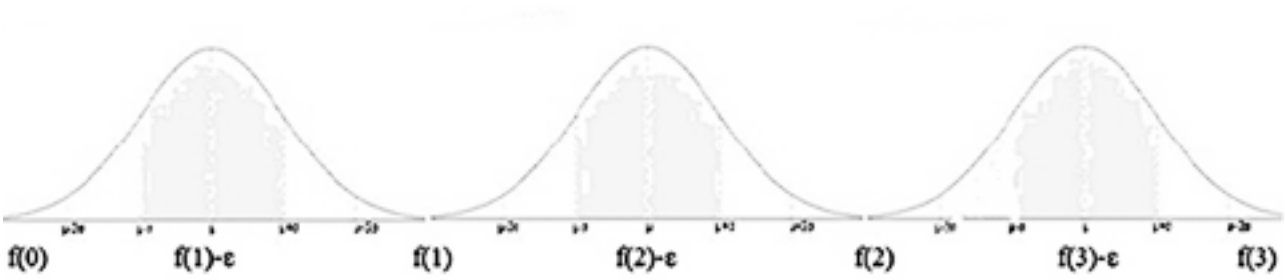
Donde $M\sqrt{f(t)}$ representa un error estimado adicional, que viene dado directamente por la desviación estándar σ, y con esto suponemos que $f(t)$ al pasar por el canal de eliminación queda en la forma:

$$f(t) - \epsilon = f(t) - \left(pf(t) \pm M\sqrt{f(t)}\right)$$

Así que se desea estimar una función $f(t)$ que satisfaga lo anterior y que adicionalmente distinga el envío de t bits a través del canal, con el envío de $t-1$ bits, para así evitar problemas adicionales con las subpalabras $S(X)$ y que este bien definida, es decir:

$$f(t) - \epsilon_0 > f(t-1) - \epsilon_1$$

Con lo anterior, y en virtud al corolario *4.3* y al teorema *4.4*, se desea que las salidas se acumulen en torno al valor recibido $f(t) - \epsilon$ como se ilustra en el siguiente diagrama:



Si se define:

$$\epsilon_0 = pf(t) + M\sqrt{f(t)}$$

$$\epsilon_1 = pf(t-1) - M\sqrt{f(t-1)}$$

Se tiene que la condición para la buena definición de $f(t)$:

$$f(t) - \epsilon_0 > f(t-1) - \epsilon_1$$

Queda en la forma:

$$f(t) - \left(pf(t) + M\sqrt{f(t)}\right) > f(t-1) - \left(pf(t-1) - M\sqrt{f(t-1)}\right)$$

Es decir:

$$(1-p)f(t) - M\sqrt{f(t)} > (1-p)f(t-1) + M\sqrt{f(t-1)}$$

Donde M es directamente proporcional a la desviación estándar en una distribución binomial.

Para estimar tal $f(t)$, supongamos que es de la forma $f(t) = kt^a$ con k constante, a entero no negativo y procedamos de forma inductiva en el exponente a:

1. ***Si $f(t)$ es lineal***: Supongamos que $f(t) = kt$, con k constante, entonces la condición de buena definición queda de la forma:

$$(1-p)f(t) - M\sqrt{f(t)} > (1-p)f(t-1) + M\sqrt{f(t-1)}$$

$$(1-p)kt - M\sqrt{kt} > (1-p)k(t-1) + M\sqrt{k(t-1)}$$

$$kt - pkt - M\sqrt{kt} > kt - k - kpt + kp + M\sqrt{k(t-1)}$$

$$k - kp > M\sqrt{kt} + M\sqrt{k(t-1)}$$

$$(1-p)k > M(\sqrt{kt} + \sqrt{k(t-1)})$$

   Lo cual no se cumple a valores grandes de t lo que restringiría mucho la longitud de los runs en la palabra X, luego se busca la siguiente potencia $a > 1$.

2. ***Si $f(t)$ es potencia con $a>1$***: Supongamos que $f(t) = kt^a$, con $a > 1$, en este caso la condición de buena definición sería:

$$(1-p)f(t) - M\sqrt{f(t)} > (1-p)f(t-1) + M\sqrt{f(t-1)}$$

$$(1-p)kt^a - M\sqrt{k}\,t^{a/2} > (1-p)k(t-1)^a + M\sqrt{k}\,(t-1)^{a/2}$$

$$kt^a - pkt^a - M\sqrt{k}\,t^{a/2} > (1-p)k\left(t^a - at^{a-1} + \cdots + at - 1\right) + M\sqrt{k}\,\left(t^{a/2} - at^{a/2-1} + \cdots + at - 1\right)$$

$$kt^a - pkt^a - M\sqrt{k}\,t^{a/2} > kt^a - kat^{a-1} + \cdots + kat - k - pkt^a + pkat^{a-1} - \cdots + pkat + pk + M\sqrt{k}\,t^{a/2} - M\sqrt{k}\,at^{a/2-1} + \cdots + M\sqrt{k}\,at - M\sqrt{k}$$

$$-M\sqrt{k}\,t^{a/2} > -kat^{a-1} + \cdots + kat - k + pkat^{a-1} - \cdots + pkat + pk + M\sqrt{k}\,t^{a/2} - M\sqrt{k}\,at^{a/2-1} + \cdots + M\sqrt{k}\,at - M\sqrt{k}$$

$$-M\sqrt{k}\,t^{a/2} > (pka - ka)t^{a-1} + \cdots + (pka - ka)t + pk + k + M\sqrt{k}\,t^{a/2} - M\sqrt{k}\,at^{a/2-1} + \cdots + M\sqrt{k}\,at - M\sqrt{k}$$

Al comparar potencias semejantes de ambas expresiones a los lados de la desigualdad y para valores muy grandes de t, se debería tener $\frac{a}{2} \approx a-1$, es decir $1-\frac{a}{2}$, luego $1 \approx \frac{a}{2}$ de donde se concluye que $a \approx 2$.

3. ***Si*** $f(t)$ ***es potencia con a=2***: Supongamos que $f(t) = kt^2$, luego la condición de buena definición sería:

$$(1-p)f(t) - M\sqrt{f(t)} > (1-p)f(t-1) + M\sqrt{f(t-1)}$$

$$(1-p)kt^2 - M\sqrt{k}\,t > (1-p)k(t-1)^2 + M\sqrt{k}\,(t-1)$$

$$kt^2 - pkt^2 - M\sqrt{k}\,t > (kt^2 - 2kt + k)(1-p) + M\sqrt{k}\,(t-1)$$

$$kt^2 - pkt^2 - M\sqrt{k}\,t > kt^2 - 2kt + k - pkt^2 + 2pkt - pk + M\sqrt{k}\,t - M\sqrt{k}$$

$$-M\sqrt{k}\,t > -2kt + k + 2pkt - pk + M\sqrt{k}\,t - M\sqrt{k}$$

$$2kt - k - 2pkt + pk > 2M\sqrt{k}\,t - M\sqrt{k}$$

$$k(2t - 1 - 2pt + p) > M(2\sqrt{k}\,t - \sqrt{k})$$

$$k((2t-1) - p(2t-1)) > M\sqrt{k}\,(2t-1)$$

$$k(2t-1)(1-p) > M\sqrt{k}\,(2t-1)$$

$$(1-p) > \frac{M}{\sqrt{k}}$$

$$\sqrt{k} > \frac{M}{1-p}$$

$$k > \left(\frac{M}{1-p}\right)^2$$

Como M es proporcional a la desviación estándar en una distribución binomial, tenemos:

$$M = \sigma = \sqrt{p(1-p)}$$

y se concluye:

$$k > \frac{p(1-p)}{(1-p)^2} = \frac{p}{1-p}$$

Por lo tanto si se escoje:

$$k = \left(T_\epsilon^2 \frac{p}{1-p} \right)$$

para T_ϵ^2 lo suficientemente grande, de tal forma que se cumpla la desigualdad anterior de forma independiente a t, se obtiene el siguiente:

Teorema 4.6. *Sea X la palabra del diccionario binario a enviar por el canal de eliminación, p la probabilidad de eliminar un bit al pasar por el canal de forma independiente e idénticamente distribuida, la función de codificación por repetición $f(t)$, donde $t = l(r)$ es la longitud de cada run de X, definida por:*

$$f(t) = \left(T_\epsilon^2 \frac{p}{1-p} \right) t^2 \qquad \text{con } T_\epsilon^2 \text{ suficientemente grande}$$

Codifica, de forma independiente a t, la palabra X a través del canal de eliminación distinguiendo el envío de t bits al envío de $t-1$ bits.

En la práctica, para usar el anterior teorema es necesario escribir la palabra X como $V_R(X)$ el vector de runs de X, luego la función de codificación por repetición $f(t)$, donde $t = l(r)$ es la longitud de cada run de X, se aplica a los exponentes del vector de runs, esos son los valores t a evaluar y así se obtiene la palabra $\acute{X}$ que es más larga que X por las repeticiones, pero a diferencia de la propuesta de Mitzenmacher, los runs se aumentan de forma *cuadrática* y no de forma aleatoria según poisson, evitando así posibles eliminaciones intencionales, y si estas se presentan ya será por efecto del error de transmisión del canal de eliminación y no por problemas en la codificación. Con esta función podemos presentar formalmente *la idea central.*

Definición 4.7. Sean $\mathbb{X} = \{0,1\}$ y $\mathbb{Y} = \{0,1\}$ los alfabetos binarios fuente y receptor respectivamente. El alfabeto $\mathbb{A}$ contiene todas las posibles palabras a ser enviadas, es decir, $\mathbb{A} = \{0,1\}^n$ y $f(t) = \left(T_\epsilon^2 \frac{p}{1-p} \right) t^2$ es una función codificadora por repetición para el canal, donde $t = l(r)$ es la longitud de cada run de X. En este caso, el canal recibe palabras de longitud aumentada en forma cuadrática según $f(t)$ y elimina cada bit aleatoriamente y de forma independiente e idénticamente distribuida, con probabilidad p y la probabilidad de recibir el bit es $(1-p)$, así la palabra en la salida puede tener menos de bits de los que se enviaron. Este es el *canal de Ω-eliminación binario independiente e idénticamente distribuido,* (COE), o simplemente *canal de Ω-eliminación* .

Esta variante del canal de eliminación original tiene la ventaja de enviar palabras más protegidas, pues al aumentar los runs de forma cuadrática se reduce la probabilidad de que se elimine todo un run complicando notablemente el proceso de decodificación, basta con determinar que función $g(t)$ cumple el proceso inverso o de decodificación. En este punto, la idea se resume en el siguiente diagrama:

Se desea enviar la palabra X por el canal de eliminación, con probabilidad de eliminación p por bit, luego:

Se comienza con enviar X por el canal de Ω-eliminación

$$\Downarrow$$

Se expande o codifica X a $\acute{X}$ usando la función codificadora $f(t) = \left(T_{\epsilon}^{2} \frac{p}{1-p}\right) t^2$

$$\Downarrow$$

Se envía $\acute{X}$ por el canal de eliminación

$$\Downarrow$$

Se decodifica la palabra recibida Y usando una función decodificadora adecuada al canal

Así, para completar el esquema anterior es necesario definir la función decodificadora $g(t)$ que tome la palabra extendida $\acute{X}$ después de la eliminación aleatoria de datos y sea capaz de llevarla, con poco margen de error, a la palabra inicial que se transmitió X y no a otra subpalabra W que produzca un error en la transmisión. Antes de esto presentaremos algunas definiciones y resultados previos:

Definición 4.8. Sea x un número real. La función que redondea x al entero más cercano, se denota $round(x)$.

Lema 4.9. *Sea* $0 < \alpha < \frac{1}{2}$, t *un entero no negativo y* $u \in \left(t^2 - \alpha\frac{t}{2}, t^2 + \alpha\frac{t}{2}\right)$, *entonces* $round(\sqrt{u}) = t$

Demostración. Sea $u \in \left(t^2 - \alpha\frac{t}{2}, t^2 + \alpha\frac{t}{2}\right)$, entonces:

$$t^2 - \alpha\frac{t}{2} < u < t^2 + \alpha\frac{t}{2}$$

$$\sqrt{t^2 - \alpha\frac{t}{2}} < \sqrt{u} < \sqrt{t^2 + \alpha\frac{t}{2}}$$

$$\sqrt{u} - t < \sqrt{t^2 + \alpha\frac{t}{2}} - t$$

Por otro lado:

$$0 < \frac{\alpha^2}{16} \;\rightarrow\; t^2 + \alpha\frac{t}{2} < t^2 + \alpha\frac{t}{2} + \frac{\alpha^2}{16} = \left(t + \frac{\alpha}{4}\right)^2$$

$$\sqrt{t^2 + \alpha\frac{t}{2}} < t + \frac{\alpha}{4} \;\rightarrow\; \sqrt{t^2 + \alpha\frac{t}{2}} - t < \frac{\alpha}{4}$$

luego:

$$\sqrt{u} - t < \sqrt{t^2 + \alpha\frac{t}{2}} - t < \frac{\alpha}{4}$$

así:

$$\sqrt{u} - t < \frac{\alpha}{4}$$

de donde:

$$round\left(\sqrt{u} - t\right) < round\left(\frac{\alpha}{4}\right) = 0$$

entonces:

$$round\left(\sqrt{u} - t\right) = round\left(\sqrt{u}\right) - round\left(t\right) = round\left(\sqrt{u}\right) - t = 0$$

y se concluye que:

$$round\left(\sqrt{u}\right) = t$$

□

La función decodificadora $g(t)$, adecuada para el canal de Ω-eliminación, es una función que debería cumplir:

$$g(f(t_i)) = t_i$$

donde t_i es la longitud del i-ésimo run de la palabra X, sin embargo en la practica, $f(t_i)$ llega después de pasar por el canal de eliminación luego en realidad $g(t)$ debe cumplir:

$$g(f(t_i) - \epsilon) \approx t_i$$

donde ϵ es la porción de datos eliminados en forma aleatoria. Así, si $\phi \approx f(t)$ entonces $g(\phi) \approx g(f(t))$, luego se quiere una función decodificadora $g(t)$ tal que:

$$g(\phi) \approx t$$

con lo que se concluye el siguiente teorema.

Teorema 4.10. *Sea X la palabra del diccionario binario a enviar por el canal de eliminación, p la probabilidad de eliminar un bit al pasar por el canal de forma independiente e idénticamente distribuida, i.i.d y la función de codificación por repetición $f(t)$, donde $t = l(r)$ es la longitud de cada run de X, definida por:*

$$f(t) = \left(T_\epsilon^2 \frac{p}{1-p}\right) t^2 \qquad \text{con } T_\epsilon^2 \text{ suficientemente grande}$$

La función decodificadora $g(t)$ que satisface $g(f(t) - \epsilon) \approx t$ es:

$$g(\phi) = round\left(\sqrt{\frac{\phi}{T_\epsilon^2\, p}}\right) \qquad \text{con } T_\epsilon^2 \text{ suficientemente grande}$$

Demostración. Sea ϕ la longitud de un run en una posible salida del canal de eliminación, se desea relacionar ϕ con $f(t) - \epsilon$ que son las salidas esperadas, luego aplicando el corolario *4.3* se obtiene:

$$f(t) - \epsilon_0 < \phi < f(t) + \epsilon_1$$

con:

$$\epsilon_1 = -pf(t) + \frac{\alpha}{2} T_\epsilon \sigma \sqrt{f(t)}$$

$$\epsilon_0 = pf(t) + \frac{\alpha}{2} T_\epsilon \sigma \sqrt{f(t)}$$

$$\sigma = \sqrt{p(1-p)}$$

por lo tanto:

$$f(t) - \epsilon_0 < \phi < f(t) + \epsilon_1$$

$$f(t) - pf(t) - \frac{\alpha}{2} T_\epsilon \sigma \sqrt{f(t)} < \phi < f(t) - pf(t) + \frac{\alpha}{2} T_\epsilon \sigma \sqrt{f(t)}$$

$$(1-p)f(t) - \frac{\alpha}{2} T_\epsilon \sigma \sqrt{f(t)} < \phi < f(t)(1-p) + \frac{\alpha}{2} T_\epsilon \sigma \sqrt{f(t)}$$

$$(1-p)kt^2 - \frac{\alpha}{2} T_\epsilon \sqrt{p(1-p)} \sqrt{kt^2} < \phi < kt^2(1-p) + \frac{\alpha}{2} T_\epsilon \sqrt{p(1-p)} \sqrt{kt^2}$$

$$(1-p)T_\epsilon^2 \frac{p}{1-p} t^2 - \frac{\alpha}{2} T_\epsilon \sqrt{p(1-p)} \sqrt{T_\epsilon^2 \frac{p}{1-p} t^2} < \phi < T_\epsilon^2 \frac{p}{1-p} t^2 (1-p) + \frac{\alpha}{2} T_\epsilon \sqrt{p(1-p)} \sqrt{T_\epsilon^2 \frac{p}{1-p} t^2}$$

$$T_\epsilon^2 pt^2 - \frac{\alpha}{2} T_\epsilon^2 pt < \phi < T_\epsilon^2 pt^2 + \frac{\alpha}{2} T_\epsilon^2 pt$$

Luego

$$T_\epsilon^2 p(t^2 - \frac{\alpha}{2}) < \phi < T_\epsilon^2 p(t^2 + \frac{\alpha}{2}t)$$

y con esto se tiene:

$$t^2 - \frac{\alpha}{2}t < \frac{\phi}{T_\epsilon^2 p} < t^2 + \frac{\alpha}{2}t$$

Si $u = \frac{\phi}{T_\epsilon^2 p}$, por el lema *4.9* se tiene:

$$round(\sqrt{u}) = round\left(\sqrt{\frac{\phi}{T_\epsilon^2 p}}\right) = t$$

y por lo tanto se toma:

$$g(\phi) = round\left(\sqrt{\frac{\phi}{T_\epsilon^2 p}}\right) = t$$

como la función decodificadora requerida. □

Con este resultado se obtienen las dos funciones requeridas para el canal de Ω-eliminación, de donde se debe verificar que $f(t)$ y $g(t)$ efectúan procesos inversos.

Corolario 4.11. *Si $f(t)$ y $g(t)$ son las funciones de codificación y decodificación para el canal de Ω-eliminación, entonces realizan procesos inversos, es decir:*

$$g(f(t) - \epsilon) = t$$

Demostración. Supongamos que $f(t) - \epsilon$ se toma de la siguiente forma:

$$f(t) - \epsilon = f(t) - \left(pf(t) \pm M\sqrt{f(t)}\right)$$

con

$$f(t) = \left(T_\epsilon^2 \frac{p}{1-p}\right) t^2, \qquad g(t) = round\left(\sqrt{\frac{t}{T_\epsilon^2 p}}\right)$$

y

$$0 < \alpha < 1, \qquad M = \frac{\alpha}{3} T_\epsilon \sqrt{p(1-p)}$$

entonces:

$$f(t) - \epsilon = f(t) - \left(pf(t) - \frac{\alpha}{3} T_\epsilon \sqrt{p(1-p)}\sqrt{f(t)}\right)$$

luego:

$$g(f(t)-\epsilon) = round\left(\sqrt{\frac{f(t)-\left(pf(t)-\frac{\alpha}{3}T_\epsilon\sqrt{p(1-p)}\sqrt{f(t)}\right)}{T_\epsilon^2 p}}\right)$$

$$g(f(t)-\epsilon) = round\left(\sqrt{\frac{f(t)(1-p)+\frac{\alpha}{3}T_\epsilon\sqrt{p(1-p)}\sqrt{f(t)}}{T_\epsilon^2 p}}\right)$$

$$g(f(t)-\epsilon) = round\left(\sqrt{\frac{\left(T_\epsilon^2\frac{p}{1-p}\right)t^2(1-p)+\frac{\alpha}{3}T_\epsilon\sqrt{p(1-p)}\sqrt{\left(T_\epsilon^2\frac{p}{1-p}\right)t^2}}{T_\epsilon^2 p}}\right)$$

$$g(f(t)-\epsilon) = round\left(\sqrt{\frac{T_\epsilon^2 pt^2+\frac{\alpha}{3}T_\epsilon^2 pt}{T_\epsilon^2 p}}\right)$$

$$g(f(t)-\epsilon) = round\left(\sqrt{t^2+\frac{\alpha}{3}t}\right)$$

según el lema *4.9*, tomando:

$$u = t^2+\frac{\alpha}{3}t \in \left(t^2-\alpha\frac{t}{2}, t^2+\alpha\frac{t}{2}\right)$$

se obtiene que:

$$g(f(t)-\epsilon) = round\left(\sqrt{t^2+\frac{\alpha}{3}t}\right) = t$$

lo cuál concluye que $f(t)$ y $g(t)$ son inversas. □

Se observa entonces que las funciones de codificación $f(t)$ y de decodificación $g(t)$ dependen del parámetro T_ϵ^2 que debe ser lo suficientemente grande como para garantizar que el run se expande lo suficiente como para que la eliminación aleatoria de bits no afecte tanto al run y se produzca el error en la decodificación, pero no tan grande como para que se sacrifique mucho la capacidad del canal.

Así se puede concluir el esquema propuesto del canal de Ω-eliminación, como una variante al canal de eliminación que puede ayudar a su estudio, según el siguiente diagrama:

Se desea enviar la palabra X por el canal de eliminación, con probabilidad de eliminación p por bit, donde t_i es la longitud del i-ésimo run de X luego:

Se comienza con enviar X por el canal de Ω-eliminación

$$\Downarrow$$

Se expande o codifica X a $\acute{X}$ usando la función codificadora:

$$f(t_i) = \left(T_\epsilon^2 \tfrac{p}{1-p}\right) t_i^2$$

$$\Downarrow$$

Se envía $\acute{X}$ por el canal de eliminación

$$\Downarrow$$

Se decodifica la palabra recibida $\acute{Y}$ usando una función decodificadora:

$$g(t) = round\left(\sqrt{\tfrac{\phi}{T_\epsilon^2 p}}\right),$$

donde la longitud de cada run de $\acute{Y}$ es $f(t_i - \epsilon)$

$$\Downarrow$$

La palabra decodificada recibida Y debe coincidir con la palabra enviada X, en caso contrario, hubo error en la transmisión y para solucionarlo se ajusta el valor de T_ϵ^2.

Para estimar la capacidad del canal de Ω-eliminación y su relación con la capacidad del canal de eliminación, se observa el efecto que produce la función de codificación $f(t) = kt^2$ en la longitud del run t, esto es, el valor esperado de la longitud de los runs aumentados, que sería:

$$E(P\left(l(r_i aumentados) = km^2\right)) = \sum_{m=1}^{\infty} \frac{km^2}{2^m}$$

Para estimar dicho valor, se usa el siguiente lema.

Lema 4.12. *Sea m un número entero no negativo, entonces:*

$$\sum_{m=1}^{\infty} \frac{m^2}{2^m} = 6$$

Demostración. Sea $S = \sum_{m=1}^{\infty} \frac{m^2}{2^m} = \sum_{m=0}^{\infty} \frac{m^2}{2^m}$, por otro lado $S = \sum_{m=1}^{\infty} \frac{m^2}{2^m} = \sum_{m=0}^{\infty} \frac{(m+1)^2}{2^{m+1}}$, luego:

$$S = 2S - S = 2\sum_{m=0}^{\infty} \frac{(m+1)^2}{2^{m+1}} - \sum_{m=0}^{\infty} \frac{m^2}{2^m} = \sum_{m=0}^{\infty} \frac{(m+1)^2}{2^m} - \sum_{m=0}^{\infty} \frac{m^2}{2^m}$$

$$= \sum_{m=0}^{\infty} \frac{m^2 + 2m + 1 - m^2}{2^m} = \sum_{m=0}^{\infty} \frac{2m+1}{2^m} = 2\sum_{m=0}^{\infty} \frac{m}{2^m} + \sum_{m=0}^{\infty} \left(\frac{1}{2}\right)^m$$

por el lema *4.1* se tiene:

$$\sum_{m=1}^{\infty} \frac{m^2}{2^m} = 2 \cdot 2 + \frac{1}{1 - \frac{1}{2}} = 6$$

□

Así el valor esperado de la longitud de los runs aumentados es $6k$ con $k = \left(T_\epsilon^2 \frac{p}{1-p}\right)$, por lo tanto se concluye el siguiente teorema:

Teorema 4.13. *El valor esperado de la suma de todas las longitudes de los runs aumentados en el canal de Ω-eliminación, es decir, el valor esperado de la longitud de las palabras enviadas es:*

$$E(\sum_i \left(l(r_i aumentados) = km^2\right)) = \frac{n}{2} \cdot 6k = 3kn \quad \textit{(bits)}$$

con $k = \left(T_\epsilon^2 \frac{p}{1-p}\right)$ *para* T_ϵ *lo suficientemente grande.*

Con esto se determina que se sacrifica la capacidad del canal de eliminación, al tomar el canal de Ω-eliminación en un factor de $3k$, pero se gana al disminuir el error por perdida de bits. Se obtiene así el siguiente resultado.

Corolario 4.14. *Si* $\mathbb{C}$ *es la capacidad del canal de eliminación, entonces la capacidad del canal de Ω-eliminación* $\mathbb{C}_\Omega$ *es:*

$$\mathbb{C}_\Omega \leq \mathbb{C} \leq 3k\mathbb{C}_\Omega$$

Aunque en este caso, el factor T_ϵ se incrementa al disminuir ϵ y aumentar n, se va a observar, numéricamente, que para valores *prácticos* de ϵ y n, T_ϵ permanece, de hecho, *acotado*.

4.3. Simulación del canal de Ω-eliminación

Luego de realizar la propuesta del canal de Ω-eliminación y estudiar su relación con el canal de eliminación es necesario mostrar aplicaciones concretas de dicha propuesta, en este caso modelar la anterior propuesta y realizar la respectiva implementación para observar su funcionamiento. Con todo lo anterior, se construye el siguiente algoritmo:

Sea C el diccionario de palabras a transmitir por el canal de Ω-eliminación, el cual elimina cada bit aleatoriamente y de forma independiente e idénticamente distribuida con probabilidad p y con probabilidad de recibir el bit de $(1-p)$, las cuales son conocidas tanto por el emisor como por el receptor, y supongamos que se encuentran ya en lenguaje binario, es decir:
$C = \{\text{palabras } X \text{ formadas por lenguaje binario con longitud } n \mid X = (x_1 x_2 \dots x_n)\ \ x_i \in \{0,1\},$
donde el número de runs de X es mayor que $k\sqrt{n},$ con $k = \left(T_\epsilon^2 \frac{p}{1-p}\right)\}$

1. Se emite X palabra del diccionario C.
2. Se construye el vector de runs $V_R(X) = (t_1, t_2, \dots, t_n)$.
3. Se aplica la función codificadora $f(t) = f(V_R(X)) = (f(t_1), f(t_2), \dots, f(t_n))$
4. Calcular la función $deruns(f(V_R(X)))$ para obtener el vector codificado por repetición, según la función $f(t_i)$ aplicado a la longitud de cada run.
5. Se manda la palabra extendida por el canal de eliminación el cual elimina términos de forma aleatoria sin dejar rastros de la eliminación, obteniendo $\acute{X}$ con menos bits.
6. Se construye el vector de runs $V_R(\acute{X}) = (f(t_1) - \epsilon_1, f(t_2) - \epsilon_2, \dots, f(t_n - \epsilon_n))$, en donde las potencias disminuyeron.
7. Se aplica la función decodificadora:

$$g\left(V_R(\acute{X})\right) = (g(f(t_1) - \epsilon_1), g(f(t_2) - \epsilon_2), \dots, g(f(t_n - \epsilon_n)))$$

8. Se calcula la función $deruns(f(V_R(\acute{X})))$ para obtener el vector decodificado por repetición que recibe el receptor.
9. Se compara X con $\acute{X}$ usando distancia de Hamming, si la distancia es cero hay éxito y se tiene $X = \acute{X}$, en caso contrario hay error de transmisión.
10. Si hay error en la transmisión, se reajusta T_ϵ a un valor mayor para tener más bits en la repetición, a fin de evitar el error.

El código de la implementación en *Matlab 8*, discriminado por la funciones antes mencionadas y donde para facilitar algunos cálculos se emplea el bit -1 en lugar del bit 0, es el siguiente:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  Programa que emula el canal de eliminación   %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%     Se ingresa la palabra W a transmitir       %
%             en binario con 0 => -1             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tic
%runs
w=[1,-1,1,1,-1,-1,1,1,1,1,1,-1,-1,1]
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%     La longitud de la palabra original es k    %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
k=length(w)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Se calcula la longitud de los runs de W      %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
rn=[];
fl=1;
ind=0;
cnt=0;
for c=1:1:k
    if w(c)==fl
        cnt=cnt+1;
    else
        ind=ind+1;
        rn(ind)=cnt;
        fl=-fl;
        cnt=1;
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%             Se crea el vector de runs          %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ind=ind+1;
rn(ind)=cnt;
disp(rn);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%     Se aplica la función codificadora f al     %
%     vector de runs con T_epsilon y p dados     %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%codificacion
T=10;
p=0.2;
k=(T^2)*p/(1-p);
t=rn;
f=k*t.^2;
disp(f);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%       Se obtiene el vector de potencias       %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%derun
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% La función deruns escribe el vector largo     %
%  en lenguaje binario usando codificación      %
%       por repetición alistando el envio       %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
t=f;
l=length(t)
w=[];
fl=1;
for c=1:1:l
    h=fl*ones(1,t(c));
    w=[w,h];
    fl=-fl;
end
disp(w)
y=length(w)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Se simula el envio por el canal de eliminación %
%    eliminando datos de la palabra codificada   %
%                 en forma aleatoria             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%canal
p=0.2;
k=length(w);
borrar=rand(1,k);
cnt=0;
wbar=[];
for c=1:1:k
    if borrar(c)>0.2
        cnt=cnt+1;
        wbar(cnt)=w(c);
  end
end
disp(wbar)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Se aplica la función runs a la palabra binaria %
%     recibida despues de pasar por el canal     %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%runs
w=wbar;
k=length(w)
rn=[];
fl=1;
ind=0;
cnt=0;
for c=1:1:k
```

```
    if w(c)==fl
        cnt=cnt+1;
    else
        ind=ind+1;
        rn(ind)=cnt;
        fl=-fl;
        cnt=1;
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%    Se obtiene el vector de potencias según la   %
%    longitud de los runs de la palabra recibida  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ind=ind+1;
rn(ind)=cnt;
disp(rn);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Se aplica la función decodificadora g al vector %
%         de runs de la palabra recibida           %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% decod
T=10;
p=0.2;
k=(T^2)*p;
t=rn;
g=round(sqrt(t./k));
disp(g)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Se aplica la función derun para obtener el    %
%    vector largo binario que es la palabra       %
%                    recibida                     %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% derun
t=g;
l=length(t)
w=[];
fl=1;
for c=1:1:l
    h=fl*ones(1,t(c));
    w=[w,h];
    fl=-fl;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Por último se obtiene la palabra transmitida  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
disp(w)
```

Después de mostrar el código en *Matlab 8* se muestran algunos ejemplos de su implementación realizados en un computador *AMD Athlon X2* de 64bits, con 4 gigas de memoria ram, sobre la plataforma *Windows 7*x64, se toma la misma palabra X y se varía el termino T_ϵ para observar su influencia en la transmisión, estos resultados pueden variar debido al carácter aleatorio de la eliminación, sin embargo sirven de referencia para la implementación.

Ejemplo 4.15. Sea X la palabra con 11 bits y 6 runs siguiente y $p = 0{,}2$ la probabilidad de que el canal de eliminación elimine un bit de forma aleatoria es independiente:

$$X = (10110011100)$$

la longitud de la palabra es $k = 11$, se toma el parámetro:

$$T_\epsilon = 10$$

el vector de runs es:

$$V_R(X) = (1^1\, 0^1\, 1^2\, 0^2\, 1^3\, 0^2)$$

y el vector de potencias:

$$\acute{X} = (1\,1\,2\,2\,3\,2)$$

la función codificadora por repetición $f(t)$ crea una palabra con 575 bits:

$$f(\acute{X}) = (f(1)\, f(1)\, f(2)\, f(2)\, f(3)\, f(2))$$

el vector de runs, listo para ser transmitido por el canal de eliminación es:

$$f(V_R(\acute{X})) = (1^{f(1)}\, 0^{f(1)}\, 1^{f(2)}\, 0^{f(2)}\, 1^{f(3)}\, 0^{f(2)}) = (1^{25}\, 0^{25}\, 1^{100}\, 0^{100}\, 1^{225}\, 0^{100})$$

y la palabra recibida luego de pasar por el canal, tiene 463 bits para una perdida aleatoria de 112 bits.

$$\acute{Y} = (1^{20}\, 0^{20}\, 1^{79}\, 0^{82}\, 1^{180}\, 0^{82})$$

al aplicar la función decodificadora $g(t)$, la palabra recibida Y con 11 bits y 6 runs es:

$$Y = (10110011100)$$

que coincide con la palabra enviada, luego hubo una transmisión correcta.

Se repite la misma palabra variando el termino $T_\epsilon = 8$

Ejemplo 4.16. Sea X la palabra con 11 bits y 6 runs siguiente y $p = 0{,}2$ la probabilidad de que el canal de eliminación elimine un bit de forma aleatoria es independiente:

$$X = (10110011100)$$

la longitud de la palabra es $k = 11$, se toma el parámetro:

$$T_\epsilon = 8$$

el vector de runs es:

$$V_R(X) = (1^1\,0^1\,1^2\,0^2\,1^3\,0^2)$$

y el vector de potencias:

$$\acute{X} = (1\,1\,2\,2\,3\,2)$$

la función codificadora por repetición $f(t)$ crea una palabra con 368 bits:

$$f(\acute{X}) = (f(1)\,f(1)\,f(2)\,f(2)\,f(3)\,f(2))$$

el vector de runs, listo para ser transmitido por el canal de eliminación es:

$$f(V_R(\acute{X})) = (1^{f(1)}\,0^{f(1)}\,1^{f(2)}\,0^{f(2)}\,1^{f(3)}\,0^{f(2)}) = (1^{16}\,0^{16}\,1^{64}\,0^{64}\,1^{144}\,0^{64})$$

y la palabra recibida luego de pasar por el canal, tiene 295 bits para una perdida aleatoria de 73 bits.

$$\acute{Y} = (1^{16}\,0^{12}\,1^{53}\,0^{50}\,1^{118}\,0^{52})$$

al aplicar la función decodificadora $g(t)$, la palabra recibida Y con 11 bits y 6 runs es:

$$Y = (10110011100)$$

que coincide con la palabra enviada, luego hubo una transmisión correcta.

Tomando la misma palabra pero con $T_\epsilon = 6$ se obtiene:

Ejemplo 4.17. Sea X la palabra con 11 bits y 6 runs siguiente y $p = 0{,}2$ la probabilidad de que el canal de eliminación elimine un bit de forma aleatoria es independiente:

$$X = (10110011100)$$

la longitud de la palabra es $k = 11$, se toma el parámetro:

$$T_\epsilon = 6$$

el vector de runs es:

$$V_R(X) = (1^1\, 0^1\, 1^2\, 0^2\, 1^3\, 0^2)$$

y el vector de potencias:

$$\acute{X} = (1\,1\,2\,2\,3\,2)$$

la función codificadora por repetición $f(t)$ crea una palabra con 207 bits:

$$f(\acute{X}) = (f(1)\, f(1)\, f(2)\, f(2)\, f(3)\, f(2))$$

el vector de runs, listo para ser transmitido por el canal de eliminación es:

$$f(V_R(\acute{X})) = (1^{f(1)}\, 0^{f(1)}\, 1^{f(2)}\, 0^{f(2)}\, 1^{f(3)}\, 0^{f(2)}) = (1^9\, 0^9\, 1^{36}\, 0^{36}\, 1^{81}\, 0^{36})$$

y la palabra recibida luego de pasar por el canal, tiene 160 bits para una perdida aleatoria de 47 bits.

$$\acute{Y} = (1^7\, 0^6\, 1^{31}\, 0^{31}\, 1^{58}\, 0^{27})$$

al aplicar la función decodificadora $g(t)$, la palabra recibida Y con 7 bits y 6 runs es:

$$Y = (1010110)$$

que es subpalabra de la palabra enviada, pero no coincide con esta, luego hubo una transmisión errónea.

Se observa que el termino T_ϵ se debe tomar lo suficientemente grande, para que las funciones $f(t)$ y $g(t)$ funcionen correctamente y la palabra recibida por el receptor efectivamente sea la palabra enviada y no una subpalabra.

Por otro lado, se efectuó una simulación en la que *Matlab* generaba palabras binarias aleatorias de una longitud dada n y se transmitían a través del canal de Ω-eliminación para intentar estimar la cantidad de posibles errores de transmisión con palabras de mayor longitud y observar el tiempo de compilación para determinar la medida en la que se afecta el rendimiento del algoritmo.

Para determinar el error de transmisión en la simulación se utilizó la distancia de Hamming entre las palabras enviadas y las recibidas, así es posible observar el comportamiento de T_ϵ en la transmisión.

A continuación se presenta el código de la simulación, con $T_\epsilon = 8$ para palabras de longitud $n = 100$ en 5000 transmisiones:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Programa que emula al canal de          %
%              \Omega-eliminación            %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  Las variables de la simulación son:     %
%T= T_epsilon de la función f Cod          %
%simul=Veces que se repite el envio creando %
%        palabras aleatorias distintas para  %
%        cada vez que se ejecuta, equivale a %
%        enviar palabras distintas de longitud %
%        dada                                %
%n= Longitud de la palabra binaria a        %
%    transmitir                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Para variar los datos deben cambiarse todas %
%las variables antes indicadas y deben ser   %
%                 iguales                    %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tic
%runs
T=6;
simul=500;
n=500;
errores=0;
for iter=1:1:simul
w=floor(2*rand(1,n))-1;
w=[1,w];
wini=w;
k=length(w);
rn=[];
fl=1;
ind=0;
cnt=0;
for c=1:1:k
    if w(c)==fl
        cnt=cnt+1;
```

```
    else
        ind=ind+1;
        rn(ind)=cnt;
        fl=-fl;
        cnt=1;
    end
end
ind=ind+1;
rn(ind)=cnt;
% disp(rn);
% pause;
% codificacion
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%p es la probabilidad de eliminación de cada %
%  bit de forma independiente e identicamente %
%                    distribuida    i.i.d      %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
p=0.2; %p probabilidad de eliminación
k=(T^2)*p/(1-p);
t=rn;
f=round(k*t.^2);
% disp(f);
% pause;
% derun
t=f;
l=length(t);
w=[];
fl=1;
for c=1:1:l
    if t(c)>0
        h=fl*ones(1,t(c));
        w=[w,h];
    end
    fl=-fl;
end
% disp(w)
% pause;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  Este p debe coincidir con el anterior      %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% canal
p=0.2; %p probabilidad de eliminación
k=length(w);
borrar=rand(1,k);
cnt=0;
wbar=[];
for c=1:1:k
    if borrar(c)>0.2 %%% Este tambien es p  %%%
        cnt=cnt+1;
        wbar(cnt)=w(c);
```

```
   end
end
 % disp(wbar)
 % runs
w=wbar;
k=length(w);
rn=[];
fl=1;
ind=0;
cnt=0;
for c=1:1:k
    if w(c)==fl
        cnt=cnt+1;
    else
        ind=ind+1;
        rn(ind)=cnt;
        fl=-fl;
        cnt=1;
    end
end
ind=ind+1;
rn(ind)=cnt;
 % disp(rn);
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 % Este p debe coincidir con el anterior     %
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 % decod
p=0.2; %p probabilidad de eliminación
k=(T^2)*p;
t=rn;
g=round(sqrt(t./k));
 % disp(g)
 % derun
t=g;
l=length(t);
w=[];
fl=1;
for c=1:1:l
    h=fl*ones(1,t(c));
    w=[w,h];
    fl=-fl;
end
 % disp(w)
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 %Los criterios para determinar el error     %
 %en la transmisión son:                     %
 %1. Longitud de palapra distinta            %
 %2. Distancia de Hamming diferente a 0      %
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if length(w)<length(wini)
```

```
    errores=errores+1;
elseif w~=wini
    errores=errores+1;
end
disp(sprintf('Total de Errores: %2.0f\n',errores));
end
disp(sprintf('Porcentaje de Errores: %1.0f\n',errores*100/simul));
toc
```

Después de efectuar 20 simulaciones para cada tipo de datos, generando palabras binarias aleatorias de longitud n, con una probabilidad aleatoria de eliminación de p y promediar resultados, se obtuvieron los siguientes datos:

Transmisiones de palabras de longitud $n = 100$ por el canal de Ω-eliminación, con p=0.2:

Datos / p	T_ϵ	n	Simulaciones	Errores	% de Errores	Tiempo
0.2	4	100	5000	483	9.66 %	0´24.75
0.2	5	100	5000	381	7.62 %	0´27.61
0.2	5.3	100	5000	88	1.76 %	0´28.73
0.2	5.5	100	5000	21	0.38 %	0´29.83
0.2	5.7	100	5000	16	0.32 %	0´29.95
0.2	6	100	5000	6	0.12 %	0´29.99
0.2	6.5	100	5000	5	0.10 %	0´31.17
0.2	7	100	5000	1	0.02 %	0´37.88
0.2	8	100	5000	0	0.00 %	0´54.19

Transmisiones de palabras de longitud $n = 500$ por el canal de Ω-eliminación, con p=0.2:

Datos / p	T_ϵ	n	Simulaciones	Errores	% de Errores	Tiempo
0.2	4	500	5000	1861	37.22 %	1´34.81
0.2	5	500	5000	1590	31.8 %	3´30.13
0.2	5.3	500	5000	411	8.21 %	4´34.35
0.2	5.5	500	5000	116	2.12 %	5´31.63
0.2	5.7	500	5000	107	2.34 %	6´29.23
0.2	6	500	5000	32	0.64 %	8´01.90
0.2	6.5	500	5000	23	0.46 %	11´37.3
0.2	7	500	5000	8	0.16 %	22´11.2
0.2	8	500	5000	0	0.00 %	27´19.5

Transmisiones de palabras de longitud $n = 1000$ por el canal de Ω-eliminación, con p=0.2:

Datos / p	T_ϵ	n	Simulaciones	Errores	% de Errores	Tiempo
0.2	4	1000	5000	3078	61.56 %	6´44.81
0.2	5	1000	5000	2739	54.78 %	16´58.5
0.2	5.3	1000	5000	908	18.16 %	21´56.5
0.2	5.5	1000	5000	191	3.82 %	26´09.1
0.2	5.7	1000	5000	163	3.26 %	29´24.1
0.2	6	1000	5000	51	1.02 %	33´34.11
0.2	6.5	1000	5000	49	0.98 %	52´10.03
0.2	7	1000	5000	17	0.34 %	68´34.1
0.2	8	1000	5000	0	0 %	117´47.2

Aumentando la probabilidad de eliminación p de 0,2 a 0,3 los resultados son:

Transmisiones de palabras de longitud $n = 100$ por el canal de Ω-eliminación, con p=0.3:

Datos / p	T_ϵ	n	Simulaciones	Errores	% de Errores	Tiempo
0.3	3	100	5000	2168	43.36 %	0´12.83
0.3	4	100	5000	912	18.24 %	0´19.51
0.3	4.2	100	5000	363	7.26 %	0´21.19
0.3	4.4	100	5000	354	7.08 %	0´23.69
0.3	4.7	100	5000	114	2.28 %	0´24.93
0.3	5	100	5000	17	0.34 %	0´27.16
0.3	5.5	100	5000	11	0.22 %	0´32.96
0.3	6	100	5000	2	0.04 %	0´41.77
0.3	7	100	5000	0	0.00 %	1´12.19

Transmisiones de palabras de longitud $n = 500$ por el canal de Ω-eliminación, con p=0.3:

Datos / p	T_ϵ	n	Simulaciones	Errores	% de Errores	Tiempo
0.3	3	500	5000	3129	62.58 %	1´57.21
0.3	4	500	5000	2376	47.52 %	4´11.32
0.3	4.2	500	5000	1431	28.62 %	4´14.65
0.3	4.4	500	5000	1270	25.41 %	4´27.54
0.3	4.7	500	5000	503	10.06 %	4´50.83
0.3	5	500	5000	68	1.36 %	8´59.82
0.3	5.5	500	5000	42	0.84 %	13´38.12
0.3	6	500	5000	13	0.26 %	19´13.43
0.3	7	500	5000	0	0.00 %	36´47.42

Transmisiones de palabras de longitud $n = 1000$ por el canal de Ω-eliminación, con p=0.3:

Datos / p	T_ϵ	n	Simulaciones	Errores	% de Errores	Tiempo
0.3	3	1000	5000	4986	99.72 %	4´54.91
0.3	4	1000	5000	4287	85.74 %	16´19.7
0.3	5	1000	5000	503	10.06 %	49´59.18
0.3	5.3	1000	5000	473	9.46 %	50´35.72
0.3	5.5	1000	5000	184	3.68 %	58´23.12
0.3	5.7	1000	5000	61	1.22 %	66´47.21
0.3	6	1000	5000	25	0.50 %	81´39.4
0.3	6.5	1000	5000	15	0.30 %	118´55´93
0.3	7	1000	5000	0	0.00 %	165´15.64

Por último, el comportamiento del canal de Ω-eliminación con $p = 0{,}5$, para $n = 100$ es el siguiente:

Transmisiones de palabras de longitud $n = 100$ por el canal de Ω-eliminación, con p=0.5:

Datos / p	T_ϵ	n	Simulaciones	Errores	% de Errores	Tiempo
0.5	3	100	5000	error	? %	?
0.5	3.3	100	5000	error	? %	?
0.5	3.5	100	5000	error	? %	?
0.5	3.6	100	5000	439	8.78 %	1´11.35
0.5	3.7	100	5000	297	5.94 %	1´18.21
0.5	3.8	100	5000	224	4.47 %	1´22.15
0.5	4	100	5000	76	1.52 %	1´27.5
0.5	5	100	5000	21	0.42 %	3´05.01
0.5	6	100	5000	2	0.04 %	6´27.04
0.5	7	100	5000	0	0 %	10´36.05

En este último caso se observa el hecho de que si p aumenta, T_ϵ puede ser tomado más pequeño pues el factor $\frac{p}{1-p}$ es creciente al acercarse a 1, sin embargo al aumentar p por arriba de 0,5 las simulaciones comienzan a ser cada vez más inestables y demoradas y tienden a generar muchos errores de ejecución, tantos que no es posible recopilar una tabla que muestre estas cantidades, salvo disminuir las simulaciones, entendiéndose que a pesar de las repeticiones, el nivel de errores es bastante alto y la transmisión no es confiable o factible.

Por lo tanto no vale la pena aumentar la longitud de la palabra X a transmitir en virtud a los resultados obtenidos con $n = 100$.

4.4. Conclusiones

Después de efectuar las diversas simulaciones y de obtener los anteriores resultados, es posible sacar algunas conclusiones del presente trabajo, las principales que se destacan son:

1. El canal de Ω-eliminación es una herramienta que puede ser útil a la hora de enfrentarse al problema del cálculo de la capacidad del canal de eliminación, pues están claramente relacionadas la una con la otra en virtud del corolario 4.14, ya que al quitar la expansión de los runs propuesta por Mitzenmacher en [4] que es de tipo poisson y aleatoria y sustituyéndola por una forma cuadrática conveniente y predecible, se podría buscar una forma explícita de la capacidad usando esta codificación, siguiendo las ideas expuestas por Fertonani y Montanari en [9] y [16] respectivamente.

2. El código por repetición obtenido, (*incluye funciones codificadora y decodificadora*) es optimo en el sentido de dar buenos resultados y una baja tasa de errores, que es susceptible de mejoras con solo modificar un termino.

3. Un trabajo que podría desarrollarse a futuro es, usando las funciones de codificación y decodificación del canal de Ω-eliminación, intentar usar la estimación de la capacidad propuesta por Mitzenmacher, Diggavi, Drinea en [10] junto con una posible implementación del algoritmo de Arimoto-Blahut [13] y [14], (que explota la concavidad de la función de información mutua), para buscar cotas mejores de la capacidad del canal de eliminación.

4. Se compilaron los códigos posteriormente en una maquina *MacBook Pro Core 2 Duo* a 2.33 Ghz con 3Gb de ram, bajo Mac OsX 10.6.8 y *Matlab 12*, obteniendo una considerable mejoría en los tiempos de compilación, con una reducción de tiempo cercana al 80 %, luego se observa que el código es muy eficiente y su demora está más en la maquina y el compilador que en sí mismo.

5. Una variante aún por estudiar es el canal de eliminación e inserción, donde no solo se eliminan bits de forma aleatoria sin dejar rastro; sino que además se producen inserciones aleatorias de bits afectando la transmisión en dos frentes, su capacidad al día de hoy también es desconocida, los trabajos de Mirghasemi y Tchamkerten en [15] y de Fertonani, Duman y Erden en [9] están enfocados en esa dirección.

Un artículo muy interesante y que muestra el estado actual del estudio del canal de eliminación es [5] de Mitzenmacher, donde se compilan algunos avances y las muchas dudas que el canal de eliminación tiene pendientes por resolver. También se recomienda el estudio de [11] de Venkataramanan y Tatikonda que ilustra los más recientes intentos por estimar la capacidad del canal de eliminación.

Anexo

Tabla 1.

k	$\binom{n}{k}$	$\binom{n}{k}p^k(1-p)^{n-k}$	$p^k(1-p)^{n-k}$	$-\frac{1}{25}\log_2 p^k(1-p)^{n-k}$
0	1	0,0000000001	0,0000000001	1,321928095
1	25	0,0000000042	0,0000000002	1,298529595
2	300	0,0000000760	0,0000000003	1,275131095
3	2300	0,0000008740	0,0000000004	1,251732595
4	12650	0,0000072103	0,0000000006	1,228334095
5	53130	0,0000454251	0,0000000009	1,204935595
6	177100	0,0002271255	0,0000000013	1,181537095
7	480700	0,0009247253	0,0000000019	1,158138595
8	1081575	0,0031209477	0,0000000029	1,134740095
9	2042975	0,0088426853	0,0000000043	1,111341595
10	3268760	0,0212224446	0,0000000065	1,087943095
11	4457400	0,0434095458	0,0000000097	1,064544595
12	5200300	0,0759667051	0,0000000146	1,041146095
13	5200300	0,1139500577	0,0000000219	1,017747595
14	4457400	0,1465072170	0,0000000329	0,994349094
15	3268760	0,1611579387	0,0000000493	0,970950594
16	2042975	0,1510855675	0,0000000740	0,947552094
17	1081575	0,1199797154	0,0000001109	0,924153594
18	480700	0,0799864769	0,0000001664	0,900755094
19	177100	0,0442030530	0,0000002496	0,877356594
20	53130	0,0198913739	0,0000003744	0,853958094
21	12650	0,0071040621	0,0000005616	0,830559594
22	2300	0,0019374715	0,0000008424	0,807161094
23	300	0,0003790705	0,0000012636	0,783762594
24	25	0,0000473838	0,0000018954	0,760364094
25	1	0,0000028430	0,0000028430	0,736965594

Bibliografía

[1] Shannon, C.E. *A Mathematical Theory of Communication.* Bell Systems Technical Journal, 27(3):379-423, MR0026286, 1948.

[2] Dobrushin, R.L. *Shannons Theorems for Channels with Synchronization Errors.* Problems of Information Transmission, 3(4):11-26, 1967. Translated from Problemy Peredachi Informatsii, vol. 3, no. 4, pp. 18-36, MR0289198, 1967.

[3] Hartley, R.V.L., *Transmission of Information.* Bell System Technical Journal, Volume 7, Number 3, pp. 535-563, 1928.

[4] Mitzenmacher, M. and Drinea, E. *A Simple Lower Bound for the Capacity of the Deletion Channel.* IEEE Transactions on Information Theory, 52:10, pp. 4657-4660, MR2300848, 2006.

[5] Mitzenmacher, M. *A Survey of Results for Deletion Channels and Related Synchronization Channels*, Probab. Surveys, 6, 1-33, 2009.

[6] Drinea, E. and Mitzenmacher, M. *On Lower Bounds for the Capacity of Deletion Channels.* IEEE Transactions on Information Theory, 52:10, pp. 4648-4657, MR2300847, 2006.

[7] Drinea, E. and Mitzenmacher, M. *Improved Lower Bounds for the Capacity of i.i.d. Deletion and Duplication Channels.* IEEE Transactions on Information Theory, 53:8, pp. 2693-2714, MR2400490, 2007.

[8] Fertonani, D. and Duman, T.M. *Novel Bounds on the Capacity of Binary Channels with Deletions and Substitutions*, In Proceedings of the 2009 IEEE International Symposium on Information Theory (ISIT), 2009.

[9] Fertonani, D. Duman, T. M. and Erden, M. F. *Bounds on the Capacity of Channels with Insertions, Deletions and Substitutions*, IEEE Trans. on Communications, vol. 59, no. 1, pp. 2-6, 2011.

[10] Diggavi, S., Mitzenmacher, M. and Pfister, H. *Capacity Upper Bounds for Deletion Channels.* In Proceedings of the International Symposium on Information Theory, pp. 1716-1720, Nice, France, June 2007.

[11] Venkataramanan, R and Tatikonda, S. *Achievable Rates for Channels with Deletions and Insertions.* Cornell University Library, abs/1102.5112, arXiv:1102.5112 [cs.IT] 2011.

[12] Rahmati, M. and Duman, T. *An Upper Bound on the Capacity of non-Binary Deletion Channels*, Cornell University Library, arXiv:1301.6599 [cs.IT], 2013.

[13] Arimoto, S. *An Algorithm for Calculating the Capacity of an Arbitrary Discrete Memoryless Channel*, IEEE Trans. Inform. Theory, vol. 18, pp. 14-20, Jan. 1972.

[14] Blahut, R. E. *Computation of Channel Capacity and Rate Distortion Functions*, IEEE Trans. Inform. Theory, vol. 18, pp. 460-473, Jan. 1972.

[15] Mirghasemi, H. and Tchamkerten, A. *On the Capacity of the One-Bit Deletion and Duplication Channel*, Cornell University Library, arXiv:1210.2704 [cs.IT], 2012.

[16] Kanoria, Y. and Montanari, A. *Optimal coding for the deletion channel with small deletion probability.* Cornell University Library, arXiv:1104.5546 [cs.IT], 2011.

[17] Gray, R. *Entropy and Information Theory.* Springer, New York, 2011.

[18] Papoulis, A. *Probability, Random Variables and Sthocastic Processes.* New York, NY: McGraw-Hill, 1991.

[19] Rudin, W. *Principles of Mathematical Analysis.* New York: McGraw-Hill, 1974.

[20] Cover, T. M. and Thomas, J. A. *Elements of Information Theory.* New York: John Wiley Sons, Inc. 2nd end, 2006.