Notas de TIC

Índice

1. Entropía

La *Entropía* es una medida de la incerteza de una variable aleatoria, o dicho de otra forma, es una medida de la cantidad de información necesaria (en promedio) para describir una variable aleatoria. Sea X una variable aleatoria que toma valores $x \in \mathcal{X}$, la entropía se define como:

$$\boxed{H(X) = -\sum_{x \in \mathcal{X}} p(x) \log(p(x)) = \sum_{x \in \mathcal{X}} p(x) \log\left(\frac{1}{p(x)}\right)}$$

En caso de que el logaritmo se encuentre en base 2, la entropía se mide en bits. Se puede ver que la entropía no depende de los valores específicos de X, sino que solo depende de las probabilidades. Por otra parte, se puede ver que la entropía es muy similar a la esperanza de una función $g(X)$:

$$\boxed{H(X) = -\mathbb{E}[\log(p(X))] = \mathbb{E}\left[\log\left(\frac{1}{p(X)}\right)\right]}$$

Así como se definió la entropía para una única variable aleatoria, es posible extender la definición para un par de variables aleatorias, dando lugar a la *Entropía Conjunta*:

$$\boxed{H(X,Y) = -\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x,y) \log(p(x,y)) = -\mathbb{E}[\log(p(x,y))]}$$

Por otra parte, también es posible definir la *Entropía Condicional* como:

$$H(Y|X) = \sum_{x \in \mathcal{X}} p(x) H(Y|X=x) = -\sum_{x \in \mathcal{X}} p(x) \sum_{y \in \mathcal{Y}} p(y|x) \log(p(y|x))$$
$$H(Y|X) = -\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x,y) \log(p(y|x))$$

Por lo tanto la entropía condicional queda como:

$$\boxed{H(Y|X) = -\mathbb{E}[\log(p(Y|X))]}$$

Propiedades de la entropía

Algunas propiedades de la entropía son:

- $H(X) \geq 0$.
- $H_b(X) = H_a(X) \log_b(a)$.
- Regla de la cadena: $H(X,Y) = H(X) + H(Y|X) = H(Y) + H(X|Y)$.
- Corolario de la regla de la cadena: $H(X,Y|Z) = H(X|Z) + H(Y|X,Z)$.
- Generalización de la regla de la cadena:

$$\boxed{H(X_1, X_2, ..., X_n) = \sum_{i=1}^{n} H(X_i|X_{i-1}, ..., X_1)}$$

- Si $Y = g(X)$ entonces $H(Y|X) = 0$.

1.1. Entropía relativa e Información Mutua

La *Entropía Relativa o Distancia de Kullback-Leibler* es una medida de la distancia entre dos distribuciones. La distancia $D(p \parallel q)$ es una medida del error de asumir que la distribución es $q(x)$, cuando la verdadera distribución es $p(x)$. Por lo tanto, se la puede expresar como:

$$\boxed{D(p \parallel q) = \sum_{x \in \mathcal{X}} p(x) \log\left(\frac{p(x)}{q(x)}\right) = \mathbb{E}_p\left[\log\left(\frac{p(X)}{q(X)}\right)\right] = \mathbb{E}_p\left[-\log\left(q(X)\right)\right] - \mathbb{E}_p\left[-\log\left(p(X)\right)\right]}$$

Como se trata de una distancia, se cumple que $D(p \parallel q) \geq 0$, cumpliéndose la igualdad solo en aquellos casos donde $p(x) = q(x)$[1]. Sin embargo, a diferencia de la distancia real, esta definición no es simétrica, cumpliéndose que $D(p \parallel q) \neq D(q \parallel p)$.

Se puede definir también a la *Entropía relativa condicional* como:

$$\boxed{D(p(y|x) \parallel q(y|x)) = \sum_{x \in \mathcal{X}} p(x) \sum_{y \in \mathcal{Y}} p(y|x) \log\left(\frac{p(y|x)}{q(y|x)}\right) = \mathbb{E}_{p(x,y)}\left[\log\left(\frac{p(Y|X)}{q(Y|X)}\right)\right]}$$

La *Información Mutua* es una medida de la cantidad de información que contiene una variable aleatoria sobre otra variable aleatoria, por lo que muestra la reducción en la aleatoriedad de una variable conociendo otra. La información mutua se define entonces como:

$$\begin{aligned} I(X;Y) &= \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x,y) \log\left(\frac{p(x,y)}{p(x)\,p(y)}\right) \\ I(X;Y) &= D(p(x,y) \parallel p(x)\,p(y)) \\ I(X;Y) &= \mathbb{E}_{p(x,y)}\left[\log\left(\frac{p(X,Y)}{p(X)\,p(Y)}\right)\right] \end{aligned}$$

Debido a que la información mutua se define como una distancia, se sabe que $I(X;Y) \geq 0$, cumpliéndose la igualdad en aquellos casos en que X e Y son independientes.

La *Información Mutua Condicional* se la puede expresar como:

$$\boxed{I(X;Y|Z) = H(X|Z) - H(X|Y,Z) = \mathbb{E}_{p(x,y,z)}\left[\log\left(\frac{p(X,Y|Z)}{p(X|Z)\,p(Y|Z)}\right)\right]}$$

Propiedades de la Entropía relativa y de la Información Mutua

- $I(X;Y) = H(X) - H(X|Y)$.
- $I(X;Y) = H(Y) - H(Y|X)$.
- $I(X;Y) = H(X) + H(Y) - H(X,Y)$.
- $I(X;Y) = I(Y;X)$.
- $I(X;X) = H(X)$.
- Regla de la cadena para la distancia:

$$\boxed{D(p(x,y) \parallel q(x,y)) = D(p(x) \parallel q(x)) + D(p(y|x) \parallel q(y|x))}$$

- Regla de la cadena de la Información Mutua:

$$\boxed{I(X_1, X_2, ...X_n; Y) = \sum_{i=1}^{n} I(X_i; Y|X_{i-1}, X_{i-2}, ..., X_1)}$$

- Regla de la cadena de la Entropía Relativa:

$$\boxed{D(p(x,y) \parallel q(x,y)) = D(p(x) \parallel q(x)) + D(p(y|x) \parallel q(y|x))}$$

[1] Se conoce como la Desigualdad de la Información: $D(p \parallel q) \geq 0 \rightarrow D(p \parallel q) = 0 \iff p = q$.

1.2. Desigualdad de Jensen

Para poder expresar esta desigualdad, es necesario primero hacer dos definiciones:

Definición: Una función $f(x)$ es *convexa* en un intervalo (a, b) si para todo $x_1, x_2 \in (a, b)$ y $0 \leq \lambda \leq 1$ se cumple que:

$$\boxed{f\left(\lambda x_1 + (1-\lambda)\, x_2\right) \leq \lambda f\left(x_1\right) + (1-\lambda)\, f\left(x_2\right)}$$

En caso de que se cumpla la igualdad y que $\lambda = 1$ o $\lambda = 0$, se dice que $f(x)$ es *estrictamente convexa*. Otra forma de definirlo es: sea $f(x)$ posee una segunda derivada no negativa (positiva) dentro del intervalo, la función es estrictamente convexa en ese intervalo.

Definición: Una función $f(x)$ es *cóncava* si $-f\left(x\right)$ es convexa.

Con estas definiciones, es posible expresar la desigualdad de Jensen: Si $f(x)$ es una función convexa y X es una variable aleatoria cualquiera, entonces:

$$\boxed{\mathbb{E}\left[f\left(X\right)\right] \geq f\left(\mathbb{E}\left[X\right]\right)}$$

En caso de que se cumpla la igualdad, siendo $f(x)$ estrictamente convexa, se cumple entonces que $\mathbb{E}\left[X\right] = X$ con probabilidad 1.

A partir de esta desigualdad (y algunas propiedades mencionadas anteriormente), es posible demostrar algunos de los siguientes teoremas:

Teorema: Sea $|\mathcal{X}|$ la cantidad de elementos en el rango de X, entonces se cumple que:

$$\boxed{H\left(X\right) \leq \log\left(|\mathcal{X}|\right)}$$

Donde se cumple la igualdad solo en aquellos casos donde X tiene una distribución uniforme.

Teorema: Condicionar una variable aleatoria, reduce la entropía:

$$\boxed{H\left(X|Y\right) \leq H\left(X\right)}$$

Donde se cumple la igualdad si y solo si X e Y son independientes.

Teorema: Sea $X_1, X_2, ...X_n$ con probabilidad conjunta $p\left(x_1, x_2, ..., x_n\right)$ entonces:

$$\boxed{H\left(X_1, X_2, ..., X_n\right) \leq \sum_{i=1}^{n} H\left(X_i\right)}$$

Donde se cumple la igualdad si y solo si las X_i son independientes.

1.3. Desigualdad log-sum

Sean $a_1, a_2, ..., a_n$ y $b_1, b_2, ..., b_n$ números no negativos, entonces se cumple que:

$$\boxed{\sum_{i=1}^{n} a_i \log\left(\frac{a_i}{b_i}\right) \geq \left(\sum_{i=1}^{n} a_i\right) \log\left(\frac{\sum_{i=1}^{n} a_i}{\sum_{i=1}^{n} b_i}\right)}$$

Donde solo se cumple la igualdad si y solo si $a_i/b_i = \text{cte}$.

A partir de esta desigualdad, es posible demostrar los siguientes teoremas:

Teorema: $H\left(p\right)$ es una función cóncava con respecto a p.

Teorema: La información mutua $I\left(X;Y\right)$ es una función cóncava con respecto a $p(x)$ para $p(y|x)$ fija, y es una función convexa con respecto a $p(y|x)$ para $p(x)$ fija.

1.4. Desigualdad del procesamiento de datos

Para poder mostrar esta desigualdad, es necesario realizar la siguiente definición:

Definición: Las variables aleatorias X, Y y Z forman una *Cadena de Markov* en ese orden ($X \to Y \to Z$), si la distribución condicional de Z depende únicamente de Y y es condicionalmente independiente de X. Por lo tanto, la función de probabilidad conjunta se puede escribir como:

$$p(x, y, z) = p(x)\, p(y|x)\, p(z|y)$$

Otra forma de pensarlo, es decir que la probabilidad de un evento dados todos los eventos anteriores es igual a la probabilidad de dicho evento dado el evento inmediatamente anterior:

$$\mathbb{P}(X_n|X_{n-1}, ..., X_{n-\infty}) = \mathbb{P}(X_n|X_{n-1})$$

Con esta definición, es posible enunciar la *Desigualdad del procesamiento de datos*: Sea $X \to Y \to Z$ una cadena de Markov, entonces:

$$\boxed{I(X;Y) \geq I(X;Z)}$$

A partir de esta desigualdad se derivan las siguientes consecuencias:

Corolario: Si $Z = g(Y)$, se cumple que $I(X;Y) \geq I(X; g(Y))$.

Corolario: Sea $X \to Y \to Z$, entonces se cumple que $I(X;Y|Z) \leq I(X;Y)$.

1.5. Desigualdad de Fano

Mediante esta desigualdad es posible estimar el error que se comete al intentar estimar una variable aleatoria X a partir de una variable aleatoria conocida Y.

Se tiene una variable aleatoria X con una distribución $p(x)$ que toma valores en $\mathcal{X}$, pero se observa a la variable aleatoria Y que se relaciona con X a través de la probabilidad condicional $p(y|x)$. De los valores de Y, es posible realizar una estimación $g(Y) = \hat{X}$ de X, la cual toma valores en $\hat{\mathcal{X}}$[2]. Con estas definiciones, la probabilidad de error es:

$$E = \begin{cases} 1 & \hat{X} \neq X \\ 0 & \hat{X} = X \end{cases} \to \mathbb{P}_E = \mathbb{P}\left(\hat{X} \neq X\right)$$

La desigualdad de Fano establece que: Para cualquier estimador $\hat{X}$ tal que $X \to Y \to \hat{X}$, $\mathbb{P}_E = \mathbb{P}\left(\hat{X} \neq X\right)$, entonces:

$$\boxed{H(\mathbb{P}_E) + \mathbb{P}_E \log(|\mathcal{X}|) \geq H\left(X|\hat{X}\right) \geq H(X|Y)}$$

Esta desigualdad se puede escribir también como:

$$\boxed{1 + \mathbb{P}_E \log(|\mathcal{X}|) \geq H(X|Y)} \qquad \boxed{\mathbb{P}_E \geq \frac{H(X|Y) - 1}{\log(|\mathcal{X}|)}}$$

Algunos corolarios de esta desigualdad son:

Corolario: Sean X y X' dos variables aleatorias iid con entropía $H(X)$, entonces:

$$\boxed{\mathbb{P}(X = X') \geq 2^{-H(X)}}$$

Donde se cumple la igualdad si y solo si X es una variable aleatoria uniforme.

Corolario: Sean X y X' dos variables aleatorias iid con $X \sim p(x)$, $X' \sim r(x')$ y $x, x' \in \mathcal{X}$, entonces:

$$\boxed{\begin{cases} \mathbb{P}(X = X') \geq 2^{-H(p) - D(p||r)} \\ \mathbb{P}(X = X') \geq 2^{-H(r) - D(r||p)} \end{cases}}$$

[2]Se observa que $X \to Y \to \hat{X}$ forman una cadena de Markov.

2. Propiedad de la equipartición asintótica (AEP)

La Ley Débil de los Grandes Números establece que para una sucesión de variables aleatorias iid, entonces el promedio converge en probabilidad a la media:

$$\overline{X}_n = \sum_{i=1}^{n} X_i \rightarrow \lim_{n\to\infty} \mathbb{P}\left(\left|\overline{X}_n - \mathbb{E}\left[X\right]\right| < \varepsilon\right) = 1$$

A partir de este resultado, la AEP establece que para una sucesión de variables aleatorias iid con probabilidad conjunta $p\left(X_1, ..., X_n\right)$, entonces $-\frac{1}{n}\log\left(p\left(X_1, ..., X_n\right)\right)$ converge a la entropía $H(X)$, y dicha probabilidad $p\left(X_1, ..., X_n\right)$ es cercana a $2^{-nH(X)}$.

Este conjunto de variables aleatorias, puede ser dividido en dos subconjuntos: el *conjunto típico* donde las variables aleatorias poseen una entropía cercana a la entropía real, y el *conjunto no típico* donde se encuentra el resto de las variables aleatorias.

A continuación se formalizan estas definiciones.

Definición: Convergencia de variables aleatorias

Sea una secuencia $X_1, X_2, ...X_n$, se dice que dicha secuencia converge a X si:

- Convergencia en probabilidad: Para cualquier $\varepsilon > 0$ se cumple que $\mathbb{P}\left(|X_n - X| > \varepsilon\right) \rightarrow 0$.
- Convergencia en media cuadrática: $\mathbb{E}\left[\left(X_n - X\right)^2\right] \rightarrow 0$.
- Con probabilidad 1 (casi segura): $\mathbb{P}\left(\lim_{n\to\infty} X_n = X\right) = 1$.

Teorema (AEP): Como se menciono anteriormente, si se tiene una secuencia de variables aleatorias iid con función de distribución $p(x)$, entonces:

$$\boxed{-\frac{1}{n}\log\left(p\left(X_1, ..., X_n\right)\right) \rightarrow H\left(X\right)} \quad \text{en probabilidad}$$

Definición: El conjunto típico $A_\varepsilon^{(n)}$ con respecto a $p(x)$, es el conjunto de secuencias $(x_1, ..., x_n) \in \mathcal{X}^n$ que cumplen que:

$$\boxed{2^{-n(H(X)+\varepsilon)} \leq p\left(x_1, ..., x_n\right) \leq 2^{-n(H(X)-\varepsilon)}}$$

Algunas propiedades del conjunto típico son:

- Si $(x_1, ..., x_n) \in A_\varepsilon^{(n)}$, entonces:

$$\boxed{H(X) - \varepsilon \leq -\frac{1}{n}\log\left(p\left(x_1, ..., x_n\right)\right) \leq H\left(X\right) + \varepsilon}$$

- $\mathbb{P}\left(A_\varepsilon^{(n)}\right) > 1 - \varepsilon$ para n tendiendo a infinito.
- $\left|A_\varepsilon^{(n)}\right| \leq 2^{n(H(X)+\varepsilon)}$, donde $\left|A_\varepsilon^{(n)}\right|$ indica la cantidad de elementos dentro del conjunto típico.
- $\left|A_\varepsilon^{(n)}\right| \geq \left(1 - \varepsilon\right) 2^{n(H(X)-\varepsilon)}$ para n tendiendo a infinito.

2.1. Consecuencias del AEP

Debido a que el conjunto típico acumula casi toda la probabilidad (para n grande), al codificar aquellos símbolos que pertenecen al conjunto típico es posible utilizar descripciones mas cortas que aquellas utilizadas para valores fuera del conjunto típico, ya que como estos últimos son poco probables no es esencial optimizar su codificación.

Teorema: Sea $X^n = (X_1, ..., X_n)$ la *extensión de la fuente*, cuya longitud es $l(X^n)$. Para n grandes y $\varepsilon > 0$, se sabe que $\mathbb{P}\left(A_\varepsilon^{(n)}\right) > 1 - \varepsilon$, por lo que la longitud promedio esperada del código es:

$$\boxed{\mathbb{E}\left[\frac{1}{n} l(X^n)\right] \leq H(X) + \varepsilon}$$

Por lo tanto, es posible representar a la secuencia X^n completa usando un promedio de $nH(X)$ bits.

Otra propiedad importante del conjunto típico, es que a pesar de acumular casi toda la probabilidad del conjunto a medida que aumenta n, el tamaño del conjunto típico es mucho mas chico que el tamaño del conjunto total. En particular, el conjunto típico posee la misma cantidad de elementos que el conjunto mas pequeño, a primer orden en el exponente.

2.2. Algunas desigualdades

2.2.1. Desigualdad de Markov

Para cualquier variable aleatoria X no negativa y $t > 0$, se cumple que:

$$\boxed{\mathbb{P}(X \geq t) \leq \frac{\mathbb{E}[X]}{t}}$$

2.2.2. Desigualdad de Chebyshev

Sea Y una variable aleatoria de media μ y varianza σ^2 y sea $X = (Y - \mu)^2$. Para cualquier $\varepsilon > 0$ se cumple que:

$$\boxed{\mathbb{P}(|Y - \mu| > \varepsilon) \leq \frac{\sigma^2}{\varepsilon^2}}$$

2.2.3. Ley débil de los grandes números

Sea $Z_1, Z_2, ...Z_n$ una secuencia de variables aleatorias iid, con media μ y varianza σ^2. Sea $\overline{Z}_n = \frac{1}{n}\sum_{i=1}^n Z_i$ el promedio muestral, se cumple que:

$$\boxed{\mathbb{P}\left(\left|\overline{Z}_n - \mu\right| > \varepsilon\right) \leq \frac{\sigma^2}{n\varepsilon^2}}$$

Se puede ver que para $n \to \infty$, esta probabilidad tiende a cero.

2.3. Cadenas de Markov

Para poder definir este tipo de variables aleatorias es necesario realizar algunas definiciones:

Definición: Un proceso estocástico se dice *estacionario* si la probabilidad conjunta de cualquier secuencia de variables aleatorias es invariante con respecto a cualquier desplazamiento en el tiempo:

$$\mathbb{P}(X_1 = x_1, X_2 = x_2, ..., X_n = x_n) = \mathbb{P}(X_{1+\ell} = x_1, X_{2+\ell} = x_2, ..., X_{n+\ell} = x_n)$$

Definición: Un proceso estocástico discreto $X_1, X_2, ..., X_n$ compone una *cadena de Markov* o *un proceso de Markov* si se cumple que:

$$\mathbb{P}(X_{n+1} = x_{n+1} | X_n = x_n, ..., X_1 = x_1) = \mathbb{P}(X_{n+1} = x_{n+1} | X_n = x_n)$$

La probabilidad conjunta se puede escribir:

$$p(x_1, x_2, ..., x_n) = p(x_1)\, p(x_2|x_1)\, p(x_3|x_2) ... p(x_n|x_{n-1})$$

Definición: Se dice que una cadena de Markov es *invariante en el tiempo* si la probabilidad condicional $p(x_{n+1}|x_n)$ no depende de n:

$$\mathbb{P}(X_{n+1} = b | X_n = a) = \mathbb{P}(X_2 = b | X_1 = a)$$

Por lo tanto, se tiene que una cadena de Markov invariante en el tiempo depende únicamente de las condiciones iniciales del problema, y de la *matriz de transición de probabilidad* $P = [P_{ij}]$ con $i, j \in \{1, 2...m\}$, donde $P_{ij} = \mathbb{P}(X_{n+1} = j | X_n = i)$. Por lo tanto, conociendo la probabilidad $p(x_n)$, es posible conocer:

$$p(x_n + 1) = \sum_{x_n} p(x_n)\, P_{x_n, x_{n+1}}$$

2.4. Tasa de entropía

La *tasa de entropía* se puede definir como la tasa de crecimiento de la entropía conjunta de una secuencia de n variables aleatorias, a medida que n crece. Se define entonces como:

$$\boxed{H_r(X) = \lim_{n\to\infty} \frac{1}{n} H(X_1, X_2, ..., X_n)}$$

Esta definición da una idea de la entropía por símbolo dada una secuencia de n variables aleatorias. Otra definición equivalente para la tasa de entropía es:

$$\boxed{H'_r(X) = \lim_{n\to\infty} H(X_n|X_{n-1}, X_{n-2}, ..., X_1)}$$

Esta definición da una idea de la entropía de la ultima variable aleatoria dado el pasado. A partir de estas definiciones es posible expresar algunos teoremas.

Teorema: Para un proceso estocástico estacionario, se cumple que:

$$H_r(X) = H'_r(X)$$

Teorema: Para una cadena de Markov estacionaria, se cumple que:

$$H_r(X) = H'_r(X) = X(X_2|X_1)$$

Teorema: Sea X una cadena de Markov con una función de distribución estacionaria p y matriz de transición P, se cumple que:

$$H_r(X) = -\sum_{ij} p_i P_{ij} \log(P_{ij})$$

3. Compresión de datos

La compresión de datos puede ser lograda asignando a aquellos símbolos mas probables, la representación mas corta, mientras que aquellos símbolos poco probables pueden ser representados por descripciones mas largas. Para poder encontrar la mejor representación, es decir aquella que mejor comprime la información, es necesario mostrar algunas definiciones.

Definición: Una *código* C para una variable aleatoria X, es un mapeo de $\mathcal{X}$ a $\mathcal{D}^*$, donde $\mathcal{D}^*$ representa el conjunto de símbolos utilizados para codificar. Se puede definir entonces a $C(x_i)$ como la codificación de x_i, y a $l(x_i)$ como la longitud de $C(x_i)$.

Definición: La longitud promedio $L(C)$ de una fuente $C(x)$, para una variable aleatoria X con una función de distribución $p(x)$, se puede expresar como:

$$\boxed{L(C) = \sum_{x\in\mathcal{X}} p(x)\, l(x)}$$

Definición: La *extensión* C^* del código C, es un mapeo de la extensión de la fuente, la cual es puede pensar como una concatenación de los códigos correspondientes a cada símbolo:

$$\boxed{C(x_1, x_2, ..., x_n) = C(x_1)\, C(x_2) ... C(x_n)}$$

Los códigos se puede clasificar en:

- *Códigos Singulares*: La codificación no es única para toda extensión de la fuente, es decir que hay símbolos que poseen la misma codificación.
- *Códigos no Singulares*: La codificación es única para cualquier extensión de la fuente:

$$x_i \neq x_j \Longrightarrow C(x_i) \neq C(x_j)$$

A partir de esta clasificación, se pueden hacer las siguientes definiciones:

Definición: Un código se dice *unívocamente decodificable* si su extensión es no singular, es decir que no hay ambigüedades para la decodificación.

Definición: Un código se dice *libre de prefijos* o *instantáneo* si ningún código es prefijo de otro. Esto permite que cualquier código sea identificado instantáneamente sin la necesidad de esperar a los códigos que vienen después.

En la figura 3.1 se tiene un resumen de los tipos de códigos.

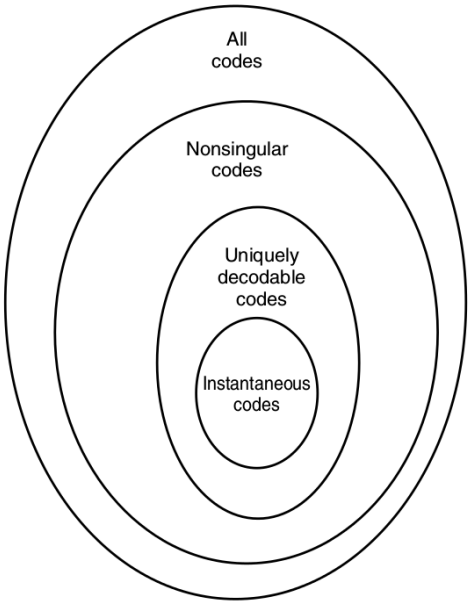

Figura 3.1: Tipos de códigos.

3.1. Desigualdad de Kraft

Los que se busca siempre es encontrar aquellos códigos instantáneos que presentan la mínima longitud promedio. Esta desigualdad establece que para cualquier código instantáneo con un alfabeto de tamaño D[3], donde cada codificación posee una longitud $l_1, l_2, ..., l_n$, se cumple que:

$$\boxed{\sum_{i=1}^{n} D^{-l_i} \leq 1}$$

Por otra parte, dado un conjunto de longitudes l_i que cumplen con esta desigualdad, existe entonces un código instantáneo con dichas longitudes.

Esta desigualdad puede ser extendida, en lo que es llamada la *desigualdad de Kraft extendida*, que establece que para cualquier conjunto infinito numerable que forma un código instantáneo se cumple que:

$$\boxed{\sum_{i=1}^{\infty} D^{-l_i} \leq 1}$$

3.2. Códigos óptimos

Mediante la desigualdad de Kraft, es posible afirmar la existencia de un código óptimo. Sin embargo, es interesante poder encontrar dicho código, el cual no solo debe cumplir con la desigualdad de Kraft, sino que también debe minimizar la longitud promedio. Por esta razón, el problema equivale a minimizar a dicha longitud:

$$L = \sum_{i=1}^{n} p_i l_i \rightarrow \text{sujeta a } \sum_{i=1}^{n} D^{-l_i} \leq 1$$

Para esto, es necesario utilizar el método de *Multiplicadores de Lagrange*:

$$J = \sum p_i l_i + \lambda \left(\sum D^{-l_i} \right) \rightarrow \frac{\partial J}{\partial l_i} = p_i - \lambda D^{-l_i} \log_e (D) = 0$$

[3]Para el caso de un código binario se tiene que $D = 2$, ya que al armar un árbol para generar el código cada rama se divide en 2.

Se puede despejar entonces:

$$D^{-l_i} = \frac{p_i}{\lambda \log_e(D)} \rightarrow \sum_{i=1}^{n} \frac{p_i}{\lambda \log_e(D)} = 1 \rightarrow \lambda = \frac{1}{\log_e(D)} \rightarrow p_i = D^{-l_i}$$

Por lo tanto, las longitudes óptimas son:

$$\boxed{l_i^* = -\log_D(p_i)}$$

La longitud promedio entonces es:

$$\boxed{L^* = \sum p_i l_i^* = -\sum p_i \log_D(p_i) = H_D(X)}$$

Esta cota encontrada es una cota mínima, ya que las longitudes así obtenidas no siempre serán números enteros. A partir de este resultado, es posible expresar el primer teorema de Shannon.

3.3. Primer Teorema de Shannon (Teorema de Codificación de la Fuente)

Sea L la longitud promedio para cualquier código instantáneo C cumple que:

$$\boxed{L \geq H_D(X)}$$

Donde la igualdad se cumple si y solo si $p_i = D^{-l_i}$. Este teorema muestra que no es posible comprimir por debajo del valor de la entropía de la fuente. Por otra parte, se sabe que las longitudes óptimas $l_i^* = -\log_D(p_i)$ pueden ser valores no enteros, por lo es es posible enunciar el siguiente teorema:

Teorema: Sean $l_1^*, l_2^*, ..., l_n^*$ las longitudes óptimas, y $L^* = \sum p_i l_i^*$ la longitud promedio óptima. Debido a que las longitudes óptimas no son números enteros, se puede tomar que cada l_i es el menor entero mayor a cada l_i^* ($l_i = \lceil l_i^* \rceil \geq l_i^*$). Se puede demostrar que estos largos redondeados cumplen con la cota de Kraft, y también cumplen que:

$$\boxed{H_D(X) \leq L^* \leq L \leq H_D(X) + 1}$$

En caso de tener una extensión de la fuente $X_1, X_2, ..., X_n$, la longitud promedio mínima por símbolo es:

$$\boxed{\frac{H(X_1, X_2, ..., X_n)}{n} \leq L_n^* \leq L_n \leq \frac{H(X_1, X_2, ..., X_n)}{n} + \frac{1}{n}}$$

En caso de que sean símbolos independientes, se puede escribir:

$$\boxed{H(X) \leq L_n^* \leq L_n \leq H(X) + \frac{1}{n}}$$

Recordando la definición de la tasa de la entropía, para n grandes se cumple que:

$$\lim_{n\to\infty} \frac{H(X_1, X_2, ..., X_n)}{n} = H_r \Longrightarrow H_r \leq L_n^* \leq L_n \leq H_r + \frac{1}{n} \Longrightarrow L_n^* \rightarrow H_r$$

De esta forma, es posible definir también a la tasa de entropía como la cantidad de bits (en promedio) por símbolo necesarios para describir el proceso.

El teorema de Shannon ofrece una cota para la longitud del código dada una determinada distribución de probabilidad. Sin embargo, en la realidad es poco común conocer la distribución real de una variable aleatoria, sino que se suele conocer una estimación. Por lo tanto, es posible expresar el siguiente teorema sobre el error en dicha estimación:

Teorema: Sea X una variable aleatoria con función de distribución $p(x)$. Sea $q(x)$ una estimación de $p(x)$, con la que se diseña un código con longitudes $l(x) = \left\lceil \log\left(\frac{1}{q(x)}\right) \right\rceil$, se cumple entonces que:

$$H(p) + D(p \parallel q) \leq \mathbb{E}_p[l(x)] < H(p) + D(p \parallel q) + 1$$

Por lo tanto, al utilizar $q(x)$ en lugar de la distribución verdadera $p(x)$, se observa un error en la longitud promedio dado por $D(p \parallel q)$.

4. Algoritmos de codificación

4.1. Método de Shannon-Fano

Este es un algoritmo que permite alcanzar la siguiente cota:

$$L(C) \leq H(X) + 2$$

Los pasos a seguir parar esta codificación son:

1. Se ordenan las probabilidades de forma creciente.
2. Seleccionar k tal que $\left|\sum_{i=1}^{k} p_i - \sum_{i=k+1}^{m} p_i\right|$ sea mínima. Es decir que divide el conjunto total en dos, de forma tal que ambos subconjuntos acumulen casi la misma probabilidad.
3. Se asigna un bit diferente a cada uno de los subconjuntos creados.
4. Cada uno de estos subconjuntos debe ser dividido en otros dos nuevos subconjuntos, repitiendo el proceso desde el inicio hasta que ya no puedan ser mas divididos.

Este método se puede expresar en la forma de un árbol, en donde se intenta equilibrar las probabilidades de cada rama para que quede balanceado.

Ejemplo

Sea una fuente que emite 5 símbolos independientes:

$$\begin{aligned} A &\rightarrow 0{,}4 \\ B &\rightarrow 0{,}2 \\ C &\rightarrow 0{,}15 \\ D &\rightarrow 0{,}15 \\ E &\rightarrow 0{,}1 \end{aligned}$$

La longitud óptima de este código es:

$$l_i^* = -\log(p_i) \rightarrow L^* = \mathbb{E}[l_i^*] = 2{,}146\,\text{bits}$$

Siguiendo los pasos mencionados, es posible diseñar un árbol para el código de Shannon-Fano, el cual se puede ver en la figura 4.1. El código queda entonces como:

$$\begin{aligned} A &\rightarrow 00 \\ B &\rightarrow 10 \\ C &\rightarrow 110 \\ D &\rightarrow 111 \\ E &\rightarrow 01 \end{aligned}$$

La longitud promedio de este código es:

$$L = \sum p_i l_i = 2{,}3\,\text{bits}$$

Se puede ver que la longitud alcanzada es cercana a la óptima, pero no tanto como seria deseable.

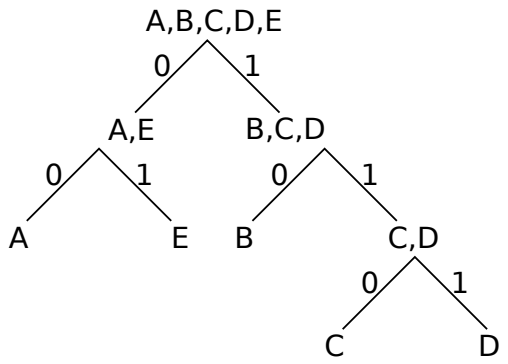

Figura 4.1: Árbol para el código de Shannon-Fano.

4.2. Método de Huffman

Este método permite alcanzar una longitud promedio menor a la del método de Shannon-Fano, y tiene la ventaja de que asigna menos bits a aquellos símbolos que son mas probables. El procedimiento para armar este código es:

1. Se ordenan las probabilidades de forma creciente.
2. Se agrupan los dos símbolos con menor probabilidad en un nuevo macro-símbolo.
3. Se reordenan nuevamente los símbolos de forma creciente (en probabilidad), y se vuelven a agrupar aquellos símbolos con mejor probabilidad. Este proceso se repite hasta que se hayan agrupado todos los símbolos.

Este método también puede ser construido en forma de árbol, pero es necesario "armarlo de atrás para adelante".

Ejemplo

Tomando los mismos símbolos del ejemplo de la sección anterior, con el método de Huffman se tiene que:

$$\begin{cases} A \to & 0{,}4 \\ B \to & 0{,}2 \\ C \to & 0{,}15 \\ D \to & 0{,}15 \\ E \to & 0{,}1 \end{cases} \Longrightarrow \begin{cases} A \to & 0{,}4 \\ DE \to & 0{,}25 \\ B \to & 0{,}2 \\ C \to & 0{,}15 \end{cases} \Longrightarrow \begin{cases} A \to & 0{,}4 \\ BC \to & 0{,}35 \\ DE \to & 0{,}25 \end{cases} \Longrightarrow \begin{cases} BCDE \to & 0{,}6 \\ A \to & 0{,}4 \end{cases} \Longrightarrow ABCDE \to 1$$

El árbol en este caso es el que se observa en la figura 4.2, y el código que se obtiene es:

$$\begin{aligned} A &\to 0 \\ B &\to 100 \\ C &\to 101 \\ D &\to 110 \\ E &\to 111 \end{aligned}$$

La longitud promedio en este caso es:

$$L = \sum p_i l_i = 2{,}2 \,\text{bits}$$

Se observa que la longitud obtenida es menor a la del ejemplo anterior, encontrándose dentro de la cota:

$$H(X) \leq L^* \leq H(X) + 1$$

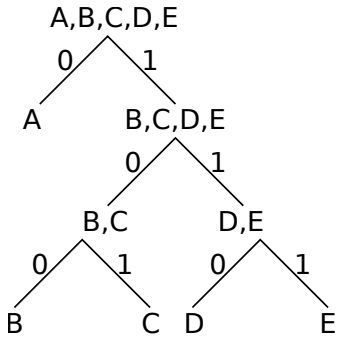

Figura 4.2: Árbol para el código de Huffman.

4.3. Método Aritmético

Este tipo de codificación en lugar de utilizar una secuencia de bits para representar un símbolo, se utiliza un subintervalo dentro del intervalo $[0, 1]$ para representar cada símbolo. Para esto, se divide el intervalo $[0, 1]$ en subintervalos definidos por las probabilidades de cada uno de los símbolos, por lo que para transmitir un determinado símbolo, solo es necesario transmitir el valor de cualquiera de los puntos dentro del intervalo correspondiente. Para transmitir un segundo símbolo, cada subintervalo se vuelve a dividir en nuevos subintervalos (también proporcionales a las probabilidades). De

esta forma, cada subintervalo se puede dividir infinitamente para cualquier extensión de la fuente (siempre y cuando sea posible alcanzar dicha precisión). En la figura 4.3 se tiene un ejemplo.

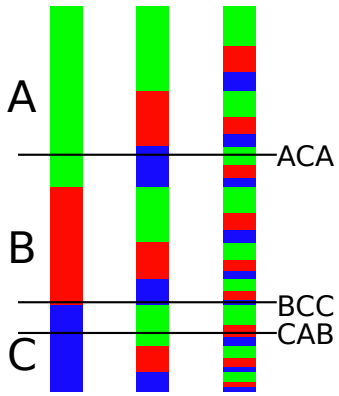

Figura 4.3: Código aritmético.

5. Codificación del canal

Un *canal discreto* es un sistema con un alfabeto $\mathcal{X}$ de entrada y un alfabeto $\mathcal{Y}$ de salida, y con una matriz de transición de probabilidad $p(y|x)$ que relaciona ambos alfabetos. Un canal se dice *sin memoria* si la distribución de probabilidad de la salida en un instante, depende únicamente de la entrada en ese instante.

En la figura 5.1 se tiene un diagrama para la comunicación de un mensaje a través de un canal. El mensaje W, trazado a partir del conjunto de índices $\{1, 2, ..., M\}$, al ser codificado resulta en la señal $X^n(W)$, la cual es recibida como una secuencia aleatoria $Y^n \sim p(y^n|x^n)$. Por lo tanto, para poder conocer el mensaje es necesario realizar una estimación $\hat{W} = g(Y^n)$.

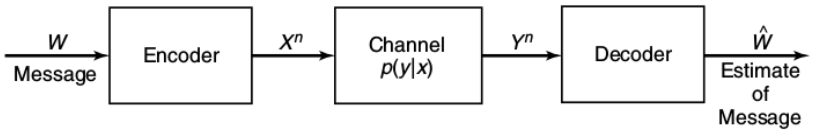

Figura 5.1: Canal de comunicación.

Definición: Un canal discreto $(\mathcal{X}, p(y|x), \mathcal{Y})$, consiste en un alfabeto de entrada $\mathcal{X}$ y uno de salida $\mathcal{Y}$, relacionados a través de $p(y|x)$.

Definición: La extensión n-ésima de un canal discreto sin memoria (DMC), se escribe como $(\mathcal{X}^n, p(y^n|x^n), \mathcal{Y}^n)$ donde:

$$p\left(y_k|x^k, y^{k-1}\right) = p(y_k|x_k) \quad k = 1, 2, ..., n$$

Definición: Un código (M, n) para el canal $(\mathcal{X}, p(y|x), \mathcal{Y})$, consiste en:

1. Un set de índices $\{1, 2, ..., M\}$.
2. Una función codificadora $X^n : \{1, 2, ..., M\} \to \mathcal{X}^n$, generando *codewords* $x^n(1), x^n(2), ..., x^n(M)$. Este conjunto de codewords se lo conoce como *codebook*.
3. Una función decodificadora $g : \mathcal{Y}^n \to \{1, 2, ..., M\}$.

Definición: La probabilidad condicional del error en la decodificación al enviar el índice i es:

$$\lambda_i = \mathbb{P}(g(Y^n) \neq i | X^n = x^n(i)) = \sum_{y^n} p(y^n|x^n(i)) \mathbf{1}\{g(y^n) \neq i\}$$

La máxima probabilidad de error queda como:

$$\lambda^{(n)} = \max_{i \in \{1,2..M\}} \lambda_i$$

Definición: La probabilidad promedio del error se puede escribir como:

$$P_e^{(n)} = \frac{1}{M} \sum_{i=1}^{M} \lambda_i$$

Definición: La tasa R del código (M, n) es:

$$\boxed{R = \frac{\log(M)}{n}}$$

Definición: Una tasa R se dice asequible si existe una secuencia de $\left(\left\lceil 2^{nR} \right\rceil, n\right)$ códigos, tales que la probabilidad de error tiende a cero a medida que $n \to \infty$.

Definición: La capacidad de un canal es el máximo absoluto para todas las tasas R asequibles.

5.1. Capacidad del canal

Definición: Se define la *capacidad del canal* como la cantidad de información que puede transmitirse por uso del canal. Para un canal discreto sin memoria se define como:

$$\boxed{C = \max_{p(x)} (I(X;Y))}$$

Por lo tanto, la capacidad del canal indica cuál es la tasa mas alta a la que puede enviarse información, minimizando la probabilidad de error. Otra forma de expresar la capacidad del canal es:

$$C = \max_{p(x)} (H(X) - H(X|Y)) = \max_{p(x)} (H(Y) - H(Y|X))$$

Definición: Un canal se dice *simétrico* si las columnas de la matriz de transición $p(y|x)$ son permutaciones unas de otras, y lo mismo ocurre con las columnas. Un canal se dice *levemente simétrico* si cada fila de la matriz de transición es una permutación de las demás filas, y la suma de todas las probabilidades de cada columna es la misma.

A partir de esta definición, es posible enunciar el siguiente teorema:

Teorema: Para un canal levemente simétrico, donde el alfabeto de entrada es uniforme, se cumple que:

$$C = \log(|\mathcal{Y}|) - H(\text{fila de la matríz de transición})$$

Propiedades de la capacidad del canal

- $C \geq 0$, ya que se sabe que $I(X;Y) \geq 0$.
- $C \leq \log(|\mathcal{X}|)$, ya que se cumple que $C = \max(I(X;Y)) \leq \max(H(X)) = \log(|\mathcal{X}|)$.
- $C \leq \log(|\mathcal{Y}|)$.
- $I(X;Y)$ es una función continua para $p(x)$.
- $I(X;Y)$ es una función cóncava para $p(x)$.

5.2. Teorema de codificación del canal

Antes de poder enunciar el teorema, es necesario realizar algunas definiciones previas.

5.2.1. Secuencias típicas conjuntas

El conjunto $A_{x,y}^{(n)}$ de secuencias típicas $\{(x^n, y^n)\}$ con distribución $p(x,y)$, es el set de n secuencias que cumplen que:

$$A_{x,y}^{(n)} = \{(x^n, y^n) \in \mathcal{X}^n \times \mathcal{Y}^n\} : \begin{cases} \left|-\frac{1}{n}\log\left(p\left(x^n\right)\right) - H(X)\right| < \varepsilon \\ \left|-\frac{1}{n}\log\left(p\left(y^n\right)\right) - H(Y)\right| < \varepsilon \\ \left|-\frac{1}{n}\log\left(p\left(x^n, y^n\right)\right) - H(X,Y)\right| < \varepsilon \end{cases}$$

donde $p\left(x^n, y^n\right) = \prod_{i=1}^{n} p\left(x_i, y_i\right)$. Esto significa que las secuencias x^n e y^n deben pertenecer al conjunto típico por separado y también de forma conjunta, para que puedan pertenecer al conjunto típico conjunto.

Teorema (AEP conjunta): Sean (X^n, Y^n) secuencias de largo n, iid, con $p\left(x^n, y^n\right) = \prod_{i=1}^{n} p\left(x_i, y_i\right)$, se cumple que:

- $\mathbb{P}\left((X^n, Y^n) \in A_{x,y}^{(n)}\right) \to 1$ cuando $n \to \infty$.
- $\left|A_{x,y}^{(n)}\right| \leq 2^{n(H(X,Y)+\varepsilon)}$.
- Sea $\left(\tilde{X}^n, \tilde{Y}^n\right) \sim p\left(x^n\right) p\left(y^n\right)$ (es decir que son independientes y poseen las mismas marginales que $p\left(x^n, y^n\right)$), entonces:

$$\mathbb{P}\left(\left(\tilde{X}^n, \tilde{Y}^n\right) \in A_{x,y}^{(n)}\right) \leq 2^{-n(I(X;Y)-3\varepsilon)} \qquad \mathbb{P}\left(\left(\tilde{X}^n, \tilde{Y}^n\right) \in A_{x,y}^{(n)}\right) \geq (1-\varepsilon)\, 2^{-n(I(X;Y)+3\varepsilon)}$$

5.2.2. Segundo Teorema de Shannon (Teorema de Codificación del Canal)

Para un canal discreto sin memoria, cualquier tasa menor que la capacidad del canal es asequible. Específicamente, para toda tasa $R < C$, existe una secuencia de $\left(2^{nR}, n\right)$ códigos donde la probabilidad de error máxima tiende a cero: $\lambda^{(n)} \to 0$. Por otra parte, es cierto también que para cualquier secuencia de $\left(2^{nR}, n\right)$ códigos donde se cumple que $\lambda^{(n)} \to 0$, entonces se cumple que $R \leq C$.

Por lo tanto, siempre y cuando se cumpla la condición de que $R < C$, es posible generar códigos con una probabilidad de error arbitrariamente baja.

Una propiedad importante a tener en cuenta, es que la capacidad por transmisión de un canal sin memoria no se incrementa si el canal es utilizado muchas veces:

Lema Sea Y^n el resultado de introducir X^n a un canal sin memoria de capacidad C, entonces:

$$\boxed{I\left(X^n; Y^n\right) \leq nC} \quad \forall p\left(x^n\right)$$

Por lo tanto, no importa la extensión de la fuente, la capacidad del canal es una propiedad del canal que no puede modificarse.

5.2.3. Desigualdad de Fano para la codificación del canal

Sea el índice de mensajes $W \in \mathcal{W} = \left\{1, 2, ..., 2^{nR}\right\}$. Sea $\hat{W} = g\left(Y^n\right)$ la decodificación obtenida a la salida del canal, de forma tal que $W \to X^n(W) \to Y^n \to \hat{W}$ forman una cadena de Markov. La probabilidad de error se puede escribir como:

$$\mathbb{P}\left(W \neq \hat{W}\right) = \mathbb{P}_E^{(n)} = \frac{1}{2^{nR}} \sum_i \lambda_i$$

Por lo tanto, la Desigualdad de Fano establece que:

Desigualdad de Fano: Para un canal discreto sin memoria con un codebook $\mathcal{C}$ y un mensaje de entrada W uniformemente distribuido en $\left\{1, 2, ..., 2^{nR}\right\}$, se tiene:

$$\boxed{H\left(W|\hat{W}\right) \leq 1 + \mathbb{P}_E^{(n)} nR}$$

5.3. Canales con realimentación (feedback)

En la figura 5.2 se tiene un canal con realimentación. Se asume que la realimentación es instantánea y libre de ruido.

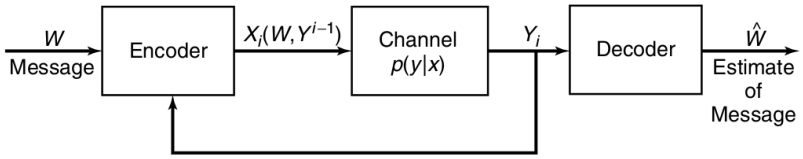

Figura 5.2: Canal con feedback.

A pesar de la realimentación, se posible probar que la capacidad no se modifica:

$$\boxed{C_{FB} = C = \max_{p(x)} \left(I\left(X;Y\right) \right)}$$

Por lo tanto, el efecto de la realimentación no introduce ninguna mejora en cuanto a la capacidad.

6. Conjuntos

6.1. Conjunto de Grupo

Un conjunto de grupo G es un sistema algebraico que posee una única operación binaria asociada. Esta operación binaria $*$ debe cumplir que:

1. La operación $*$ es asociativa.
2. Dentro del conjunto de grupo existe un elemento e, tal que para cualquier elemento a dentro del conjunto se cumple que:

   $$a * e = e * a = a$$

   Este elemento se lo conoce como el elemento identidad de $*$. Este elemento es único.
3. Por cada elemento $a \in G$, existe un elemento $a' \in G$ tal que:

   $$a * a' = a' * a = e$$

   El elemento a' se lo conoce como la inversa de a. Este elemento es único.

Un conjunto de grupo puede ser finito o infinito, dependiendo de la cantidad de elementos que contiene. La cantidad de elementos que posee un conjunto de grupo indica el orden del grupo: orden$= |G|$.

6.2. Campos

Sea F un conjunto de elementos con 2 operaciones binarias asociadas: multiplicación y suma. Se dice que F es un campo si se cumple que:

1. F es un conjunto conmutativo para la suma. El elemento identidad para la suma es el cero.
2. F es un conjunto conmutativo para la multiplicación. El elemento identidad para la multiplicación es el uno.
3. Para tres elementos $a, b, c \in F$ se cumple que:

   $$a\left(b + c\right) = a \cdot b + a \cdot c$$

   Es decir, que se cumple la propiedad distributiva.

A partir de esta definición, se puede ver que un campo F esta compuesto por dos grupos: grupo aditivo y grupo multiplicativo. Como cada uno de estos conjuntos tienen que contener a sus respectivos elementos identidad, un campo debe contener como mínimo dos elementos. Un campo es un sistema algebraico en el cual se pueden realizar las operaciones de suma, resta (inversa de la suma), multiplicación y división (inversa de la multiplicación).

Definición: Sea F un campo donde 1 es la identidad del grupo multiplicativo. La *característica* de F se define como el mínimo valor entero de λ que cumple que:

$$\sum_{i=1}^{\lambda} 1 = 0$$

Si no existe un valor de λ tal que se cumple esto, se dice que F es de característica cero ($\lambda = 0$), y F es un campo infinito. Una propiedad que cumple λ, es que es un numero primo.

6.2.1. Campos de Galois

Se tratan de campos finitos. Sea p un numero primo. Es posible demostrar que el conjunto $\{0, 1, ..., p-1\}$ forma un grupo conmutativo para la suma de modulo p. Por otra parte, también puede demostrarse que el conjunto $\{1, 2, ..., p-1\}$ forma un grupo conmutativo para la multiplicación de modulo p. En base a esto, se puede demostrar que el conjunto $\{0, 1, ..., p-1\}$ forma un campo finito de Galois $GF(p)$ de orden p:

$$\sum_{i=1}^{p-1} 1 = p - 1 \rightarrow \sum_{i=1}^{p} 1 = 0$$

Por lo tanto, p representa la característica del campo. Por otra parte, para cualquier entero m, es posible crear un campo de Galois $GF(p^m)$, que posee p^m elementos. Este campo $GF(p^m)$ contiene a $GF(p)$ como un subcampo.

Sea $a \neq 0 \in GF(q)$, se cumple que:

$$a^n \in GF(q) \quad n = 1, 2, ...$$

Como se trata de un campo finito, es necesario que las potencias de a comiencen a repetirse, es decir que $a^m = a^k$. Tomando que $m > k$, entonces se puede plantear:

$$a^m = a^k \rightarrow a^k a^{m-k} = a^k \rightarrow a^{m-k} = 1 \rightarrow n = m - k \rightarrow a^n = 1$$

Definición: Sea $a \neq 0 \in GF(q)$. El mínimo valor entero de n que cumple que $a^n = 1$ se lo conoce el orden de a.

Teorema: Sea $a \neq 0 \in GF(q)$. Se cumple entonces que:

$$a^{q-1} = 1$$

Por esta razón, sea n el orden de a, se cumple que n es un divisor de $q - 1$.

Definición: Se dice que $a \in GF(q)$ es un *elemento primitivo* si es de orden $q - 1$: $a^{q-1} = 1$. Todo campo posee al menos un elemento primitivo.

6.2.2. Campos vectoriales

Sea F un campo. Sea V un conjunto de elementos donde existe la operación binaria de la suma. Se define también a la operación de multiplicación entre los elementos de F y de V. El conjunto V es un espacio vectorial sobre F si se cumple que:

1. V es conmutativo para la suma.
2. Para todo $a \in F$ y $\mathbf{v} \in V$, se cumple que $a \cdot \mathbf{v} \in V$.
3. Se cumple la regla asociativa: $\forall a, b \in F$ y $\mathbf{v} \in V$, se cumple que:

$$(a \cdot b) \cdot \mathbf{v} = a \cdot (b \cdot \mathbf{v})$$

4. Se cumple la regla distributiva: $\forall \mathbf{u}, \mathbf{v} \in V$ y $a, b \in F$, se cumple que:

$$a \cdot (\mathbf{u} + \mathbf{v}) = a \cdot \mathbf{u} + a \cdot \mathbf{v} \qquad (a + b) \cdot \mathbf{v} = a \cdot \mathbf{v} + b \cdot \mathbf{v}$$

5. Existe un elemento identidad en F tal que: $\mathbf{v} \cdot 1 = 1 \cdot \mathbf{v} = \mathbf{v}$.
6. Existe un elemento nulo en F tal que: $\mathbf{v} \cdot 0 = 0 \cdot \mathbf{v} = 0$.

Definición: Un conjunto de vectores $\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_k \in V$ se dice linealmente independiente si se cumple que:

$$a_1\mathbf{v}_1 + a_2\mathbf{v}_2 + \ldots + a_k\mathbf{v}_k = 0 \Longleftrightarrow a_1 = a_2 = \ldots = a_k = 0$$

6.2.3. Polinomios para campos finitos

Cuando se trabaja con campos de Galois de orden q $GF(q)$, es posible expresarlos como:

$$p(X) = a_0 + a_1 X + a_2 X^2 + \ldots + a_n X^n$$

donde los coeficientes $a_i \in GF(q)$ con $0 \leq i \leq n$. Esta descripción a través de polinomios debe cumplir todas las propiedades enunciadas anteriormente (asociatividad, distributividad, conmutatividad).

Definición: Un polinomio $p(X)$ de grado m en $GF(q)$, se dice que es irreducible si no es divisible por ningún otro polinomio en $GF(q)$ que tenga un grado menor que m.

Teorema: Cualquier polinomio $p(X)$ de grado m irreducible en $GF(q)$, divide $X^{q^m-1} - 1$.

Definición: Un polinomio $p(X)$ de grado m irreducible en $GF(q)$, se dice primitivo si el mínimo entero positivo n para el cual $p(X)$ divide $X^n - 1$ es $n = q^m - 1$.

6.2.4. Construcción de un campo de Galois

Para construir un campo de Galois es necesario partir de un polinomio primitivo de orden m sobre $GF(p) = \{0, 1, \ldots, p-1\}$:

$$p(X) = p_0 + p_1 X + \ldots + p_{m-1} X^{m-1} + X^m$$

Este polinomio presenta m raíces. Sean 0 y 1 los elementos unitarios, se puede escribir:

$$\begin{aligned}
0 \cdot 0 &= 0 \\
0 \cdot 1 = 1 \cdot 0 &= 0 \\
0 \cdot \alpha = \alpha \cdot 0 &= 0 \\
1 \cdot 1 &= 1 \\
1 \cdot \alpha = \alpha \cdot 1 &= \alpha \\
\alpha^2 &= \alpha \cdot \alpha \\
\vdots &= \vdots \\
\alpha^j &= \prod_j \alpha
\end{aligned}$$

Dado que α es una raíz de $p(X)$, se cumple que:

$$p(\alpha) = p_0 + p_1 \alpha + \ldots + p_{m-1} \alpha^{m-1} + \alpha^m = 0$$

Por otra parte, se sabe que $p(X)$ divide $X^{p^m-1} - 1$ por ser un polinomio primitivo, por lo tanto:

$$X^{p^m-1} - 1 = q(X) \cdot p(X) \rightarrow \alpha^{p^m-1} - 1 = q(\alpha) \cdot \underbrace{p(\alpha)}_{0} = 0 \rightarrow \boxed{\alpha^{p^m-1} = 1}$$

Se puede ver entonces que la secuencia se potencias de α debe repetirse comenzar a repetirse para $\alpha^{k(p^m-1)}$ con $k = 1, 2, \ldots$. Por esta razón, el campo posee p^m elementos diferentes:

$$\mathcal{F} = \left\{0, 1, \alpha, \alpha^2, \ldots, \alpha^{p^m-2}\right\}$$

Se dice entonces que se tiene un $GF(p^m)$. Se puede ver que se trata de un grupo cerrado, ya que la secuencia debe repetirse. Sea $i + j \geq p^m - 1$, se tiene que $\alpha^i \cdot \alpha^j = \alpha^{i+j}$ no cae dentro del conjunto de valores, por lo que se debe escribir:

$$i + j = (p^m - 1) + r \rightarrow \alpha^{i+j} = \alpha^{(p^m-1)+r} = \underbrace{\alpha^{(p^m-1)}}_{1} \alpha^r = \alpha^r$$

De esta forma, queda demostrado que la multiplicación es una operación cerrada. Es posible también demostrar que lo mismo se cumple para la suma, y por lo tanto queda demostrado que $\mathcal{F} = \left\{0, 1, \alpha, \alpha^2, \ldots, \alpha^{p^m-2}\right\}$ forma un campo de Galois $GF(p^m)$. Otra forma de describir este campo, es a través de tuplas formadas por los coeficientes a_i:

$$\alpha^i = a_0 + a_1 \alpha + \ldots + a_{m-1} \alpha^{m-1} \rightarrow (a_0, a_1, \ldots a_{m-1})$$

Utilizando esta representación, al realizar una suma, se debe sumar cada componente según las reglas de $GF(p)$.

7. Códigos lineales en bloques

Se tiene una fuente que emite símbolos de forma continuo dentro de $GF(2)$, los cuales forman una secuencia. Cada uno de estos símbolos emitidos, se los suele conocer como bits. En la codificación por bloques, cada secuencia de símbolos es segmentada en bloques de mensajes de longitud k fija, por lo que cada mensaje $\mathbf{u} = (u_0, u_1, \ldots, u_{k-1})$ tienen 2^k posibles combinaciones diferentes. Para transmitir estos mensajes, se codifican en un codeword $\mathbf{v} = (v_0, v_1, \ldots, v_n)$ con $n > k$. Este conjunto de 2^k codewords, se dice que forman un código en bloques (n, k). Los $n - k$ bits que se agregan a cada mensaje en la codificación, se los conoce como bits de redundancias, los cuales permiten detectar y corregir errores. La tasa R del código se puede obtener como:

$$R = \frac{k}{n}$$

Esta tasa da una idea de la cantidad de información (en promedio) que contiene cada codeword.

Definición: Un código binario en bloques de longitud n con 2^k codewords, se dice que es un código lineal en bloques (n, k) si y solo si los 2^k codewords forman un subespacio de k dimensiones en $\mathbf{v}$ para las n tuplas en $GF(2)$.

7.1. Matriz generadora y matriz de paridad

Como los códigos lineales en bloques forman un subespacio dentro del espacio generado por las $n-$tuplas de $GF(2)$, significa que existen k codewords linealmente independientes tal que cada codeword $\mathbf{v} \in \mathcal{C}$ es una combinación lineal de estos k codewords independientes. Se dice entonces que estos k codewords linealmente independientes forman una base de $\mathcal{C}$.

Se desea transmitir el mensaje $\mathbf{u} = (u_0, u_1, \ldots, u_{k-1})$, el cual debe ser codificado como $\mathbf{v} = (v_0, v_1, \ldots, v_{n-1})$. Este codeword $\mathbf{v} \in \mathcal{C}$ se puede escribir como la combinación lineal de $\mathbf{g}_0, \mathbf{g}_1, \ldots, \mathbf{g}_{k-1}$, los cuales forman una base de $\mathcal{C}$:

$$\mathbf{v} = u_0\mathbf{g}_0 + u_1\mathbf{g}_1 + \ldots + u_{k-1}\mathbf{g}_{k-1}$$

Esto se puede escribir también como:

$$\boxed{\mathbf{v} = \mathbf{u} \cdot G \rightarrow G = \begin{bmatrix} \mathbf{g}_0 \\ \mathbf{g}_1 \\ \vdots \\ \mathbf{g}_{k-1} \end{bmatrix} = \begin{bmatrix} g_{0,0} & g_{0,1} & \cdots & g_{0,n-1} \\ g_{1,0} & g_{1,1} & \cdots & g_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ g_{k-1,0} & g_{k-1,1} & \cdots & g_{k-1,n-1} \end{bmatrix}}$$

Esta matriz G se la conoce como matriz generadora. Dado que no existe una única base para el código $\mathcal{C}$, tampoco existe una única matriz generadora.

Aquellos $n - k$ bits de redundancia, forman el espacio nulo de $\mathcal{C}$, el cual se puede definir como:

$$\mathcal{C}_d = \{\mathbf{w} \in V : \langle \mathbf{w}, \mathbf{v} \rangle = 0 \forall \mathbf{v} \in \mathcal{C}\}$$

Sea $\mathcal{B}_d$ una base de $\mathcal{C}_d$, ésta contiene $n-k$ codewords linealmente independientes: $\mathbf{h}_0, \mathbf{h}_1, \ldots, \mathbf{h}_{n-k-1}$. Se puede definir entonces a la matriz de chequeo de paridad como:

$$\boxed{H = \begin{bmatrix} \mathbf{h}_0 \\ \mathbf{h}_1 \\ \vdots \\ \mathbf{h}_{n-k-1} \end{bmatrix} = \begin{bmatrix} h_{0,0} & h_{0,1} & \cdots & h_{0,n-1} \\ h_{1,0} & h_{1,1} & \cdots & h_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ h_{n-k-1,0} & h_{n-k-1,1} & \cdots & h_{n-k-1,n-1} \end{bmatrix}}$$

Se puede ver que $G \cdot H^T = \mathbf{0}$. Gracias a esta matriz, es posible verificar que no haya errores en la transmisión, ya que se debe cumplir que:

$$\boxed{\mathcal{C} = \left\{\mathbf{v} \in V : \mathbf{v} \cdot H^T = 0\right\}}$$

Por lo tanto, un código lineal en bloques queda unívocamente determinado por estas dos matrices.

7.1.1. Códigos sistemáticos

Un código se dice sistemático, si cada codeword esta dividido en dos partes: k bits de mensaje y $n-k$ bits de redundancias, de acuerdo a lo que se muestra en la figura 7.1.

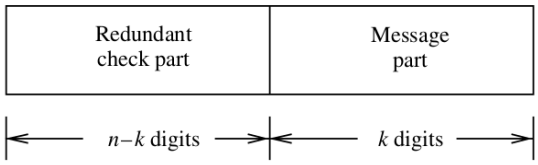

Figura 7.1: Estructura de un código sistemático. Por supuesto que puede invertirse el orden de ambas partes, pero cambiarán las matrices G y H.

Por lo tanto, la matriz generadora se puede escribir como:

$$G = \begin{bmatrix} \mathbf{g}_0 \\ \mathbf{g}_1 \\ \vdots \\ \mathbf{g}_{k-1} \end{bmatrix} = \left[\underbrace{\begin{matrix} p_{0,0} & p_{0,1} & \cdots & p_{0,n-k-1} \\ p_{1,0} & p_{1,1} & \cdots & p_{1,n-k-1} \\ \vdots & \vdots & \ddots & \vdots \\ p_{k-1,0} & p_{k-1,1} & \cdots & p_{k-1,n-k-1} \end{matrix}}_{\text{matriz } P} \underbrace{\begin{matrix} 1 & 0 & \ldots & 0 \\ 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 1 \end{matrix}}_{I_{k\times k}} \right] = [P\, I_{k\times k}]$$

Sea $\mathbf{u} = (u_0, u_1, \ldots, u_{k-1})$ el mensaje a transmitir, y sea $\mathbf{v} = (v_0, v_1, \ldots, v_{n-1})$ el mensaje codificado, se tiene que:

$$\mathbf{v} = \mathbf{u} \cdot G$$

Por lo tanto, se puede ver que:

$$\begin{cases} v_j = u_0 p_{0,j} + u_1 p_{1,j} + \ldots + u_{k-1} p_{k-1,j} & j = 0, 1, \ldots, n-k-1 \\ v_{n-k+\ell} = u_\ell & \ell = 0, 1, \ldots, k-1 \end{cases}$$

Esta expresión muestra que los últimos k bits de $\mathbf{v}$ coinciden con el mensaje $\mathbf{u}$, mientras que los primeros $n-k$ bits son una combinación lineal de los bits de $\mathbf{u}$. Estos $n-k$ bits son los bits de paridad, y por esta razón la matriz P se la conoce como matriz de paridad de G.

La matriz H se puede escribir como:

$$H = \left[I_{n-k}\, P^T\right] = \left[\underbrace{\begin{matrix} 1 & 0 & \ldots & 0 \\ 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 1 \end{matrix}}_{I_{n-k}} \underbrace{\begin{matrix} p_{0,0} & p_{1,0} & \cdots & p_{k-1,0} \\ p_{0,1} & p_{1,1} & \cdots & p_{k-1,1} \\ \vdots & \vdots & \ddots & \vdots \\ p_{0,n-k-1} & p_{1,n-k-1} & \cdots & p_{k-1,n-k-1} \end{matrix}}_{P^T} \right]$$

Se puede ver que se cumple que $G \cdot H^T = \mathbf{0}$. Notar además que cualquiera de las siguientes relaciones es válida (cuál se usa depende del orden de las partes del código, según si se invierte el esquemita de la figura 7.1)

$$\text{Sistemáticas} \rightarrow \begin{cases} G = \left[I_k \quad P\right] \iff H = \left[P^T \quad I_{n-k}\right] \\ G = \left[P \quad I_k\right] \iff H = \left[I_{n-k} \quad P^T\right] \\ G = \left[I_k \quad P^T\right] \iff H = \left[P \quad I_{n-k}\right] \\ G = \left[P^T \quad I_k\right] \iff H = \left[I_{n-k} \quad P\right] \end{cases}$$

7.2. Detección de errores

Se desea transmitir $\mathbf{v} = (v_0, v_1, \ldots, v_{n-1}) \in \mathcal{C}$ por un canal BSC. A la salida del canal se obtiene el vector $\mathbf{r} = (r_0, r_1, \ldots, r_{n-1})$. Por lo tanto, el error en la transmisión se puede escribir como:

$$\mathbf{e} = \mathbf{r} + \mathbf{v} = (e_0, e_1, \ldots, e_{n-1}) = (r_0 + v_0, r_1 + v_1, \ldots, r_{n-1} + v_{n-1})$$

Como se transmiten bits, cada una de las sumas debe hacerse en modulo 2, y por lo tanto se tiene:

$$\begin{cases} e_j = 1 \longleftrightarrow v_j \neq r_j \\ e_j = 0 \longleftrightarrow v_j = r_j \end{cases}$$

Por lo tanto, el vector $\mathbf{e}$ contiene unos en las posiciones donde el vector recibido difiere del vector enviado. Al vector $\mathbf{e}$ se lo conoce como patrón de error o vector de error.

Cuando el receptor recibe un mensaje $\mathbf{r}$, debe poder detectar errores y corregirlos. Según lo visto anteriormente, las columnas de la matriz H son ortogonales a los codewords posibles en el código ($G \cdot H^T = \mathbf{0}$), por lo que es posible utilizarla para poder detectar los errores en la transmisión:

$$\begin{aligned} \mathbf{r} &= \mathbf{e} + \mathbf{v} \\ \mathbf{r} \cdot H^T &= (\mathbf{e} + \mathbf{v}) \cdot H^T \\ \mathbf{s} = (s_0, s_1, \ldots, s_{n-k-1}) &= \mathbf{e} \cdot H^T + \underbrace{\mathbf{v} \cdot H^T}_{\mathbf{0}} \\ \mathbf{s} = (s_0, s_1, \ldots, s_{n-k-1}) &= \mathbf{e} \cdot H^T \end{aligned}$$

El vector $\mathbf{s}$ se lo conoce como *síndrome*, y es nulo solo en el caso de que no haya habido errores en la transmisión, o que el error haya sido tal que se recibió un codeword valido pero diferente al enviado. Esto último ocurre únicamente cuando el patrón de error $\mathbf{e}$ coincide con un codeword valido, y se lo conoce como patrón de error indetectable.

7.2.1. Distancia de un código

Sea $\mathbf{v}$ un codeword en $GF(2)$. El peso de Hamming $w(\mathbf{v})$ de este codeword, se define como la cantidad de componentes no nulas que posee ese codeword (cantidad de unos). Para $0 \leq i \leq n$, se pueden definir los A_i como la cantidad de codewords en $\mathcal{C}$ con peso i. Por lo tanto, las cantidades $A_0, A_1, \ldots, A_n$ se las puede definir como la distribución de pesos del código $\mathcal{C}$[4].

El peso mínimo de todos los codewords en $\mathcal{C}$ se define como:

$$w_{min}(\mathcal{C}) = \min\{w(\mathbf{v}) : \mathbf{v} \in \mathcal{C}, \mathbf{v} \neq \mathbf{0}\}$$

Asumiendo que se tiene un canal BSC con probabilidad de error p, se tienen A_i patrones de errores indetectables por cada peso i, y cada uno ocurre con probabilidad $p^i(1-p)^{n-i}$. Por lo tanto, la probabilidad de que ocurran i errores en la transmisión es $A_i p^i (1-p)^{n-i}$. De esta forma, la probabilidad de tener un error indetectable es:

$$P_u(E) = \sum_{i=1}^{n} A_i p^i (1-p)^{n-i}$$

Se puede probar, que existen códigos con una cota máxima de probabilidad de error indetectable:

$$P_u(E) \leq 2^{-(n-k)}$$

A aquellos códigos que cumplen con esta cota, se dice que son buenos códigos de detección de errores.

Sean $\mathbf{v}$ y $\mathbf{w}$ dos codewords en $GF(2)$. La distancia entre estos dos codewords se corresponde con la cantidad de bits en que difieren. Algunas propiedades importantes para la distancia son:

$$\begin{cases} d(\mathbf{v}, \mathbf{w}) + d(\mathbf{w}, \mathbf{x}) \geq d(\mathbf{v}, \mathbf{x}) \\ d(\mathbf{v}, \mathbf{w}) = w(\mathbf{v} + \mathbf{w}) \end{cases}$$

A partir de estas propiedades se puede demostrar que la distancia mínima de un código es:

$$\boxed{d_{min}(\mathcal{C}) = w_{min}(\mathcal{C})}$$

Teorema: Para cada codeword en $\mathcal{C}$ con peso i, existen i columnas en H cuyo vector suma es nulo. Contrario a esto, si existen i columnas en H cuyo vector suma es nulo, significa que que existe un codeword valido en $\mathcal{C}$ con peso i.

Teorema: El peso mínimo (o distancia mínima) de un código con matriz H, es igual a la mínima cantidad de columnas en H cuyo vector suma es nulo.

[4] Se puede ver que $A_0 + A_1 + \ldots + A_n = 2^k$

Teorema: Dado el espacio nulo de la matriz H, si no hay $d-1$ o menos columnas en H que sumen el espacio nulo, entonces la distancia mínima es al menos d.

La capacidad de un código de detectar y corregir errores esta determinada por la distancia mínima del código. Un código con distancia mínima d_{min}, ningún patrón de error con $d_{min}-1$ o menos errores puede transformar un codeword valido en otro codeword valido. Por lo tanto, si se tiene una cantidad de errores menor o igual a $d_{min}-1$, el síndrome no sera nulo, y el decodificador podrá detectar el error. En caso de tener mas errores que $d_{min}-1$, ya no se puede garantizar que se detecte el error. Por esta razón, se define a la capacidad de detección de errores como $d_{min}-1$, y la cantidad de patrones de errores detectables es:

$$\boxed{\binom{n}{1}+\binom{n}{2}+\ldots+\binom{n}{d_{min}(\mathcal{C})-1}}$$

Para $n\to\infty$, esta cantidad converge a 2^n-2^k+1.

7.3. Decodificación de un código

Una forma de decodificación, es por máxima verosimilitud, es decir que al recibir un codeword $\mathbf{r}$, se lo decodifica en un vector $\mathbf{v}$ tal que $\mathbb{P}(\mathbf{r}|\mathbf{v})$ sea máxima. Para un BSC, esto es equivalente a elegir un $\mathbf{v}$ tal que la distancia $d(\mathbf{r},\mathbf{v})$ sea mínima. Este tipo de decodificación requiere comparar $\mathbf{r}$ con los 2^k posibles codewords antes de poder tomar una decisión, lo cual lo vuelve un método lento para altos valores de k. Las regiones para esta codificación están determinadas por:

$$D(\mathbf{v}_i)=\{\mathbf{r}\in V:\mathbb{P}(\mathbf{r}|\mathbf{v}_i)\geq\mathbb{P}(\mathbf{r}|\mathbf{v}_j),i\neq j\}$$

Otro tipo de decodificación, es en base a la minimización de la distancia entre el vector recibido y los codewords validos del código (como se dijo anteriormente, una un BSC esta decodificación es igual a la anterior). Este tipo de decodificación, se puede visualizar en la figura 7.2.

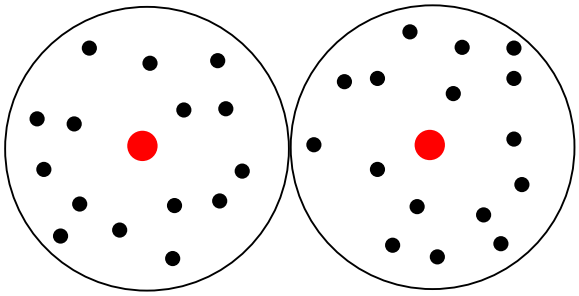

Figura 7.2: Regiones de decodificación: Cada punto dentro de la nube, permite reconocer el codeword valido enviado a partir de la tupla recibida.

Las regiones de la figura 7.2 se pueden obtener como:

$$D(\mathbf{v}_i)=\{\mathbf{r}\in V:d(\mathbf{r},\mathbf{v}_i)\leq d(\mathbf{r},\mathbf{v}_j),i\neq j\}$$

Para hacer este tipo de decodificación se debe hacer:

1. Se organizan los 2^k codewords como una matriz de $2^{n-k}\times 2^k$, con el vector nulo arriba de todo.
2. En la primer fila, en cada una de las columnas, se colocan los codewords válidos.
3. En la primer columna, en cada una de las 2^{n-k} filas, se colocan los patrones de error.
4. Debajo de cada codeword válido, se debe colocar la suma entre el codeword válido y el patrón de error correspondiente.
5. Cada posición en la matriz, representa un codeword recibido que puede ser decodificado por máxima verosimilitud.

Esta construcción se la conoce como standard array, el cual puede verse en la figura 7.3.

Cosets	Coset leaders					
$\mathcal{C}$	$\mathbf{e}_0 = \mathbf{v}_0 = \mathbf{0}$	$\mathbf{v}_1$	$\cdots$	$\mathbf{v}_i$	$\cdots$	$\mathbf{v}_{2^k-1}$
$\mathbf{e}_1 + \mathcal{C}$	$\mathbf{e}_1$	$\mathbf{e}_1 + \mathbf{v}_1$	$\cdots$	$\mathbf{e}_1 + \mathbf{v}_i$	$\cdots$	$\mathbf{e}_1 + \mathbf{v}_{2^k-1}$
$\mathbf{e}_2 + \mathcal{C}$	$\mathbf{e}_2$	$\mathbf{e}_2 + \mathbf{v}_1$	$\cdots$	$\mathbf{e}_2 + \mathbf{v}_i$	$\cdots$	$\mathbf{e}_2 + \mathbf{v}_{2^k-1}$
$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$	$\cdots$	$\vdots$
$\mathbf{e}_{2^{n-k}-1} + \mathcal{C}$	$\mathbf{e}_{2^{n-k}-1}$	$\mathbf{e}_{2^{n-k}-1} + \mathbf{v}_1$	$\cdots$	$\mathbf{e}_{2^{n-k}-1} + \mathbf{v}_i$	$\cdots$	$\mathbf{e}_{2^{n-k}-1} + \mathbf{v}_{2^k-1}$

Figura 7.3: Standard Array.

Cada fila del standard array se la conoce como *coset*, mientras que a los patrones de error se los conoce como *coset leader*. Para poder decodificar un mensaje recibido, se debe hacer:

1. Se busca el codeword recibido $\mathbf{r}$ en el standard array.
2. El patrón de error es el coset leader del codeword correspondiente.
3. Para encontrar el codeword válido, se debe sumar el patrón de error al codeword recibido: $\mathbf{c} = \mathbf{r} + \mathbf{e}$.

Algunas propiedades del standard array son:

- La suma de dos vectores en un mismo coset resulta en un codeword válido.
- No hay dos vectores iguales en ningún coset, es decir que no se repiten las tuplas.
- Todas las tuplas en un mismo coset, presentan un mismo síndrome. Por otra parte, cada coset tiene un síndrome diferente.

Dado que cada coset tiene un único síndrome, es posible agregar una columna a la tabla de la figura 7.3, en donde se coloca el síndrome correspondiente. Por lo tanto, para realizar una decodificación, se puede proceder:

1. Se calcula el síndrome del codeword recibido: $\mathbf{s} = \mathbf{r} \cdot H^T$.
2. El patrón de error es el coset leader según el síndrome obtenido.
3. Para encontrar el codeword válido, se debe sumar el patrón de error al codeword recibido: $\mathbf{c} = \mathbf{r} + \mathbf{e}$.

7.4. Códigos Cíclicos

Sea $\mathbf{v} = (v_0, v_1, \ldots, v_{n-1})$ una n-tupla de $GF(2)$. Al hacer un shifteo cíclico de $\mathbf{v}$ se obtiene:

$$\mathbf{v}^{(1)} = (v_{n-1}, v_0, \ldots, v_{n-1})$$

Definición: Un código (n, k) en bloques se dice *cíclico* si al realizar un shifteo cíclico del codeword $c \in \mathcal{C}$, el nuevo codeword obtenido también pertenece al codebook $\mathcal{C}$.

Al igual que ocurría con los códigos lineales, los códigos cíclicos pueden ser expresados en forma de polinomios:

$$\mathbf{v} = (v_0, v_1, \ldots, v_{n-1}) \rightarrow \mathbf{v}(X) = v_0 + v_1 \cdot X + \ldots + v_{n-1} \cdot X^{n-1}$$

En un código cíclico (n, k), todos los codewords distintos de cero tienen un grado mayor o igual que $n - k$, pero no mayor que $n - 1$. De todos estos posibles polinomios, existe un polinomio conocido como el *polinomio generador* $\mathbf{g}(X)$, el cual es único, de grado $n - k$, y siempre presenta el termino independiente 1. Todos los posibles codewords del codebook, pueden ser obtenidos a partir de este polinomio generador, por lo que se puede escribir:

$$\mathbf{v}(X) = \mathbf{m}(X)\,\mathbf{g}(X)$$

Donde $\mathbf{m}(X) = m_0 + m_1 \cdot X + \ldots + m_{k-1} \cdot X^{k-1}$ es un polinomio de $GF(2)$ de grado menor o igual a $k-1$, y representa el mensaje que se desea enviar. En forma matricial, se puede escribir la matriz generadora:

$$G = \begin{bmatrix} 1 & g_1 & g_2 & \cdots & g_{n-k-1} & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & g_1 & \cdots & g_{n-k-2} & g_{n-k-1} & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & g_1 & g_2 & g_3 & \cdots & g_{n-k-1} & 1 \end{bmatrix}$$

Se puede ver que cada fila de la matriz generadora, se corresponde con un shifteo circular de las demás filas. Por otra parte, esta matriz así expresada, no compone un código sistemático, pero es posible obtener uno haciendo operaciones entre filas.

Una propiedad importante del polinomio generador, es que es múltiplo de $X^n + 1$:

$$X^n + 1 = \mathbf{g}(X)\,\mathbf{f}(X) \rightarrow \mathbf{f}(X) = 1 + f_1 \cdot X + \ldots + f_{k-1} \cdot X^{k-1} + f_k \cdot X^k$$

El reciproco de $\mathbf{f}(X)$ se puede escribir como:

$$\mathbf{h}(X) = X^k \cdot \mathbf{f}\left(X^{-1}\right) = 1 + h_1 \cdot X + \ldots + h_{k-1} \cdot X^{k-1} + X^k$$

El polinomio $\mathbf{h}(X)$ se lo conoce como polinomio chequeador de paridad, y se puede probar que también es múltiplo de $X^n + 1$. La matriz chequeadora de paridad queda:

$$H = \begin{bmatrix} 1 & h_1 & h_2 & \cdots & h_{k-1} & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & h_1 & \cdots & h_{k-2} & h_{k-1} & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & h_1 & h_2 & h_3 & \cdots & h_{k-1} & 1 \end{bmatrix}$$

Nuevamente, se puede ver que todas las filas se corresponden con shifteos circulares de las demás filas.

Como se dijo anteriormente, este método de generación de codewords no permite una codificación sistemática. Para lograr esto es necesario realizar ciertos pasos. Suponiendo que se desea enviar el mensaje $\mathbf{m}$ con su respectivo polinomio $\mathbf{m}(X)$. Si a este polinomio se lo multiplica por X^{n-k} se obtiene:

$$X^{n-k} \cdot \mathbf{m}(X) = m_0 \cdot X^{n-k} + m_1 \cdot X^{n-k+1} + \ldots + m_{k-1} \cdot X^{n-1}$$

Si a este polinomio se lo divide por el polinomio generador, se obtiene:

$$X^{n-k} \cdot \mathbf{m}(X) = \mathbf{a}(X) \cdot \mathbf{g}(X) + \mathbf{b}(X)$$

Donde $\mathbf{a}(X)$ y $\mathbf{b}(X)$ son el cociente y el resto de la división, respectivamente. Dado que el grado de $\mathbf{g}(X)$ es $n-k$, el grado de $\mathbf{b}$ debe ser menor o igual que $n-k-1$:

$$\mathbf{b}(X) = b_0 + b_1 \cdot X + \ldots + b_{n-k-1} \cdot X^{n-k-1}$$

Reescribiendo las expresiones, se puede escribir:

$$\mathbf{b}(X) + X^{n-k} \cdot \mathbf{m}(X) = \mathbf{a}(X) \cdot \mathbf{g}(X)$$

En esta expresión se puede ver que $\mathbf{g}(X)$ es divisor de todo el término del lado izquierdo del igual, y como $\mathbf{b}(X) + X^{n-k} \cdot \mathbf{m}(X)$ es de grado menor o igual que $n-1$, esto significa que $\mathbf{b}(X) + X^{n-k} \cdot \mathbf{m}(X)$ es un polinomio de un codeword valido en $\mathcal{C}$. En forma de tupla, se tiene:

$$\mathbf{b}(X) + X^{n-k} \cdot \mathbf{m}(X) \rightarrow (b_0, b_1, \ldots, b_{n-k-1}, m_0, m_1, \ldots, m_{k-1})$$

Donde se puede ver que la expresión obtenida esta escrita de forma sistemática.

Asumiendo que se desea transmitir mensajes de la forma $\mathbf{m}_i(X) = X^i$, es decir que son nulos salvo en la posición i-ésima, se puede escribir:

$$X^{n-k} \cdot \mathbf{m}_i(X) = X^{n-k-i} = \mathbf{a}_i(X)\mathbf{g}(X) + \mathbf{b}(X) \rightarrow \mathbf{b}(X) + X^{n-k-i} = \mathbf{a}_i(X)\mathbf{g}(X)$$

Por lo tanto, $\mathbf{b}(X) + X^{n-k-i}$ es un codeword valido de $\mathcal{C}$. Las matrices generadora y chequeadora de paridad son:

$$G_{sis} = \begin{bmatrix} b_{0,0} & b_{0,1} & \cdots & b_{0,n-k-1} & 1 & 0 & 0 & \cdots & 0 \\ b_{1,0} & b_{1,1} & \cdots & b_{1,n-k-1} & 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ b_{k-1,0} & b_{k-1,1} & \cdots & b_{k-1,n-k-1} & 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$$

$$H_{sis} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & b_{0,0} & b_{1,0} & \cdots & b_{k-1,0} \\ 0 & 1 & 0 & \cdots & 0 & b_{0,1} & b_{1,1} & \cdots & b_{k-1,1} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & b_{0,n-k-1} & b_{1,n-k-1} & \cdots & b_{k-1,n-k-1} \end{bmatrix}$$

7.4.1. Detección de errores en ráfagas

1. Todo código cíclico (n, k) detecta ráfagas de longitud menor o igual que $n - k$.
2. Si la longitud de la ráfaga de error es igual a $n - k + 1$, entonces se pueden detectar todas menos la que coincide con el polinomio $g(X)$. Detecta con una probabilidad de $p = 1 - 2^{-(n-k)}$.
3. Si la longitud de la ráfaga de error es mayor que $n - k + 1$, entonces se puede detectar todas las ráfagas que no coincidan con palabras del código, es decir todas las ráfagas que no sean múltiplos de $g(X)$. Detecta con probabilidad $p = 1 - 2^{-(n-k+1)}$.

Para detectar un error, se puede hacer:

$$r(X) = c(X) + \varepsilon(X) \rightarrow \frac{r(X)}{g(X)} = \frac{\cancel{c(X)}}{\cancel{g(X)}} + \frac{\varepsilon(X)}{g(X)} = \frac{\varepsilon(X)}{g(X)}$$

Por lo tanto, si el resto de dividir $\frac{r(X)}{g(X)}$ es nulo, significa que no se produjo ningún error.