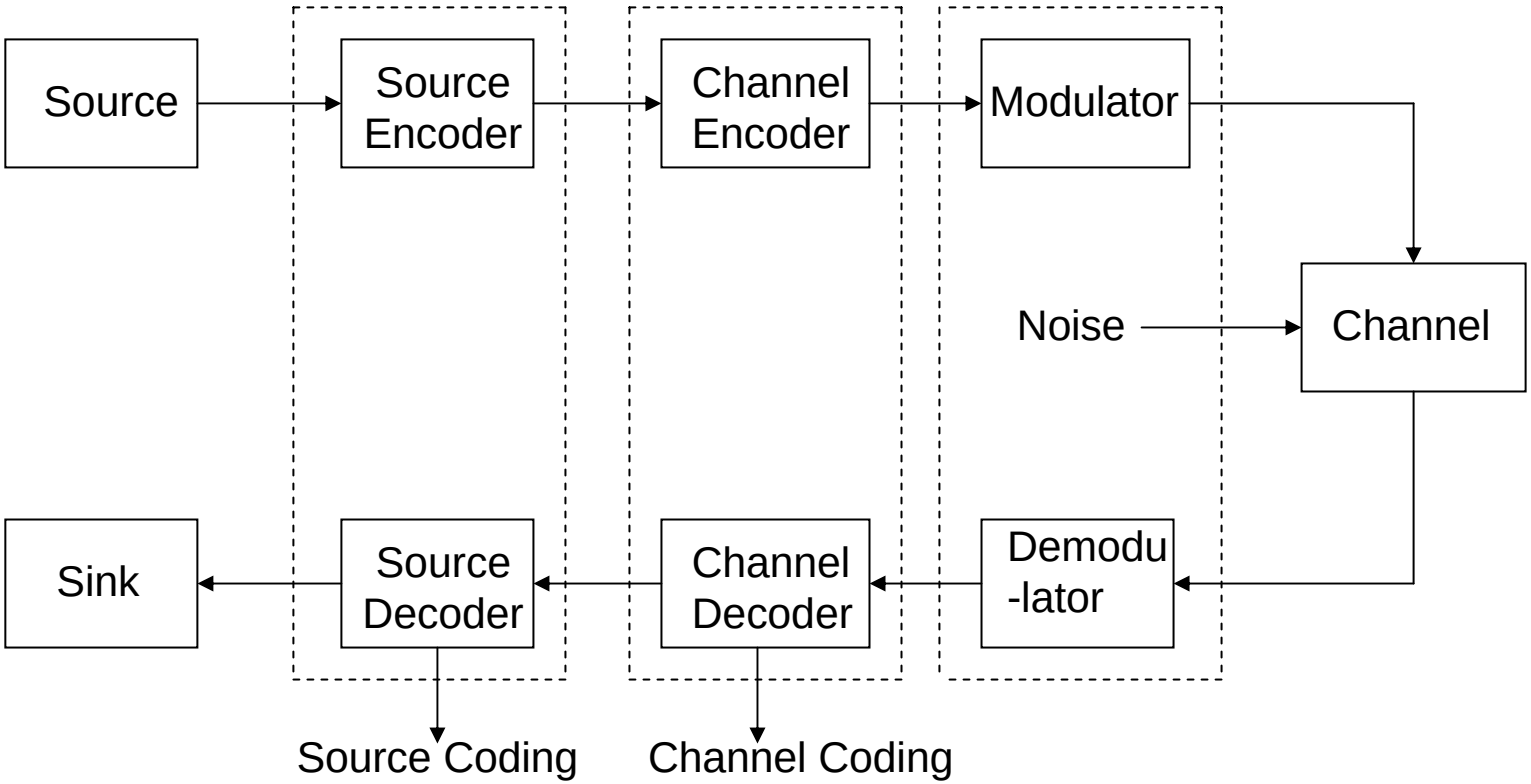
Information Theory and Coding

Communication System Block Diagram



Modulation converts bits into coding analog waveforms suitable for transmission over physical channels. We will not discuss modulation in any detail in this course.

Information Theory and Coding

Lecture 1

Probability Review

Origin in gambling
Laplace - combinatorial counting, circular discrete geometric probability - continuum

A N Kolmogorous 1933 Berlin

Notion of an experiment

Let Ω be the set of all possible outcomes. This set is called the sample set.

Let $\mathcal{A}$ be a collection of subsets of Ω with some special properties. $\mathcal{A}$ is then a collection of events $(\Omega, \mathcal{A})$ are jointly referred to as the sample space.

Then a probability model/space is a triplet $(\Omega, \mathcal{A}, P)$ where P satisfies the following properties

1 Non negativity : $P(A) \geq 0 \ \ \forall A \ \epsilon \ \mathcal{A}$

2 Additivity : If $\{A_n, n \geq 1\}$ are disjoint events in $\mathcal{A}$,

$$\text{then } P(U_{n=1}^{\infty} A_n) = \sum_{n=1}^{\infty} P(A_n)$$

3 Bounded : $P(\Omega) = 1$

* Example : $\Omega = \{T, H\} \ \ \mathcal{A} = \{\phi, \{T, H\}, \{T\}, \{H\}\}$

$$P(\{H\}) = 0.5$$

When Ω is discrete (finite or countable) $\mathcal{A} = lP(\Omega)$, where lP is the power set. When Ω takes values from a continiuum, $\mathcal{A}$ is a much smaller set. We are going to hand-wave out of that mess. Need this for consistency.

* Note that we have not said anything about how events are assigned probabilities. That is the engineering aspect. The theory can guide in assigning these probabilities, but is not overly concerned with how that is done.

There are many consequences

1. $P(A^{\subset}) = 1 - P(A) \Rightarrow P(\phi) = 0$
2. $P(A \cup B) = P(A) + P(B) - P(A \cap B)$
3. Inclusion - Exclusion principle :

   If $A_1, A_2, \ldots, A_n$ are events, then

$$P(\cup_{j=1}^{n} A_j) = \sum_{j=1}^{n} P(A_j) - \sum_{1 \le i < j \le n} P(A_i \cap A_j)$$
$$+ \sum_{1 \le i < j < k \le n} P(A_i \cap A_j \cap A_k) .. (-1)^n P(A_1 \cap A_2 \ldots \cap A_n)$$

   Can be proved using induction
4. Monotonivity : $A \subset B \Rightarrow (P)A \le P(B)$
5. Continuity : First define limits : $A = \cup A_n$ or $A = \cap A_n$
   (a) If$A_n \uparrow A$, then $P(A_n) \uparrow P(A)$

$$A_1 \subset A_2 \ldots \subset A, \qquad \lim_{n \to \infty} P(A_n) = P(\lim_{n \to \infty} A_n)$$
$$fcont \Rightarrow \lim_{x_n \to x} f(X_n) = f(\lim_{x_n \to x} X_n) = f(x)$$

   (b) $A_n \downarrow A$, then $P(A_n) \downarrow P(A)$

Proof :

a) $A_1 \subset A_2 \ldots \subset A_n$
$B_1 = A_1, B_2 = A_2 \backslash A_1 = A_2 \cap A_1^{\subset}$
$B_3 = A_3 \backslash A_2$

B_n is a sequence of disjoint "annular rings". $\boxed{\cup_{k=1}^{n} B_n = A_n}$

$\cup_{n=1}^{\infty} B_n = \cup_{n=1}^{\infty} A_n = A$

By additivity of P

$$P(A) = P(\cup_{n=1}^{\infty} B_n) = \sum_{n=1}^{\infty} P(B_n) = \lim_{n \to \infty} \sum_{k=1}^{n} P(B_n)$$
$$= \lim_{n \to \infty} P(\cup_{k=1}^{n} B_k) = \lim_{n \to \infty} P(A_n)$$

We have, $P(A) = \lim_{n \to \infty} p(A_n)$

b)

$$A_1 \supset A_2 \supset A_n \ldots \supset A \qquad A \subset B \quad A \supset B$$

$$A^{\subset} \supset B^{\subset} \quad A^{\subset} \subset B^{\subset}$$

$$A_1^{\subset} \subset A_z^{\subset} \ldots \subset A^{\subset}$$

$$P(A^{\subset}) = \lim_{n\to\infty} P(A_n^{\subset})$$

$$\Rightarrow 1 - P(A) = \lim_{n\to\infty} 1 - P(A_n) = 1 - \lim_{n\to\infty} P(A_n)$$

$$\Rightarrow P(A) = \lim_{n\to\infty} P(A_n)$$

Limits of sets. Let $A_n \subset \mathcal{A}$ be a sequence of events

$$inf_{k\geq n} A_k = \cap_{k=n}^{\infty} A_k \qquad sup_{k\geq n} A_k = \cup_{k=n}^{\infty} A_k$$

$$\liminf_{n\to\infty} A_n = \cup_{n=1}^{\infty} \cap_{k=n}^{\infty} A_k \qquad \limsup_{n\to\infty} A_n = \cap_{n=1}^{\infty} \cup_{k=n}^{\infty} A_k$$

If $\liminf_{n\to\infty} A_n = \limsup_{n\to\infty} A_n = A$, then we say A_n converges to A

Some useful interpretations :

$$\begin{aligned}
\limsup A_n &= \{W : \sum 1_{A_n}(W) = \infty\} \\
&= \{W : W \epsilon A_{n_k}, K = 1, 2, \ldots\} \text{for some sequences} n_k \\
&= \{A_n 1 : 0\} \\
\liminf A_n &= \{W : A.W \epsilon A_n\} \text{for all n except a finite number} \\
&= \{W : \sum 1_{A_n}^{\subset}(W) < \infty\} \\
&= \{W : W \epsilon A_n \quad \forall_n \geq n_o(W)\}
\end{aligned}$$

Borel Cantelli Lemma : Let $\{A_n\}$ be a sequence of events.
$If \sum_{n=1}^{\infty} P(A_n) < \infty$ then $P(A_n 1 : 0) = P(\limsup A_n) = 0$

$$\begin{aligned}
P(A_n 1 : 0) &= P(\lim_{n\to\infty} \cup_{j\geq n} A_j) \\
&= \sum_n P(A_n) \leq \infty \\
&= \lim_{n\to\infty} P(\cup_{j\geq n} A_j) \leq \lim_{n\to\infty} \sum_{j=n}^{\infty} P(A_j) = 0
\end{aligned}$$

Converse to B-C Lemma

If $\{A_n\}$ are independent events such that
$\sum_n P(A_n) = \infty$, then $P\{A_n 1 : 0\} = 1$

$$\begin{aligned} P(A_n 1:0) &= P(\lim_{n\to\infty} \cup_{j\geq n} A_j) \\ &= \lim_{n\to\infty} P(\cup_{j\geq n} A_j) \\ &= \lim_{n\to\infty} (1 - P(\cap_{j\geq n} A_j^C)) \\ &= 1 - \lim_{n\to\infty} \lim_{m\to\infty} \Pi_{k=n}^m (1 - P(A_k) \end{aligned}$$

$1 - P(A_k) \leq e_m^{-P(A_k)}$

$$\begin{aligned} \text{therefore} \lim_{m\to\infty} \Pi_{k=n}^m (1 - P(A_k)) &\leq \lim_{m\to\infty} \Pi_{k=n}^m e^{-P(A_k)} \\ &= \lim_{m\to\infty} e^{-\sum_{k=n}^m P(A_k)} \\ &= e^{-\sum_{k=n}^\infty P(A_k)} = e^{-\infty} = 0 \end{aligned}$$

Random Variable :
Consider a random experiment with the sample space $(\Omega, \mathcal{A})$. A random variable is a function that assigns a real number to each outcome in Ω.

$X : \Omega \longrightarrow \mathcal{R}$

In addition for any interval (a, b) we want $X^{-1}((a, b)) \quad \epsilon \quad \mathcal{A}$. This is an technical condition we are stating merely for completeness.

Such a function is called Borel measurable cumulative.

The cumulative distribution function for a random variable X is defined as

$$F(x) = P(X \leq x)^{\sim}, P(\{W 1_{\epsilon\Omega} X(W) \leq x\}), X \epsilon \mathcal{R}$$

A random variable is said to discrete if it can take only a finite or countable/denumerable. The probability mass function (PMF) gives the probability that X will take a particular value.

$$P_X(x) = P(X = x)$$

We have $F(x) = \sum_{y \leq x} P_x(y)$

A random variable is said to be continuous if there exists a function $f(x)$, called the probability distribution function such that

$$F(x) = P(X \leq x) = \int_{-\infty}^{x} f(y)\mathrm{d}y$$

differentiating, we get $f(x) = \frac{\mathrm{d}}{\mathrm{d}x}F(x)$

The distribution function $F_x(x)$ satisfies the following properties

1) $F(x) \geq 0$
2) $F(x)$ is right contininous $\lim_{x_n \downarrow x} F(x) = F(x)$
3) $F(-\infty) = 0, F(+\infty) = 1$

Let $A_k = \{W : X \leq x_n\}, A = \{W : X \leq x\}$

Clearly $A_1 \supset A_2 \supset A_4 \supset A_n \supset A \quad LetA = \cap_{k=1}^{\infty} A_k$

Then $A = \{W : X(W) \leq x\}$

We have $\lim_{n\to\infty} P(A_n) = \lim_{x_n \downarrow x} F(x_n)$

By continuity of P, $P(\lim_{n\to\infty} A_n) = P(A) = F(x)$

Independence

Suppose $(\Omega, \mathcal{A}, P)$ is a probability space. Events $A, B \, \epsilon \, \mathcal{A}$ are independent if

$$P(A \cap B) = P(A).P(B)$$

In general events $A_1, A_2, \ldots A_n$ are said to be independent if

$$P(\cap_{i \epsilon I} A_i) = \Pi_{i \epsilon I} P(A_i)$$

for all finite I C {1, . . . , n}

Note that pairwise independence does not imply independence as defined above.
Let $\Omega = \{1, 2, 3, 4\}$, each equally mobable let $A_1 = \{1, 2\} A_2 = \{1, 3\}$ and $A_3 = \{1, 4\}$.
Then only two are independent.

A finite collection of random variables $X_1, \ldots, X_k$ is independent if

$$P(X_1 \leq x_1, \ldots, X_k \leq x_k) = \Pi_{i=1}^{k} P(X_i \leq x_i) \ \ \forall \, x_i \, \epsilon \, \mathcal{R}, 1 \leq i \leq k$$

Independence is a key notion in probability. It is a technical condition, don't rely on intuition.

Conditional Probability
The probability of event A occuring, given that an event B has occured is given by

$$P(A|B) \ = \ \frac{P(A \cap B)}{P(B)}, \quad P(B) > 0$$

If A and B are independent, then

$$P(A|B) \ = \ \frac{P(A)P(B)}{P(B)} \ = \ P(A) \text{ as expected}$$

In general $P(\cap_{i=1}^{n} A_1) = \ \ P(A_1)P(A_2|A_1) \ \ P(A_n|A_1, A_2, \ldots, A_{n-1})$

Expected Value

The expectation, average or mean of a random variable is given by

$$\text{EX} = \begin{cases} = \sum x P(X = x) & X \text{is discreet} \\ \int_{-\infty}^{\infty} x f(x) \mathrm{d}x & \text{continuous} \end{cases}$$

In general $EX = \int_{x=-\infty}^{\infty} x \mathrm{d}F(x)$ This has a well defined meaning which reduces to the above two special cases when X is discrete or continuous but we will not explore this aspect any further.

We can also talk of expected value of a function

$$Eh(X) = \int_{-\infty}^{\infty} h(x)\mathrm{d}F(x)$$

Mean is the first moment. The n^{th} moment is given by
$EX^n = \int_{-\infty}^{\infty} x^n \mathrm{d}F(x)$ if it exists
$VarX = E(X - EX)^2 = EX^2 - (EX)^2$ $\sqrt{VarX}$ is called the std deviation

Conditional Expectation :
If X and Y are discrete, the conditional p.m.f. of X given Y is defined as

$$P(X = x|Y = y) \; = \quad \frac{P(X = x, Y = y)}{P(Y = y)} \quad P(Y = y) > 0$$

The conditional distribution of X given Y=y is defined as $F(x|y) = P(X \leq x|Y = y)$ and the conditional expectation is given by

$$E[X|Y = y] \; = \; \sum xP(X = x|Y = y)$$

If X and Y are continuous, we define the conditional pdf of X given Y as

$$f_{X(Y)}\,(x|y) = \frac{f(x, y)}{f(y)}$$

The conditional cumulative distribution in cdf is given by

$$F_{X|Y}\,(x|y) = \int_{-\infty}^{x} f_{X|Y}\,(x|y)\mathrm{d}x$$

Conditional mean is given by

$$E[X|Y = y] = \int x\, f_{X|Y}\,(x|y)\mathrm{d}x$$

It is also possible to define conditional expectations functions of random variables in a similar fashion.

Important property If X & Y are rv.

$$EX \;=\; E[EX|Y] \;=\; \int E(X|Y=y)\mathrm{d}F_Y(y)$$

Markov Inequality :

Suppose $X \geq 0$. Then for any $a > 0$

$P(X \geq a) \leq \frac{EX}{a}$

$EX = \int_o^a x\mathrm{d}F(x) + \int_a^\infty x\mathrm{d}F(x)$

$\int_a^\infty a\mathrm{d}F(x) = a.p(X \geq a)$

$P(X \geq a) \leq \frac{EX}{a}$

Chebyshev's inequality :

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

Take $Y = |X - a|$

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

The weak law of Large Number

Let $X_1, X_2, \ldots$ be a sequence of independent and identically distributed random variables with mean N and finite variance σ^2

Let $S_n = \sum_{\frac{k=1}{n}}^{n} X_k$

Then $\quad P(|S_n - N| \geq \delta|) \;\Rightarrow 0$ as n $\Rightarrow \infty \;\; \forall\delta$

Take any $\delta > 0$

$P(|S_n - N| \geq \delta) \leq \frac{VarS_n}{\delta^2}$

$= \frac{1}{n^2} \;\; \frac{n\sigma^2}{\delta^2} = \frac{1}{n} \;\; \frac{\sigma^2}{\delta^2}$

$\lim_{n\to\infty} P(|S_n N| \geq \delta) \longrightarrow 0$

Since δ rvar pushed arbitrarily

$$\lim_{n\to\infty} P(|S_n - N| \geq 0) = 0$$

The above result holds even when σ^2 is infinite as long as mean is finite.

Find out about how L-S work and also about WLLN

We say $S_n \Rightarrow N$ in probability

Information Theory and Coding

Lecture 2

Entropy

The entropy H(X) of a discrete random variable is given by

$$\begin{aligned} H(X) &= \sum_{x\epsilon\mathcal{X}} P(x)\, log\frac{1}{P(x)} \\ &= -\sum_{x\epsilon\mathcal{X}} P(x)\, logP(x) \\ &= E\, log\frac{1}{P(X)} \end{aligned}$$

$log\,\frac{1}{P(X)}$ is called the self-information of X. Entropy is the expected value of self information.

Properties :

1. $H(X) \geq 0$

   $P(X) \leq 1 \Rightarrow \frac{1}{P(X)} \geq 1 \Rightarrow log\frac{1}{P(X)} \geq 0$

2. Let$H_a(X) = Elog_a\frac{1}{P(X)}$

   Then $H_a(X) = (log_{a^2}).H(X)$

3. Let $|\mathcal{X}| = M$ Then $H(X) \leq logM$

$$\begin{aligned} H(x) - logM &= \sum_{x\epsilon\mathcal{X}} P(x)\, log\frac{1}{P(x)} - logM \\ &= \sum_{x\epsilon\mathcal{X}} P(x)\, log\frac{1}{P(x)} - \sum_{x\epsilon\mathcal{X}} P(x)\, logM \\ &= \sum_{x\epsilon\mathcal{X}} P(x)\, log\frac{1}{MP(x)} \\ &= E\, log\frac{1}{MP(x)} \\ Jensens' &\leq logE\left(\frac{1}{MP(x)}\right) \\ &= log\sum P(x)\frac{1}{MP(x)} \\ &= 0 \end{aligned}$$

therefore $H(X) \leq logM$ When $P(x) = \frac{1}{M} \ \forall \ x \ \epsilon \ \mathcal{X}$

$H(X) = \sum\limits_{x\sigma\mathcal{X}} \frac{1}{M} \, logM = logM$

4. $H(X) = 0 \Rightarrow X$ is a constant

Example : $\mathcal{X} = \{0, 1\}$ $P(X = 1) = P$, $P(X = 0) = 1 - P$

$H(X) = P \, log\frac{1}{P} + \ (1 - P) \, log\frac{1}{1-P} \ = \ H(P)$(the binary entropy function)

We can easily extend the definition of entropy to multiple random variables. For example, let $Z = (X, Y)$ where X and Y are random variables.

Definition : The joint entropy H(X,Y) with joint distribution P(x,y) is given by

$$\begin{aligned} H(X,Y) &= +\sum_{x\epsilon\mathcal{X}}\sum_{y\epsilon\mathcal{Y}} P(x,y) log\left[\frac{1}{P(x,y)}\right] \\ &= E \, log\frac{1}{P(X,Y)} \end{aligned}$$

If X and Y are independent, then

$$\begin{aligned} H(X,Y) &= \sum_{x\epsilon\mathcal{X}}\sum_{y\epsilon\mathcal{Y}} P(x).P(y) \, log\frac{1}{P(x), P(y)} \\ &= \sum_{x\epsilon\mathcal{X}}\sum_{y\epsilon\mathcal{Y}} P(x).P(y) \, log\frac{1}{P(x)} + \sum_{x\epsilon\mathcal{X}}\sum_{y\epsilon\mathcal{Y}} P(x)P(y) \, log\frac{1}{P(y)} \\ &= \sum_{y\epsilon\mathcal{Y}} P(y)H(X) + \sum_{x\epsilon\mathcal{X}} P(x)H(Y) \\ &= H(X) + H(Y) \end{aligned}$$

In general, given $X_1, \ldots, X_n$. i.i.d. random variables,

$$H(X1, \ldots, X_n) = \sum_{i=1^n} H(X_i)$$

We showed earlier that for optimal coding

$$H(X) \leq L^{-} < H(X) + 1$$

What happens if we encode blocks of symbols?

Lets take n symbols at a time

$X^n = (X1, \ldots, X_n)$ Let L^-n be the optimal code

$H(X1, \ldots, X_n) \leq L^{-n} < H(X1, \ldots, X_n) + 1$

$H(X1, \ldots, X_n) = \sum H(X_i) = nH(X)$

$H(X) \leq L^{-n} \leq nH(X) + 1$

$H(X) \leq \frac{L^{-n}}{n} \leq H(X) + \frac{1}{n}$

Therefore, by encoding a block of source symbols at a time, we can get as near to the entropy bound as required.

Information Theory and Coding

Lecture 3

Asymptotic Equipartition Property

The Asymptotic Equipartition Property is a manifestation of the weak law of large numbers.

Given a discrete memoryless source, the number of strings of length $n = |\mathcal{X}|^n$. The AEP asserts that there exists a typical set, whose cumulative probability is almost 1. There are around $2^{nh(X)}$ strings in this typical set and each has probability around $2^{-nH(X)}$

"Almost all events are almost equally surprising."

Theorem : Suppose $X_1, X_2, \ldots$ are iid with distribution p(x)

Then $-\frac{1}{n} log\, P(X_1, \ldots, X_n) \rightarrow H(X)$ in probability

Proof : Let $Y_k = log\left[\frac{1}{P(X_k)}\right]$. Then Y_k are iid and $EY_k = H(X)$

Let $S_n = \frac{1}{n} \sum_{k=1}^{n} Y_k$. By WLLN $S_n \rightarrow H(x)$ in probability

But $S_n = \frac{1}{n} \sum_{k=1}^{n} log\frac{1}{P(X_k)} = -\sum_{k=1}^{n} \frac{log}{n} P(X_k)$

$= -\frac{1}{n} log(P(X_1, \ldots, X_n))$

Definition : The typical set A_ϵ^n is the set of sequences $x^n = (x_1, \ldots, x_n)\, \epsilon\, \mathcal{X}^n$ such that $2^{-n(H(X)+\epsilon)} \leq P(x_1, \ldots, x_n) \leq 2^{-n(H(X)-\epsilon)}$

Theorem :

a) If $(x_1, \ldots, x_n)\, \epsilon\, A_\epsilon^n$, then
$H(X) - \epsilon \leq -\frac{1}{n} logP(x_1, \ldots, x_n) \leq H(X) + \epsilon$

b) $Pr(A_\epsilon^n) > 1 - \epsilon$ for large enough n

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

d) $|A_\epsilon^n| \geq (1-\epsilon)\, 2^{n(H(X)-\epsilon)}$ for large enough n

Remark

1. Each string in A_ϵ^n is approximately equiprobable
2. The typical set occur with probability 1
3. The size of the typical set is roughly $2^{nH(X)}$

Proof :

a) Follows from the definition

b) AEP

$\Rightarrow -\frac{1}{n} log\, P(X_1, \ldots, X_n) \rightarrow H(X)$ in prob

$\Pr\left[|-\frac{1}{n} log\, P(X_1, \ldots, X_n) - H(X)| < \epsilon\right] > 1-\delta$ for large enough n

Take $\delta = \epsilon_1 \;\; Pr(A_\epsilon^n) > 1-\delta$

c)

$$\begin{aligned} 1 &= \sum_{x^n \epsilon \mathcal{X}^n} P(x^n) \\ &\geq \sum_{x^n \epsilon A_\epsilon^n} P(x^n) \\ &\geq \sum_{x^n \epsilon A_\epsilon^n} 2^{-n(H(X)+\epsilon)} \\ &= |A_\epsilon^n| \,.\, 2^{-n(H(X)+\epsilon)} \Rightarrow |A_\epsilon^n| \;\leq\; 2^{n(H(X)+\epsilon)} \end{aligned}$$

d)

$$\begin{aligned} Pv(A_\epsilon^n) &> 1-\epsilon \\ \Rightarrow 1-\epsilon &< Pv(A_\epsilon^n) \\ &= \sum_{x^n \epsilon A_\epsilon^n} Pv(x^n) \\ &\leq \sum_{x^n \epsilon A_\epsilon^n} 2^{-n(H(X)-\epsilon)} \\ &= |A_\epsilon^n| \,.\, 2^{-n(H(X)-\epsilon)} \\ |A_\epsilon^n| &\geq (1-\epsilon) \,.\, 2^{-n(H(X)-\epsilon)} \end{aligned}$$

strings of length $n = |\mathcal{X}|^n$

typical strings of length $n \cong 2^{nH(X)}$

$\lim \frac{2^{nH(X)}}{|X|n}$

$= \lim 2^{-n(log|X|-H(X))} \rightarrow 0$

One of the consequences of AEP is that it provides a method for optimal coding. This has more theoretical than practical significance.

Divide all strings of length n into A_ϵ^n and $A_\epsilon^{n\subset}$

We know that $|A_\epsilon^n| \leq 2^{n(H(X)+\epsilon)}$

Each sequence in A_ϵ^n is represented by its index in the set. Instead of transmitting the string, we can transmit its index.

$$\#\text{bits required} = \lceil log(|A_\epsilon^n|)\rceil < n(H(X)+\epsilon)+1$$

Prefix each sequence by a 0, so that the decoder knows that what follows is an index number.

$$\#\text{bits} \leq n(H(X)+\epsilon)+2$$

For $X^n \, \epsilon \, A_\epsilon^{n\subset}$,

$$\#\text{bits required} = nlog|\mathcal{X}|+1+1$$

Let $l(x^n)$ be the length of the codeword corresponding to x^n. Assume n is large enough that $Pv(A_\epsilon^n) > 1-\epsilon$

$$\begin{aligned}
El(x^n) &= \sum_{x^n} P(x^n)l(x^n) \\
&= \sum_{x^n \epsilon A_\epsilon^n} P(x^n)l(x^n) + \sum_{x^n \epsilon A_\epsilon^{n\subset}} P(x^n)l(x^n) \\
&\leq \sum_{x^n + A_\epsilon^n} P(x^n)[(nH+\epsilon)+2] \; + \; \sum P(x^n)(n\, log|X|+2) \\
&= Pv(A_\epsilon^n)\,.\,(n(H+\epsilon)+2) + Pv(A_\epsilon^{n\subset})\,.\,(n\, log|X|+2) \\
&\leq n(H+\epsilon)+2+\epsilon.n\, log|X| \\
&= n(H+\epsilon^1) \quad \epsilon^1 = \epsilon + \epsilon\, log|X| + \frac{2}{n}
\end{aligned}$$

Theorem : For a DMS, there exists a UD code which satisfies

$E\left(\frac{1}{n}\, l(x^n)\right) \leq H(X) + \epsilon$ for n sufficiently large

Information Theory and Coding

Lecture 4

Data Compression

The conditional entropy of a random variable Y with respect to a random variable X is defined as

$$\begin{aligned} H(Y|X) &= \sum_{x\epsilon\mathcal{X}} P(x)H(Y|X=x) \\ &= \sum_{x\epsilon\mathcal{X}} P(x) \sum_{y\epsilon\mathcal{Y}} P(y|x) log \frac{1}{P(y|x)} \\ &= \sum_{x\epsilon\mathcal{X}} \sum_{y\epsilon\mathcal{Y}} P(x,y) log \frac{1}{P(y|x)} \\ &= E \frac{1}{logP(y|x)} \end{aligned}$$

In general, suppose $X = (X_1, \ldots, X_n) \quad Y = (Y_1, \ldots, Y_m)$

Then $H(X|Y) = E\frac{1}{logP(Y|X)}$

Theorem : (Chain Rule)

$$H(XY) = H(X) + H(Y|X)$$

$$\begin{aligned} H(X,Y) &= -\sum_{x\epsilon\mathcal{X}} \sum_{y\epsilon\mathcal{Y}} P(x,y) logP(x,y) \\ &= -\sum_{x\epsilon\mathcal{X}} \sum_{y\epsilon\mathcal{Y}} P(x,y) logP(x).P(y|x) \\ &= -\sum_{x} \sum_{y} P(x,y) logP(x) - \sum_{x\epsilon\mathcal{X}} \sum_{y\epsilon\mathcal{Y}} P(x,y) log(y|x) \\ &= -\sum_{x} P(x) logP(x) - \sum_{x\epsilon\mathcal{X}} \sum_{y\epsilon\mathcal{Y}} P(x,y) log(y|x) \\ &= H(X) + H(Y|X) \end{aligned}$$

Corollary :

1)

$$\begin{aligned} H(X,Y|Z) &= H(X|Z) + H(Y|X,Z) \\ &= E \frac{1}{logP(y|x,z)} \end{aligned}$$

2)

$$\begin{aligned} H(X_1, \ldots, X_n) &= \sum_{k=1}^{n} H(X_k|X_{k-1}, \ldots, X_1) \\ H(X_1, X_2) &= H(X_1) + H(X_2|X_1) \\ H(X_1, X_2, X_3) &= H(X_1) + H(X_2, X_3|X_1) \\ &= H(X_1) + H(X_2|X_1) + H(X_3|X_1, X_2) \end{aligned}$$

3) $H(Y) \leq H(Y|X)-$

Stationary Process : A stochastic process is said to be stationery if the joint distribution of any subset of the sequence of random variables is invariant with respect to shifts in the time index.

$$Pr(X_1 = x_1, \ldots, X_n = x_n) = Pr(X_{1+t} = x_1, \ldots, X_{n+t} = x_n)$$

$$\forall\ t\ \epsilon\ \mathcal{Z} \text{ and all } x_1,\ \ldots,\ x_n\ \epsilon\ \mathcal{X}$$

Remark : $H(X_n|X_{n-1}) = H(X_2|X_1)$

Entropy Rate : The entropy rate of a stationery stochastic process is given by

$$H = \lim_{n\to\infty} \frac{1}{n} H(X_1, \ldots, X_n)$$

Theorem : For a stationery stochastic process, H exists and further satisfies

$$H = \lim_{n\to\infty} \frac{1}{n} H(X_1, \ldots, X_n) = \lim_{n\to\infty} H(X_n|X_{n-1}, \ldots, X_1)$$

Proof : We will first show that $\lim H(X_n|X_{n-1}, \ldots X_1)$ exists and then show that

$$\lim_{n\to\infty} \frac{1}{n} H(X_1, \ldots, X_n) = \lim_{n\to\infty} H(X_n|X_{n-1}, \ldots, X_1)$$

Suppose $\lim_{n\to\infty} x_n = x$. Then we mean for any $\epsilon > 0$, there exists a number N_ϵ such that

$$|x_n - x| < \epsilon \qquad \forall n \geq N_\epsilon$$

Theorem : Suppose x_n is a bounded monotonically decreasing sequence, then $\lim\limits_{n\to\infty} x_n$ exists.

$$\begin{aligned} H(X_{n+1}|X_1,\ldots,X_n) &\leq H(X_{n+1}|X_2,\ldots,X_n) \\ &= H(X_n|X_1,\ldots,X_{n-1}) \text{by stationarity} \end{aligned}$$

$\Rightarrow H(X_n|X_1,\ldots,X_{n-1})$ is monotonically decreasing with n

$$0 \leq H(X_n|X_1,\ldots,X_{n-1}) \leq H(X_n) \leq log|\mathcal{X}|$$

Cesaro mean

Theorem : If $a_n \to a,$ then $b_n = \frac{1}{n}\sum\limits_{k=1}^{n} a_k \to a$

WTS. $\forall\epsilon > 0, \exists N_\epsilon$ s.t. $\quad |b_n - a| < \epsilon \quad \forall_n \geq N_\epsilon$

We know $a_n \to a \qquad \exists N_{\frac{\epsilon}{2}}$ s.t $n \geq N_{\frac{\epsilon}{2}}$

$$\begin{aligned} |a_n - a| &\leq \frac{\epsilon}{2} \\ |b_n - a| &= \left|\frac{1}{n}\sum_{k=1}^{n}(a_k - a)\right| \\ &\leq \frac{1}{n}\sum_{k=1}^{n}|a_k - a| \\ &\leq \frac{1}{n}\sum_{k=1}^{N_{\epsilon|Z}}|a_k - a| + \frac{n - N(\epsilon)}{n}\frac{\epsilon}{Z} \\ &\leq \frac{1}{n}\sum_{k=1}^{N_{\epsilon|Z}}|a_k - a| + \frac{\epsilon}{Z} \end{aligned}$$

Choose n large enough that the first term is less than $\frac{\epsilon}{Z}$

$$|b_n - a| \leq \frac{\epsilon}{Z} + \frac{\epsilon}{Z} = \epsilon \;\; \forall n \geq N_\epsilon^*$$

Now we will show that

$$\begin{aligned} \lim H(X_n|X_1,\ldots,X_{n-1}) &\rightarrow \lim \frac{1}{n}H(X_1,\ldots,X_n) \\ \frac{H(X_1,\ldots,X_n)}{n} &= \frac{1}{n}\sum_{k=1}^{n} H(X_1|X_{k-1},\ldots,X_1) \\ & \quad \downarrow \\ \lim \frac{H(X_1,\ldots,X_n)}{n} &= \lim_{n\rightarrow\infty} H(X_n|X_1,\ldots,X_{n-1}) \end{aligned}$$

Why do we care about entropy rate H ? Because $A\epsilon P$ holds for all stationary ergodic process,

$$-\frac{1}{n}logP(X_1,\ldots,X_n) \rightarrow H \text{ in prob}$$

This can be used to show that the entropy rate is the minimum number of bits required for uniquely decodable lossless compression.

Universal data/source coding/compression are a class of source coding algorithms which operate without knowledge of source statistics. In this course, we will consider Lempel. Ziv compression algorithms which are very popular Winup & Gup use version of these algorithms. Lempel and Ziv developed two version LZ78 which uses an adaptive dictionary and LZ77 which employs a sliding window. We will first describe the algorithm and then show why it is so good.

Assume you are given a sequence of symbols $x_1, x_2 \ldots$ to encode. The algorithm maintains a window of the W most recently encoded source symbols. The window size is fairly large $\simeq 2^{10} - 2^{17}$ and a power of 2. Complexity and performance increases with W.

a) Encode the first W letters without compression. If $|X| = M$, this will require $\lceil WlogM \rceil$ bits. This gets amortized over time over all symbols so we are not worried about this overhead.

b) Set the pointer P to W

c) Find the largest n such that

$$x_{P+1}^{P=n} = x_{P-u}^{P-u-1+n} \text{ for some u, } 0 \leq u \leq W-1$$

Set $n = 1$ if no match exists for $n \geq 1$

d) Encode n into a prefix free code word. The particular code here is called the unary binary code. n is encoded into the binary representation of n preceded by $\lfloor logn \rfloor$ zeros

$1 : \lfloor log1 \rfloor = 0 \quad 1$

$2 : \lfloor log2 \rfloor = 1 \quad 010$

$3 : \lfloor log3 \rfloor = 1 \quad 011$

$4 : \lfloor log4 \rfloor = 2 \quad 00100$

e) If $n > 1$ encode u using $\lceil logW \rceil$ bits. If $n = 1$ encode x_{p+1} using $\lceil logM \rceil$ bits.

f) Set $P = P + n$; update window and iterate.

Let R(N) be the expected number of bits required to code N symbols

Then $\lim_{W\to\infty} \lim_{N\to\infty} \frac{R(N)}{N} = H(X)$

"Baby LZ- algorithm"
Assume you have been given a data sequence of $W + N$ symbols where $W = 2^{n^*(H+2\epsilon)}$, where H is the entropy rate of the source. n^* divides $N|X| = M$ is a power of 2.

Compression Algorithm
If there is a match of n^* symbols, send the index in the window using $logW(= n^*(H+2\epsilon))$ bits. Else send the symbols uncompressed.

Note : No need to encode n, performance sub optimal compared to LZ more compression n needs only log n bits.

$Y_k = \#$ bits generated by the K^{th} segment

$\#\text{bits sent} = \sum_{k=1}^{N/n^*} Y_k$

$Y_k = logW$ if match

$= \quad n^* logM$ if no match

$E(\#\text{bits sent}) = \sum_{k=1}^{N/n^*} EY_k$

$= \quad \frac{N}{n^*}(P(match).logW + P(No\ match).n^* logM)$

$\frac{E(\#\text{bits sent})}{N} = P(match).\frac{logW}{n^*} + P(no\ match)logM$

claim $P(no\ match) \to 0$ as $n^* \to \infty$

$\lim_{n^* \to \infty} \frac{E(\#\text{bits sent})}{N} = \frac{logW}{n^*} = \frac{n^*(H+2\epsilon)}{n^*} = H + 2\epsilon$

Let S be the minimum number of backward shifts required to find a match for n^* symbols

Fact : $E(S|X_{P+1}, X_{P+2}, \ldots, X_{P+n^*}) = \frac{1}{P(X_{P+1,\ldots,X_{P+n^*}})}$

for a stationery ergodic source. This result is due to kac.

By Markov inequality

$$
\begin{aligned}
P(No\ match|X_{P+1}^{P+n^*}) &= P(S > W|X_{P+1}^{P+n^*}) \\
&= \frac{ES}{W} = \frac{1}{P(X_{P+1}^{P+n^*}).W} \\
P(No\ match) &= P(S > W) \\
&= \sum P(X_{P+1}^{P+n^*}) P(S > W|X_{P+1}^{P+n^*}) \\
&= \sum P(n^*) P(S > W|X^{n^*}) \\
&= \sum_{X^{n^*} \epsilon A_\epsilon^{n^*}} P(X^{n^*}) P(S > W|X^{n^*}) + \underbrace{\sum_{X^{n^*} \epsilon A_\epsilon^{n^* \subset}} P(X^{n^*})(S > W|X^{n^*})} \\
&\leq P(A_\epsilon^{n^* \subset}) \to 0 \text{ as } n^* \to \infty \\
X^{n^*} \epsilon\ A_\epsilon^{n^*} &\Rightarrow P(X^{n^*}) \geq 2^{-n^*(H+\epsilon)} \\
&= therefore\ \frac{1}{P(X^{n^*})} \leq 2^{n^*(H+\epsilon)} \\
&\leq \sum_{X^{n^*} \epsilon A_\epsilon^{n^*}} P(X^{n^*}).\frac{1}{P(X^{n^*}).W} \\
&\leq \frac{2^{n^*(H+\epsilon)}}{W} \sum_{X^{n^*} \epsilon A_\epsilon^{n^*}} P(X^{n^*}) = \frac{2^{n^*(H+\epsilon)}}{W}.P(A_\epsilon^{n^*}) \\
&\leq \frac{2^{n^*(H+\epsilon)}}{W} = 2^{n^*(H+\epsilon-H-2\epsilon)} = 2^{n^*(-\epsilon)} \to 0 \text{ as } n^* \to \infty
\end{aligned}
$$

Information Theory and Coding

Lecture 5

Channel Coding

Source coding deals with representing information as concisely as possible. Channel coding is concerned with the "reliable" "transfer" of information. The purpose of channel coding is to add redundancy in a controlled manner to "manage" error. One simple approach is that of repetition coding wherein you repeat the same symbol for a fixed (usually odd) number of time. This turns out to be very wasteful in terms of bandwidth and power. In this course we will study linear block codes. We shall see that sophisticated linear block codes can do considerably better than repetition. Good LBC have been devised using the powerful tools of modern algebra. This algebraic framework also aids the design of encoders and decoders. We will spend some time learning just enough algebra to get a somewhat deep appreciation of modern coding theory. In this introductory lecture, I want to produce a bird's eye view of channel coding.

Channel coding is employed in almost all communication and storage applications. Examples include phone modems, satellite communications, memory modules, magnetic disks and tapes, CDs, DVD's etc.

Digital Foundation : Tornado codes Reliable data transmission over the Internet
Reliable DSM VLSI circuits

There are tow modes of error control.

Error detection → Ethernet CRC
Error correction → CD

Errors can be of various types : Random or Bursty

There are two basic kinds of codes : Block codes and Trellis codes

This course : Linear Block codes

Elementary block coding concepts

Definition	:	An alphabet is a discrete set of symbols
Examples	:	Binary alphabet$\{0, 1\}$
		Ternary alphabet$\{0, 1, 2\}$
		Letters$\{a, \ldots, z\}$

Eventually these symbols will be mapped by the modulator into analog wave forms and transmitted. We will not worry about that part now.

In $a(n, k)$ block code, the incoming data source is divided into blocks of k symbols.

Each block of k symbols called a dataword is used to generate a block of n symbols called a codeword. $(n-k)$ redundant bits.

Example : $(3,1)$ binary repetition code
$0 \rightarrow 000 \qquad n=3, k=1$
$1 \rightarrow 111$

Definition : A block code G of blocklength n over an alphabet $\mathcal{X}$ is a non empty set of n-tuples of symbols from $\mathcal{X}$. These n-tuples are called codewords.

The rate of the code with M symbols is given by

$$R = \frac{1}{n} \, logq \; M$$

Let us assume $|\mathcal{X}| = q$. Codewords are generated by encoding messages of k symbols.

$\#$ messages $= q^k = |G|$

Rate of code $= \frac{k}{n}$

Example : Single Parity check code SPC code
Dataword : 010
Codeword : 0101
$k=3, n=4, \quad Rate = \frac{3}{4}$

This code can detect single errors.

Ex : All odd number of errors can be detected. All even number of errors go undetected
Ex : Suppose errors occur with prob P. What is the probability that error detection fails?

Hamming distance : The Hamming distance $d(x,y)$ between two q-ary sequences x and y is the number of places in which x and y differ.

Example:

$$\begin{aligned} x &= 10111 \\ y &= 01011 \\ d(x,y) &= 1+1+1=3 \end{aligned}$$

Intuitively, we want to choose a set of codewords for which the Hamming distance between each other is large as this will make it harder to confuse a corrupted codeword with some other codeword.

Hamming distance satisfies the conditions for a metric namely

1. $d(x, y) \geq 0$ with equality if $x = y$
2. $d(x, y) = d(y, x)$ symmetry
3. $d(x, y) \leq d(x, z) + d(z, y)$ (triangle inequality)

Minimum Hamming distance of a block code is the distance of the two closest codewords

$$\begin{aligned} d_{min} &= min\ d(c_i, c_j) \\ & \quad c_i, c_j\ \epsilon\ G \\ &= i \neq j \end{aligned}$$

An (n, k) block code with $d_{min} = d$ is often referred to as an (n, k, d) block code.

Some simple consequences of d_{min}

1 An (n, k, d) block code can always detect up to d-1 errors
Suppose codeword c was transmitted and r was received (The received word is often called a senseword.)

The error weight = # symbols changed / corrupted
$= d(c, r)$

If $d(c, r) < d$, then r cannot be a codeword. Otherwise c and r would be two codewords whose distance is less than the minimum distance.

Note :

(a) Error detection $\neq$ Error correction
$d(c_1, r) < d \qquad d(c_2, r) < d$

(b) This is the guaranteed error detecting ability. In practise, errors can be detected even if the error weight exceeds d. e.g. SPC detects all odd patterns of errors.

2 An (n, k, d) block code can correct up to
$t = \lfloor \frac{d-1}{2} \rfloor$ errors

Proof : Suppose we detect using nearest neighbor decoding i.e. given a senseword r, we choose the transmitted codeword to be

$$\begin{aligned} \hat{c} &= argnum\ d(r,c) \\ &= c\ \epsilon\ G \end{aligned}$$

A Hamming sphere of radius r centered at an n tuple c is the set of all n tuples, $c^{'}$ satisfying $d(c,c^{'}) \leq r$

$$t = \lfloor \frac{d_{min}-1}{2} \rfloor \Rightarrow d_{min} \geq 2t = 1$$

Therefore, Hamming spheres of radius t are non-intersecting. When $\leq t$ errors occur, the decoder can unambiguously decide which codeword was transmitted.

Singleton Bound : For an (n,k) block code $n-k \geq d_{min}-1$

Proof :
Remove the first $d-1$ symbols of each codeword in $\mathcal{C}$, Denote the set of modified codewords by $\hat{\mathcal{C}}$

Suppose $x\ \epsilon\ \mathcal{C}$, denote by $\hat{x}$ its image in $\hat{\mathcal{C}}$
Then $x \neq y \Rightarrow \hat{x} \neq \hat{y}$
Therefore If $\hat{x} = \hat{y}$, then $d(x,y) \leq d-1$

Therefore $q^k = |\mathcal{C}| = |\hat{\mathcal{C}}|$

$$\begin{aligned} \text{But } |\mathcal{C}| &\leq q^{n-d_{min}+1} \\ \Rightarrow q^k &\leq q^{n-d_{min}+1} \\ \Rightarrow k &\leq n-d_{min}+1 \\ \text{or } n-k &\geq d_{min}-1 \end{aligned}$$

\# possible block codes $= 2^{n.2^k}$

We want to find codes with good distance structure. Not the whole picture.

The tools of algebra have been used to discover many good codes. The primary algebraic structure of interest are Galois fields. This structure is exploited not only in discovering good codes but also in designing efficient encoders and decoders.

Information Theory and Coding

Lecture 6

Algebra

We will begin our discussion of algebraic coding theory by defining some important algebraic structures.

Group, Ring, Field and Vector space.

Group : A group is an algebraic structure $(G, *)$ consisting of a set G and a binary operator * satisfying the following four axioms.

1. Closure : $\forall a, b \, \epsilon \, G, a * b \, \epsilon \, G$
2. Associative law : $(a * b) * c = a * (b * c) \quad \forall a, b, c \, \epsilon \, G$
3. Identity : $\exists \, e \, \epsilon \, G$ such that $e * a = a * e = a \; \forall \; a \, \epsilon \, G$
4. Inverse : $\forall \; a \, \epsilon \, G, \exists \; b \, \epsilon \, G$ such that $b * a = a * b = e$

A group with a finite number of element is called a finite group. If $a * b = b * a \;\; \forall \; a, b \, \epsilon \, G$, then G is called a commutative or abelian group. For abelian groups * is usually denoted by + and called addition. The identity element is called 0.

Examples :

$$(\mathcal{Z}, +), (\mathcal{R} \backslash \{0\}, .\}, \quad (\mathcal{Z} \backslash n, +)$$

How about $(\mathcal{Z}, -)$. Ex: Prove $(\mathcal{Z}, -)$ is not a group.

An example of a non commutative group : Permutation Groups

Let $X = \{1, 2, \ldots, n\}$. A 1-1 map of X onto itself is called a permutation. The symmetric group S_n is made of the set of permutations of X.

eg : $n = 3 \quad S_n = \{123, 132, 213, 231, 312, 321\}$

132 denotes the permutation $1 \rightarrow 1, 2 \rightarrow 3, 3 \rightarrow 2$. The group operation is defined by the composition of permutations. $b * c$ is the permutation obtained by first applying c and then applying b.

For example :

$132 * 213 = 312$
$b \quad\quad c$
$213 * 132 = 231$ Non-commutative

A finite group can be represented by an operation table. e.g. $\mathcal{Z}/2 = \{0, 1\}$ $\quad (\mathcal{Z}/2, +)$

$+$	0	1
0	0	1
1	1	0

Elementary group properties

1. The identity element is unique
   Let $e_1 \& e_2$ be identity elements
   Then $e_1 = e_1 * e_2 = e_2$

2. Every element has a unique inverse
   b and b' are two inverses of a. Then
   $b = b * e = b * (a * b') = (b * a) * b' = e * b' = b'$

3. Cancellation
   $a * b = a * c \Rightarrow a = c \qquad b * a = c * a \Rightarrow b = c$
   $a^{-1} * a * b = a^{-1} * a * c$
   $\Rightarrow b = c \rightarrow$ No duplicate elements in any row or column of operation table

Exercise : Denote inverse of $x \, \epsilon \, G$ by x'
Show that $(a * b)' = b' * a'$

Definition : The order of a finite group is the number of elements in the group

Subgroups : A subgroup of G is a subset H of G that is itself a group under the operations of G

1) Closure : $a \, \epsilon \, H, b \, \epsilon \, H \;\; \Rightarrow \;\; a * b \, \epsilon \, H$

2) Associative : $(a * b) * c \;\; = \;\; a * (b * c)$

3) Identity : $\exists \, e' \, \epsilon \, H$ such that $a * e' \; = \; e' * a \; = \; a \qquad \forall \, a \, \epsilon \, H$
   Note that $e' = e,$ the identity of G $1 \; a * e' = axe$

4) Inverse : $\forall \, a \, \epsilon \, H, \;\; \exists \, b$ such that $\;\; a * b \; = \; b * a \; = \; e$

   Property (2) holds always because G is a group.

   Property (3) follows from (1) and (4) provided H is non-empty.

H non empty $\Rightarrow$ $\exists\, a \,\epsilon\, H$
$(4) \Rightarrow a^{-1} \,\epsilon\, H$
$(1) \Rightarrow a * a^{-1} = e \,\epsilon\, H$

Examples : $\{e\}$ is a subgroup, so is G itself. To check if a non-empty subset H is a subgroup we need only check for closure and inverse (Properties 1 and 4)

More compactly $a * b^{-1} \epsilon\, H \qquad \forall a, b \,\epsilon\, H$

For a finite group, enough to show closure.

Suppose G is finite and $h \,\epsilon\, G$ consider the set $H = \{h, h * h, h * h * h, \ldots\}$. We will denote this compactly as $\{h, h^2, h^3, \ldots\}$. Consists of all powers of h.

Let the inverse of h be h'

Then $(h^k)' = (h')^k$

Why? $h^2 * (h')^2$
$h * h * h' * h' = e$

Similarly $(h')^2 * h^2 = e$ Closure $\Rightarrow$ inverse exists
Since the set H is finite

$$\begin{aligned} h^i &= h^j \\ h^i (h')^i &= h^j (h')^j \\ h^{i-j} &= e \end{aligned}$$

$\exists\, n$ such that $h^n = e$

$H = \{h, h^2, \ldots, h^n\} \rightarrow$ cyclic group, subgroup generated by H

$$\begin{aligned} h^n &= e \\ h.h^{n-1} &= e \quad \text{In general,} (h^k)' = h^{n-k} \\ h' &= h^{n-1} \end{aligned}$$

Order of an element H is the order of the subgroup generated by H

Ex: Given a finite subset H of a group G which satisfies the closure property, prove that H is a subgroup.

Cosets : A left coset of a subgroup H is the set denoted by $g * H = \{g * H : h \epsilon H\}$.
Ex : $g * H$ is a subgroup if $g \epsilon H$

A right coset is $H * g = \{h * g : h \epsilon H\}$

Coset decomposition of a finite group G with respect to H is an array constructed as follows :

a) Write down the first row consisting of elements of H

b) Choose an element of G not in the first row. Call it g_2. The second row consists of the elements of the coset $g * H$

c) Continue as above, each time choosing an element of G which has not appeared in the previous rows. Stop when there is no unused element left. Because G is finite the process has to terminate.

$$\begin{array}{ccccc} h_1 = 1 & h_2 & \dots & h_n \\ g_2 & g_2 & g_2 * h_2 & \dots & g_2 * h_n \\ \vdots & & & & \\ g_m & g_m & g_m * h_2 & \dots & g_m * h_n \end{array}$$

$h_1, g_2, g_3, \dots, g_m$ are called coset leaders

Note that the coset decomposition is always rectangular. Every element of G occurs exactly once in this rectangular array.

Theorem : Every element of G appears once and only once in a coset decomposition of G.

First show that an element cannot appear twice in the same row and then show that an element cannot appear in two different rows.

Same row : $g_k h_1 = g_k h_2 \Rightarrow h_1 = h_2$ a contradiction

Different row : $g_k h_1 = g_l h_2$ where $k > l$
$= g_k = g_l h_2 h_1 \Rightarrow g_k \epsilon g_l * H$, a contradiction

$|G| = |H|$. (number of cosets of G with respect to H)

Lagrange's Theorem : The order of any subgroup of a finite group divides the order of the group.

Corr : Prime order groups have no proper subgroups
Corr : The order of an element divides the order of the group

Rings : A ring is an algebraic structure consisting of a set R and the binary operations, + and . satisfying the following axioms

1. $(R, +)$ is an abelian group
2. Closure : $a.b \, \epsilon \, R \quad \forall a, b \, \epsilon \, R$
3. Associative Law : $a.(b.c) = (a.b).c$
4. Distributive Law : $\begin{array}{rcl} (a+b).c & = & a.c + b.c \\ c.(a+b) & = & c.a + c.b \end{array}$
   Two Laws ; need not be commutative

0 is additive identity, 1 is multiplicative identity

Some simple consequences

1. $a \; 0 = 0 \; a = 0$
   $a.0 = a.(0+0) = a.0 + a.0$
   therefore $0 = a.0$
2. $a.(-b) = (-a).b = -(a.b)$
   $0 = a.0 = a(b-b) = a.b + a(-b)$
   therefore $a(-b) = -(a.b) \qquad 0.b = (a-a)b$
   $Similarly (-a)b = -(a.b) \qquad = ab + (-a).b$
   $. \rightarrow$ multiplication $\quad + \rightarrow$ addition $\; a.b = ab$

Examples $(\mathcal{Z}, +, .) \quad (\mathcal{R}, +, .) \quad (\mathcal{Z} \backslash n, +, .)$
$(\mathcal{R}_{nxn}, +, .)$ noncommutative ring

$\mathcal{R}[x]$: set of all polynomials with real coefficients under polynomial addition and multiplication

$$\mathcal{R}[x] = \{a_0 + a_1 x + \ldots + a_n x^n : n \geq 0, a_k \, \epsilon \, \mathcal{R}\}$$

Notions of commutative ring, ring with identity. A ring is commutative if multiplication is commutative.

Suppose there is an element $l \, \epsilon \, R$ such that $1.a = a.1 = a$
Then R is a ring with identity

Example $(2\mathcal{Z}, +, .)$ is a ring without identity

Theorem :
In a ring with identity

i) The identity is unique

ii) If an element a has an multiplicative inverse, then the inverse is unique.

Proof : Same as the proof for groups.
An element of R with an inverse is called a unit.

$(\mathcal{Z}, +, .)$	units	$\neq 1$
$(\mathcal{R}, +, .)$	units	$\mathcal{R} \backslash \{0\}$
$(\mathcal{R}_{nxn}, +, .)$	units	nonsingular or invertible matrices
$\mathcal{R}[x]$	units	polynomals of order 0 except the zero polynomal

If $ab = ac$ and $a \neq 0$ Then is $b = c$?

Zero divisors, cancellation Law, Integral domain

Consider $\mathcal{Z}/4. = \{0, 1, 2, 3\}$ suppose $a.b = ac$. Then is $b = c$? $a = b = 2$. A ring with no zero divisor is called when $a \neq 0$ an integral domain. Cancellation holds in an integral domain.

Fields :
A field is an algebraic structure consisting of a set F and the binary operators + and . satisfying

a) $(F, +)$ is an abelian group

b) $(F - \{0\}, .)$ is an abelian group

c) Distributive law : $a.(b + c) = ab + ac$

	addition	multiplication	substraction	division
Conventions	0	1	$a + (-b)$	$a \vert b$
	$-a$	a^{-1}	$a - b$	ab^{-1}

Examples : $(\mathcal{R}, +, .), (\mathcal{C}, +, .), (\mathcal{Q}, +, .)$

A finite field with q elements, if it exists is called a finite field or Galois filed and denoted by $GF(q)$. We will see later that q can only be a power of a prime number. A finite field can be described by its operation table.

$GF(2)$	$+$ 0 1	. 0 1
	0 0 1	0 0 0
	1 1 0	1 0 1
$GF(3)$	$+$ 0 1 2	. 0 1 2
	0 0 1 2	0 0 0 0
	1 1 2 0	1 0 1 2
	2 2 0 1	2 0 2 1
$GF(4)$	$+$ 0 1 2 3	. 0 1 2 3
	0 0 1 2 3	0 0 0 0 0
	1 1 0 3 2	1 0 1 2 3
	2 2 3 0 1	2 0 2 3 1
	3 3 2 1 0	3 0 3 1 2

multiplication is not modulo 4.

We will see later how finite fields are constructed and study their properties in detail. Cancellation law holds for multiplication.

Theorem : In any field

$$ab = ac \text{ and } a \neq 0$$

$$\Rightarrow b = c$$

Proof : multiply by a^{-1}
Introduce the notion of integral domain

$$\text{Zero divisors } <\Rightarrow \text{ Cancellation Law}$$

Vector Space :
Let F be a field. The elements of F are called scalars. A vector space over a field F is an algebraic structure consisting of a set V with a binary operator + on V and a scalar vector product satisfying.

1. $(V, +)$ is an abelian group
2. Unitary Law : $1.V = V$ for $\forall\ V\ \epsilon\ V$
3. Associative Law : $(C_i C_2).V = C_i(C_2 V)$
4. Distributive Law : $\begin{array}{rcl} C.(V_1 + V_2) & = & C.V_1 + C.V_2 \\ (C_1 + C_2).V & = & C_1.V + C_2.V \end{array}$

A linear block code is a vector subspace of $GF(q)^n$.

Suppose $\bar{V}_1, \ldots, \bar{V}_m$ are vectors in $GF(q)^n$.

The span of the vectors $\{\bar{V}_1, \ldots, \bar{V}_m\}$ is the set of all linear combinations of these vectors.

$$\begin{aligned} S &= \{a_1\bar{v}_1 + a_2\bar{v}_2 + \ldots + a_m\bar{v}_m : a_1, \ldots, a_m \,\epsilon\, GF(q)\} \\ &= LS(\bar{v}_1, \ldots, \bar{v}_m) \qquad LS \rightarrow \text{ Linear span} \end{aligned}$$

A set of vectors $\{\bar{v}_1, \ldots, \bar{v}_m\}$ is said to be linearly independent (LI) if

$$a_1\bar{v}_1 + \ldots + a_m\bar{v}_m = 0 \Rightarrow a_1 = a_2 = \ldots = a_m = 0$$

i.e. no vector in the set is a linear combination of the other vectors.

A basis for a vector space is a linearly independent set that spans the vector space.
What is a basis for $GF(q)^n$. $\qquad V = LS$ (Basis vectors)

$$\text{Take} \quad \begin{aligned} \bar{e}_1 &= (1, 0, \ldots, 0) \\ \bar{e}_2 &= (0, 1, \ldots, 0) \\ \bar{e}_n &= (0, 0, \ldots, 1) \end{aligned}$$

Then $\{\bar{e}_k : 1 \leq k \leq n\}$ is a basis

To prove this, need to show that e'_k are LI and span $GF(q)^n$.

$$\text{Span :} \quad \begin{aligned} \bar{v} &= (v_1, \ldots, v_n) \qquad \text{Independence : consider } e_1 \\ &= \sum_{k=1}^{n} v_k e_k \end{aligned}$$

$\{e_k\}$ is called the standard basis.

The dimension of a vector space is the number of vectors in its basis.
dimension of $GF(q)^n = n \qquad$ a vector space VC

Suppose $\{\bar{b}_1, \ldots, \bar{b}_m\}$ is a basis for $GF(q)^n$

Then any $\bar{v}\,\epsilon\, V$ can be written as

$$\begin{aligned} \bar{V} &= V_1\bar{b}_1 + V_2\bar{b}_2 + \ldots + V_m\bar{b}_m \qquad V_1, \ldots, V_m \,\epsilon\, GF(q) \\ &= (V_1 V_2 \ldots V_m) \begin{pmatrix} \bar{b}_1 \\ \bar{b}_2 \\ \bar{b}_m \end{pmatrix} \\ &= (V_1, V_2 \quad V_m)B \end{aligned}$$

$$= \quad \bar{a}.B \qquad \text{where } \bar{a} \,\epsilon\, (GF(q))^m$$

Is it possible to have two vectors $\bar{a}$ and $\bar{a}'$ such that $\bar{a}B = \bar{a}'B$

Theorem : Every vector can be expressed as a linear combination of basis vectors in exactly one way

Proof : Suppose not.
Then

$$\begin{aligned}
\bar{a}.B &= \bar{a}'.B \\
\Rightarrow \quad (\bar{a} - \bar{a}').B &= 0 \\
\Rightarrow \quad (\bar{a} - \bar{a}') \begin{pmatrix} \bar{b}_1 \\ \bar{b}_2 \\ \vdots \\ \bar{b}_m \end{pmatrix} &= 0 \\
(\bar{a}_1 = \bar{a}'_1)\bar{b}_1 + (a_2 - a'_2)\bar{b}_2 + \ldots (a_m - a'_m)\bar{b}_m &= 0
\end{aligned}$$

But $\bar{b}'_k$ are LI

$$\begin{aligned}
\Rightarrow a_k &= a'_k \qquad 1 \leq k \leq m \\
\Rightarrow \bar{a} &= \bar{a}'
\end{aligned}$$

Corollary : If $(b_1, \ldots, b_m)$ is a basis for V, then V consists of q^m vectors.
Corr : Every basis for V has exactly m vectors

Corollary : Every basis for $GF(q)^n$ has n vectors. True In general for any finite dimensional vector space Any set of K LI vectors forms a basis.

Review : Vector Space Basis
$\{b_1, \ldots, b_m\}$

$$v = \bar{a}B \qquad \bar{a} \,\epsilon\, GF(q)^m B = \begin{pmatrix} \bar{b}_1 \\ \bar{b}_2 \\ \bar{b}_m \end{pmatrix}$$

$|v| = q^m$

Subspace : A vector subspace of a vector space V is a subset W that is itself a vector space. All we need to check closed under vector addition and scalar multiplication.

The inner product of two n-tuples over $GF(q)$ is

$$\begin{aligned}
(a_1, \ldots, a_n).(b_1, \ldots, b_m) &= a_1 b_1 + \ldots + a_n b_n \\
&= \sum a_k b_k \\
&= \bar{a}.\bar{b}^\top
\end{aligned}$$

Two vectors are orthogonal if their inner product is zero

The orthogonal complement of a subspace W is the set $W^{\perp}$ of n-tuples in $GF(q)^n$ which are orthogonal to every vector in W.

$$V \epsilon W^{\perp} \text{ iff} \qquad v.w = 0 \qquad \forall\, w \epsilon W^{\perp}$$

$$\begin{array}{lllll} & W & = & \{00, 10, 20\} & GF(2)^2 \\ \text{Example : } GF(3)^2 : & W^{\perp} & = & \{00, 01, 02\} & W = \{00, 11\} \\ & & & & W^{\perp} = \{00, 11\} \end{array}$$

$$\begin{array}{lllll} dim\, W & = & 1 & 10 & \\ dim\, W^{\perp} & = & 1 & 01 & \end{array}$$

Lemma : $W^{\perp}$ is a subspace

Theorem : If $dimW = k,$ then $dimW^{\perp} = n - k$

Corollary : $W = (W^{\perp})^{\perp}$ Firstly $WC(W^{\perp})^{\perp}$

Proof : Let

$$\begin{array}{ll} dimW & = k \\ \Rightarrow dimW^{\perp} & = n - k \\ \Rightarrow dim(W^{\perp})^{\perp} & = k \\ dimW & = dim(W^{\perp})^{\perp} \end{array}$$

Let $\{g_1, \ldots, g_k\}$ be a basis for W and $\{h_1, \ldots, h_{n-k}\}$ be a basis for $w^{\perp}$

$$\text{Let } G = \underset{k \times n}{\begin{bmatrix} g_1 \\ \vdots \\ g_k \end{bmatrix}} \quad H = \underset{n-k \times n}{\begin{bmatrix} h_1 \\ \vdots \\ h_{n-k} \end{bmatrix}}$$

Then $GH^{\top} = O_{k \times n-k}$

$$Gh_1^{\top} = \begin{bmatrix} g_1 h_1^{\top} \\ \vdots \\ g_k h_1^{\top} \end{bmatrix} = O_{k \times 1}$$

$GH^{\top} = O_{k \times n-k}$

Theorem : A vector $V \epsilon W\ iff\ V\ H^{\top} = 0$

$vh_1^{\top} = 0 \qquad v \epsilon W$ and $h_1 \epsilon W^{\perp}$

$\Rightarrow V[h_1^\top \quad h_2^\top \qquad h_{n-k}^\top] = 0$

i.e. $VH^\top = 0$

$\Leftarrow$ Suppose $VH^\top = 0 \qquad \Rightarrow \qquad Vh_j^\top = 0 \quad 1 \leq j \leq n-k$

Then $V \; \epsilon \; (W^\perp)^\perp = W$

$WTS \; V \; \epsilon \; (W^\perp)^\perp$

i.e. $v.w = 0 \qquad \forall \; w \; \epsilon \; W^\perp$

But $w = \sum_{j=1}^{n-k} a_j h_j$

$v.w \;\; = \;\; vw^\top \;\; = \;\; v. \sum_{j=1}^{n-k} a_j h_j^\top \;\; = \;\; \sum_{j=1}^{n-k} a_j v_j . h_j^\top = 0$

We have two ways of looking at a vector V in W
$V \; \epsilon \; W \Rightarrow V = a \; G \qquad$ for some a
Also $VH^\top = 0$

How do you check that a vector w lies in W ?

Hard way : Find a vector $\bar{a} \; \epsilon \; GF(q)^k$ such that $v = aG-$

Easy way : Compute $VH^\top$. H can be easily determined from G.

Information Theory and Coding

Lecture 7

Linear Block Codes

A linear block code of blocklength n over a field $GF(q)$ is a vector subspace of $GF(q)^n$.

Suppose the dimension of this code is k. Recall that rate of a code with M codewords is given by

$$R = \frac{1}{n}\, log_q\, M$$

Here $M = q^k \Rightarrow R = \frac{k}{n}$

Example : Repetition, SPC,

Consequences of linearity
The hamming weight of a codeword, $w(c)$, is the number of nonzero components of c . $w(c) = d_H(o, c)$

Er: $w(0110) = 2, \quad w(3401) = 3$

The minimum hamming weight of a block code is the weight of the nonzero codeword with smallest weight w_{min}

Theorem : For a linear block code, minimum weight = minimum distance

Proof : $(V, +)$ is a group
$w_{min} = min_c\, w(c) = d_{min} = min_{c_i \neq c_j}\, d(c_i \supset c_j)$

$w_{min} \geq d_min \qquad d_{min} \geq w_{min}$

Let c_o be the codeword of minimum weight. Since o is a codeword

$w_{min} = w(c_o) = d(o, c_o) \geq min_{c_i \neq c_j}\, d(c_i \supset c_j)$

$= d_{min}$

$d_{min} \geq w_{min}$

Suppose C_1 and C_2 are the closest codewords. Then $C_1 - C_2$ is a codeword.

$$\begin{aligned}\text{Therefore } d_{min} &= d(c_1, c_2) &&= d(o, c_1 - c_2)\\ &&&= w(c_1 - c_2)\\ &&&= min_c\, w(c)\\ &&&= w_{min}\\ \text{Therefore } d_{min} &\geq w_{min}\end{aligned}$$

Key fact : For LBCs, weight structure is identical to distance structure.

Matrix description of LBC

A LBC $\mathcal{C}$ has dimension k

$\Rightarrow \exists$ basis set with k vectors or n-tuples. Call these $g_o, \ldots, g_{k-1}$. Then any $C\, \epsilon\, \mathcal{C}$ can be written as

$$\begin{aligned}C &= \alpha_0 g_0 + \alpha_1 g_1 + \ldots + \alpha_{k-1}\, g_{k-1}\\ &= [\alpha_o \alpha_1 \ldots \alpha_{k-1}] \begin{bmatrix} g_o \\ g_1 \\ \vdots \\ g_{k-1} \end{bmatrix}\\ \text{i.e. } C &= \bar{\alpha} G \qquad \alpha\, \epsilon\, GF(q)^k \quad G\text{is called the generator matrix}\end{aligned}$$

This suggests a natural encoding approach. Associate a data vector α with the codeword αG. Note that encoding then reduces to matrix multiplication. All the trouble lies in decoding.

The dual code of $\mathcal{C}$ is the orthogonal complement $\mathcal{C}^{\perp}$

$\mathcal{C}^{\perp} = \{h : \quad ch^{\top} = o\ \forall\ c\, \epsilon\, \mathcal{C}\}$

Let $h_o, \ldots, h_{n-k-1}$ be the basis vectors for $C^{\perp}$ and H be the generator matrix for $C^{\perp}$. H is called the parity check matrix for $\mathcal{C}$

Example : $\mathcal{C} = (3, 2)$ parity check code

$$\mathcal{C} = \begin{matrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{matrix} \quad G = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} \qquad k = 2, n - k = 1$$

$$\mathcal{C}^{\perp} = \begin{matrix} 0 & 0 & 0 \\ 1 & 1 & 1 \end{matrix} \quad H = [1\ 1\ 1]$$

H is the generator matrix for the repetition code i.e. SPC and RC are dual codes.

Fact : C belongs to $\mathcal{C}$ iff $CH^{\top} = 0$

Let $\mathcal{C}$ be a linear block code and $\mathcal{C}^{\perp}$ be its dual code. Any basis set of $\mathcal{C}$ can be used to form G. Note that G is not unique. Similarly with H.

Note that $\bar{C}H^{\top} = 0 \quad \forall \bar{C}\, \epsilon\, \mathcal{C}$ in particular true for all rows of G

Therefore $GH^{\top} = 0$

Conversely suppose $GH^{\top} = 0$, then H is a parity check matrix if the rows of H form a LI basis set.

	$\mathcal{C}$	$\mathcal{C}^{\perp}$
Generator matrix	G	H
Parity check matrix	H	G
	$\bar{C}\, \epsilon\, \mathcal{C}\ iff\ CH^{\top} = 0$	$\bar{V}\, \epsilon\, \mathcal{C}^{\perp}\ iff\ VG^{\top} = 0$

Equivalent Codes : Suppose you are given a code $\mathcal{C}$. You can form a new code by choosing any two components and transposing the symbols in these two components for every codeword. What you get is a linear block code which has the same minimum distance. Codes related in this manner are called equivalent codes.

Suppose G is a generator matrix for a code $\mathcal{C}$. Then the matrix obtained by linearly combining the rows of G is also a generator matrix.

$b_1 \qquad G = \begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_{k-1} \end{bmatrix}$

elementary row operations -

Interchange any tow rows

Multiplication of any row by a nonzero element in $GF(q)$

Replacement of any row by the sum of that row and a multiple of any other rows.

Fact : Using elementary row operations and column permutation, it is possible to reduce G to the following form

$$G = [I_{k \times k} \quad P]$$

This is called the systematic form of the generator matrix. Every LBC is equivalent to a code has a generator matrix in systematic form.

Advantages of systematic G

$$\begin{aligned} C &= a.G \\ &= (a_0 \dots a_{k-1})[I_{k \times k} \quad P] \qquad k \times n-k \\ &= (a_0 \dots a_{k-1}, C_k, \dots, C_{k-1}) \end{aligned}$$

Check matrix
$H = [-P^\top I_{n-k \times n-k}]$

1) $GH^\top = 0$

2) The row of H form a LI set of $n-k$ vectors.

Example

$$G = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix} \qquad P = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}$$

$$P^\top = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

$$-P^\top = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

$$H = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 \end{bmatrix} \qquad n-k \geq d_{min} - 1$$

Singleton bound (revisited)

$$d_{min} = min_c \; w(c) \leq 1 + n - k$$

Codes which meet the bound are called maximum distance codes or maximum-distance separable codes.

Now state relationship between columns of H and d_{min}

Let $\mathcal{C}$ be a linear block code (LBC) and $\mathcal{C}^\perp$ be the corresponding dual code. Let G be the generator matrix for $\mathcal{C}$ and H be the generator matrix for $\mathcal{C}^\perp$. Then H is the parity check matrix for $\mathcal{C}$ and G is the parity check matrix for H.

$\bar{C}.H^\top = 0 \quad \Leftrightarrow \quad \bar{C} \; \epsilon \; (\mathcal{C}^\perp)^\perp = \mathcal{C}$
$\bar{V}.G^\top = 0 \quad \Leftrightarrow \quad \bar{V} \; \epsilon \; \mathcal{C}^\perp$

Note that the generator matrix for a LBC $\mathcal{C}$ is not unique.

Suppose

$$G = \begin{bmatrix} \bar{g}_0 \\ \bar{g}_1 \\ \vdots \\ \bar{g}_{k-1} \end{bmatrix} \quad \text{Then} \quad \begin{aligned} \mathcal{C} &= LS(\bar{g}_0, \ldots, \bar{g}_{k-1}) \\ &= LS(G) \end{aligned}$$

Consider the following transformations of G

a) Interchange two rows $\mathcal{C}' = LS(G') = LS(G) = \mathcal{C}$

b) Multiply any row of G by a non-zero element of $GF(q)$.

$$G' = \begin{bmatrix} \alpha\bar{g}_0 \\ \vdots \\ \bar{g}_{k-1} \end{bmatrix} \quad \begin{aligned} LS(G') &= ? \\ &= \mathcal{C} \end{aligned}$$

c) Replace any row by the sum of that row and a multiple of any other row.

$$G' = \begin{bmatrix} \bar{g}_0 + \alpha\bar{g}_1 \\ \bar{g}_1 \\ \vdots \\ \bar{g}_{k-1} \end{bmatrix} \quad LS(G') = \mathcal{C}$$

Easy to see that $GH^\top = O_{k\times n-k} \quad HG^\top = O_{n-k\times k}$

Suppose G is a generator matrix and H is some $n-k\times n$ matrix such that $GH^\top = O_{k\times n-k}$. Is H a parity check matrix.

The above operations are called elementary row operations.

Fact 1 : A LB code remains unchanged if the generator matrix is subjected to elementary row operations. Suppose you are given a code C. You can form a new code by choosing any two components and transposing the symbols in these two components. This gives a new code which is only trivially different. The parameters (n, k, d) remain unchanged. The new code is also a LBC. Suppose $G = [f_0, f_1, \ldots f_{n-1}]$ Then $G' = [f_1, f_0, \ldots f_{n-1}]$. Permutation of the components of the code corresponds to permutations of the columns of G.

Defn : Two block codes are equivalent if they are the same except for a permutation of the codeword components (with generator matrices G & G') G' can be obtained from G

Fact 2 : Two LBC's are equivalent if using elementary row operations and column permutations.

Fact 3 : Every generator matrix G can be reduced by elementary row operations and column operations to the following form :

$$G = [I_{k\times k} \;\; P_{n-k\times k}]$$

Also known as row-echelon form

Proof : Gaussian elimination
Proceed row by row and then interchange rows and columns.

A generator matrix in the above form is said to be systematic and the corresponding LBC is called a systematic code.

Theorem : Every linear block code is equivalent to a systematic code.

Proof : Combine Fact3 and Fact2

There are several advantages to using a systematic generator matrix.

1) The first k symbols of the codeword is the dataword.
2) Only $n-k$ check symbols needs to be computed $\Rightarrow$ reduces decoder complexity.
3) If $G = [I\ P]$, then $H = [-P^{\top}_{n-k\times k} \;\; I_{n-k\times n-k}]$

$$NTS: \qquad GH^{\top} = 0 \qquad GH^{\top} = [IP]\begin{bmatrix} -P \\ I \end{bmatrix} = -P + P = 0$$

Rows of H are LI

Now let us study the distance structure of LBC

The Hamming weight, $w(\bar{c})$ of a codeword $\bar{c}$, is the number of non-zero components of $\bar{c}$. $w(\bar{c}) = d_H(o, \bar{c})$

The minimum Hamming weight of a block code is the weight of the non-zero codeword with smallest weight.

$$w_{min} = min_{\bar{c}\,\epsilon\,\mathcal{C}} \;\; w(\bar{c})$$

Theorem : For a linear block code, minimum weight = minimum distance

Proof : Use the fact that $(\mathcal{C}, +)$ is a group

$$w_{min} = min_{\bar{c}\,\epsilon\,\mathcal{C}}\; w(\bar{c}) \qquad d_{min} = min_{(c_i \neq c_j)_{c_i, c_j\,\epsilon\,\mathcal{C}}}\; d(\bar{c}_i, \bar{c}_j)$$

$$w_{min} \geq d_{min} \qquad d_{min} \geq w_{min}$$

Let $\bar{c}_o$ be the minimum weight codeword
$O\,\epsilon\,\mathcal{C}$

$$w_{min} = w(c_o) = d(o, \bar{c}_o) \geq min_{(c_i \neq c_j)_{c_i, c_j\,\epsilon\,\mathcal{C}}}$$

$$\Rightarrow w_{min} \geq d_{min}$$

Suppose $\bar{c}_1$ and $\bar{c}_2$ are the two closest codewords
Then $\bar{c}_1 - \bar{c}_2\,\epsilon\,\mathcal{C}$

$$\begin{array}{rcccl} \text{therefore } d_{min} & = & d(\bar{c}_1, \bar{c}_2) & = & d(o, \bar{c}_1, \bar{c}_2) \\ & & & = & w(\bar{c}_1, \bar{c}_2) \\ & & & \geq & min_{\bar{c}\,\epsilon\,\mathcal{C}}\; w(\bar{c}) \quad = \quad w_{min} \end{array}$$

Key fact : For LBC's, the weight structure is identical to the distance structure.

Given a generator matrix G, or equivalently a parity check matrix H, what is d_{min}.

Brute force approach : Generate $\mathcal{C}$ and find the minimum weight vector.

Theorem : (Revisited) The minimum distance of any linear (n, k) block code satisfies

$$d_{min} \leq 1 + n - k$$

Proof : For any LBC, consider its equivalent systematic generator matrix. Let $\bar{c}$ be the codeword corresponding to the data word $(1\,0\ldots0)$
Then $w_{min} \leq w(\bar{c}) \leq 1+n-k$
$\Rightarrow d_{min} \leq 1+n-k$

Codes which meet this bound are called maximum distance seperable codes. Examples include binary SPC and RC. The best known non-binary MDS codes are the Reed-Solomon codes over $GF(q)$. The RS parameters are

$$(n,k,d) = (q-1, q+d, d+1) \qquad q = 256 = 2^8$$

Gahleo Mission $(255, 223, 33)$

A codeword $\bar{c}\ \epsilon\ \mathcal{C}$ iff $\bar{c}H^\top = 0$. Let $H = [\bar{f}_0, \bar{f}_1 \ldots \bar{f}_{n-1}]$ where $\bar{f}_k$ is a $n-k \times 1$ column vector.

$\bar{c}H^\top = 0 \Rightarrow \sum_{i=0}^{n-1} c_i f_i = 0$ when $f_k^\top$ is a $1 \times n-k$ vector corresponding to a column of H.

therefore each codeword corresponds to a linear dependence among the columns of H. A codeword with weight w implies some w columns are linearly dependent. Similarly a codeword of weight at most w exists, if some w columns are linearly dependent.

Theorem : The minimum weight $(= d_{min})$ of a LBC is the smallest number of linearly dependent columns of a parity check matrix.

Proof : Find the smallest number of LI columns of H. Let w be the smallest number of linearly dependent columns of H. Then $\sum_{k=0}^{w-1} a_{n_k}\ \bar{f}_{n_k} = 0$. None of the a_{n_k} are o. (violate minimality).

Consider the codeword $\begin{matrix} C_{n_k} & = & a_{n_k} \\ C_1 & = & 0 \end{matrix}$ otherwise
Clearly $\bar{C}$ is a codeword with weight w.

Examples
Consider the code used for ISBN (International Standardized Book Numbers). Each book has a 10 digit identifying code called its ISBN. The elements of this code are from $GF(11)$ and denoted by $0, 1, \ldots, 9, X$. The first 9 digits are always in the range 0 to 9. The last digital is the parity check bit.

The parity check matrix is

$H = [1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10]$

$GF(11)$ is isomorphic to Z/11 under addition and multiplication modulo11

$d_{min} = 2$

$\Rightarrow$ can detect one error

Ex : Can also detect a transposition error i.e. two codeword positions are interchanged.

Blahut : $\begin{matrix}[0521553741] \\ 12345678910\end{matrix} \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{bmatrix} = 65$

Hamming Codes

Two binary vectors are independent iff they are distinct and non zero. Consider a binary parity check matrix with m rows. If all the columns are distinct and non-zero, then $d_{min} \geq 3$. How many columns are possible? $2^m - 1$. This allows us to obtain a binary $(2^m - 1, 2^m - 1 - m, 3)$ Hamming code. Note that adding two columns gives us another column, so $d_{min} \leq 3$.

Example : $m = 3$ gives us the $(7, 4)$ Hamming code

$$H = \begin{bmatrix} 0111 & 100 \\ 1101 & 010 \\ \underbrace{1011} & \underbrace{001} \end{bmatrix} \quad G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$-P^\top \qquad I_{3\times 3}$

$G = [I \quad P] \Rightarrow$

Hamming codes can correct single errors and detect double errors used in SIMM and DIMM.

Hamming codes can be easily defined over larger fields.

Any two distinct & non-zero m-tuples over $GF(q)$ need not be LI. e.g. $\bar{a} = 2.\bar{b}$

Question : How many m-tuples exists such that any two of them are LI.

$\frac{q^m-1}{q-1}$ Defines a $\left(\frac{q^m-1}{q-1}, \frac{q^m-1}{q-1} - m, 3\right)$ Hamming code over GF(q)

Consider all nonzero m-tuples or columns that have a 1 in the topmost non zero component. Two such columns which are distinct have to be LI.

Example : $(13, 10)$ Hamming code over $GF(3)$

$$H = \begin{bmatrix} 1 & 1 & 1 & 111 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 112 & 2 & 2 & 1 & 1 & 0 & 1 & 0 \\ 1 & 2 & 0 & 120 & 1 & 2 & 1 & 2 & 0 & 0 & 1 \end{bmatrix}$$

$3^3 = 27 - 1 = \frac{26}{2} = 13$

$d_{min} of C^{\perp}$ is always $\geq k$

Suppose a codeword $\bar{c}$ is sent through a channel and received as senseword $\bar{r}$. The errorword $\bar{e}$ is defined as

$$\bar{e} = \bar{r} - \bar{c}$$

The decoding problem : Given $\bar{r}$, which codeword $\bar{c} \; \epsilon \; \mathcal{C}$ maximizes the likelihood of receiving the senseword $\bar{r}$?

Equivalently, find the most likely error pattern $\hat{e}$. $\hat{c} = \bar{r} - \bar{e}$ Two steps.

For a binary symmetric channel, most likely, means the smallest number of bit errors. For a received senseword $\bar{r}$, the decoder picks an error pattern $\bar{e}$ of smallest weight such that $\bar{r} - \bar{e}$ is a codeword. Given an (n, k) binary code, $P(w(\bar{e}) = j) = (N_j)P^j(1 - P)^{N-j}$ function of j.

This is the same as nearest neighbour decoding. One way to do this would be to write a lookup table. Associate every $\bar{r} \; \epsilon \; GF(q)^n$, to a codeword $\hat{c} \; \epsilon \; GF(q)$, which is $\bar{r}$'s nearest neighbour. A systematic way of constructing this table leads to the (Slepian) standard array.

Note that $\mathcal{C}$ is a subgroup of $GF(q)^n$. The standard array of the code $\mathcal{C}$ is the coset decomposition of $GF(q)^n$ with respect to the subgroup $\mathcal{C}$. We denote the coset $\{\bar{g} + \bar{c} : \forall \; \epsilon \; \mathcal{C}\}$ by $\bar{g} + \mathcal{C}$. Note that each row of the standard array is a coset and that the cosets completely partition $GF(q)^n$.

$$\text{The number of cosets } = \frac{|GF(q)^n|}{\mathcal{C}} = \frac{q^n}{q^k} = q^{n-k}$$

Let $o, \bar{c}_2, \ldots, \bar{c}_{q^k}$ be the codewords. Then the standard array is given by

$$\begin{array}{llll} o & \bar{c}_2 & \bar{c}_3 & \ldots \quad \bar{c}_{q^k} \\ \bar{e}_2 & \bar{c}_2 + \bar{e}_2 & \bar{c}_3 + \bar{e}_2 & \ldots \quad \bar{c}_{q^k} + \bar{e}_2 \\ \bar{e}_3 & \vdots & & \\ \vdots & & & \\ \bar{e}_{q^{n-k}} & \bar{e}_{q^{n-k}} + \bar{c}_2 & & \ldots \quad \bar{c}_{q^k} + \bar{e}_{q^{n-k}} \end{array}$$

1. The first row is the code $\mathcal{C}$ with the zero vector in the first column.
2. Choose as coset leader among the unused n-tuples, one which has least Hamming weight, "closest to all zero vector".

Decoding : Find the senseword $\bar{r}$ in the standard array and denote it as the codeword at the top of the column that contains $\bar{r}$.

Claim : The above decoding procedure is nearest neighbour decoding.

Proof : Suppose not.
We can write $\bar{r} = \bar{c}_{ijk} + \bar{e}_j$. Let $\bar{c}_1$ be the nearest neighbor.
Then we can write $\bar{r} = \bar{c}_i + \bar{e}_i$ such that $w(\bar{e}_i) < w(e_j)$

$$\begin{array}{llll} \Rightarrow & \bar{c}_j + \bar{e}_j & = & \bar{c}_i + \bar{e}_i \\ \text{i.e.} & \bar{e}_i & = & \bar{e}_j + \bar{c}_j - \bar{c}_i \quad \text{But } \bar{c}_j - \bar{c}_i \in \mathcal{C} \\ \Rightarrow & \bar{e}_i & \in & \bar{e}_j + \mathcal{C} \text{ and } w(\bar{e}_i) < w(\bar{e}_j), \text{ a contradiction.} \end{array}$$

Geometrically, the first column consists of Hamming spheres around the all zero codeword. The k^{th} column consists of Hamming spheres around the k^{th} codeword.

Suppose $d_{min} = 2t + 1$. Then Hamming spheres of radius t are non-intersecting.

In the standard array, draw a horizontal line below the last row such that $w(e_k) \leq t$. Any senseword above this codeword has a unique nearest neighbour codeword. Below this line, a senseword will have more than one nearest neighbour codeword.

A Bounded-distance decoder corrects all errors up to weight t. If the senseword falls below the Lakshman Rekha, it declares a decoding failure. A complete decoder assigns every received senseword to a nearby codeword. It never declares a decoding failure.

Syndrome detection
For any senseword $\bar{r}$, the syndrome is defined by $\bar{S} = \bar{r}H^\top$.

Theorem : All vectors in the same coset have the same syndrome. Two distinct cosets have distinct syndromes.

Proof : Suppose $\bar{r}$ and $\bar{r}'$ are in the same coset
Then $\bar{r} = \bar{c} + \bar{e}$. Let $\bar{e}$ be the coset leader and $\bar{r}' = \bar{c}' + \bar{e}$

therefore	$S(\bar{r})$	$=$	$\bar{r}H^\top$	$=$ $\bar{e}H^\top$
and	$S(\bar{r}')$	$=$	$\bar{r}'H^\top$	$=$ $\bar{e}H^\top$

Suppose two distinct cosets have the same syndrome. Then Let $\bar{e}$ and $e^\top$ be the corresponding coset leaders.

$$\bar{e}H^\top = \bar{e}'H^\top$$
$$\Rightarrow \quad \bar{e} - \bar{e}' \ \epsilon \ \mathcal{C}$$
therefore $\bar{e} = \bar{e}' + \bar{c} \Rightarrow \bar{e} \ \epsilon \ \bar{e}' + \mathcal{C}$ a contradiction

This means we only need to tabulate syndromes and coset leaders.

Suppose you receive $\bar{r}$. Compute syndrome $S = rH^\top$. Look up table to find coset leader $\bar{e}$

Decide $\hat{c} = \bar{r} - \bar{e}$

Example : $(1,3)RC$

Hamming Codes :

Basic idea : Construct a Parity Check matrix with as many columns as possible such that no two columns are linearly dependent.

Binary Case : just need to make sure that all columns are nonzero and distinct.

Non-binary Case : $\bar{V}_1 \neq 0$
Pick a vector $\bar{V}_1 \ \epsilon \ GF(q^m)$, The set of vectors LD with $\bar{V}_1$ are $\{\bar{O}, \bar{V}_1, 2\bar{V}_1, \ldots, (q-1)\bar{V}_1\} \triangleq H_1$

Pick $\bar{V}_2 \ \epsilon ? H_1$ and form the set of vectors LD with $\bar{V}_2\{\bar{O}, \bar{V}_2, 2\bar{V}_2, \ldots, (q-1)V_2\}$

Continue this process till all the m-tuples are used up. Two vectors in disjoint sets are L.I. Incidentally $\{H_n, +\}$ is a group.

$\#columns = \frac{q^m-1}{q-1}$

Two non-zero distinct m tuples that have a 1 as the topmost or first non-zero component are LI Why?

$\#mtuples = q^{m-1} + q^{m-2} + \ldots + 1 = \frac{q^m - 1}{q-1}$

Example : $m = 2, q = 3 \quad n = \frac{3^2-1}{3-1} = 4 \quad k = n - m = 2 \qquad (4, 2, 3)$

Suppose a codeword $\bar{c}$ is sent through a channel and received as senseword $\bar{r}$. The error vector or error pattern is defined as

$$\bar{e} = \bar{r} - \bar{c}$$

The Decoding Problem : Given $\bar{r}$, which codeword $\hat{c}\,\epsilon\,\mathcal{C}$ maximizes the likelihood of receiving the senseword $\bar{r}$? Equivalently, find the most likely valid errorword $\hat{e}$, $\hat{c} = \bar{r} - \hat{e}$.

For a binary symmetric channel, with $Pe < 0.5$ "most likely" error pattern is the error pattern with least number of 1's, i.e. the pattern with the smallest number of bit errors. For a received senseword $\bar{r}$, the decoder picks an error pattern $\hat{e}$ of smallest weight such that $\bar{r} - \hat{e}$ is a codeword.

This is the same as nearest neighbour decoding. One way to do this would be to write a look-up table. Associate every $\bar{r}\,\epsilon\,GF(q)^n$ to a codeword $\hat{c}(\bar{r})\,\epsilon\,\mathcal{C}$, which is $\bar{r}$'s nearest neighbour in $\mathcal{C}$.

There is an element of arbitrariness in this procedure because some $\bar{r}$ may have more than one nearest neighbour. A systematic way of constructing this table leads us to the (slepian) standard array.

We begin by noting that $\mathcal{C}$ is a subgroup of $GF(q)^n$. For any $\bar{g}\,\epsilon\,GF(q)^n$, the coset associated with $\bar{g}$ is given by the set $\bar{g} + \mathcal{C} = \{\bar{g} + \bar{c} : \bar{c}\,\epsilon\,\mathcal{C}\}$

Recall :

1) The cosets are disjoint completely partition $GF(q)^n$

2) # cosets $= \frac{|GF(q)^n|}{|\mathcal{C}|} = \frac{q^n}{q^k} = q^{n-k}$

The standard array of the code $\mathcal{C}$ is the coset decomposition of $GF(q)^n$ with respect to the subgroup $\mathcal{C}$.

Let $\bar{o}, \bar{c}_2, \ldots, \bar{c}_{q^k}$ be the codewords. Then the standard array is constructed as follows :

a) The first row is the code $\mathcal{C}$ with the zero vector in the first column. $\bar{o}$ is the coset leader.

b) Among the vectors not in the first row, choose an element or vector $\bar{e}_2$, which has least Hamming weight. The second row is the coset $\bar{e}_2 + \mathcal{C}$ with $\bar{e}_2$ as the coset leader.

c) Continue as above, each time choosing an unused vector $g \, \epsilon \, GF(q)^n$ of minimum weight.

Decoding : Find the senseword $\bar{r}$ in the standard array and decode it as the codeword at the top of the column that contains $\bar{r}$.

Claim : Standard array decoding is nearest neighbor decoding.

Proof : Suppose not. Let $\bar{c}$ be the codeword obtained using standard array decoding and let $\bar{c}'$ be any of the nearest neighbors. We have $\bar{r} = \bar{c} + \bar{e} = \bar{c}' + \bar{e}'$ where $\bar{e}$ is the coset leader for $\bar{r}$ *and* $w(\bar{e}) > w(\bar{e}')$
$\Rightarrow \bar{e}' = \bar{e} + \bar{c} - \bar{c}'$
$\Rightarrow \bar{e}' \, \epsilon \, \bar{e} + \mathcal{C}$
By construction, $\bar{e}$ has minimum weight in $\bar{e} + \mathcal{C}$
$\Rightarrow w(\bar{e}) \leq w(\bar{e}')$, a contradiction

What is the geometric significance of standard array decoding. Consider a code with $d_{min} = 2t + 1$. Then Hamming spheres of radius t, drawn around each codeword are non-intersecting. In the standard array consider the first n rows. The first n vectors in the first column constitute the Hamming sphere of radius 1 drawn around the $\bar{o}$ vector. Similarly the first n vectors of the k^{th} column correspond to the Hamming sphere of radius 1 around $\bar{c}_k$. The first $n + (n_2)$ vectors in the k^{th} column correspond to the Hamming sphere of radius 2 around the k^{th} codeword. The first $n + (n_2) + \ldots (n_t)$ vectors in the k^{th} column are elements of the Hamming sphere of radius t around $\bar{c}_k$. Draw a horizontal line in the standard array below these rows. Any senseword above this line has a unique nearest neighbor in $\mathcal{C}$. Below the line, a senseword may have more than one nearest neighbor.

A Bounded Distance decoder converts all errors up to weight t, i.e. it decodes all sensewords lying above the horizontal line. If a senseword lies below the line in the standard array, it declares a decoding failure. A complex decoder simply implements standard array decoding for all sensewords. It never declares a decoding failure.

Example : Let us construct a $(5, 2)$ binary code with $d_{min} = 3$

$$H = \underbrace{\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix}}_{-P^\top \quad I} \qquad G = \underbrace{\begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \end{bmatrix}}_{I : P}$$

Codewords $\mathcal{C} = \{00000, 01101, 10110, 11011\}$

00000	01101	10110	11011
00001	01100	10111	11010
00010	01111	10100	11001
00100	01001	10010	11111
01000	00101	11110	10011
10000	11101	00110	01011
00011	01110	10101	11000
01010	00011	11100	11011
		00011	$= 11011 + 11000$
			$= 00000 + 00011$

Syndrome detection :
For any senseword $\bar{r}$, the syndrome is defined by $\bar{s} = \bar{r}H^\top$

Theorem : All vectors in the same coset (row in the standard array) have the same syndrome. Two different cosets/rows have distinct syndromes.

Proof : Suppose $\bar{r}$ and $\bar{r}'$ are in the same coset.
Let $\bar{e}$ be the coset leader.
Then $\bar{r} = \bar{c} + \bar{e}$ and $\bar{r}' = \bar{c}' + \bar{e}$
$\bar{r}H^\top = \bar{c}H^\top + \bar{e}H^\top = \bar{e}H^\top = \bar{c}'H^\top + \bar{e}H^\top = \bar{r}'H^\top$

Suppose two distinct cosets have the same syndrome.
Let $\bar{e}$ and $\bar{e}'$ be the coset leaders
Then $\bar{e}H^\top = \bar{e}'H^\top$
$\Rightarrow (\bar{e} - \bar{e}')H^\top = 0 \;\Rightarrow\; \bar{e} - \bar{e}' \,\epsilon\, \mathcal{C} \;\Rightarrow\; \bar{e} \,\epsilon\, \bar{e}' + \mathcal{C}$, a contradiction.

This means we only need to tabulate syndromes and coset leaders. The syndrome decoding procedure is as follows :

1) Compute $S = \bar{r}H^\top$
2) Look up corresponding coset leader $\bar{e}$
3) Decode $\hat{c} = \bar{r} - \bar{e}$

q^{n-k} $RS(255, 223, 33)$, $q^{n-k} = (256)^{32} = 2$
$2^{8\times 32} = 2^{256} > 10^{64}$ more than the number of atoms on earth?

Why can't decoding be linear. Suppose we use the following scheme : Given syndrome S, we calculate $\hat{e} = S.B$ where B is a $n - k \times n$ matrix.

Let $E = \{\hat{e} : \hat{e} = SB \text{ for some } S \, \epsilon \, GF(q)^{n-k}\}$

Claim : E is a vector subspace of $GF(q)^n$
$|E| \leq q^{n-k} \Rightarrow dimE \leq n - k$
Let $E_1 = \{$single errors where the non zero component is 1 which can be detected$\}$
$E_1 \subset E$

We note that no more $n - k$ single errors can be detected because E_1 constitutes a LI set. In general not more than $(n - k)(q - 1)$ single errors can be detected. Example $(7, 4)$ Hamming code can correct all single errors (7) in contrast to $7 - 4 = 3$ errors with linear decoding.

Need to understand Galois Field to devise good codes and develop efficient decoding procedures.