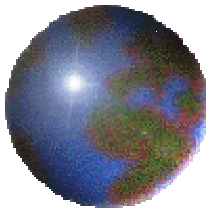
Syllabus for the Course Information Theory and Coding

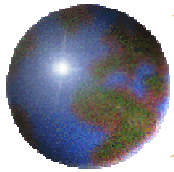
- Review of probability theory
- Entropy
- Mutual information
- Data compression
- Huffman coding
- Asymptotic equipartition property
- Universal source coding
- Channel capacity
- Differential entropy
- Block codes and Convolutional codes.



Information Theory and Coding

Pavan S. Nuggehalli

CEDT, IISc, Bangalore

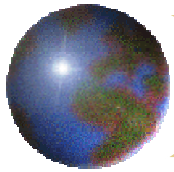


Course Outline - I

Information theory is concerned with the fundamental limits of communication.

- What is the ultimate limit to data compression? e.g. how many bits are required to represent a music source.

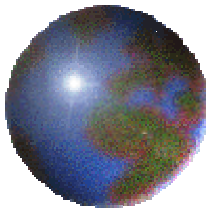
- What is the ultimate limit of reliable communication over a noisy channel, e.g. how many bits can be sent in one second over a telephone line.



Course Outline - II

Coding theory is concerned with practical techniques to realize the limits specified by information theory

- Source coding converts source output to bits.
  - Source output can be voice, video, text, sensor output …
- Channel coding adds extra bits to data transmitted over the channel
  - This redundancy helps combat the errors introduced in transmitted bits due to channel noise

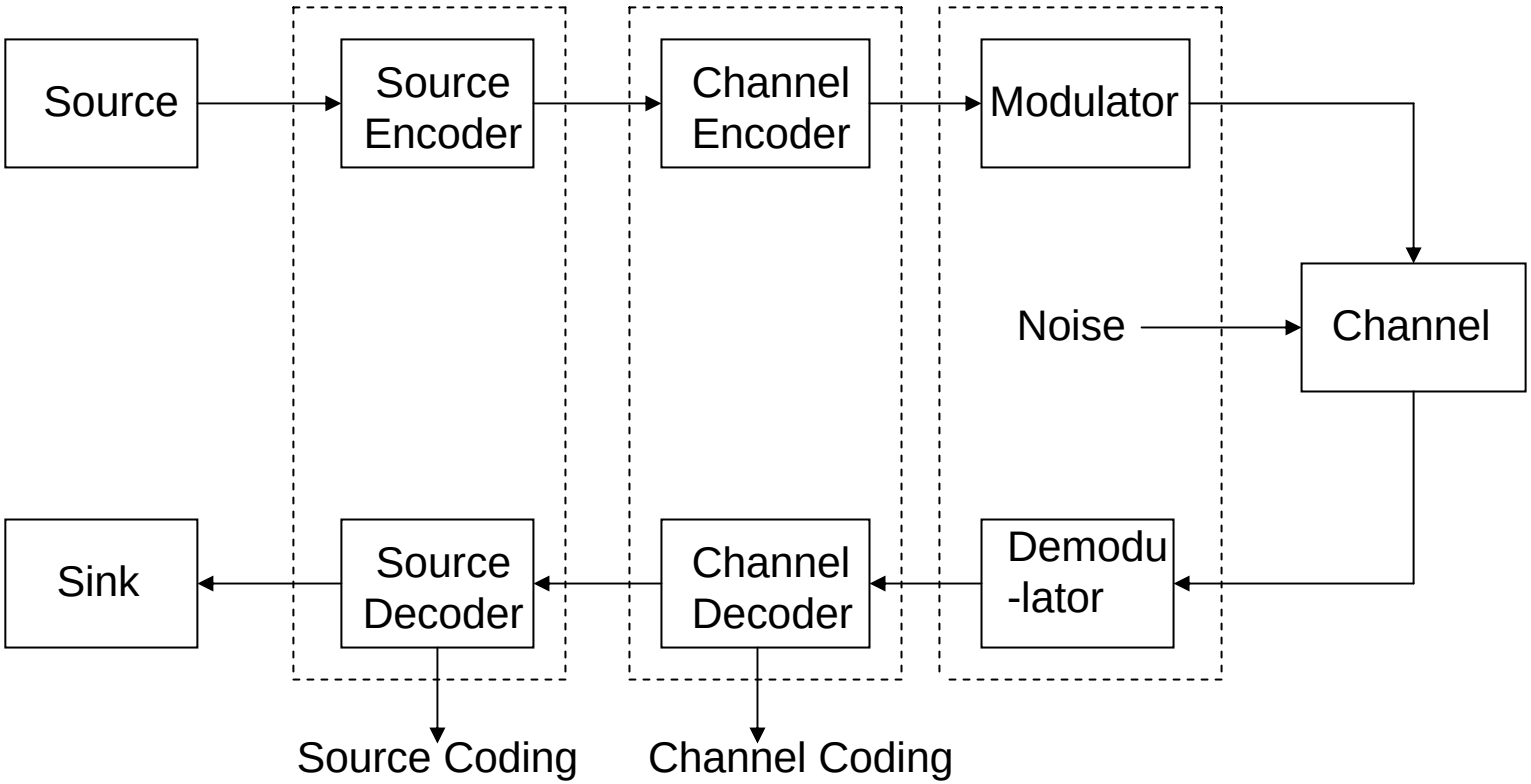


Information Theory and Coding

Pavan S. Nuggehalli

CEDT, IISc, Bangalore

Communication System Block Diagram

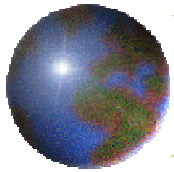


Modulation converts bits into coding analog waveforms suitable for transmission over physical channels. We will not discuss modulation in any detail in this course.

What is Information?

- Sources can generate “information” in several formats
  - sequence of symbols such as letters from the English or Kannada alphabet, binary symbols from a computer file.
  - analog waveforms such as voice and video signals.
- Key insight : Source output is a random process

  * This fact was not appreciated before Claude Shannon developed information theory in 1948



Randomness

- Why should source output be modeled as random?
- Suppose not x. Then source output will be a known determinative process.
  x simply reproduces this process at the risk without bothering to communicate?
- The number of bits required to describe source output depends on the probability distribution of the source, not the actual values of possible outputs.

Information Theory and Coding

Lecture 1

Pavan Nuggehalli *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

Pavan Nuggehalli *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

Pavan Nuggehalli *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

Pavan Nuggehalli *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

Pavan Nuggehalli *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 $\rightarrow$ Ethernet CRC
Error correction $\rightarrow$ 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

Pavan Nuggehalli *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 : $\forall a, b \in G, a * b \in G$
2. Associative law : $(a * b) * c = a * (b * c) \quad \forall a, b, c \in G$
3. Identity : $\exists\ e \in G$ such that $e * a = a * e = a\ \forall\ a \in G$
4. Inverse : $\forall\ a \in G, \exists\ b \in G$ such that $b * a = a * b = e$

A group with a finite number of element is called a finite group. If $a * b = b * a \ \ \forall\ a, b \in G$, 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 :

$132 \quad * \quad 213 \quad = \quad 312$
$b \qquad\quad c$
$213 \quad * \quad 132 \quad = \quad 231$ 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

Pavan Nuggehalli *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

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

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

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

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

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

$G = [I \quad P]$

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.

	$\bar{e}H^\top = \bar{e}'H^\top$
$\Rightarrow$	$\bar{e} - \bar{e}' \ \epsilon \ \mathcal{C}$
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.

Information Theory and Coding

Lecture 8

Pavan Nuggehalli *Finite Fields*

Review : We have constructed two kinds of finite fields. Based on the ring of integers and the ring of polynomials.

$(GF(q) - \{0\}, .)$ is cyclic $\Rightarrow \exists\, \alpha \in GF(q)$ such that
$GF(q) - \{0\} = \{\alpha, \alpha^2, \ldots, \alpha^{q-1} = 1\}$. There are $Q(q-1)$ such primitive elements.

Fact 1 : Every finite field is isomorphic to a finite field $GF(p^m), P$ prime constructed using a prime polynomial $f(x) \in GF(p)[x]$

Fact 2 : There exist prime polynomials of degree m over $GF(P)$, p prime, for all values of p and m. Normal addition, multiplication, smallest non-trivial subfield.

Let $GF(q)$ be an arbitrary field with q elements. Then $GF(q)$ must contain the additive identity 0 and the multiplicative identity 1. By closure $GF(q)$ contains the sequence of elements $0, 1, 1+1, 1+1+1, \ldots$ and so on. We denote these elements by $0, 1, 2, 3, \ldots$ and call them the integers of the field. Since q is finite, the sequence must eventually repeat itself. Let p be the first element which is a repeat of an earlier element, r. i.e. $p = r\ m\ GF(q)$.

But this implies $p - r = 0$, so if $r \neq 0$, then these must have been an earlier repeat at $p - r = 0$. We can then conclude that $p = 0$. The set of field integers is given by $G = \{0, 1, 2, \ldots, P-1\}$.

Claim : Addition and multiplication in G is modulo p.

Addition is modulo p because G is a cyclic group under addition. Multiplication is modulo p because of distributive law.

$$a.b \;=\; \underbrace{(1+1+\ldots+1)}_{\text{a times}}.b \;=\; b+b+\ldots+b = a \times b (mod\ p)$$

Claim : $(G, +, .)$ is a field
$(G, +)$ is an abelian group, $(G - \{0\}, .)$ is an abelian group, distributive law holds.

To show $(G - \{0\}, .)$ is a group, NTS closure & inverse.

This is just the field of integers modulo p. We can then conclude that p is prime. Then we have

Theorem : Each finite field contains a unique smallest subfield which has a prime

number of elements.

The number of elements of this unique smallest subfield is called the characteristic of the field.

Corr. If q is prime, the characteristic of $GF(q)$ is q. In other words $GF(q) = \mathcal{Z}/q$. Suppose not G is a subgroup of $GF(q) \Rightarrow p/q \Rightarrow p = q$.

Defn : Let $GF(q)$ be a field with characteristic p. Let $f(x)$ be a polynomial over $GF(p)$. Let $\alpha \in GF(q)$. Then $f(\alpha), \alpha \in GF(q)$ is an element of $GF(q)$. The monic polynomial of smallest degree over $GF(p)$ with $f(\alpha) = 0$ is called the minimal polynomial of α over $GF(p)$.

Theorem :

1) Every element $\alpha \in GF(q)$ has a unique minimal polynomial.

2) The minimal polynomial is prime

Proof : Pick $\alpha \in GF(q)$

Evaluate the zero, degree.0, degree.1, degree.2 monu polynomials with $x = \alpha$, until a repetition occurs. Suppose the first repetition occurs at $f(x) = h(x)$ where $deg\ f(x) > deg\ h(x)$. Otherwise $f(x) - h(x)$ has degree $< deg\ f(x)$ and evaluates to 0 for $x = \alpha$. Then $f(x) - h(x)$ is the minimal polynomial for $\alpha \in GF(q)$.

Any other lower degree polynomial cannot evaluate to 0. Otherwise a repetition would have occurred before reaching $f(x)$.

Uniqueness : $g(x)$ and $g'(x)$ are two monu polynomials of lowest degree such that $g(\alpha) = g'(\alpha) = 0$. Then $h(x) = g(x) - g'(x)$ has degree lower than $g(x)$ and evaluates to 0, a contradiction.

Primatily : Suppose $g(x) = p_1(x)\ .\ p_2(x)\ .\ p_N(x)$. Then $g(\alpha) = 0 \Rightarrow p_k(\alpha) = 0$. But $p_k(x)$ has lower degree than $g(x)$, a contradiction.

Theorem : Let α be a primitive element in a finite field $GF(q)$ with characteristic p. Let m be the degree of the minimal polynomial of α over $GF(p)$. Then $q = p^m$ and every element $\beta \in GF(q)$ can be written as

$$\beta = a_{m-1}\ \alpha^{m-1} + a_{m-2}\ \alpha^{m-2} + \ldots + a_1\ \alpha + a_0$$

where $a_{m-1}, \ldots, a_0 \in GF(p)$

Proof : Pick some combination of $a_{m-1}, a_{m-2}, \ldots, a_0 \in GF(p)$

Then $\beta = a_{m-1}\ \alpha^{m-1} + \ldots + a_1\ \alpha + a_0 \in GF(q)$

Two different combinations cannot give rise to the same field element. Otherwise we would have
$\beta = a_{m-1}\ \alpha^{m-1} + \ldots + a_0 = b_{m-1}\ \alpha^{m-1} + \ldots + b_0$
$\Rightarrow (a_{m-1} - b_{m-1})\alpha^{m-1} + \ldots + (a_0 - b_0) = 0$
$\Rightarrow \alpha$ is a zero of a polynomial of degree $m-1$, contradicting the fact that the minimal polynomial of α has degree m.

There are p^m such combinations, so $q \geq p^m$. Pick any $\beta \in GF(q) - \{0\}$. Let $f(\alpha)$ be the *deg* m, minimal polynomial of α. Then $\beta = \alpha^l \quad 1 \leq l \leq q-1$

Then $\beta = \alpha^l = Q(\alpha).f(\alpha) + r(\alpha), deg\ r(\alpha) \leq m-1$ division algorithm
i.e. $\beta = r(\alpha)$
$\Rightarrow$ every element β can be expressed as a linear combination of $\alpha^{m-1}, \alpha^{m-2}, \ldots, \alpha^0$
Therefore $q \leq p^m$. Hence proved.

Corr : Every finite field is isomorphic to a field $GF(p)/p(x)$

Proof : By theorem, every element of $GF(q)$ can be associated with a polynomial of *deg* $m-1$ replacing α with the indeterminate x. These polynomials can be thought of as field elements. They are added and multiplied modulo $f(x)$, the minimal polynomial of α. This field is then isomorphic to the field $GF(p)[x]/f(x)$.

Theorem : A finite field exists of size p^m for all primes p and $m \geq 1$

Proof : Handwaving
Need to show that there exists a prime polynomial over $GF(p)$ for every degree m.
The scene of Eratosthenes

Information Theory and Coding

Lecture 9

Pavan Nuggehalli *Cyclic Codes*

Cyclic codes are a kind of linear block codes with a special structure. A LBC is called cyclic if every cyclic shift of a codeword is a codeword.

Some advantages

- amenable to easy encoding using shift register
- decoding involves solving polynomial equations not based on look-up tables
- good burst correction and error correction capabilities. Ex-CRC are cyclic
- almost all commonly used block codes are cyclic

Definition : A linear block code $\mathcal{C}$ is cyclic if

$$(c_0\ c_1 \ldots c_{n-1}) \in \mathcal{C} \Rightarrow (c_{n-1}\ c_0\ c_1 \ldots c_{n-2}) \in \mathcal{C}$$

Ex : equivalent def. with left shifts

It is convenient to identify a codeword $c_0\ c_1\ \ldots\ c_{n-1}$ in a cyclic code $\mathcal{C}$ with the polynomial $c(x) = c_0 + c_1 x + \ldots + c_{n-1}\ x^{n-1}$. If $c_i \in GF(q)$ Note that we can think of $\mathcal{C}$ as a subset of $GF(q)[x]$. Each polynomial in $\mathcal{C}$ has degree $m \leq n-1$, therefore $\mathcal{C}$ can also be thought of as a subset of $GF(q)|x^n - 1$, the ring of polynomials modulo $x^n - 1$ $\mathcal{C}$ will be thought of as the set of n tuples as well as the corresponding codeword polynomials.

In this ring a cyclic shift can be written as a multiplication with x in the ring.

Suppose $c = (c_0 \ldots c_{n-1})$
Then $c(x) = c_0 + c_1 x + \ldots c_{n-1}\ x^{n-1}$
Then $x\ c(x) = c_0 x + c_1 x^2 + \ldots c_{n-1}\ x^n$
and $x\ c(x)\ mod\ x^n - 1 = c_0 x + \ldots + c_{n-1}$
which corresponds to the code $(c_{n-1}\ c_0\ \ldots\ c_{n-2})$
Thus a linear code is cyclic iff

$$c(x) \in \mathcal{C} \Rightarrow x\ c(x)\ mod\ x^n - 1 \in \mathcal{C}$$

A linear block code is called cyclic if

$$(c_0\ c_1\ \ldots\ c_{n-1}) \in \mathcal{C} \Rightarrow (c_{n-1}\ c_0\ c_1\ \ldots\ c_{n-2}) \in \mathcal{C}$$

Equivalently, in terms of codeword polynomials,

$$c(x) \in \mathcal{C} \Rightarrow x\ c(x)\ mod\ x^n - 1 \in \mathcal{C}$$

Theorem : A set of codeword polynomials $\mathcal{C}$ is a cyclic code iff

1) $\mathcal{C}$ is a subgroup under addition
2) $c(x) \in \mathcal{C} \Rightarrow a(x)\ c(x)\ mod\ x^n - 1 \in \mathcal{C} \quad \forall a(x) \in GF(q)[x]$

Theorem : Let $\mathcal{C}$ be an (n, k) cyclic code. Then

1) There exists a unique monu polynomial $g(x) \in \mathcal{C}$ of smallest degree among all non zero polynomials in $\mathcal{C}$
2) $c(x) \in \mathcal{C} \Rightarrow c(x) = a(x)\ g(x)$
3) $deg.\ g(x) = n - k$
4) $g(x) \mid x^n - 1$

Let $h(x) = \frac{x^n-1}{g(x)}$. $h(x)$ is called the check polynomial. We have

Theorem : $c(x) \in \mathcal{C} \Leftrightarrow c(x)\ h(x) = 0\ mod\ x^n - 1$

Theorem : The generator and parity check matrices for a cyclic code with generator polynomial $g(x)$ and check polynomial $h(x)$ are given by

$$G = \begin{bmatrix} g_0 & g_1 & \dots & g_{n-k} & 0 & 0 \dots & 0 \\ 0 & g_0 & g_1 \dots & g_{n-k-1} & g_{n-k} & 0 \dots & 0 \\ 0 & 0 & g_0 & & & g_{n-k} \dots & 0 \\ \dots & \vdots & \ddots & & & & \\ 0 & 0 & & g_0 & g_1 \dots & & g_{n-k} \end{bmatrix}$$

$$H = \begin{bmatrix} h_k & h_{k-1} & h_{k-2} & \dots & \dots & h_0 & 0 & 0 & 0 & 0 \\ 0 & h_k & h_{k-1} & \dots & \dots & & h_0 & 0 & 0 & 0 \\ \vdots & 0 & h_k & \ddots & & & & & & \\ & \vdots & 0 & & \ddots & & & & & \\ 0 & 0 & 0 & & & h_k & h_{k-1} & \dots & & h_0 \end{bmatrix}$$

Proof : Note that $g_0 \neq 0$ Otherwise, we can write

$$\begin{array}{lcll} g(x) = xg'(x) & \Rightarrow & x^n g(x) & = & xg'(x) \bmod x^n - 1 \\ & \Rightarrow & g'(x) & = & x^{n-1} g(x) \bmod x^n - 1 \\ & \Rightarrow & g'(x) & \in & \mathcal{C} \text{ a contradiction} \end{array}$$

Each row in G corresponds to a codeword $(g(x), xg(x), \ldots, x^{k-1}g(x))$. These codewords are LI. For H to be the parity check matrix, we need to show that $GH^T = 0$ and that the $n-k$ rows of H are LI. $deg\ h(x) = k \Rightarrow h_k \neq 0$. Therefore, each row is LI. Need to show $GH^T = 0_{k \times n-k}$

We know $g(x)\ h(x) = x^n - 1 \Rightarrow$ coefficients of $x^1, x^2, \ldots, x^{n-1}$ are 0
i.e. $u_l = \sum_{k=0}^{l} g_k\ h_{l-k} = 0 \qquad 1 \leq l \leq n-1$

It is easy to see by inspection that

$$GH^T = \begin{bmatrix} u_k & u_{k+1} & \ldots & u_{n-1} \\ u_{k-1} & u_k & & u_{n-2} \\ \vdots & & & \\ u_1 & u_2 & & u_{n-k} \end{bmatrix} = O_{k \times n-k}$$

These matrices can be reduced to systematic form by elementary row operations.

Encoding and decoding : polynomial multiplication
Let $a(x)$ be the data polynomial, degree $\leq k-1$

$$c(x) = a(x)\ g(x)$$

Decoding :
Let $v(x) = c(x) + e(x)$ be the received senseword. $e(x)$ is the errorword polynomial

Definition : Syndrome polynomial $s(x)$ is given by $s(x) = v(x)\ mod\ g(x)$

We have $$\begin{array}{lcl} s(x) & = & [c(x) + e(x)] \bmod g(x) \\ & = & e(x) \bmod g(x) \end{array}$$

Syndrome decoding : Find the $e(x)$ with the least number of nonzero coefficients satisfying

$$s(x) = e(x) \bmod g(x)$$

Syndrome decoding can be implemented using a look up table. There are q^{n-k} val-

ues of $s(x)$, store corresponding $e(x)$

Theorem : Syndrome decoding is nearest neighbour decoding.

Proof : Let $e(x)$ be the error vector obtained using syndrome detection. Syndrome detection differs from nearest neighbour decoding if there exists an error polynomial $e'(x)$ with weight strictly less than $e(x)$ such that

$$\begin{array}{rrcl} & v(x) - e'(x) & = & c'(x) \in \mathcal{C} \\ \text{But} & v(x) - e(x) & = & c(x) \in \mathcal{C} \\ \Rightarrow & e'(x) & - & e(x) \in \mathcal{C} \\ \Rightarrow & e'(x) & = & e(x) \; mod \; g(x) \\ But & s(x) & = & e(x) \; mod \; g(x) \\ \Rightarrow & s(x) & = & e'(x) \; mod \; g(x), \text{ a contradiction} \end{array}$$

By definition $e(x)$ is the smallest weight error polynomial for which $s(x) = e(x) \; mod \; g(x)$

Let us construct a binary cyclic code which can correct two errors and can be decoded using algebraic means Let $n = 2^m - 1$. Suppose α is a primitive element of $GF(2^m)$. Define

$$\mathcal{C} = \{c(x) \in GF(2)/x^n - 1 : c(\alpha) = 0 \text{ and } c(\alpha^3) = 0 \; in \; GF(2^m) = GF(n+1)\}$$

Note that $\mathcal{C}$ is cyclic : $\mathcal{C}$ is a group under addition and $c(x) \in \mathcal{C} \Rightarrow a(x) \; c(x) \; mod \; x^n - 1 = 0$

$$a(\alpha) \; c(\alpha) = 0 \text{ and } \alpha(\alpha^3) \; c(\alpha^3) = 0 \Rightarrow a(x) \; c(x) \; mod \; x^n - 1 \in \mathcal{C}$$

We have the senseword $v(x) = c(x) + e(x)$

Suppose at most 2 errors occurs. Then $e(x) = 0$ or $e(x) = x^i$ or $e(x) = x^i + x^j$

Define $X_1 = \alpha^i$ and $X_2 = \alpha^j$. X_1 and X_2 are called error location numbers and are unique because α has order n. So if we can find $X_1 \& X_2$, we know i and j and the senseword is properly decoded.

Let $S_1 = V(\alpha) = \alpha^i + \alpha^j = X_1 + X_2$
and $S_2 = V(\alpha^3) = \alpha^{3i} + \alpha^{3j} = X_1^3 + X_2^3$

We are given S_1 and S_2 and we have to find X_1 and X_2. Under the assumption that at

most 2 errors occur, $S_1 = 0$ iff no errors occur.

If the above equations can be solved uniquely for X_1 and X_2, the two errors can be corrected. To solve these equations, consider the polynomial

$$\begin{array}{lcll}
(x - X_1)(x - X_2) & = & x^2 + (X_1 + X_2)x + (X_1X_2) & \\
X_1 + X_2 & = & S_1 & \\
(X_1 + X_2)^3 & = & (X_1 + X_1)^2(X_1 + X_2) & = (X_1^2 + X_2^2)(X_1 + X_2) \\
 & & & = X_1^3 + X_1^2X_2 + X_2^2X_1 + X_2^3 \\
 & & & = S_3 + X_1X_2(S_1) \\
 & & \Rightarrow X_1X_2 & = \frac{S_1^3+S_3}{S_1} \\
\text{therefore } (x - X_1)(x - X_2) & = & x^2 + S_1x + \frac{S_1^3+S_3}{S_1} &
\end{array}$$

We can construct the RHS polynomial. By the unique factorization theorem, the zeros of this quadratic polynomial are unique.

One easy way to find the zeros is to evaluate the polynomial for all 2^m values in $GF(2^m)$

Solution to Midterm II

1. Trivial
2. 51, not abelian $(a - b)^{-1} = ab = b^{-1}a^{-1} = ba$
3. Trivial
4. $1, 3, 7, 9, 21, 63$,
   $\alpha \in GF(p) \quad \alpha^{p-1} = 1 \Rightarrow \alpha^p = \alpha$
   Look at the polynomial $x^p - x$. This has only p zeros given by elements of $GF(p)$. If $\alpha \in GF(p^m)$ and $\beta^p = \beta$ and $\beta \notin GF(p)$, then $x^p - x$ will have $p + 1$ roots, a contradiction.

Procedure for decoding

1. Compute syndromes $S_1 = V(\alpha)$ and $S_2 = V(\alpha^3)$
2. If $S_1 = 0$ and $S_2 = 0$, assume no error
3. Construct the polynomial $x^2 + s_1x + \frac{s_1^3+s_3}{s_1}$
4. Find the roots X_1 and X_2. If either of them is 0, assume a single error. Else, let $X_1 = \alpha^i$ and $X_2 = \alpha^j$. Then errors occur in locations i & j

Many cyclic codes are characterized by zeros of all codewords.

$$\mathcal{C} = \{c(x) : c(\beta_1) = 0, c(\beta_2) = 0, \ldots, c(\beta_l) = 0\}$$

$$c_0, \ldots, c_{n-1} \in GF(q) \text{ and } \beta_1, \ldots, \beta_l \in GF(Q) \supset GF(q) \Rightarrow Q = q^m$$

Note that the above definition imposes constraint on values of q and n.

$$c(\beta) = 0 \; \forall c \in \mathcal{C}(x - \beta) \mid c(x) \Rightarrow x - \beta \mid g(x)$$

$$\Rightarrow x - \beta \mid x^n - 1 \Rightarrow \beta^n = 0 \Rightarrow n \mid Q - 1 \Rightarrow n | q^m - 1$$

because $GF(Q)$ is an extension field of $GF(q)$

Lemma : Suppose n and q are co-prime. Then there exists some number m such that $n \mid q^m - 1$

Proof :

$$\begin{array}{lcl} q & = & Q_1 n + S_1 \\ q^2 & = & Q_2 n + S_2 \\ \vdots & & \\ q^{n+1} & = & Q_{n+1} n + S_n + 1 \end{array}$$

All the remainders lie between 0 and $n-1$. Because we have $n+1$ remainders, at least two of them must be the same, say S_i and S_j.

Then we have

$$\begin{array}{lcl} q^j - q^i & = & Q_j n + S_j - Q_i n - S_i \\ & = & (Q_j - Q_i) n \\ \text{or } q^j(q^{j-1} - 1) & = & (Q_j - Q_i) n \\ n \not| q^i & \Rightarrow & n | q^{j-i} - 1 \end{array}$$

Put $m = j - 1$, and we have $n | q^m - 1$ for some m

Corr : Suppose n and q are co-prime and $n | q^m - 1$. Let $\mathcal{C}$ be a cyclic code over $GF(q)$ of length n.

Then $g(x) = \Pi_{i=1}^{l}(x - \beta_l)$ where $\beta_l \in GF(q^m)$
$n | q^m - 1 \Rightarrow x^n - 1 | x^{q^m - 1} - 1$

Proof : $z^k - 1 = (z-1)(z^{k-1} + z^{k-2} + \ldots + 1)$
Let $q^m - 1 = n.r.$
Put $z = x^n$ and $k = r$
Then $x^{nr} - 1 = x^{q^m-1} - 1 = (x^n - 1)(x^{n(k-1)} + \ldots + 1)$
$\Rightarrow x^n - 1 | x^{q^m-1} \Rightarrow g(x) | x^{q^m-1}$

But $x^{q^m-1} - 1 = \Pi_{i=1}^{q^m-1}(x - \alpha_i), \alpha_i \in GF(q^m), \alpha_i \neq 0$
Therefore $g(x) = \Pi_{i=1}^{l}(x - \beta_i)$

Note : A cyclic code of length n over $GF(q)$, such that $n \neq q^m - 1$ is uniquely specified by the zeros of $g(x)$ in $GF(q^m)$.

$$\mathcal{C} = \{c(x) : c(\beta_i) = 0, 1 \leq i \leq l\}$$

Defn. of BCH Code : Suppose n and q are relatively prime and $n | q^m - 1$. Let α be a primitive element of $GF(q^m)$. Let $\beta = \alpha^{q^m-1/n}$. Then β has order n in $GF(q^m)$.

An (n, k) BCH code of design distance d is a cyclic code of length n given by

$$\mathcal{C} = \{c(x) : c(\beta) = c(\beta^2) \ldots = c(\beta^{d-1}) = 0\}$$

Note : Often we have $n = q^m - 1$. In this case, the resulting BCH code is called primitive BCH code.

Theorem : The minimum distance of an (n, k) BCH code of design distance d is at least d.

Proof :

$$\text{Let } H = \begin{bmatrix} 1 & \beta & \beta^2 & \ldots & \beta^{n-1} \\ 1 & \beta^2 & \beta^4 & \ldots & \beta^{2(n-1)} \\ \vdots & & & & \\ 1 & \beta^{d-1} & \beta^{2(d-1)} & \ldots & \beta^{(d-1)(n-1)} \end{bmatrix}$$

$$c(\beta^1) = c \begin{bmatrix} 1 \\ \beta^i \\ \beta^2 i \\ \beta^{(n-1)i} \end{bmatrix}$$

Therefore $C \in \mathcal{C} \Leftrightarrow CH^T = 0$ and $C \in GF(q)[x]$

Digression :
Theorem : A matrix A has an inverse iff $detA \neq 0$

Corr : $CA = 0$ and $C \neq 0_{1\times n} \Rightarrow detA = 0$

Proof : Suppose $detA \neq 0$. Then A^{-1} exists
$\Rightarrow CAA^{-1} = 0 \Rightarrow C = 0$, a contradiction.

Lemma : $\begin{array}{lcll} det(A) & = & det(A^T) & \\ det(KA) & = & KdetA & K scalar \end{array}$

Theorem : Any square matrix of the form
$A = \begin{bmatrix} 1 & 1 & \dots & 1 \\ X_1 & X_2 & & X_d \\ \vdots & \vdots & & \vdots \\ X_1^{d-1} & X_2^{d-1} & & X_d^{d-1} \end{bmatrix}$ has a non-zero determinant iff all the X_1 are distinct vandermonde matrix.

Proof : See pp 44, Section 2.6
End of digression

In order to show that weight of c is at least d, let us proceed by contradiction. Assume there exists a nonzero codeword c, with weight $w(c) = w < d$ i.e. $c_i \neq 0$ only for $i \in \{n_1, \dots, n_w\}$

$CH^T = 0$

$$\Rightarrow (c_0 \dots c_{n-1}) \begin{bmatrix} 1 & 1 & \dots & 1 \\ \beta & \beta^2 & & \beta^{d-1} \\ \beta^2 & \beta^4 & & \beta^{2(d-1)} \\ \vdots & \vdots & & \vdots \\ \beta^{n-1} & \beta^{2(n-1)} & & \beta^{(d-1)(n-1)} \end{bmatrix} = 0$$

$$\Rightarrow (c_{n_1} \dots c_{n_w}) \begin{bmatrix} \beta^{n_1} & \beta^{2n_1} & \dots & \beta^{(d-1)n_1} \\ \beta^{n_2} & \beta^{2n_2} & & \beta_{(d-1)n_2} \\ \vdots & \vdots & & \vdots \\ \beta^{n_w} & \beta^{2n_w} & & \beta^{(d-1)n_w} \end{bmatrix} = 0$$

$$\Rightarrow (c_{n_1} \dots c_{n_w}) \begin{bmatrix} \beta^{n_1} & \dots & \beta^{wn_1} \\ \beta^{n_2} & & \beta^{wn_2} \\ \vdots & & \vdots \\ \beta^{n^w} & & \beta^{wn_w} \end{bmatrix} = 0$$

$$\Rightarrow det \begin{bmatrix} \beta^{n_1} & \dots & \beta^{wn_1} \\ \beta^{n_2} & & \beta^{wn_2} \\ \vdots & & \vdots \\ \beta^{n_w} & & \beta^{wn_w} \end{bmatrix} = 0$$

$$\Rightarrow \beta^{n_1}\beta^{n_2}\dots\beta^{n_w}.det \begin{bmatrix} 1 & \beta^{n_1} & \dots & \beta^{(w-1)n_1} \\ 1 & \beta^{n_2} & \dots & \beta^{(w-1)n_2} \\ \vdots & & & \\ 1 & \beta^{n_w} & \dots & \beta^{(w-1)n_w} \end{bmatrix} = 0$$

$det(A) = det(A^T)$
But

$$\Rightarrow det \begin{bmatrix} 1 & 1 & \dots & 1 \\ \beta^{n_1} & \beta^{n_2} & & \beta^{n_w} \\ \vdots & & & \vdots \\ \beta^{(w-1)n_1} & \beta^{(w-1)n_2} & & \beta^{(w-1)n_w} \end{bmatrix} = 0, \text{ a contradiction. Hence proved.}$$

Suppose $g(x)$ has zeros at $\beta_1, \dots, \beta_l$ Can we write

$$\mathcal{C} = \{c(x) : c(\beta_1) = \dots = c(\beta_l) = 0\}?$$

No! in general. Example Take $n = 4$ $g(x) = (x+1)^2 = x^2 + 1 | x^4 - 1$
$\mathcal{C}(x) = \{c(x) : c(1) = 0\} = \{a(x)(x+1) \; mod \; x^4 - 1\} \neq < g(x) >$

Some special cases :
1) Binary Hamming codes : let $q = 2$ and $n = 2^m - 1$. Clearly n and q are co-prime. Let α be a primitive element of $GF(2^m)$. Then the (n, k) BCH code of design distance 3 is given by $\mathcal{C} = \{c(x) : c(\alpha) = 0, c(\alpha^2) = 0\}$

In $GF(2)[x]$ $\quad c(x^2) = c(x)^2$

$$(a+b)^2 = a + b \Rightarrow [\sum_{i=1}^{n-1} c_i x^i]^2 = \sum_{i=1}^{n-1} c_1^2 x^{2i} = \sum_{i=1}^{n-1} c_i x^{2i} = c(x^2)$$

Therefore $\mathcal{C} = \{c(x) : c(\alpha) = 0\}$

Let $H = [\alpha \; \alpha^2 \dots \alpha^n = 1]$
Then $C \in \mathcal{C} \Leftrightarrow CH^T = 0$

$\beta \in GF(2^m) \Rightarrow \beta = a_0 + a_1 x + \dots a_{m-1} \; x^{m-1}$ $a_0, \dots, a_{m-1} \in GF(2)$ $n = 7, \alpha \in GF(8)$

$$H = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

2) R.S. codes : Take $n = q - 1$

$$n = q - 1 \quad \mathcal{C} = \{c(x) : c(\alpha) = c(\alpha^2) \dots c(\alpha^{d-1}) = 0\}$$

The generator polynomial for this code is
$g(x) = (x - \alpha)(x - \alpha^2) \dots (x - \alpha^{d-1})\alpha$ is primitive

Ex : Prove that $g(x)$ is the generator polynomial of $\mathcal{C}$
$deg\ g(x) = n - k = d - 1$
$\Rightarrow d = n - k + 1$
Therefore, we have an $(n, k, n - k + 1)$ MDS code

R.S. codes are optimal in this sense

Example : 3 error correcting RS code over $GF(11)$
$n = q - 1 = 10 \quad d = 7 \quad k = n - d + 1 = 10 - 7 + 1 = 4$
$(10, 4, 7)$ RS code over $GF(11)$.

$g(x) = (x - \alpha)(x - \alpha^2)(x - \alpha^3)(x - \alpha^*)(x - \alpha^5)(x - \alpha^6)$
2 is a primitive element
$= (x - 2)(x - 4)(x - 8)(x - 5)(x - 10)(x - 9)$
$= x^6 + 6x^5 + 5x^4 + 7x^3 + 2x^2 + 8x + 2$

EX : dual of an MDS code is dual : Hint columns are linearly dependent. $\Leftrightarrow$ rows are linearly dependent.

Real-life examples :
music CD standard RS codes are $(32, 28)$ and $(28, 24)$ double error correcting codes over $GF(256)$.
NASA standard RS code is a $(255, 223, 33)$ code.

Review : n, q co-prime, $n|q^m - 1$. $\beta \in GF(q^m)$ has order n. A (n, k) BCH code of design d is defined as

$$\mathcal{C} = \{c(x) : c(\beta) = \dots c(\beta^{d-1})\} = 0$$

This definition is useful in deriving the distance structure of the code. The obvious questions to ask are

1) Encoding : need to find the generator polynomial

2) Decoding : will not be covered. PGZ decoder described in section 6.6 is an extension of the $2\varepsilon c$ we described earlier. Peterson - Gorenstein - Zierler

Minimal polynomials : Let $\alpha \in GF(q^m)$. The monic polynomial of smallest degree over $GF(q)$ with α as a zero is called the minimal polynomial of α over $GF(q)$. This is a slight generalization of the definition given previously where q was equal to p.

Recall : Existence and uniqueness, primality

Lemma : Let $f(x)$ be the minimal polynomial of $\alpha \in GF(q^m)$. Suppose $g(x)$ is a polynomial over $GF(q)$ such that $g(\alpha) = 0$. Then $f(x)|g(x)$

Proof : $g(x) = f(x)Q(x) + r(x)$ where $deg\ r(x) < deg\ f(x)$
$g(\alpha) = 0 \Rightarrow r(\alpha) = g(\alpha) - f(\alpha)Q(\alpha) = 0$. Hence $r(x)$ must be 0. Alternate Proof : Appal to primarity of $f(x)$.

Primitive polynomial The minimal polynomial of a primitive element is called primitive polynomial. Suppose α is a primitive element of $GF(q^m)$ and $p(x)$ is the minimal polynomial of α over $GF(q)$. Then $GF(q^m) \sim GF(q)[x]/p(x)$.

The polynomial $x \in GFq[x]/p(x)$ is the primitive element of $GF(q)[x]/p(x)$

Corr : If $p(\beta) = 0$ and $p(x)$ is prime, then $p(x)$ is the minimal polynomial of β.

Theorem : Consider a BCH code with zeros at $\beta, \beta^2, \ldots, \beta^{d-1}$. Let $f_k(x)$ be the minimal polynomial of $\beta^k \in GF(q^m - 1)$ over $GF(q)[x]$. Then the generator polynomial is given by

$$g(x) = LCM(f_1(x), f_2(x) \ldots, f_{d-1}(x))$$

Proof :

$$c(\alpha^k) = 0 \Rightarrow f_k(x)|c(x) \qquad 1 \leq k \leq d-1$$

$$\Rightarrow g(x) = LCM(f_1(x), \ldots, f_k(x)|c(x) \quad \text{by } UPF$$

Therefore $deg.g(x) \leq deg\ c(x) \quad \forall\ c(x) \in \mathcal{C}$
Also $g(x) \in \mathcal{C}$
Therfore $g(x)$ is the generator polynomial

Finding the generator polynomials reduces to finding the minimal polynomials of the zeros of the generator polynomial.

Brute force approach : Factorize $x^n - 1$ into its prime polynomials over $GF(q)[x]$

$$x^n - 1 = b_1(x)b_2(x)\ldots b_l(x)$$

Since $g(x)|x^n - 1, g(x)$ has to be a product of a subset of these factors. $g(x)$ is zero for $x = \beta, \beta^2, \ldots, \beta^{d-1}$.

We know that $g(\beta^k) = 0 \Rightarrow f_k(x)$ is a factor of $g(x)$ and hence of $x^n - 1$. By Unique Prime Factorization $f_k(x)$ must be equal to one of the $b_j(x)\ \ 1 \leq j \leq l$.

We only need to find the polynomial $b_j(x)$ for which $b_j(\beta^k) = 0$. Repeat this procedure to find the minimal polynomials of all zeros and take their LCM to find the generator matrix.

Problem : factorizing $x^n - 1$ is hard

Theorem : Let p be the characteristic of $GF(q)$. Let $f(x)$ be a polynomial over $GF(q).f(x) = \sum_{i=o}^{n} f_i x^i \ \ f_i \in GF(q)$
Then $f(x)^{p^m} = \sum_{i=0}^{n} f_1^{p^m} x^{ip^m} \qquad \forall_m$

Proof :
$$\begin{array}{rrcl} & (a+b)^p & = & a^p + b^p \\ & (a+b)^{p^2} & = & (a^p + b^p)^p = a^{p^2} + b^{p^2} \\ & (a+b)^{p^m} & = & a^{p^m} + b^{p^m} \\ \text{In general} & (a+b+c)^p & = & (a+b)^p + c^p = a^p + b^p + c^p \\ & (a+b+c)^{p^m} & = & a^{p^m} + b^{p^m} + c^{p^m} \end{array}$$

In general
$$\begin{array}{rrcl} & (\sum_{i=0}^{n} a_i)^{p^m} & = & a_0^{p^m} + a_1^{p^m} + \ldots + a_n^{p^n}.a_i \in GF(q) \\ \text{therefore } & f(x)^{p^m} = (\sum_{i=0}^{n} f_i x^i)^{p^m} & = & (f_0 x^0)^{p^m} + \ldots (f_n x^n)^{p^m} \\ & & = & \sum_{i=0}^{n} f_i^{p^m} x^{ip^m} \end{array}$$

Corr : Suppose $f(x)$ is a polynomial over $GF(q)$
Then $f(x)^q = f(x^q)$

Proof : $f(x)$ is prime. If we show that $f(\beta^q) = 0$, then we are done. Suppose $g(x)$ is the minimal polynomial of β^q. Then $f(\beta^q) = 0 \Rightarrow g(x)|f(x)$. $f(x)$ prime $\Rightarrow g(x) = f(x)$

Now,

$$\begin{aligned} f(x)^q &= [\sum_{i=0}^{r} f_i x^1]^q \quad r = deg f(x) \quad \text{because } q = p^m \\ &= \sum_{i=0}^{r} f_i^q x^{iq} \qquad f_i \in GF(q) \Rightarrow f_i^q = f_i \\ &= \sum_{i=0}^{r} f_i x^{iq} \;=\; f(x^q) \\ Therefore f(\beta^q) &= f(\beta)^q = 0 \end{aligned}$$

Conjugates : The conjugates of β over $GF(q)$ are the zeros of the minimal polynomial of β over $GF(q)$ (includes β itself)

Theorem : The conjugates of β over $GF(q)$ are $\beta, \beta^q, \beta^{q^2}, \ldots, \beta^{q^{r-1}}$
where r is the least positive integer such that $\beta^{q^r} = \beta$

First we check that β^{q^k} are indeed conjugates of β
$f(\beta) = 0 \Rightarrow f(\beta^q) = 0 \Rightarrow f(\beta^{q^2}) = 0$ and so on
Therefore $f(\beta) = 0 \Rightarrow f(\beta^{q^k}) = 0$ Repeatedly use the fact that $f(x)^q = f(x^q)$

Let $f(x) = (x - \beta)(x - \beta^q) \ldots (x - \beta^{q^{r-1}})$

The minimal polynomial of β has zeros at $\beta, \beta^q, \ldots, \beta^{q^{r-1}}$

Therefore $f(x) = (x - \beta)(x - \beta^q) \ldots (x - \beta^{q^{r-1}})|$ minimal polynomial of β over $GF(q)$

If we show that $f(x) \in GF(q)[x]$, then $f(x)$ will be the minimal polynomial and hence $\beta, \beta^q, \ldots, \beta^{q^{r-1}}$ are the only conjugates of β

$$\begin{aligned} f(x)^q &= (x - \beta)^q (x - \beta^q)^q \ldots (x - \beta^{q^{r-1}})^q \\ &= (x^q - \beta^q)(x^q - \beta^{q^2}) \ldots (x^q - \beta^{q^r}) \\ &= (x^q - \beta)(x^q - \beta^q) \ldots (x - \beta^{q^{r-1}}) \\ &= f(x^q) \\ \text{But } f(x)^q &= f_0^q + f_1 x^q + \ldots + f_r^q x^{rq} \\ \&f(x^q) &= f_0 + f_1 x^q + \ldots + f_r x^{rq} \end{aligned}$$

Equating the coefficients, we have $f_i^q = f_i$
$\Rightarrow f_i \in GF(q) \Rightarrow f(x) \in GF(q)[x]$.

Examples :
$GF(4) = \{0, 1, 2, 3\}$

Conjugates of 2 in $GF(2)$ are $\{2, 2^2\} = \{2, 3\}$

Therefore minimal polynomial of 2 and 3 is

$$\begin{aligned}(x-2)(x-3) &= x^2 + (2+3)x + 2 \times 3 \\ &= x^2 + x + 1\end{aligned}$$

$GF(8) = \{0, 1, 2, \ldots, 7\} = \{0, 1, \alpha, x+1, \ldots, \alpha^2 + \alpha + 1\}$

The conjugates of 2 in $GF(2)$ are $\{2, 2^2, 2^4\}$ and the minimal polynomial is $x^3 + x + 1$

Cyclic codes are often used to correct burst errors and detect errors.

A cyclic burst of length t is a vector whose non-zero components are among t cyclically consecutive components, the first and last of which are non-zero.

We can write such an errorword as follows :

$$e(x) = x^i b(x) \; mod \; x^n - 1$$

where $deg \; b(x) = t - 1$ and $b_o \neq 0$

In the setup we have considered so far, the optimal decoding procedure is nearest neighbor decoding. For cyclic codes this reduces to calculating the syndrome and then doing a look up (at least conceptually). Hence a cyclic code which corrects burst errors must have syndrome polynomials that are distinct for each correctable error pattern.

Suppose a linear code (not necessarily cyclic) can correct all burst errors of length t or less. Then it cannot have a burst of length $2t$ or less as a codeword. A burst of length t could change the codeword to a burst pattern of length t or less which could also be obtained by a burst pattern of length $\leq t$ operating on the all zero codeword.

Rieger Bound : A linear block code that corrects all burst errors of length t or less must satisfy

$$n - k \geq 2t$$

Proof : Consider the coset decomposition of the code $\#cosets = q^{n-k}$. No codeword is burst of length t. Consider two vectors which are non zero in their first $2t$ components. $d(c_1, c_2) \leq 2t \Rightarrow c_1 - c_2 \notin \mathcal{C}$. $c_1 - c_2$ is a burst of length $\leq 2t \Rightarrow c_1 - c_2 \notin \mathcal{C}$
$\Rightarrow c_1$ and c_2 are in different cosets.

Therefore $\#cosets \geq \#n - tuples$ different in their first $2t$ components

Therefore $q^{n-k} \geq q^{2t}$
$\Rightarrow n - k \geq 2t$

For small n and t, good error-correcting cyclic codes have found by computer search. These can be augmented by interleaving.
Section 5.9 contains more details.

$$g(x) = x^6 + x^3 + x^2 + x + 1 \qquad n - k = 6 \;\; 2t = 6$$

Cyclic codes are also used extensively in error detection where they are often called cyclic redundancy codes.

The generator polynomial for these codes of the form

$$g(x) = (x + 1)\, p(x)$$

where $p(x)$ is a primitive polynomial of degree m. The blocklength $n = 2^m - 1$ and $k = 2^m - 1 - (m + 1)$. Almost always, these codes are shortened considerably. The two main advantages of these codes are :

1) error detection is computing the syndrome polynomial which can be efficiency implemented using shift registers.

2) $m = 32 \qquad (2^{32} - 1, 2^{32} - 33)$
good performance at high rate

$x + 1 | g(x) \Rightarrow x + 1 | c(x) \Rightarrow c(1) = 0$, so the codewords have even weight.

$p(x)|c(x) \Rightarrow c(x) = 0 \Rightarrow c(x)$ are also Hamming codewords. Hence $\mathcal{C}$ is the set of even weight codewords of a Hamming code $d_{min} = 4$

Examples

$$CRC - 16 : g(x) = (x + 1)(x^{15} + x + 1) = x^{16} + x^{15} + x^2 + 1$$

$$CRC - CCITT \qquad g(x) = x^{16} + x^{12} + x^5 + 1$$

Error detection algorithm : Divide senseword $v(x)$ by $g(x)$ to obtain the syndrome polynomial (remainder) $s(x)$. $s(x) \neq 0$ implies an error has been detected.

No burst of length less than $n-k$ is a codeword.

$$\begin{aligned} e(x) &= x^i b(x) \; mod \; x^n - 1 \quad deg \; b(x) < n-k \\ s(x) &= e(x) \; mod \; g(x) \\ &= x^i b(x) \; mod \; g(x) \\ &= [x^i \; mod \; g(x)] \; . \; [b(x) \; mod \; g(x)] \; mod \; g(x) \\ s(x) &= 0 \; * \end{aligned}$$

$\Rightarrow b(x) \; mod \; g(x) = 0$ which is impossible if $deg \; b(x) < deg \; g(x)$

$* \; g(x) \;\; /\!\!/ \; x^i$ Otherwise $g(x)|x^i \Rightarrow g(x)|x^i - x^{n-i}$
$\Rightarrow g(x)|x^n$
But $g(x)|x^n - 1$

If $n-k = 32$ every burst of length less than 32 will be detected.

We looked at both source coding and error control coding in this course.

Probalchstic techniques like turbo codes and trellis codes were not covered.

Modulation, lossy coding.

E2 231: Data Communication Systems January 7, 2005

Homework 1

1. In communication systems, the two primary resource constraints are transmitted bandwidth and channel bandwidth. For example, the capacity (in bits/s) of a band-limited additive white Gaussian noise channel is given by

$$C = W \log_2(1 + \frac{P}{N_0 W})$$

where W is the bandwidth, $\frac{N_0}{2}$ is the two-sided power spectral density and P is the signal power. Find the minimum energy required to transmit a signal bit over this channel. Take $W = 1$, $N_0 = 1$.

(Hint: Energy = Power$\times$Time)

2. Assume that there are exactly $2^{1.5n}$ equally likely, grammatically correct strings of n characters, for every n (this is, of course, only approximately true). What is then the minimum bit-rate required to transmit spoken English? You can make any reasonable assumption about how fast people speak, language structure and so on. Compare this with the 64 Kbps rate bit-stream produced by sampling speech 8000 times/s and then quantizing each sample to one of 256 levels.

3. Consider the so-called Binary Symmetric Channel (BSC) shown below. It shows that bits are erroneously received with probability p. Suppose that each bit is transmitted several times to improve reliability (Repetition Coding). Also assume that, at the input, bit are equally likely to be 1 or 0 and that successive bits are independent of each other. What is the improvement in bit error rate with 3 and 5 repetitions, respectively?

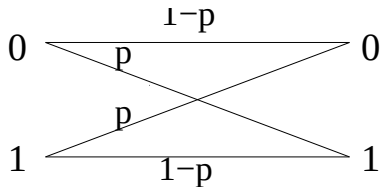


Figure 1: Binary Symmetric Channel

E2 231: Data Communication Systems January 17, 2005

Homework 2

1. Suppose $f(x)$ be a strictly convex function over the interval (a, b). Let $\lambda_k, 1 \leq k \leq N$ be a sequence of non-negative real numbers such that $\sum_{k=1}^{N} \lambda_k = 1$. Let $x_k, 1 \leq k \leq N$, be a sequence of numbers in (a, b), all of which are not equal. Show that

a) $f(\sum_{k=1}^{N} \lambda_k x_k) < \sum_{k=1}^{N} \lambda_k f(x_k)$

b) Let X be a discrete random variable. Suppose $Ef(X) = f(EX)$. Use Part(a) to show that X is a constant.

2-8. Solve the following problems from Chapter 5 of Cover and Thomas : 4, 6, 11, 12, 15, 18 and 22.

9. Solve Parts (a) and (b) of 5.26.

E2 231: Data Communication Systems February 3, 2005

Homework 3

1. A random variable X is uniformly distributed between -10 and +10. The probability density function for X is given by

$$f_X(x) = \begin{cases} \frac{1}{20} & -10 \leq x \leq 10 \\ 0 & \text{otherwise} \end{cases}$$

Suppose $Y = X^2$. What is the probability density function of Y?

2. Suppose $X_1, X_2, \ldots, X_N$ are independent and identically distributed continuous random variables. What is the probability that $X_1 = \min(X_1, X_2, \ldots, X_N)$.
(Hint: Solve for $N = 2$ and generalize)

3-6. Solve the following problems from Chapter 3 of Cover and Thomas : 1, 2, 3 and 7.

7-12. Solve the following problems from Chapter 4 of Cover and Thomas : 2 (Hint: $H(X, Y) = H(Y, X)$), 4, 8, 10, 12 and 14.

E2 231: Data Communication Systems February 18, 2005

Homework 4

1-4. Solve the following problems from Chapter 4 of Cover and Thomas : 2, 4, 9, and 10.

E2 231: Data Communication Systems March 6, 2005

Homework 5

1.

a) Perform a Lempel-Ziv compression of the string given below. Assume a window size of 8.

0001110000111001100010100111001011101110111100011011011101110000100011000 11

b) Assume that the above binary string is obtained from a discrete memoryless source which can take 8 values. Each value is encoded into a 3-bit segment to generate the given binary string. Estimate the probability of each 3-bit segment by its relative frequency in the given binary string. Use this probability distribution to devise a Huffman code. How many bits do you need to encode the given binary string using this Huffman code?

2.

a) Show that a code with minimum distance $d_{\min}$ can correct any pattern of p erasures if

$$d_{\min} \geq p+1$$

b) Show that a code with minimum distance $d_{\min}$ can correct any pattern of p erasures and t errors if

$$d_{\min} \geq 2t+p+1$$

3. For any q-ary $(n.k)$ block code with minimum distance $2t+1$ or greater, show that the number of data symbols satisfy

$$n-k \geq \log_q \left[1+\binom{n}{1}(q-1)+\binom{n}{2}(q-1)^2+\ldots+\binom{n}{t}(q-1)^t\right]$$

4. Show that only one group exists with three elements. Construct this group and show that it is abelian.

5.

a) Let G be a group and $a^{'}$ denote the inverse of any element $a \in G$. Prove that $(a*b)^{'} = b^{'}*a^{'}$.

b) Let G be an arbitrary finite group (not necessarily abelian). Let h be an element of G and H be the subgroup generated by h. Show that H is abelian

6-15. Solve the following problems from Chapter 2 of Blahut : 5, 6, 7, 11, 14, 15, 16, 17, 18 and 19.

16. (Putnam, 2001) Consider a set S and a binary operation $*$, i. e., for each $a, b \in S$, $a * b \in S$. Assume $(a * b) * a = b$ for all $a, b \in S$. Prove that $a * (b * a) = b$ for all $a, b \in S$.

E2 231: Data Communication Systems March 14, 2005

Homework 6

1-6. Solve the following problems from Chapter 3 of Blahut : 1, 2, 4, 5, 11 and 13.

7. We defined maximum distance separable (MDS) codes as those linear block codes which satisfy the Singleton bound with equality, i.e., $d = n - k + 1$. Find all binary MDS codes. (Hint: Two binary MDS codes are the repetition codes $(n, 1, n)$ and the single parity check codes $(n, n-1, 2)$. It turns out that these are the only binary MDS codes. All you need to show now is that a binary $(n, k, n-k+1)$ linear block code cannot exist for $k \neq 1, k \neq n-1$).

8. A perfect code is one for which there are Hamming spheres of equal radius about the codewords that are disjoint and that completely fill the space (See Blahut, pp. 59-61 for a fuller discussion). Prove that all Hamming codes are perfect. Show that the (9,7) Hamming code over GF(8) is a perfect as well as MDS code.

E2 231: Data Communication Systems March 28, 2005

Homework 7

1-12. Solve the following problems from Chapter 4 of Blahut : 1, 2, 3, 4, 5, 7, 9, 10, 12, 13, 15 and 18.

13. Recall that the Euler totient function, $\phi(n)$, is the number of positive numbers less than n, that are relatively prime to n. Show that

a) If p is prime and $k >= 1$, then $\phi(p^k) = (p-1)p^{k-1}$.

b) If n and m are relatively prime, then $\phi(mn) = \phi(n)\phi(n)$.

c) If the factorization of n is $\prod_i q_i^{k_i}$, then $\phi(n) = \prod_i (q_i - 1)q_i^{k_i - 1}$

14. Suppose $p(x)$ is a prime polynomial of degree $m > 1$ over $GF(q)$. Prove that $p(x)$ divides $x^{q^m-1} - 1$. (Hint: Think minimal polynomials and use Theorem 4.6.4 of Blahut)

15. In a finite field with characteristic p, show that

$$(a+b)^p = a^p + b^p, \quad a, b \in GF(q)$$

E2 231: Data Communication Systems April 6, 2005

Homework 8

1. Suppose $p(x)$ is a prime polynomial of degree $m > 1$ over $GF(q)$. Prove that $p(x)$ divides $x^{q^m-1} - 1$. (Hint: Think minimal polynomials and use Theorem 4.6.4 of Blahut)

2-9. Solve the following problems from Chapter 5 of Blahut : 1, 5, 6, 7, 10, 13, 14, 15.

10-11. Solve the following problems from Chapter 6 of Blahut : 5, 6

12. This problem develops an alternate view of Reed-Solomon codes. Suppose we construct a code over $GF(q)$ as follows. Take $n = q$ (Note that this is different from the code presented in class where n was equal to $q-1$). Suppose the dataword polynomial is given by $a(x)=a_0+a_1x+\ldots+a_{k-1}x^{k-1}$. Let α_1, α_2, ..., α_q be the q different elements of $GF(q)$. The dataword polynomial is then mapped into the n-tuple $(a(\alpha_1), a(\alpha_2), \ldots, a(\alpha_q)$. In other words, the j^{th} component of the codeword corresponding to the data polynomial $a(x)$ is given by

$$a(\alpha_j) \quad = \quad \sum_{i=0}^{k-1} a_i \alpha_j^i \in GF(q), 1 \leq j \leq q$$

a) Show that the code as defined is a linear block code.

b) Show that this code is an MDS code. It is often called an extended RS code.

c) Suppose this code is punctured in $r <= q - k = d - 1$ places (r components of each codeword are removed) to yield an $(n = q - r, k, d = n - k + 1 - r)$ code. Show that the punctured code is also an MDS code.

Hint: $a(x)$ is a polynomial of order less than or equal to $k - 1$. By the Fundamental Theorem of Algebra, $a(x)$ can have at most $k - 1$ zeros. Therefore, a non-zero codeword can have at most $k - 1$ symbols equal to 0. Hence $d >= n - k + 1$. The proof follows.

E2 231: Data Communication Systems February 16, 2004

Midterm Exam 1

1.

a) Find the binary Huffman code for the source with probabilities $(\frac{1}{3}, \frac{1}{5}, \frac{1}{5}, \frac{2}{15}, \frac{2}{15})$. (*2 marks*)

b) Let $l_1, l_2, \ldots, l_M$ be the codeword lengths of a binary code obtained using the Huffman procedure. Prove that

$$\sum_{k=1}^{M} 2^{-l_k} = 1$$

(*3 marks*)

2. A code is said to be suffix free if no codeword is a suffix of any codeword.

a) Show that suffix free codes are uniquely decodable. (*2 marks*)

b) Suppose you are asked to develop a suffix free code for a discrete memoryless source. Prove that the minimum average length over all suffix free codes is the same as the minimum average length over all prefix free codes. (*2 marks*)

c) State a disadvantage of using suffix free codes. (*1 mark*)

3. Let $X_1, X_2, \ldots$ be an i.i.d sequence of discrete random variables with entropy H, taking values in the set $\mathcal{X}$. Let

$$B_n(s) = \{x^n \in \mathcal{X}^n : p(x^n) \geq 2^{-ns}\}$$

denote the subset of n-tuples which occur with probability greater than 2^{-ns}

a) Prove that $|B_n(s)| \leq 2^{ns}$. (*2 marks*)

b) For any $\epsilon > 0$, show that $lim_{n\to\infty} Pr(B_n(H+\epsilon)) = 1$ and $lim_{n\to\infty} Pr(B_n(H-\epsilon)) = 0$ (*3 marks*)

4.

a) A fair coin is flipped until the first head occurs. Let X denote the number of flips required. Find the entropy of X. (*2 marks*)

b) Let $X_1, X_2, \ldots$ be a stationary stochastic process. Show that (*3 marks*)

$$\frac{H(X_1, X_2, \ldots, X_n)}{n} \leq \frac{H(X_1, X_2, \ldots, X_{n-1})}{n-1}$$

E2 231: Data Communication Systems March 29, 2004

Midterm Exam 2

1.

a) For any q-ary (n, k) block code with minimum distance $2t+1$ or greater, show that the number of data symbols satisfies

$$n - k \;\geq\; \log_q \left[1 + \binom{n}{1}(q-1) + \binom{n}{2}(q-1)^2 + \ldots + \binom{n}{t}(q-1)^t \right]$$

(*2 marks*)

b) Show that a code with minimum distance d_{min} can correct any pattern of p erasures and t errors if

$$d_{min} \;\geq\; 2t + p + 1$$

(*3 marks*)

2.

a) Let S_5 be the symmetric group whose elements are the permutations of the set $X = \{1, 2, 3, 4, 5\}$ and the group operation is the composition of permutations.

- How many elements does this group have? (*1 mark*)
- Is the group abelian? (*1 mark*)

b) Consider a group where every element is its own inverse. Prove that the group is abelian. (*3 marks*)

3.

a) You are given a binary linear block code where every codeword c satisfies $cA^T = 0$, for the matrix A given below.

$$A \;=\; \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

- Find the parity check matrix and the generator matrix for this code.(*2 marks*)
- What is the minimum distance of this code? (*1 mark*)

b) Show that a linear block code with parameters, $n = 7$, $k = 4$ and $d_{min} = 5$, cannot exist. *(1 mark)*

c) Give the parity check matrix and the generator matrix for the $(n, 1)$ repetition code. *(1 mark)*

4.

a) Let α be an element of $GF(64)$. What are the possible values for the multiplicative order of α? *(1 mark)*

b) Evaluate $(2x^2 + 2)(x^2 + 2)$ in $GF(3)[x]/(x^3 + x^2 + 2)$. *(1 mark)*

c) Use the extended Euclidean algorithm to find the multiplicative inverse of 9 in $GF(13)$. *(2 marks)*

d) Suppose p is prime and α is an element of $GF(p^m)$, $m > 1$. Show that $\alpha^p = \alpha$ implies that α belongs to the subfield $GF(p)$. *(2 marks)*

E2 231: Data Communication Systems April 22, 2004

Final Exam

1. For each statement given below, state whether it is true or false. Provide brief (1-2 lines) explanations.

a) The source code $\{0, 01\}$ is uniquely decodable. (*1 mark*)

b) There exists a prefix-free code with codeword lengths $\{2, 2, 3, 3, 3, 4, 4, 4\}$. (*1 mark*)

c) Suppose $X_1, X_2, \ldots, X_n$ are discrete random variables with the same entropy value. Then $H(X_1, X_2, \ldots, X_n) <= nH(X_1)$. (*2 marks*)

d) The generator polynomial of a cyclic code has order 7. This code can detect all cyclic bursts of length at most 7. (*1 mark*)

2. Let $\{p_1 > p_2 >= p_3 >= p_3\}$ be the symbol probabilities for a source which can assume four values.

a) What are the possible sets of codeword lengths for a binary Huffman code for this source? (*1 mark*)

b) Prove that for any binary Huffman code, if the most probable symbol has probability $p_1 > \frac{2}{5}$, then that symbol must be assigned a codeword of length 1.(*2 marks*)

c) Prove that for any binary Huffman code, if the most probable symbol has probability $p_1 < \frac{1}{3}$, then that symbol must be assigned a codeword of length 2.(*2 marks*)

3. Let $X_0, X_1, X_2, \ldots,$ be independent and indentically distributed random variables taking values in the set $\mathcal{X} = \{1, 2, \ldots, m\}$ and let N be the waiting time to the next occurence of X_0, where $N = \min_n\{X_n = X_0\}$.

a) Show that $EN = m$. (*2 marks*)

b) Show that $E\log(N) \leq H(X)$. (*3 marks*)

Hint: $EN = \sum_{n>=1} P(N >= n)$, $EY = E(EY|X)$.

4. Consider a sequence of independent and identically distributed binary random variables, $X_1, X_2, \ldots, X_n$. Suppose X_k can either be 1 or 0. Let $A_\epsilon^{(n)}$ be the typical set associated with this process.

a) Suppose $P(X_k = 1) = 0.5$. For each ϵ and each n, determine $|A_\epsilon^{(n)}|$ and $P(A_\epsilon^{(n)})$. (*1 mark*)

b) Suppose $P(X_k = 1) = \frac{3}{4}$. Show that

$$\begin{aligned} H(X_k) &= 2 - \frac{3}{4}\log 3 \\ -\frac{1}{n}\log P(X_1, X_2, \ldots, X_n) &= 2 - \frac{Z}{n}\log 3 \end{aligned}$$

where Z is the number of ones in $X_1, X_2, \ldots, X_n$. (*2 marks*)

c) For $P(X_k = 1) = \frac{3}{4}$, $n = 8$ and $\epsilon = \frac{1}{8}\log 3$, compute $|A_\epsilon^{(n)}|$ and $P(A_\epsilon^{(n)})$. (*2 marks*)

5. Imagine you are running the pathology department of a hospital and are given 5 blood samples to test. You know that exactly one of the samples is infected. Your task is to find the infected sample with the minimum number of tests. Suppose the probability that the k^{th} sample is infected is given by $(p_1, p_2, \ldots, p_5) = (\frac{4}{12}, \frac{3}{12}, \frac{2}{12}, \frac{2}{12}, \frac{1}{12})$.

a) Suppose you test the samples one at a time. Find the order of testing to minimize the expected number of tests required to determine the infected sample. (*1 mark*)

b) What is the expected number of tests required? (*1 mark*)

c) Suppose now that you can mix the samples. For the first test, you draw a small amount of blood from selected samples and test the mixture. You proceed, mixing and testing, stopping when the infected sample has been determined. What is the minimum expected number of tests in this case? (*2 marks*)

d) In part (c), which mixture will you test first? (*1 mark*)

6.

a) Let C be a binary Hamming code with $n = 7$ and $k = 4$. Suppose a new code C' is formed by appending to each codeword $\bar{x} = (x_1, \ldots, x_7) \in C$, an over-all parity check bit equal to $x_1 + x_2 + \ldots + x_7$. Find the parity check matrix and the minimum distance for the code C'? (*2 marks*)

b) Show that in a binary linear block code, for any bit location, either all the codewords contain 0 in the given location or exactly half have 0 and half have 1 in the given location. (*3 marks*)

7.

a) Suppose α and β are elements of order n and m respectively, of a finite Abelian group and that $GCD(n, m) = 1$. Prove that the order of the element $\alpha * \beta$ is nm. (*2 marks*)

b) Prove that $f(x) = x^3 + x^2 + 1$ is irreducible over $GF(3)$. (*1 mark*)

c) What are the multiplicative orders of the elements of $GF(3)/x^3+x^2+1$? (*1 mark*)

8.

a) Suppose α is a primitive element of $GF(p^m)$, where p is prime. Show that all the conjugates of α (with respect to $GF(p)$ are also primitive. (*2 marks*)

b) Find the generator polynomial for a binary double error correcting code of block-length $n = 15$. Use a primitive α and the primitive polynomial $p(x) = x^4 + x + 1$. (*2 marks*)

c) Suppose the code defined in part(b) is used and the received senseword is equal to $x^9 + x^8 + x^7 + x^5 + x^3 + x^2 + x$. Find the error polynomial. (*1 mark*)

9. A linear (n, k, d) block code is said to be maximum distance seperable (MDS) if $d_{\min} = n - k + 1$.

a) Construct an MDS code over $GF(q)$ with $k = 1$ and $k = n - 1$. (*1 mark*)

b) Suppose an MDS code is punctured in $s <= n - k = d - 1$ places (s check symbols are removed) to yield an $(n = q - s, k)$ code. Show that the punctured code is also an MDS code. (*2 marks*)

c) Suppose C is an MDS code over $GF(2)$. Show that no MDS code can exist for $2 \leq k \leq n - 2$. (*2 marks*)

10.

a) Find the generator polynomial of a Reed-Solomon (RS) code with $n = 10$ and $k = 7$. What is the minimum distance of this code? (*2 marks*)

b) Consider an (n, k) RS code over $GF(q)$. Suppose each codeword $(c_0, c_2, \ldots, c_{n-1})$ is extended by adding an overall parity check c_n given by

$$c_n = -\sum_{j=0}^{n-1} c_j$$

Show that the extended code is an $(n + 1, k)$ MDS code. (*3 marks*)