ELEC 6131

Error Detecting and Correcting Codes

Lecture 1: Jan 13, 2015

- Course outline.
- Exams: grading scheme.
- Projects.

Why an How of Channel Coding:

In this course we will mainly talk about how we perform channel Coding. I assume that through taking Communications courses you know, or at least have some clue about, why we perform Channel Coding. So, I will only spend a lillle on this subject.

You may say: "What type of question is this? We do error control Coding to reduce the error rate." Of course as the name indicates error Control coding's goal is to reduce the number of errors. However, is this the only way to control errors?

You may recall that in a BPSK (binary phase shift keying) scheme, the probabity of error is:

$$P_e = Q\left(\sqrt{2\frac{E_b}{N_o}}\right)$$

Nothing* prevents you from increasing E_b to reduce P_e to any value you wish no matter how small it is. Let's investigate this option. Assume that you are using a BPSK modem and your $\frac{E_b}{N_0}$ is 4dB. Then your bit error rate (BER) is 10^{-2}. That is 1 bit in 100. This may not be so bad if we used old digital techniques like PCM. But today's systems are packet oriented. That is the data is not sent sample after sample. For example in LPC (Linear Predictive Coding) of speech, each 20 msec. of speech is represend by a frame of length 180 bits. With $BER = 10^{-2}$, almost all frames will have one or more errors.

* Nothing except practical considerations.

For image and video the situation is much worse as the blocklength is 188 bites (1504 bits) for MPE

For new services a so called "error-free" channel, i.e., one with BER $\leq 10^{-9}$ or 10^{-10} is required. Returning to the BPSK, if we look for BER of 10^{-9} we have to increase the $\frac{E_b}{N_0}$ to 12.5 dB. This means an increase of 8.5 dB or 7 times increase.

For a cellular phone user, everything being the same, it means that his/her battery drains 7 times faster. So, instead of say, every 24 hours, he/she has to charge every 3.5 hours.

Here is where error Control Coding Comes handy. You may easily get 10^{-9} or better BER with 4 dB of $\frac{E_b}{N_0}$ or less. The price paid is an increase in bandwidth requirement.

A bit of information theory:

In 1948 Claude Shannon defined the capacity of a channel as the maximum rate at which you may communicate over a channel with absolutely no error (of course asymptotically, i.e., as the codewords' length tends to infinity).

In another word: channel capacity, C, is a quantity related to a specific channel. With appropriate signalling (coding) you may approach C as tightly as you wish. However, you cannot exceed this rate.

In particular, Shannon showed that the capacity of the Additive White Gaussian Noise (AWGN) channel is:

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

where W is the bandwidth available, P is the signal power at the receiver and N is the noise power.

Assume that, we transmit at maximum rate $R = C$. Then $R = W \log_2(1 + \frac{P}{N})$

We can write $P = E_b R$, that is energy per bit times the number of bit per second gives the energy in a second or power.

We can also write $N = N_o W$.

So, we have

$$\frac{R}{W} = \log_2\left(1 + \frac{E_b}{N_o}\frac{R}{W}\right)$$

Solving for $\frac{E_b}{N_o}$ we get:

$$\frac{E_b}{N_o} = \frac{2^{R/W} - 1}{R/W}$$

Let's take $\frac{R}{W} = 1$ bits/sec./Hz.

Then $$\frac{E_b}{N_o} = \frac{2^1 - 1}{1} = 1 \text{ or } \underline{\underline{0}} \text{ dB}$$

This means that sending one bit per second on each Hz. of bandwidth requires 0 dB of SNR!

Going back to the previous example, if we take the information code it with a rate $\frac{1}{2}$ code, i.e., one parity bit for each data bit, and then use QPSK, we have $\frac{R}{W} = 1$. So, anything we get for $\frac{E_b}{N_o}$ that is far from 0 dB is a waste of power.

Fortunately, in recent years Turbo Codes, LDPC Codes and similar coding schemes have closed the gap considerably.

Error Control Strategies:

The title of this course is "Error Detecting and Correcting Codes".

You may wonder what is the value of detecting errors if we do not correct them. For one thing, you may use the knowledge that the information is erroneous to discard it. But more importantly, You (a receiver you use) can ask for retransmiss This is basis of Automatic-Repeat-Request (ARQ) ARQ is very efficient as error detection is simpler and requires less overhead (parity) than error correction and you only have to retransmit when there is error. In fact, it makes ARQ adaptive to channel condition. The problem with ARQ, however, is the delay when the channel condition requires retransmission. This rules out the use of ARQ for delay sensitive (real-time) data such as voice calls and live video.

The other strategy is Forward Error Correction (FEC). In this scheme, we add enough parity so that the errors expected to occur are correctable. The problem is that most often the channel is better than what you have designed the code for and, hence, the parity bits are wasted. On the other hand if the channel becomes worse than what you have considered the errors remain uncorrected or it is even possible that you add more errors by trying to correct errors.

Types of Codes:

There are basically two types of Codes:

- Block Codes
- Convolutional Codes
  (in general trellis Codes)

In a block Code a number of bits say k enter the encoder an $n \geq k$ bits leave as the output. That is $n-k$ parity bits are added to k information bits. The ratio $r = \frac{k}{n}$ is the rate of the Code. Do not mix this with the transmission rate which is the number of bits transmitted in one second.

Block Codes may be <u>Systematic</u> or non-Systematic. A Systematic Code has all its k information bits, next to each other, usually at the beginning or end of the codeword.

Block Codes may be <u>linear</u> or not. In A linear Code sum of (in general the linear combination of every two Codewords is another Codeword.

In a linear code parities are formed by modulo two (X-oring) addition of information bits. That is no other gate such as AND and OR is used.

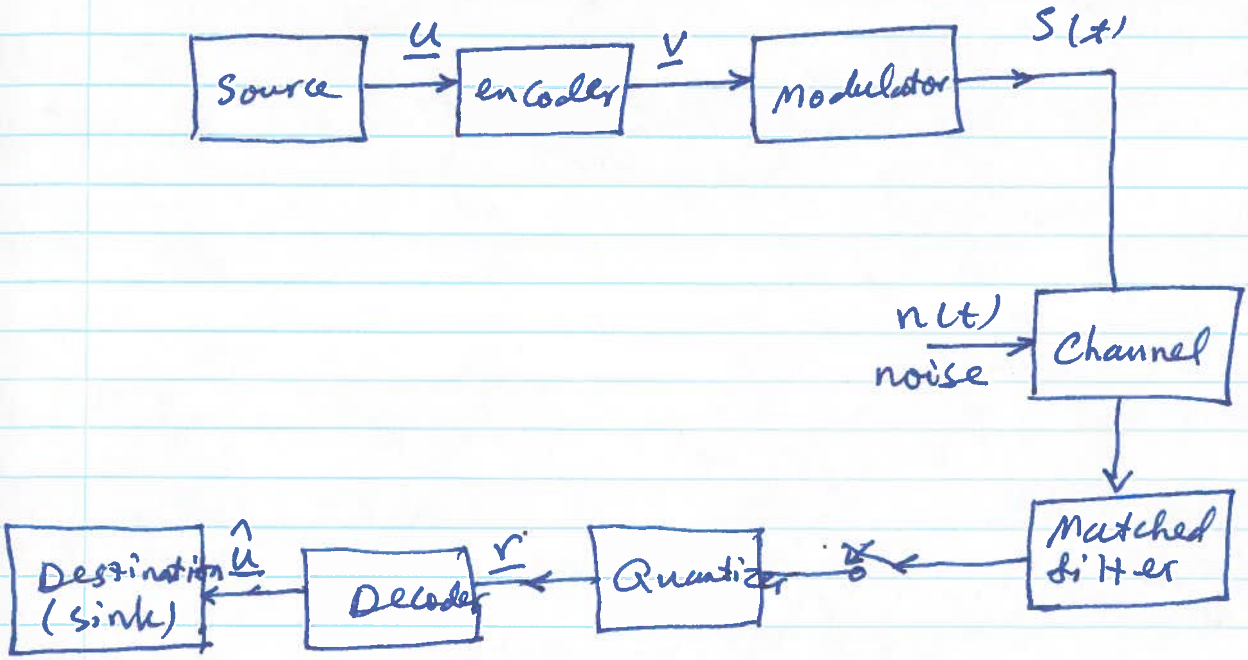
In this course, we only talk about linear Codes. That is why the first thing we do after this introduction is to go over Some basic concepts from linear algebra.

The second type of Codes are Convolutional codes. Convolutional Codes are linear trellis Codes.

In a convolutional Code the input to the encoder is k bits, the same as block Codes. (k is usually ~~and~~ small). The output however, is not only a function of the input bits, but also on previous inputs (or previous outputs).

Maximum likelihood decoding:

When decoding a received signal to recover the information bits, it is obvious that our goal is to minimize expected (average) probability of error.



A vector of binary (or in general multi-valued) symbols $\underline{v}$ is encoded to generate the Codeword $\underline{v}$. After modultion we have a waveform S(t) which gets corrupted by noise n(t).

After filtering, we sample the Signal to get a vector of samples from which we can recover a (hopefully correct) copy of the original data, $\underline{u}$

Depending on whether we would like to perform hard-decision decoding or soft-decision decoding, we take the number of the quantizer levels 2 (for Hard-decision) and mor for soft-decision decoding.

we would like to minimize the probability of error

$$P(E|\underline{r}) = P(\hat{\underline{u}} \neq \underline{u} | \underline{r})$$

or equivalently

$$P(E|\underline{r}) = P(\hat{\underline{v}} \neq \underline{v} | \underline{r})$$

Since there is a one-to-one Correspondance between $\underline{u}$ and $\underline{v}$.

To minimize probability of error $P(\hat{\underline{v}} \neq \underline{v} | \underline{r})$ is equivalent to maximizing $P(\hat{\underline{v}} = \underline{v} | \underline{r})$.

But,

$$P(\underline{v}|\underline{r}) = \frac{P(\underline{r}|\underline{v})P(\underline{v})}{P(\underline{r})}$$

and we need to choose $\hat{\underline{v}}$ as the $\underline{v}$ that maximizes $P(\underline{v}|\underline{r})$. Note that $P(\underline{r})$ is common to all $P(\underline{v}|\underline{r})$ for ~~all~~ every $\underline{v}$. So, we do not have to consider it. Also if the input vectors $\underline{u}$ and consequently $\underline{v}$ have all the same probability, we can ignore $P(\underline{V})$ and choose $\underline{v}$ that maximizes $P(\underline{r}|\underline{V})$

This is called maximum-likelihood detection (MLD). The original one we started with, i.e., maximizing $P(\underline{V}|\underline{r})$ is called maximum a priori probability (MAP) and is needed when the input bits are not equally probable. This is true in case of Turbo-decoding in iterations >1 when we have some information about the values of bits.

If the channel is memoryless, i.e., it does affect bits without any relationship to what it has done to previously (or later) transmitted bits, we have

$$p(\underline{r}|\underline{V}) = \prod_i p(r_i|V_i)$$

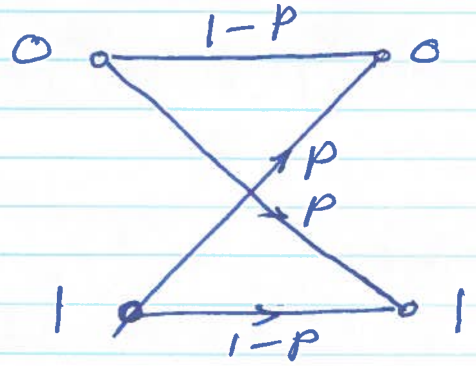
Taking log of $p(\underline{r}|\underline{V})$, we get log-likelihood functi

$$\log P(\underline{r}|\underline{V}) = \sum_i \log p(r_i|V_i)$$

Now let's see what happens when we use MLD for a channel ove which we have used binary modulation such as BPSK in previous example. Then a $\underline{0}$ or a $\underline{1}$ changes with probability $p = Q(\sqrt{2\frac{E_b}{N_0}})$

So $P(r_i \neq v_i) = p$

This can be shown as



This is called a binary symmetric channel (BSC

Denote
~~Assume~~ the number of places $\underline{r}$ and $\underline{v}$ are different by $d(\underline{r}, \underline{v})$. This is called the Hamming distance.

For the BSC Channel, we can write ~~the~~ log-likelihood function as:

$$\log P(\underline{r}|\underline{v}) = d(\underline{r},\underline{v})\log p + [n - d(\underline{r},\underline{v})]\log(1-p)$$

$$= d(\underline{r},\underline{v})\log\frac{p}{1-p} + n\log(1-p)$$

Assuming $p < \frac{1}{2}$ we have $\log\frac{p}{1-p} < 0$ and Therefore maximizing $\log p(\underline{r}|\underline{v})$ is equivalent to minimizing ~~der~~ $d(\underline{r},\underline{V})$. That is to choose the codeword that is closes in Hamming distance to $\underline{r}$.

Introduction to Algebra

You are certainly familiar with the concept of set. A set is a "shapeless" ensemble of objects. What gives shape (structure) to a set is a relationship between its elements or an operation transforming elements of the set to each other.

A binary operation (it does not have anything to do with zero and one) is an operation that takes two elements (that is why it is called binary and gives another element. Addition or multiplication are operations defined for set of integers, reals, rational numbers.

A set is said to be closed under an operation if the result of applying the operation to two elements of the set results in an element in The set.

Take a set G with operation $*$ we say

G is closed under $*$ if:

for any $a, b \in G$ we have $a*b \in G$.

Example: The set of positive integers is closed under addition but not closed under subtraction.

Example: The set of integers is closed under multiplication but not under division.

Definition: A set G with a binary operation $*$ is a group if; the set is closed under $*$ and:

i) The binary operation is associative:

$$a*(b*c) = (a*b)*c$$

i.e., no need for paranthesis.

ii) G contains an element e such that:

for all $a \Rightarrow a*e = e*a$

iii) For any element $a \in G$ there is $a' \in G$ such that $a*a' = a'*a = e$.

a' is called the inverse of a.

Theorem: The identity element of a group G is unique.

assume that G has two identities, say e and $\hat{e}$. Then

$$\hat{e} = \hat{e} * e = e.$$

Theorem: The inverse of an element of Group is unique.

Proof:

Assume that $a \in G$ has two inverses a' and a''. Then

$$a' = a' * e = a' * (a * a'') = (a' * a) * a'' = e * a'' = a''.$$

Example: The set of integers with operation $+$ forms a group. The identity element is zero.

Example: set of Integers with multiplication is not a group. Why?

Question: . Is the set of rationals with multiplication a group? If not what should we do to form a group under multiplication?

Finite groups:

Take a set Consisting of 5 elements, $G = \{0,1,2,3,4\}$. It is obvious that under addition (usual addition) this is not a group. For one thing it is not even closed under addition. For example $2+3=5 \notin G$. Secondly, there is no way we can define inverse.

To find an operation that when applied to G makes it a group, we can use addition followed by some "truncation" or in proper terms make "modulo" operation. This means to divide the result of ordinary addition by 5 and take the remainder as the modulo-5 sum.

$\boxplus_5$	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

ingeneral for $G=\{0,1,2,\dots,m-1\}$ we use modulo-m addition. That is we divide the result $a+b$ by m and take the remainder r as the modulo-m sum.

$G=\{0,1,2,\dots,m-1\}$ forms a group under modulo-m addition:

a) First of all when we divide $a+b$ by m

we get $\quad a+b=mq+r$

where $\quad 0\leqslant r\leqslant m-1$

that is $\quad r=a\boxplus b\in G$

So G is <u>closed</u> under $\boxplus$

b) Since the ordinary is associative we have

$$a+b+c=(a+b)+c=a+(b+c)$$

Dividing $a+b+c$ by m, we get

$$a+b+c=qm+r \quad \text{with} \quad 0\leqslant r<m$$

Dividing $a+b$ by m, we get

$$a+b=q_1m+r_1 \quad \text{again} \quad 0\leqslant r_1<m$$

So $a\boxplus b=r_1$

Dividing $r_1 + c$ by m, we get

$$r_1 + c = q_2 m + r_2, \qquad 0 \leq r_2 < m$$

So

$$r_1 \boxplus c = r_2$$

and

$$(a \boxplus b) \boxplus c = r_2$$

Adding $a + b = q_1 m + r_1$ to

$$r_1 + c = q_2 m + r_2$$

we get

$$a + b + c = (q_1 + q_2) m + r_2$$

So

$$(a \boxplus b) \boxplus c = r$$

we can also show:

$$a \boxplus (b \boxplus c) = r$$

So

$$(a \boxplus b) \boxplus c = a \boxplus (b \boxplus c)$$

That is modulo-m addition is associative

c) 0 is the identity element since

$0 \boxplus a = a \boxplus 0 = a$ for any $a \in G$.

d) $a + (m - a) = m$

So $a \boxplus (m - a) = 0$

So $m - a$ is the inverse of a

and since $0 \leqslant m - a \leqslant m - 1$

we can say that each element of G has an inverse.

This is called an additive group.

Multiplicative Group

Let p be a prime the set $G = \{1, 2, \dots, p-1\}$ under modulo p multiplication $\boxdot$ form a group.

$a \boxdot b$ is defined as the remainder of dividing the real multiplication resute of $a \cdot b$ by p.

Since $a \cdot b = qp + r$

$0 < r < p$ the set G is closed under $\boxdot$.

It is easy to show that $\boxdot$ is also cummutative, associative and the identity element is 1.

Take $i \in G$. It is clear that since p is a prime i and G are mutually prime and we can find a and b such that

$$a.i+b.p=1$$

or

$$a.i=-b.p+1$$

that is $a \boxdot i = i \boxdot a = 1$ if $a \in G$

if $a \notin G$ then we divide a by p to get

$$a=q.p+r$$

and r is the inverse of i since:

$$a.i=(q.p+r)i=-b.p+1$$

$$\Rightarrow r.i=-(b+qi)p+1$$

Modulo-5 multiplication

$\boxdot$	1	2	3	4
1	1	2	3	4
2	2	4	1	3
3	3	1	4	2
4	4	3	2	1