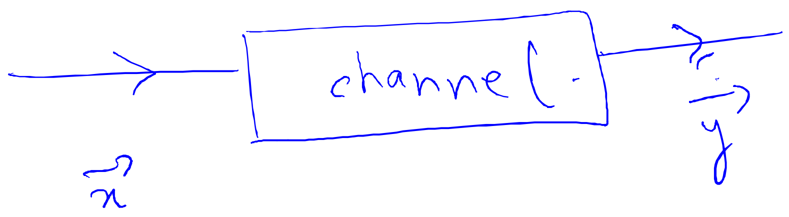
Week 1

Monday, January 11, 2016 12:52 PM



Channel → a mathematical model for communication & storage systems

Input alphabet → F

Output alphabet → Φ

$\vec{x} \in F^n$

$\vec{y} \in \Phi^m$

Two types of channels

→ probabilistic

→ Adversarial → later

probabilistic channel

$P_{\vec{Y}/\vec{X}}(\vec{y}/\vec{x})$ → probability distribution of output, given input.

Example → $\boxed{\text{BINARY SYMMETRIC CHANNEL}}$ (BSC)

$F = \{0,1\}$

$\Phi = \{0,1\}$

$m = n.$

upper case

$P_{\vec{Y}/\vec{X}}(\vec{y}/\vec{x})$ $\left[\begin{array}{l}\vec{y} = (y_1, y_2, \dots y_n) \\ \vec{x} = (x_1, x_2, \dots x_n)\end{array}\right]$

$$= \prod_{i=1}^{n} p_{Y/X}(y_i/x_i)$$

Lower case

$$p_{Y/X}(y/x) = \begin{cases} 1-p & \text{if } y = x \\ p & \text{if } y = \text{NOT}(x) \end{cases}$$

cross-over probability.

For instance, if $n=2$

$$P_{\vec{Y}/\vec{X}}(00/00) = (1-p) \times (1-p) = (1-p)^2.$$

$$P_{\vec{Y}/\vec{X}}(01/00) = (1-p)\,p$$

Alternate view

$\vec{Y} = \vec{X} + \vec{Z}$ → Always XOR !

$n \times 1$

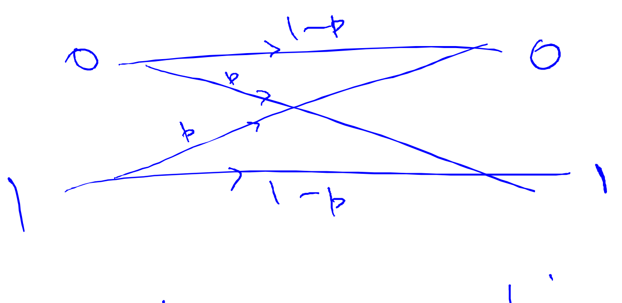
$\vec{Z}$ → i.i.d components

Ber. (p)

$\vec{Z} \rightarrow \begin{pmatrix} z_1 \\ \vdots \\ z_n \end{pmatrix}$ → $P(z_i = 1) = p$

$P(z_i = 0) = 1-p$

i.e all operations are modulo 2



Independence assumption

Y_i is independent of $\{X_j : j \neq i\}$ given X_i

Memoryless channels

given X_i

channels

$$\Pr\left(Y_i = y_i \mid X_1 = x_1, X_2 = x_2 \ldots X_n = x_n \right)$$

$$= \Pr(Y_i = y_i \mid X_i = x_i)$$

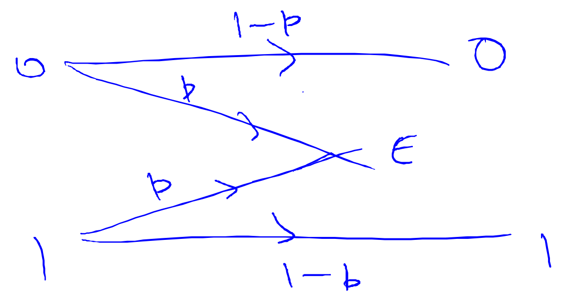
$$= p_{Y/X}(y_i / x_i)$$

Binary erasure channel. (BEC)

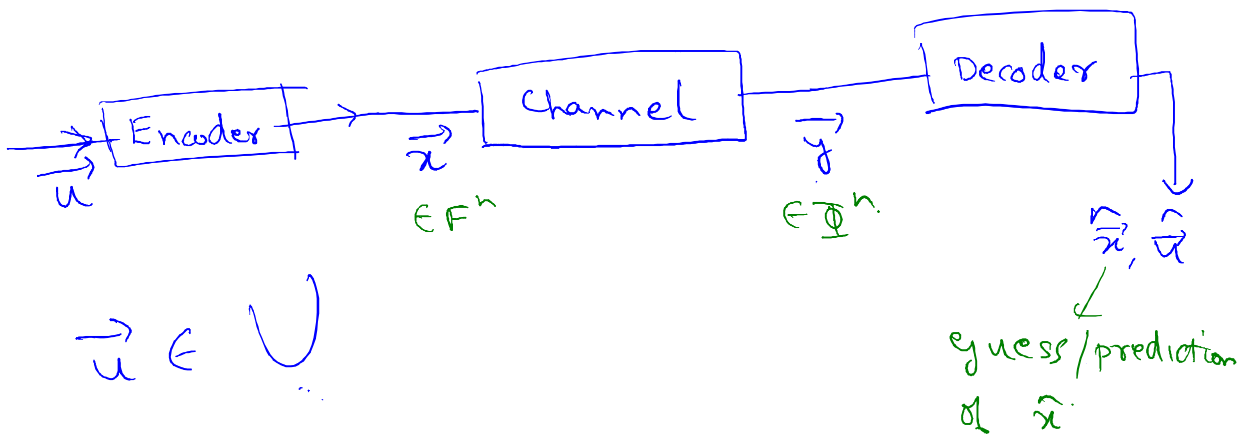
$F = \{0, 1\}, \quad \Phi = \{0, 1, \epsilon\}$ erasure symbol

$$P_{\vec{Y}/\vec{X}} = \prod_{i=1}^{n} p_{Y/X}(y_i / x_i)$$

$$p_{Y/X}(y/x) = \begin{cases} 1-p & \text{if } y = x \\ p & \text{if } y = \epsilon \\ 0 & \text{if } y = \text{NOT}(x) \end{cases}$$



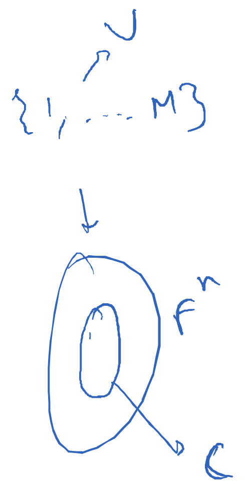
Error correcting code



$\vec{u} \in U$

$|U| = M$

An (n, M) error correcting code contains the following



→ A one-to-one mapping from U to F^n, called encoder

→ A one-to-one mapping from Φ^{m} to U. (or from Φ^m to F^n) called decoder.

usually, we assume $m = n$.

1/13/2016

Formally, an (n, M) code contains a set $C \subseteq F^n$ where $|C| = M$

{Encoder mapping from $U \rightarrow C$ is implicit}

For a given channel, and a code C, a decoder is a mapping from Φ^m to C

$$D(\vec{y}) = \vec{x}_0, \quad \text{where } \vec{x}_0 \in C$$

$\hookrightarrow$ Decoding function

Hamming distance

→ imposes some geometry on F^n, Φ^m etc.

Given a set F, and a number n, the Hamming distance between strings $\vec{x}, \vec{y} \in F^n$ is

$$d_H(\vec{x}, \vec{y}) = \left| \{ i : x_i \neq y_i \} \right|$$

[where $\vec{x} = (x_1, \cdots x_n)$
$\vec{y} = (y_1, \cdots y_n)$]

Example

$$d_H(0010, 0111) = 2.$$

Hamming weight

For any set F with a reference element "0", the Hamming weight of a vector $\vec{x} \in F^n$ is

$$d_H(\vec{x}, \vec{0})$$

Rate & Min. distance of a code

Rate of an (n, M) code C over alphabet F is

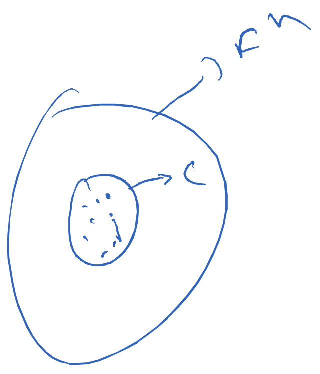
$$\boxed{R = \frac{\log_{|F|} M}{n}}$$

$$M = |C| \leq |F|^n \Rightarrow \log_{|F|} M \leq n$$

$$\Rightarrow R \leq 1$$

Min distance of a code C is

$$d_{min} = \min_{\substack{\vec{x} \neq \vec{y} \\ \vec{x}, \vec{y} \in C}} d_H(\vec{x}, \vec{y})$$



(n, M, d) code has block length n, and min. distance d.

Examples

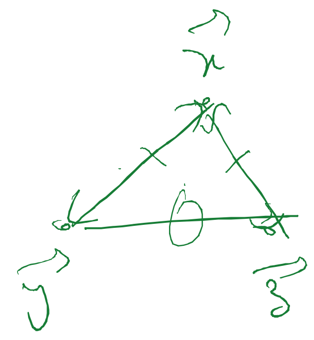
Properties

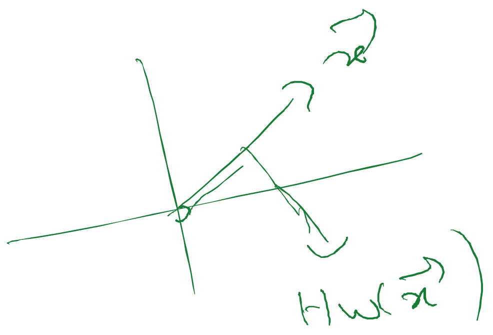
1) $d_H(\vec{x}, \vec{y}) \geq 0$ with equality if $\vec{x} = \vec{y}$

2) $d_H(\vec{x}, \vec{y}) = d_H(\vec{y}, \vec{x})$

3) Triangle inequality

$$d_H(\vec{x}, \vec{y}) + d_H(\vec{y}, \vec{z}) \geq d_H(\vec{x}, \vec{z})$$





$$d_H(\vec{x}, \vec{y}) = H.w(\vec{x} + \vec{y})$$

for binary vectors $\vec{x}, \vec{y}$

Repetition code over $F = \{0,1\}$

$M = 2$

arbitrary n.

$C = \{0000\cdots0,\ 11\cdots1\}$

Rate $= \frac{1}{n}$

$[\longrightarrow 0 \text{ as } n \to \infty]$

Min. distance $= n$

Single parity code over $F = \{0,1\}$

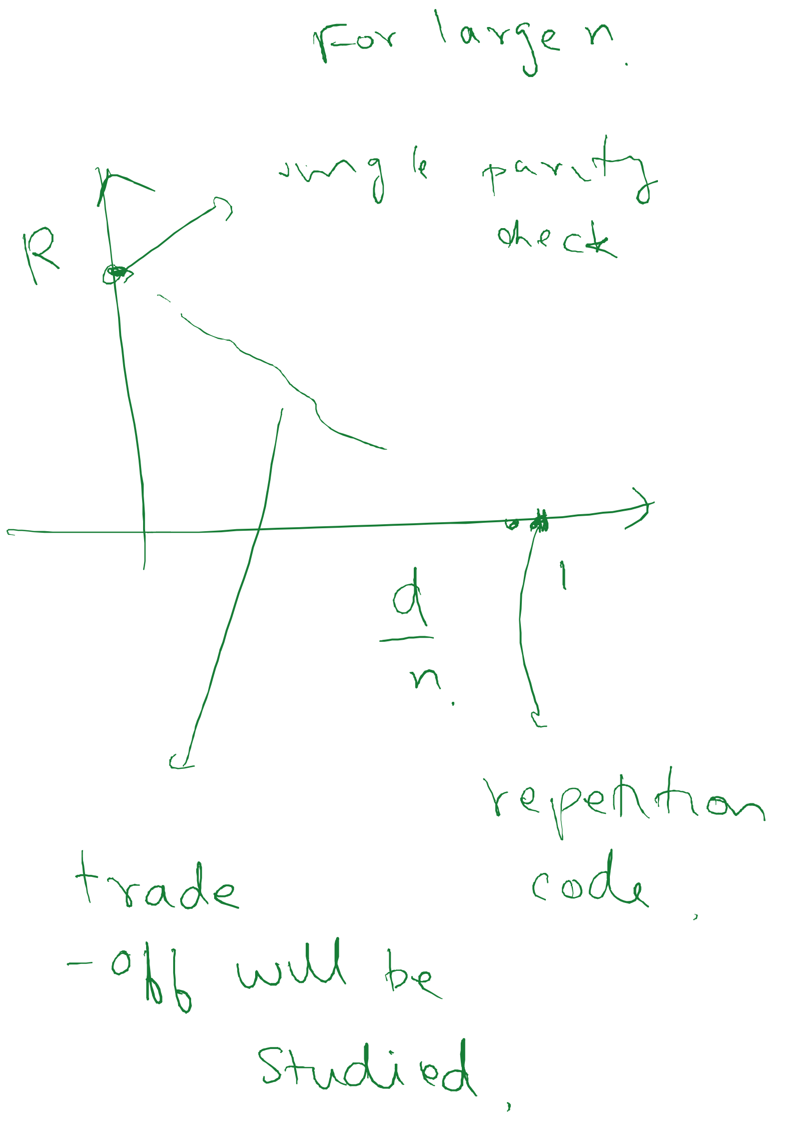
$M = 2^{n-1}$, arbitrary n.

$U = \{0,1\}^{n-1}$

$$\vec{u} = (u_1, u_2, \cdots u_{n-1}) \to \underbrace{(u_1, u_2, \cdots u_{n-1}, \overset{\downarrow \text{XOR}}{u_1 + u_2 + \cdots u_{n-1}})}_{\vec{x}}$$

Rate $= \frac{n-1}{n} = 1 - \frac{1}{n}$

$\longrightarrow 1$ as $n \to \infty$



Example

$n = 3$.

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

min. distance $= 2$

In fact, in general, for a single parity check code,

$d_{min} = 2$ [does not depend on n.]

$$\left.\begin{matrix} 00\cdots\overset{\text{parity check}}{\overset{\downarrow}{0}}0 \\ 00\cdots 11 \end{matrix}\right\} \text{distance} = 2$$

Note that for a single parity code

$\vec{x} \in C$ if and only if

$x_1 + x_2 \cdots x_n = 0 \longrightarrow$ called a parity check equation.

$\hookrightarrow$ always XOR.

Decoder for probabilistic channels

For a channel $P_{\vec{y}/\vec{x}}$, the probability of error of a codeword $\vec{x}_0 \in C$.

$$P_{err}(\vec{x}_0) = P(\underset{\hookrightarrow \text{Decoder}}{D}(\vec{y}) \neq \vec{x}_0 / \vec{x} = \vec{x}_0)$$

1) Maximal prob. of error

$$P_{err}(C) = \max_{\vec{x}_0 \in C} \hat{P}_{err}(\vec{x}_0)$$

2) Average prob. of error

$$P(\;) = \frac{1}{\;} \Sigma\, P_{err}(\vec{x}_0)$$

$$P_{err}(C) = \frac{1}{M} \sum_{\vec{x}_0 \in C} P_{err}(\vec{x}_0)$$

MAP decoder

minimizes prob. of error for a given $P_{\vec{X}}(\vec{x})$ →

$$D(\vec{y}) = \arg\max_{\vec{x}_0 \in C} P_r(\vec{X} = \vec{x}_0 / \vec{Y} = \vec{y})$$

Sketch of proof of optimality

$$p(\text{error}) = \sum_{\vec{y} \in \Phi^n} P_r(\text{error} / \vec{Y} = \vec{y}) P_r(\vec{Y} = \vec{y})$$

$$= \sum_{\vec{y}} P_r(\vec{x} \neq D(\vec{y}) / \vec{Y} = \vec{y}) P_{\vec{Y}}(\vec{y})$$

$$= \sum_{\vec{y}} P_{\vec{Y}}(\vec{y}) [1 - P_r(\vec{x} = D(\vec{y}) / \vec{Y} = \vec{y})]$$

$$= \sum_{\vec{y}} P_{\vec{Y}}(\vec{y}) [1 - \underbrace{P_{\vec{X}/\vec{Y}}(D(\vec{y}) / \vec{y})}_{}]$$

larger this is, the smaller the pr(error), hence the MAP rule

ML Decoder

minimizes prob. of error If $P_{\vec{X}}(\vec{x})$ is uniform.

$$D(\vec{y}) = \arg\max_{\vec{x}_0 \in C} P_{\vec{Y}/\vec{X}}(\vec{y} / \vec{x}_0)$$

BSC(p)

It can be shown that, if $p < \frac{1}{2}$

$$D(\vec{y}) = \arg\min_{\vec{x}_0 \in C} d_H(\vec{y}, \vec{x}_0)$$

↳ ML decoder.

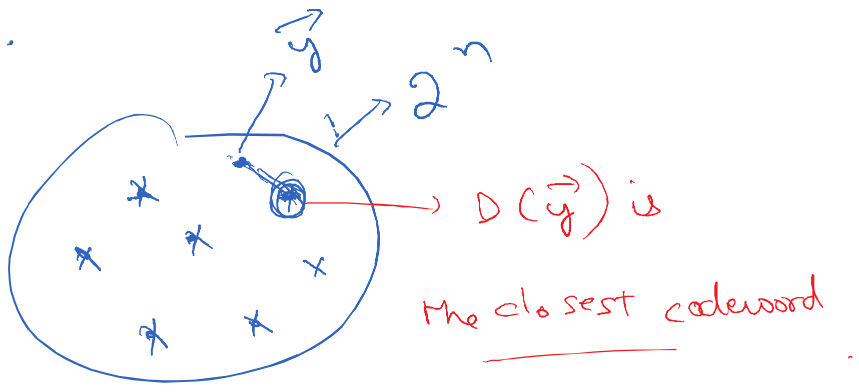
Note that for a BSC

$$\boxed{P_{\vec{Y}/\vec{X}}(\vec{y} / \vec{x}) = p^{d_H(\vec{y}, \vec{x})} (1-p)^{n - d_H(\vec{y}, \vec{x})}}$$

Also, if $p < \frac{1}{2}$,

$$p^a (1-p)^{n-a} > p^b (1-p)^{n-b}$$

iff $a < b$



Example

n=3, repetition code.

$C = \{000, 111\}$

$\vec{y}$	→	$D(\vec{y})$
000	→	000
001	→	000
010	→	000
011	→	111
100	→	000
101	→	111
110	→	111
111	→	111
011	→	111

$D(\vec{y})$ → ML decoder.

$$P_{err}(000) = P(\vec{y} = 101 \text{ or } \vec{y} = 110 \text{ or } \vec{y} = 111 / \vec{x} = 000)$$

$$P_{err}(000) = 3p^2(1-p) + p^3$$

1/15/2016

$P_{err}(000) \lesssim p$

can be easily verified that

$\boxed{P_{err}(000) < p}$ → longer Repetition increases reliability

As $n \to \infty$, $P_{err} \to 0$ for repetition code

given a number ϵ, what is the highest rate R, s.t $\exists$ a (n, M) code of rate R. with $P_{err} \leq \epsilon$.

Shannon's results for BSC(p)

Achievability

1) If $R < \boxed{1 - H(p)}$, then for every ϵ, there exists an (n, M) code of rate R. s.t. $P_{err} \leq \epsilon$, for sufficiently large n

$H(x) = -x \log_2 x - (1-x) \log_2 (1-x)$

capacity of Binary Symmetric channel BSC(p)

allowitz

2) If $R > \boxed{1 - H(p)}$, then. as $n \to \infty$, $P(err) \to 1$ for every sequence of (n, M) codes with rate R.

w converse.

Remarks

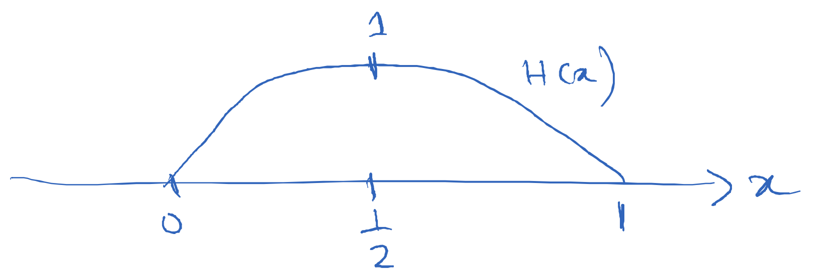
1) Shannon's results are non-constructive. In this course, we will learn how to construct codes

2) Shannon's approach ignored computational complexity.

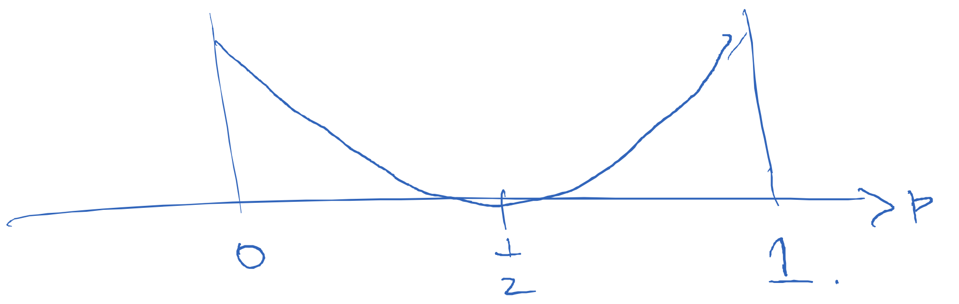
Eg. comp. complexity of ML decoding techniques BSC $\sim$ order of $M = |F|^{nR}$.

we will learn low-complexity decoding

3) $H(x) \to$ called the binary entropy function.



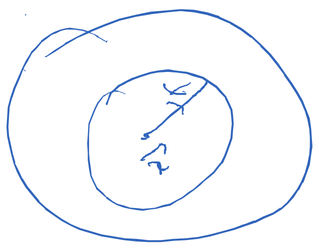
capacity (p)



Interpretation of H(x)

For $\vec{x} \in \{0,1\}^n$

let $S(n,t) = \{ \vec{y} : d_H(\vec{y},\vec{x}) = t \}$



$$|S(n,t)| = \binom{n}{t}$$

as $n \to \infty$

$$|S(n,pn)| \approx 2^{nH(p)}$$

more precisely

can be shown using stirling approximation ← $|S(n,pn)| = 2^{nH(p) + \underbrace{o(n)}_{\text{sublinear fn. of } n}}$

$$\lim_{n \to \infty} \frac{o(n)}{n} = 0$$

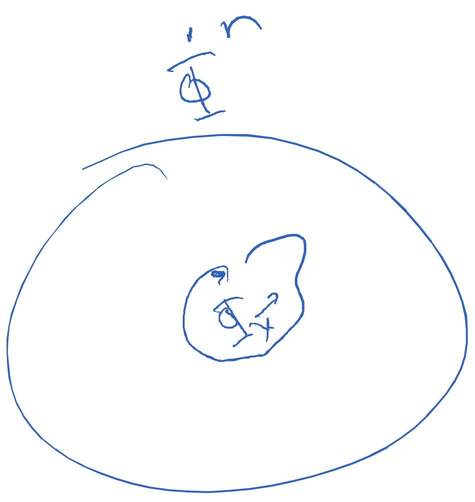
Adversarial channels

$$\underset{\in \mathbb{F}^n}{\vec{x}} \longrightarrow \boxed{\text{channel}} \longrightarrow \vec{y} \in \Phi^n$$

For every $\vec{x}$, define a set

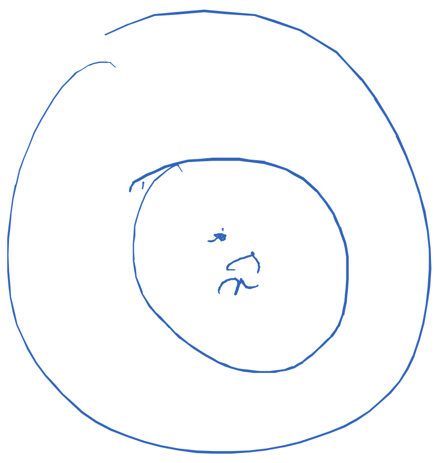
$\Phi_{\vec{x}} \subseteq \Phi^n$

↳ set of possible output strings given input was $\vec{x}$



Example

t - error channel



Define

$B(\vec{x}, r) = \{ \vec{z} \in \mathbb{F}^n : d_H(\vec{z},\vec{x}) \leq r \}$

Ball of radius r centered at $\vec{x}$

t - error channel has

$$\Phi_{\vec{x}} = B(\vec{x}, t)$$