2

20th class

Source coding: (a) fixed-fixed length (typical sets - formally not covered)
(b) Arithmetic coding, Lempel Ziv algorithm (compression without knowing P_X)
Runlength coding

Zip algorithm

So far we discussed this $X \in \mathcal{X} \xrightarrow{\text{coded}} \mathcal{C}$

It can be naturally mapped to

$\underline{X} \in \mathcal{X}^n \xrightarrow{\text{Source code}} \mathcal{C}$

$$|\mathcal{X}^n| = |\mathcal{X}|^n$$

For $\underline{x} = (x_1, \ldots, x_n)$, $x_i \in \mathcal{X}$,

$$P_{\underline{X}}(\underline{x}) = \prod_{i=1}^{n} P_X(x_i)$$

$\underline{x} \in \mathcal{X}^n$, $x \in \mathcal{X}$

We just do source coding in the alphabet $\mathcal{Y} = \mathcal{X}^n$

with prob distribution $P_{\underline{X}}(\underline{x}) \triangleq \prod_{i=1}^{n} P_X(x_i) \rightarrow (1)$, $\underline{x} \in \mathcal{Y}$

Recall that $\bar{L}(X)_{\text{Shannon-Fano}} < H(X) + 1$

Also $H(X) \le \bar{L}(X)_{\text{Huff}} \le \bar{L}_{\text{Sh.Fano}}(X) < H(X) + 1 \rightarrow (2)$

Huff: Optimal

For $\underline{X} = (X_1, \ldots, X_n) \in \mathcal{X}^n$ with dist as in (1),

$$H(\underline{X}) \le \bar{L}_{\text{Huff}}(\underline{X}) \le \bar{L}_{S\text{-}F}(\underline{X}) < H(X_1, \ldots, X_n) + 1$$

$H(\underline{X}) = nH(X)$

$$= \sum_{i=1}^{n} H(X_i) + 1$$

$$= nH(X) + 1$$

$$H(X) \le \frac{1}{n} \bar{L}_{\text{Huff}}(\underline{X}) \le \frac{\bar{L}_{S\text{-}F}(\underline{X})}{n} \le H(X) + \frac{1}{n} \rightarrow (3)$$

Comparing with (2), this upper bound in (3) is better.
(avg no of bits required to represent the source X.)

Shannon's Source coding theorem (achievability)

As $n \to \infty$, we see that

$$\frac{\bar{L}_{Huffman}(X)}{n}, \frac{\bar{L}_{SF}(X)}{n} \to \underline{H(X)}$$

$\Rightarrow$ Optimal compression (Min possible length required to represent X) is achieved by either of the 2 schemes asymptotically in n.

Converse: no matter what code we use, avg no of bits / source symbol $\geq H(X)$

As n grows, the algorithm for compress is complicated (Complexity $\approx$ No of operations $\approx$ exponential in n, this is very bad)

To counter this, use 'streaming compression' like Run length coding, Lempel Ziv algo etc.

Channel Coding:

Prob that $Y = y$ gn input $X = x$

$X \in \mathcal{X} \to$ [$p(y|x)$ Channel] $\to Y \in \mathcal{Y}$

channel code

Pick $\mathcal{C} \subseteq \mathcal{X}^n$ & transmit sequence only from $\mathcal{C}$. (instead of $\mathcal{X}^n$)

$\to$ We hope to reduce the prob of error

$(y_1, \dots, y_n)$ $(x_1, \dots, x_n)$

$$p(\underline{y} | \underline{x}) = \prod_{i=1}^{n} \underline{p(y_i | x_i)}$$

Prob that $\underline{Y} = \underline{y}$ gn $\underline{X} = \underline{x}$

Prob that $Y_i = y_i$ gn $X_i = x_i$

Tx side | Rx side

$c \in \mathcal{C} \subseteq \mathcal{X}^n \to$ [$p(\underline{y}|\underline{c})$] $\xrightarrow{\underline{Y}}$ [Decoder $g(\cdot)$] $\to g(\underline{Y})$ = estimate of $\hat{\underline{c}}$.

$P_{error}(\underline{c}) = P(\hat{c} \neq \underline{c})$ where $\hat{c}$ is the estimate for $\underline{c}$ derived by $\underline{y}$ (ch o/p) which is obtained from the channel when $\underline{c}$ is transmitted.

$= P(g(\underline{y}) \neq \underline{c})$

→ Our code $\mathcal{C}$ must be chosen so that the $P_{error}(\underline{c})$ is small, $\forall \underline{c} \in \mathcal{C}$

→ Also we want $R = \dfrac{\log_2 |\mathcal{C}|}{n}$ bits.

Shannon's Channel Coding theorem:

Converse: The rate of any code $\mathcal{C}$ (in the above channel) which has $P(error) \to 0$, should satisfy

$$R < \underbrace{\max_{P_X} I(X;Y) = C}_{\text{Channel Capacity.}}$$

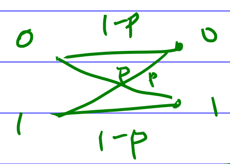
Achievability: For any $\epsilon > 0$, $\exists$ some 'sequence of channel codes' (one for every n) with rate $R = C - \epsilon$, such that

Vanishing prob of error ← $\left(P_{(error)} \to 0 \text{ as } n \to \infty. \right)$

→ (P_{error} decreases exponentially with n)

($P_{error} \approx 2^{-n\delta}$).

Previously Seen: For Binary Symmetric Channel



For BSC, we say that

$p(y|x) = p$ if $y \neq x$

① $C_{BSC} = \max_{P_X} I(X;Y) = 1 - H_2(p)$

Binary entropy $H_2(p) \triangleq p\log\frac{1}{p} + (1-p)\log\frac{1}{1-p}$

(2) We also saw converse to the S-Channel Coding thm for BSC.

That means we showed that for any code $\mathcal{C}$ with negligible prob of error, the rate R of the code has to satisfy $R < C_{BSC} = 1 - H_2(p)$.

—— x ——

Achievability : Class 21

for BSC, achievability part of Channel Coding theorem is stated as follows.

For any $\epsilon > 0$, there exists a sequence of codes $\mathcal{C}_n$ (one for each value of n) with rate $R = 1 - H_2(p) - \epsilon$ and & same decoding function such that $P(\text{error}) \to 0$ as $n \to \infty$

This doesn't change as n changes.

(more specifically we will say, for some constant $\delta > 0$)

$$P(\hat{\underline{c}} \neq \underline{c}) \leq 2^{-n\delta} \text{ for each } n; \text{ \& for each } \underline{c} \in \mathcal{C}_n$$

estimate of tx codeword

transmitted codeword (particular $\underline{c} \in \mathcal{C}$)

For any specific tx codeword $\underline{c}$, Randomness lies here in the estimate because decoder's input = rx vector is a random vector chosen according to distribution $p(\underline{y}|\underline{c})$

Proof argument of achievability (for BSC):

We will use a 'random' code.

Entire argument is for a fixed value of n. We will assume n is large (so that no of flips is $\approx np$)

We will pick a code $\mathcal{C}_n$ of rate $R = 1 - H_2(p) - \epsilon$.

So we want $|\mathcal{C}_n| = 2^{nR}$ [assume that nR is an integer, which will be true for large enough n an rational R].

$\mathcal{C}_n$ = set of vectors we picked in this process

Random code construction: ① Pick each codeword in the code $\mathcal{C}_n$ from $\{0,1\}^n$ uniformly at random

(n-length sequences over $\{0,1\}$)

$\Rightarrow P($ codeword = any specific n-length seq in $\{0,1\}^n) = \frac{1}{2^n}$.

② Repeat the process 2^{nR} times.

Repeat the ①② steps until we get $|\mathcal{C}_n| = 2^{nR}$

Now we want to prove $P(\hat{\underline{c}} \neq \underline{c}) \leq 2^{-n\delta}$ . . . . . . for every $\underline{c} \in \mathcal{C}_n$ for some $\delta > 0$.

For getting a handle on the probability of error, we need to specify how the decoding function is defined (i.e how is estimate $\hat{c}$ calculated given a particular received vector)

Let the decoding function be denoted by

$$D : \underbrace{\{0,1\}^n}_{\text{domain of decoding fn}} \longrightarrow \underbrace{\mathcal{C}_n}_{\text{codomain}}$$

We will define the fn D as follows

For any $y \in \{0,1\}^n$,

$$D(\underline{y}) \triangleq \underset{\underline{c}' \in \mathcal{C}_n}{\operatorname{argmax}} \; p(\underbrace{\underline{y}}_{\text{Output}} \mid \underbrace{\underline{c}'}_{\text{Inp}})$$

('break the ties arbitrarily')

This can be found at decoder! because $p(\underline{y} \mid \underline{c})$ is known to decoder $\forall\, \underline{y}, \underline{c}$.

$\boxed{D(\underline{y}) \text{ is the estimate } \hat{\underline{c}} \text{ of the trx codeword when received vector is } \underline{y}}$

↳ "Maximum Likelihood Decoding Rule"

To show that $P(\hat{\underline{c}} \neq \underline{c})$ is small, we will show that

$P(D(\underline{y}) \neq \underline{c})$ is small where $\underline{y}$ is the random

received vector when $\underline{c}$ is transmitted.

Note

$$p(\underline{y} \mid \underline{c}') = p((y_1 \cdots y_n) \mid (c_1', \ldots, c_n'))$$

$$= \prod_{i=1}^{n} p(\underbrace{y_i}_{i\text{th op bit}} \mid \underbrace{c_i'}_{i\text{th input bit}}) \quad \text{[We assume this true]}$$

$$= p^{d_H(\underline{y}, \underline{c}')} (1-p)^{n - d_H(\underline{y}, \underline{c}')} \qquad \begin{cases} = p & \text{if } y_i \neq c_i \\ = 1-p & \text{if } y_i = c_i \end{cases}$$

$$p(\underline{y} \mid \underline{c}') = \left(\frac{p}{1-p}\right)^{d_H(\underline{y}, \underline{c}')} (1-p)^n \Rightarrow (A)$$

$d_H(\underline{y}, \underline{c}') =$ no of positions where $\underline{y}$ & $\underline{c}'$ differ

If $p < 0.5$,
By (A) max $p(\underline{y} \mid \underline{c}')$ across all $\underline{c}' \in \mathcal{C}_n$ is same as

minimizing $d_H(\underline{y}, \underline{c}')$ '' '' '' '' '' '' '' ''

$$\Rightarrow \hat{\underline{c}} = \underset{\underline{c}' \in \mathcal{C}}{\operatorname{argmin}}\ d_H(\underline{y}, \underline{c}') \rightarrow \text{Minimum Hamming distance decoder}$$

Class 22

Continuing proof of achievability

For the above decoder, we want to show $P(\hat{\underline{c}} \neq \underline{c}) \leq 2^{-n\delta}$ (for $\delta > 0$) $\forall \underline{c}$

For any specific $\underline{c} \in \mathcal{C}_n$ as the transmitted codeword,

$P(\hat{\underline{c}} \neq \underline{c}) \rightarrow$ we want to find out an upper bound for this

random

$\hat{\underline{c}} \neq \underline{c}$ occurs when $\underline{c}$ is not the closest codeword to $\underline{y}$.

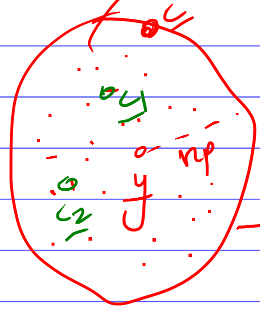
But by Law of large numbers, (To be done in problem)

$$d_H(y, \underline{c}) \approx np$$

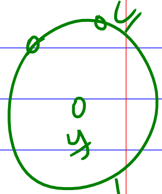
→ If there are other codewords within this ball $B(y, np)$, then decoder can make an error.

each dot here is codeword

Hamming ball around y of radius np



$\boxed{\text{If there are no other codewords in this ball, decoder will not make an error}}$



all codewords except c lie outside the ball ⇓ MDD will decode correctly.

$$B(y, np) = \{ x \in \{0,1\}^n : d_H(x, y) \le np \}$$

$$P(\underbrace{\hat{c} \neq c}_{\text{actual transmitted codeword}}) \le P(\exists\, \underline{c}' \in B(y, np) : \underline{c}' \neq \underline{c})$$

$$\le \frac{|B(y, np)|}{2^n}$$

(this is because the codewords are picked uniformly at random)

$$\le \frac{\sum_{i=0}^{np} \binom{n}{i}}{2^n} = \frac{\binom{n}{np} + \sum_{i=0}^{np-1} \binom{n}{i}}{2^n} \rightarrow ①$$

$$\binom{n}{0} + \binom{n}{1} \cdots + \binom{n}{np}$$

$np < \frac{n}{2}$ (we assume $p < 0.5$, otherwise we can change the channel to $BSC(p')$ where $p' = 1 - p$)

As n grows large, this value is dominated by the first term which is $\binom{n}{np}$

$$\text{RHS of } ① \approx \frac{2^{nH_2(p)}}{2^n}$$

(as $n \uparrow$, $\binom{n}{np} + \cdots + \binom{n}{0} \approx 2^{nH_2(p)}$)

$$\approx 2^{-n(1 - H_2(p))}$$

By ① ⇒ $P(\hat{c} \neq c) \le 2^{-n(1 - H_2(p))}$ → This is for a specific codeword $\underline{c} \in \mathcal{C}_n$ → Ⓐ

We want to show that for all codewords simultaneously (A) has to hold

We want $P\left(\bigcup_{\underline{c}\in\mathcal{C}_n}\left(\hat{\underline{c}}\neq\underline{c}\right)\right)\leq 2^{-n\delta}$ for some $\delta>0$,

Note that if $P\left(\bigcup_{\underline{c}\in\mathcal{C}_n}(\hat{\underline{c}}\neq\underline{c})\right)\to 0$ then $P(\hat{c}\neq c)\to 0$ $\forall\, \underline{c}\in\mathcal{C}_n$

Now note: (Union bound)

WKT: $P\left(\bigcup_{\underline{c}\in\mathcal{C}_n}(\hat{\underline{c}}\neq\underline{c})\right)\leq\sum_{\underline{c}\in\mathcal{C}_n}P(\hat{\underline{c}}\neq\underline{c})$

we have bound in (A) on this

$$\leq\sum_{\underline{c}\in\mathcal{C}_n}\underline{\underline{2^{-n(1-H_2(p))}}}$$

this value doesn't depend on $\underline{c}\in\mathcal{C}_n$

$$\leq 2^{nR}\cdot 2^{-n(1-H_2(p))}$$

$$\leq 2^{-n(1-H_2(p)-R)}$$

$$P\left(\bigcup_{\underline{c}\in\mathcal{C}_n}(\hat{\underline{c}}\neq\underline{c})\right)\leq 2^{-n\epsilon}$$

[as our rate $R=1-H_2(p)-\epsilon$] (ϵ was given constant >0)

$\Rightarrow P(\hat{\underline{c}}\neq\underline{c})\leq 2^{-n\epsilon}$ for all $\underline{c}\in\mathcal{C}_n$.

$\hookrightarrow$ Hence proved //

— x —

In practice using Random codes + MDD (or MLD) for BSC is very complex (complexity of encoder / decoder is extremely large (extremely large $=\exp(n)$))

In practice, we use structured (deterministic) codes which have low encoding/decoding perform → Most interesting class of structured codes called "linear codes"

→ Difficult problem for BSC channel (not considerably solved today)

↓

get linear codes of small probability of error & rate close to capacity