||Quantum Information Theory|| (Rough Notes)
from: John Preskill's Notes (Further shortened)

"Defn":
Quantum Information: It deals with (a) Transmission of classical info over quantum channels
(b) Tradeoff b/w acquisition of inform^n about a Quantum system & disturbing it.
(c) Quantifying Quantum Entanglement.
(d) Transmission of quantum info over quantum channels.

Remark: These themes are united by: Interpretation & applications of the Von Neumann entropy.

§5.1 Shannon for Dummies

(1) How much can a message be compressed ("noiseless coding thm") i.e. how redundant is the information.
(2) At what 'rate' can we communicate over a noisy channel; how much redundancy to add to protect against errors. (The "noisy channel coding thm")

Key: Entropy suitable quantification

§5.1.1 Shannon Entropy & Data Compression.

Consider: A message is a string from the letter set

$$\{a_1, a_2 \ldots a_k\}$$

& each letter occurs with "apriori" probability $p(a_x)$ independently.

NB: $\#$ typical messages $\sim \dfrac{n!}{\prod_x [n p(a_x)]!} \overset{\text{demand}}{\simeq} 2^{-nH(a_x)}$

$\because$ Typically, the letter a_x occurs $n.p(a_x)$ times.

Claim: Using Sterling approx. $H(X) = \sum_x -p(x) \log p(x)$ "$\langle \log p(x) \rangle$"

(I changed a_x to x for simplicity)

Remark: See full notes for ES & SS.

§ 5.2.1 Mutual Information

NB: $H(X)$ quantifies info ∵ it tells us # bits we need to encode (we encode only typical messages)

Motivation: Have a noisy device. You input x, it sends me y. How much information did I gain about x after learning y?

Formally: I know characteristics of the machine/channel
$p(y|x)$
I know the apriori probabilities of the letters
$p(x)$
⇒ I can compute $p(y) = \sum_x p(y|x)\, p(x)$
I therefore have $p(x|y) = \dfrac{p(y|x) \cdot p(x)}{p(y)}$

NB: Once I know ys, you must send typically
$$H(X|Y) := \langle -\log p(x|y) \rangle$$
bits for each letter of the n-bit string, x.

NB: $\because\ p(x|y) = \dfrac{p(x,y)}{p(y)}$

(a) $H(X|Y) = \langle -\log p(x,y) + \log p(y) \rangle$
$= H(X,Y) - H(Y)$

(b) $H(Y|X) = \langle -\log p(y|x) \rangle$
$= H(X,Y) - H(X)$

Intuition: $H(X|Y)$ is the # bits more, required to specify X & Y, if X is known.

Def^n: (Motivation: Information I gain about X is avg # bits needed to specify X minus avg. # bits needed to specify X after I learnt Y)

: $I(X ; Y) \equiv H(X) - H(X|Y)$
$= H(X) + H(Y) - H(X,Y)$
$= H(Y) - H(Y|X)$

: Mutual Information

Remark: I learn as much about X by learning Y as I learn about Y by learning X.
(symmetry)
∴ Learning Y can't reduce my knowledge of X; so
$I(X;Y) \geq 0$

Claim: $H(X) \geq H(X|Y) \geq 0$ (can be proved using convexity of log).

NB: For X Y uncorrelated

$$I(X;Y) = \left\langle \log \frac{P(x,y)}{P(x)P(y)} \right\rangle = 0$$

as expected.

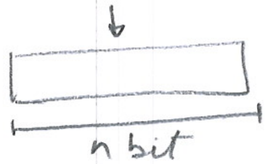
§ 5.1.3 The noisy channel coding theorems

Question: How many bits per letter are needed to send a message (code length $n \to \infty$) with arbitrary reliability?

Eg.: Given: Letters $\{0, 1\}$ $p(0) = p(1) = \frac{1}{2}$

Channel: Binary symmetric: $P(0|0) = 1-p$ | $P(0|1) = p$
$P(1|0) = p$ | $P(1|1) = 1-p$

Objective: Encode k bits



Have 2^k codewords for 2^n strings, s.t. we transmit reliably.

∴ Def^n: Rate := $R = \frac{k}{n}$

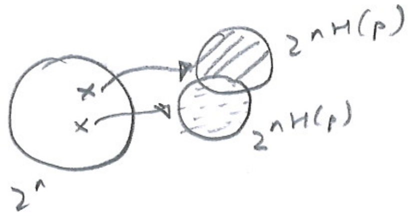
Sol^n: NB: An n-bit input string (from 2^n possible strings) will get mapped to a set of typical strings by the channel. These will be $2^{nH(p)}$ in number (recall: typically np bits get flipped: that gives $2^{nH(p)}$ # of typical strings)

Rationale: We would want our codeword to not change with anything less than np flips.

If I assume the same keyword for each such set without an overlap, then the best I can do is have (see the figure)

Concl^n: $2^k \cdot 2^{nH(p)} < 2^n$ $\therefore 2^{nR} 2^{nH(p)} \leq 2^n$

$\Rightarrow R \leq 1 - H(p)$



pick a set of size 2^k s.t. $2^k \cdot 2^{nH(p)}$ can be fit into the # of distinct messages allowed by n bits.

Def^n: $C(p) := 1 - H(p)$: Channel Capacity

NB: We can't expect our error rate to be better than $C(p)$.

Question: Is $C(p)$ achievable?

Remark: The set of elements at a "distance" np are referred to as a "Hamming sphere".

Constr^n: Assume : 2^{nR} codewords are chosen at random (from 2^n elements)

: To decode a message, draw a "hamming sphere" of radius $nH(p) + \delta$.

Claim : The "Hamming Sphere" will contain only $\underline{1}$ codeword (on an average) essentially.

"Proof" : Fraction of strings inside the Hamming Sphere : $\dfrac{2^{n(H(p)+\delta)}}{2^n} = 2^{-n(C(p)-\delta)}$

(We neglect the possibility of no keywords $\because$ the sphere is quite large)
(Not typical for this to happen)

Prob. of accidental occurance of an additional codeword in the sphere $= 2^{-n(C(p)-R-\delta)}$

(That's the prob. of making an error).

So we can choose R as close to C while the error dissappears in the $n \to \infty$ limit.

NB: We showed on an average (over the codewords) the errors should be less than say ϵ.

Claim: We can choose keywords so that error for each keyword is less than ϵ.

"Proof": We know $\frac{1}{2^{nR}} \sum P_i < \epsilon$ where P_i is the prob of errors in decoding a keyword numbered i.

: L. Def^n: $N_{2\epsilon}$:= # keywords with error $> 2\epsilon$ each. $(P_i > 2\epsilon)$

$\Rightarrow \frac{1}{2^{nR}} N_{2\epsilon} \cdot 2\epsilon < \epsilon$ (see rough if confused)

$\Rightarrow N_{2\epsilon} < (2^{nR})/2$

: Concl^n: We need to throw, worst case, half the codewords to achieve $P_i < 2\epsilon$ $\forall$ remaining codewords!

NB: $2\epsilon = 2^{-n(C(p) - R - \delta)} \Rightarrow R = C(p) - \frac{1}{n}$ for the new errors.

Result: We can achieve the rate $C(P) = 1 - H(p)$ (asymptotically) with an arbitrarily small error.

† General Constr^n: Given: $p(y|x)$ for the channel
$X = \{x, p(x)\}$ for the letters
Send n letters
assume: channel acts independently on each bit. ("memoryless channel")

: Choose a random codeword set from X^n. Each of these will typically be from the set of typical strings. This would be of size $2^{nH(x)}$.

NB: For a typical message in Y, aprox. $2^{nH(X|Y)}$ messages could've been sent.

: To decode: associate a sphere with Y containing $2^{n(H(X|Y)+\delta)}$ inputs

case (0): no codeword; atypical
case (i): $\exists$ a unique keyword/codeword; done.

case (b): prob. of more than 1 codeword is small
claim:
proof: $\dfrac{2^{n(H(X|Y)+\delta)}}{2^{nH(X)}}$ = fraction of strings in the decoding sphere (typically) of the input strings

$= 2^{-n(H(X|Y)-H(X)+\delta)}$

$= 2^{-n(I(X;Y)-\delta)}$

Prob. of a codeword accidentally falling in the sphere $= 2^{nR}\,2^{-n(I(X;Y)-\delta)} = 2^{-n(I-R-\delta)}$

concl^n: We can get, at best $R \simeq I(X;Y)$

Remark: I is the inform^n per letter that can be sent.

: Results about all errors $< \epsilon$ etc can be proven same as before.

Def^n: Channel Capacity $:= C = \max\limits_{\{p(x)\}} I(X;Y)$

NB: $P(y|x)$ defines the channel.

Remark/Motivation: We haven't shown we can't do better than C.

Claim: C is an upperbound to the attainable rate.

"Proof": Assume: 2^{nR} strings are our codewords.

Consider: A prob dist^n. $\tilde{X}^n$ s.t. each codeword is equi-probable (i.e. 2^{-nR})

NB: $H(\tilde{X}^n) = nR$

Assume: We send these codewords through the channel & obtain $\tilde{Y}^n$.

NB: $p(y_1 y_2 \cdots y_n | x_1 \cdots x_n) = \prod\limits_i p(y_i|x_i)$

$\because$ The channel acts independently on each letter.

$\therefore$ $H(\tilde{Y}^n|\tilde{X}^n) = \langle -\log P(y^n|x^n)\rangle$

$= \sum\limits_i \langle -\log(p_i|x_i)\rangle$

$= \sum\limits_i H(\tilde{Y}_i|\tilde{X}_i)$

Not clear:

where 2-Def^n: $\tilde{X}_i$ $\tilde{Y}_i$ are marginal distr for the i^{th} letter.

(from $\tilde{X}^n$ & $\tilde{Y}^n$)

$\because$ Recall: $H(X,Y) \le H(X) + H(Y)$

$\Rightarrow H(\tilde{Y}^n) \le \sum_i H(\tilde{Y}_i)$

NB:
$$\begin{aligned} I(\tilde{Y}^n;\tilde{X}^n) &= H(\tilde{Y}^n) - H(\tilde{Y}^n|\tilde{X}^n) \\ &\le \sum_i \left[H(\tilde{Y}_i) - H(\tilde{Y}_i|\tilde{X}_i)\right] \\ &= \sum_i I(\tilde{Y}_i;\tilde{X}_i) \le nC \end{aligned}$$

$\because$ def^n of C

NB:
$$\begin{aligned} I(\tilde{X}^n;\tilde{Y}^n) &= I(\tilde{Y}^n;\tilde{X}^n) \\ &= H(\tilde{X}^n) - H(\tilde{X}^n|\tilde{Y}^n) \\ &= nR - H(\tilde{X}^n|\tilde{Y}^n) \le nC \end{aligned}$$

Final Argument: For zero error as $n \to \infty$, we must have either (a) the input codeword is determined by the output $H(\tilde{X}^n|\tilde{Y}^n) = 0$ effectively

or (b) $\frac{1}{n} H(\tilde{X}^n|\tilde{Y}^n) \to 0$ for $n \to \infty$.

NB: $\frac{1}{n} H(\tilde{X}^n|\tilde{Y}^n) \to 0 \not\Rightarrow \frac{1}{n}(\tilde{Y}^n|\tilde{X}^n) \to 0$

We may be able to decode perfectly $\not\Rightarrow$ There's no uncertainty in the channel.

$\therefore$ In any case, $R \le nC$

regardless of how one constructs the codes & the decoding scheme.

Rough

$2^{nH(p)}$

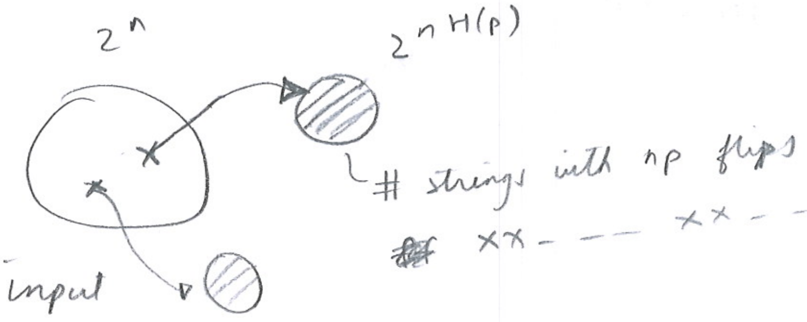
$$\frac{n!}{\prod np} =$$

$$\frac{1}{2^{nR}} \sum P_i < \epsilon$$

$$\frac{N \cdot \epsilon}{N} = \epsilon$$

$$\frac{\left(\frac{N}{2}\right) 2\epsilon}{N} = \epsilon$$

n!



$N_\epsilon \cdot P_i$

$$N_\epsilon \cdot 2\epsilon < \sum_{N_\epsilon} P_i$$

N_ϵ have $P_i > 2\epsilon$

Imagine all others have $P_i = 0$

$$\frac{(N_\epsilon \cdot 2\epsilon)}{N} < \epsilon$$