Tuesday, November 10, 2020 6:00 PM

DMS transmitting n symbols from N-length dictionary X having $f(x) = \begin{cases} p_1, & x = a_1 \\ \vdots & \vdots \\ p_N, & x = a_N \end{cases}$, need $nH(X)$ bits to rep. message of n symbols

$\longrightarrow$ $\boxed{H(X) \text{ bits per symbol}}$

Ex. X has N equiprobable outputs. How many effective outputs in an n-length transmission?

from last week $\rightarrow$ $2^{nH(X)} = 2^{-n\sum_{i=1}^{N} \frac{1}{N}\log\frac{1}{N}} = 2^{-n\log 1/N} = 2^{n\log N}$

$= N^n$

= total # of outputs

$\longrightarrow$ uniform = max entropy $\Rightarrow$ you save nothing by knowing $H(X)$

$\longrightarrow$ cannot compress to smaller # effective outputs

[illegible] last week: $H(X) \leq \log N \;\; \forall X$

BUT result from last week: $H(X) \leq \log N \ \forall X$ w/ N-length support

So it's never worse than this

Uniform = worst case scenario

Briefly - if not memoryless, then each symbol depends on past, sequence of RVs (discrete-time random process)

and $H \equiv \lim_{n \to \infty} H(X_n | X_1, \ldots, X_{n-1})$ entropy rate

then - # effective outputs $= 2^{nH}$

H bits/symbol needed

Theorem (Source Coding): A source w/ entropy (or entropy rate, if not memoryless) H can be encoded with an arbitrarily small error probability at any rate R (bits/symbol) s.t.

$$R > H$$

Conversely, if $R < H$, the error will not approach 0.

H(X) not even necessarily an integer (or even rational)

R more-or-less has to be (not exactly)

$H(X)$ not even necessarily an integer (or even rational)

R more-or-less has to be (not exactly)

— constrained to rational #s

in practice, $R \neq H$

How do we encode a DMS output in bits s.t. R is near H

Source Coding Algorithms

Famous/practical example: Huffman coding

an example of a Variable-length code (VLC) — each symbol gets a binary codeword, but not necessarily same length

length is a function of symbol probability

Necessary that any message is uniquely Decodable!

ex.

$a \to 0$

$b \to 1$ I receive message 101

$c \to 01$

is this bc or bab?

don't know! not uniquely decodable!

$a \to 1$

$b \to 01$

$c \to 00$

— satisfies prefix condition

Prefix condition: No codeword "is the start" of another codeword

Prefix condition No codeword "is the start" of another codeword

average word length should be approximately $H = -\sum p_i \log p_i$

Huffman

optimal code in that it gives codewords w/ the **smallest average length** that satisfies prefix condition

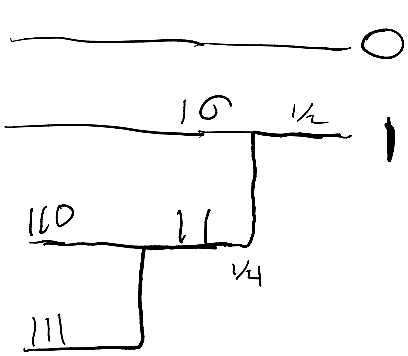
Algorithm

1. Sort symbols in order of probability
2. Merge the least probable 2 symbols into a single output, and repeat until there are only 2
3. assign 1 and 0 to each of the two outputs (arbitrarily)
4. Work backwards, unmerging and appending 0 and 1 to each pair until each symbol has a codeword

on the right

Ex. 4-PSK, $\{a_1, a_2, a_3, a_4\} \xrightarrow{\text{probs}} \{\overbrace{1/2}^{p_1}, \overbrace{1/4}^{p_2}, \overbrace{3/16}^{p_3}, \overbrace{1/16}^{p_4}\}$

Symbol	prob
a_1	1/2
a_2	1/4
a_3	3/16
a_4	1/16



$a_1 \to 0$

$a_2 \to 10$

$a_3 \to 110$

$a_4 \to 111$

Kraft

$\frac{1}{2} + \frac{1}{4} + \frac{1}{4} = 1$

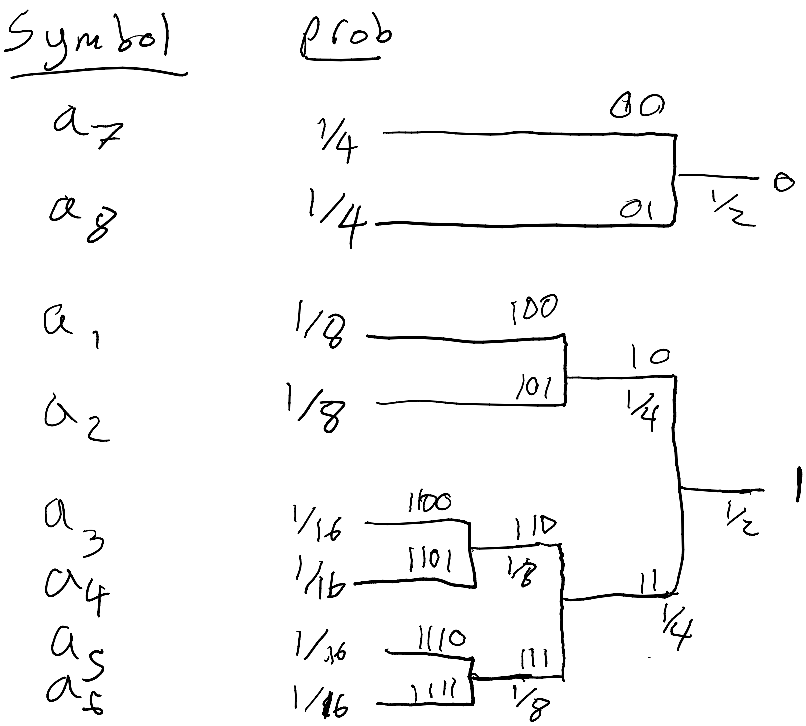
avg length $\bar{L} = 1 \cdot p_1 + 2 \cdot p_2 + 3(p_3 + p_4)$

$= 1/2 + 1/2 + 3/4 = 1.75 = \bar{L}$

$H = -\sum p_i \log p_i \approx 1.7$, pretty good!

<u>Ex</u> 8-ary $\{a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8\}$

probs $\{1/8, 1/8, 1/16, 1/16, 1/16, 1/16, 1/4, 1/4\}$



$a_1 \to 100$
$a_2 \to 101$
$a_3 \to 1100$
$a_4 \to 1101$
$a_5 \to 1110$
$a_6 \to 1111$
$a_7 \to 00$
$a_8 \to 01$

<u>Kraft</u>

$2 \cdot 2^{-3} + 4 \cdot 2^{-4} + 2 \cdot 2^{-2}$

$= \frac{1}{4} + \frac{1}{4} + \frac{1}{2}$

$= 1$

$\bar{L} = 3(1/8) + 3(1/8) + \frac{4}{16}(4) + \frac{2}{4} \cdot 2 = 2.75$ bits/symbol

$H(X) = \frac{1}{8}\log_2 8 + \frac{1}{8}\log_2 8 + \frac{4}{16}\log_2 16 + \frac{2}{4}\log_2 4$

$= 3/8 + 3/8 + 16/16 + 4/4 = 2.75 = \bar{L}$

here, we matched optimum length 

note: NBC or gray coding would give

$l_i = 3$, $i = 1, \ldots, 8$, so $\bar{L} = 3 > H = \bar{L}_{Huffman}$

<u>Result I won't prove:</u>

Huffman satisfies: $H(X) \le \bar{L} \le H(X) + 1$

Now extending Huffman to sequences of n symbols (same algorithm) $\rightarrow$ Huffman satisfies

$$H(X^n) \le \bar{L} \le H(X^n) + \frac{1}{n}$$

So as $n \rightarrow \infty$, $\bar{L} \rightarrow H(X^n)$ ✓

Last result: All uniquely decodable VLCs satisfy

$$\boxed{\sum_{i=1}^{N} 2^{-l_i} \le 1}$$ Kraft inequality

l_i is length of each word

Noise! (sorry)

Tuesday, November 10, 2020 7:21 PM

Digital Channels

- I transmit either a 1 or a 0,
- I receive either a 1 or a 0, only so many ways to be wrong!

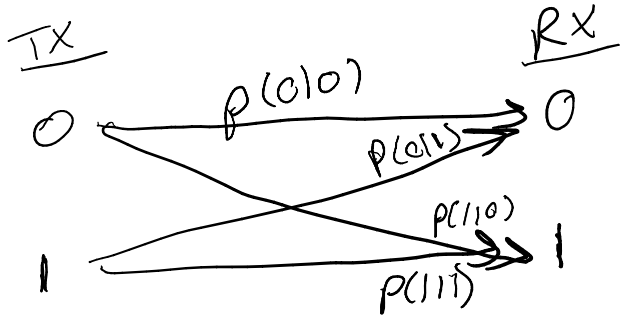
an error can come from ISI, AWGN, etc.

we only care here about the discrete error

Discrete Memoryless Channel (DMC) (model)

- The probability of an error for any given bit is indep. of other bits.

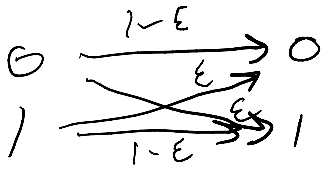
Governed by 2 parameters: $P(0|1) = 1 - P(1|1)$
$P(1|0) = 1 - P(0|0)$



$p(0|0) \equiv$ true zero
$p(1|0) \equiv$ false one
$p(0|1) \equiv$ false zero
$p(1|1) \equiv$ true one

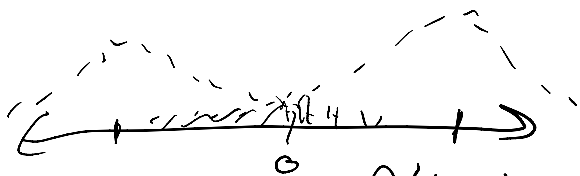
Special case: 1-Parameter Binary Symmetric channel (BSC)

Special case: 1-parameter binary symmetric channel (BSC)



"crossover probability" ε

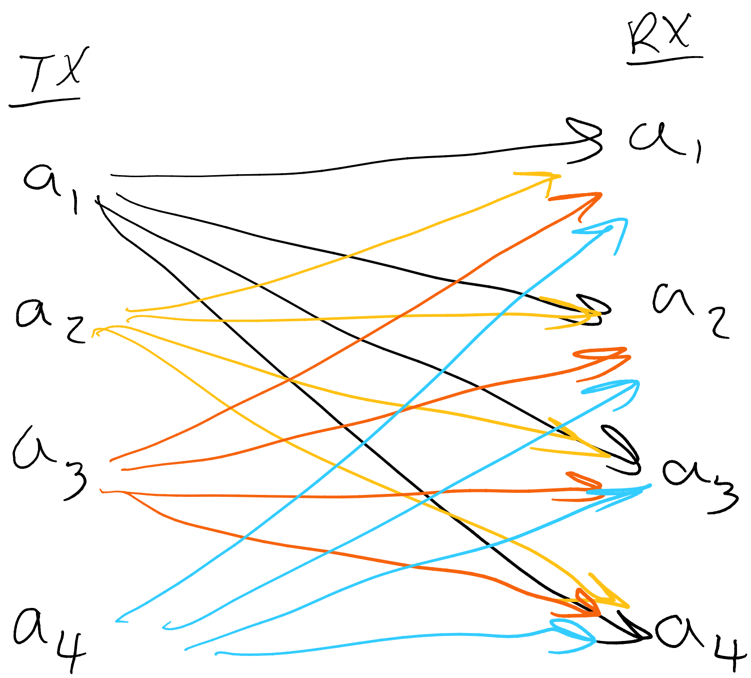
Ex: Binary antipodal under AWGN is BSC (if equiprob)



$$P(1|0) = P(0|1) = \varepsilon = Q\left(\sqrt{\frac{2E_b}{N_0}}\right)$$

Send many symbols, can consider a more complex

DMS



Still tractable model.

---

already: used info theory to give us the limit on the Compression rate (bits/symbol) for DMS w/ arbitrarily

Compression rate (bits/symbol) for DMS w/ arbitrarily smallerror

Remarkably — can also use info theory to find a limit on transmission rate in transmitting across DMC w/ arbitrarily smallerror.

→ Even w/ noise, can achieve entirely reliable transmission simply by transmitting slowly.

Ex.

scheme 1 $\quad R = 1\ b/sym$

$a_0 \to 0$

$a_1 \to 1$

scheme 2

$a_0 \to 000000 \quad R = 6\ b/sym$

$a_1 \to 111111$

Channel 1 BSC $\varepsilon = 0$

reliable every time

scheme 2 is wasteful

Channel 2 BSC $\varepsilon = 0.2$

Scheme one sees error 20% of time

scheme 2 is more robust to error

$000000 \xrightarrow{error} 010000$

$\xrightarrow{\text{nearest 0}} 000000$

allows for error correction

i.e. transmitting slower

allows for error correction
by transmitting slower

we want to formalize this idea
and find fastest we can transmit while still reliable

Consider a channel w/ input alphabet $X = \{x_1, \dots, x_m\}$

transmission probabilities $P[x_j | x_k]$, $k = 1, \dots, m$, $j = 1, \dots, m$

we define the n^{th} extension channel — the channel where

n symbols $(a_1, \dots, a_n) \in X^n$ are transmitted

probs: $\prod_{i=1}^{n} P[y_i | x_i]$

How many ways can input and output disagree at a locations?

$$\begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix} \rightsquigarrow \begin{pmatrix} \tilde{a}_1 \\ \vdots \\ \tilde{a}_n \end{pmatrix}$$

$a_i = \tilde{a}_i$ for $n-a$ vals of i

$a_j \neq \tilde{a}_j$ for a vals of j

this can occur in $\binom{n}{a}$ ways

as $n \nearrow$, assuming error occurs w/ prob. ε (symmetric channel)

then it is increasingly likely input and output disagree in precisely

$n\varepsilon$ locations

$\#$ ways to occur $= \binom{n}{n\varepsilon} = \frac{n!}{(n-n\varepsilon)!\,(n\varepsilon)!}$, Stirling: $\underbrace{\log N! \approx N\log N - N}_{\text{for large } N}$

So $\log_2 \binom{n}{n\varepsilon} = \log_2(n!) - \log_2((n-n\varepsilon)!) - \log_2((n\varepsilon)!)$

Stirling $\rightsquigarrow \approx (n\log n - n) - ((n-n\varepsilon)\log(n-n\varepsilon) - (n-n\varepsilon)) - (n\varepsilon \log n\varepsilon - n\varepsilon)$

$= n(\log n \cancel{-1} - \log(n-n\varepsilon) + \varepsilon\log(n-n\varepsilon) \cancel{+1} \cancel{-\varepsilon} - \varepsilon\log n\varepsilon \cancel{+\varepsilon})$

$= n(\log n - \varepsilon(\log n\varepsilon - \log(n-n\varepsilon)) - \log(n-n\varepsilon))$

$= n(\cancel{\log n} - \varepsilon(\cancel{\log n} + \log\varepsilon \cancel{-\log n} - \log(1-\varepsilon)) - \cancel{\log n} - \log(1-\varepsilon))$

$= n(-\varepsilon\log\varepsilon - (1-\varepsilon)\log(1-\varepsilon))$

$= n\,H_b(\varepsilon)$ binary entropy function

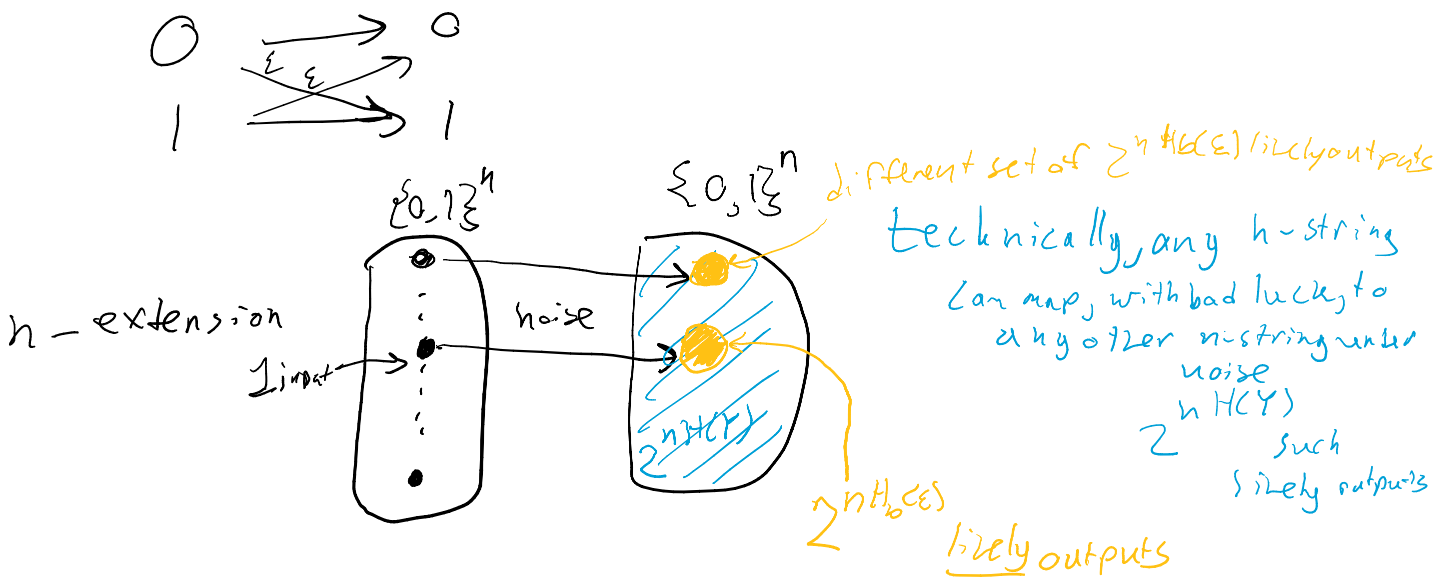
so $\binom{n}{n\varepsilon} \approx 2^{n H_b(\varepsilon)}$ = # probable outputs for any single input n-sequence

Source Coding: $2^{nH(Y)}$ "typical" outputs of a DMS Y

Source Coding: $2^{nH(Y)}$ "typical" outputs of a DMS Y

DMS $\longleftrightarrow$ DMC both governed in high n by entropy.

To understand better:



If I send only a subset of all possible n-strings the hope: there is no overlap between probable outputs

→ There will be error

→ each erroneous signal corresponds to only one input! (with prob → 1 as $n \to \infty$)

Can separate outputspace into M "error regions"

$$\frac{\#\text{ total likely outputs}}{\#\text{ likely for one input}} = \frac{2^{nH(Y)}}{2^{nH_b(\varepsilon)}} = 2^{n(H(Y)-H_b(\varepsilon))}$$

of the 2^n possible inputs, to achieve reliable comms, use only $2^{n(H(Y)-H_b(\varepsilon))} = M$. <u>Wasteful</u>, as we can

rep. w/ $H(Y)-H_b(\varepsilon)$ bits/symbol worse than would do w/ no noise

but in return → <u>reliable</u> w/ prob 1 as $n \to \infty$

$$\boxed{R = \frac{\log_2 M}{n} = H(Y) - H_b(\varepsilon)}$$ bits/sym for reliable comms

analogous to Source coding

---

to get the most out of this want M to be <u>big</u>

- most possible codewords per n bits

M max when $H(Y)$ is max ($H_b(\varepsilon)$ is fn of channel only)

→ $H(Y)$ is max when <u>$P[0]=P[1]=1/2$</u> → $H(Y) = \log 2 = 1$

so max $\boxed{R = 1 - H_b(\varepsilon)}$

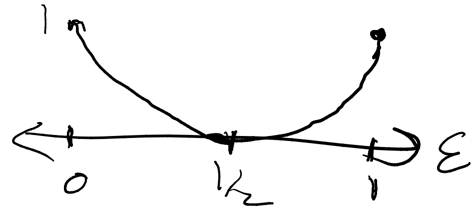
can't control this

can't control this
worst at 1/2

$R \le 1$, 1 only if $\varepsilon = 0$ or 1

max rate for reliable transmission for BSC is defined to be

$$\boxed{C_{BSC} = 1 - H_b(\varepsilon)}$$



"channel capacity"

More generally

Noisy Channel Coding THM: The channel capacity of a DMC is given by

$$C = \max_{f_X(x)} I(X;Y)$$

$$= \max_{f_X(x)} \left(H(Y) - H(Y|X) \right)$$

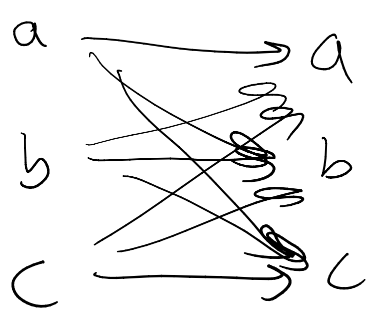
where $f_X(x)$ – the pmf of X which TX controls (not always uniform) if we transmit at

$R<C$, reliable comm <u>is</u> possible

but, a poor choice of $f_X(x)$ may make it IMPOSSIBLE

if we transmit at $R>C$, reliable comm is <u>never possible</u>

<u>Ex</u>



$P(a|a)=P(b|b)=P(c|c)=.5$

$P(b|a)=\ldots=P(a|c)=0.25$

<u>Find C</u>

$$I(X;Y)=H(Y)-H(Y|X)$$

$$H(Y|X)=\sum_{i=1}^{3} P[X=x_i]\,H(Y|X=x_i)$$

$$H(Y|X=a)=-\frac{1}{2}\log_2\left(\frac{1}{2}\right)-\frac{1}{4}\log_2\left(\frac{1}{4}\right)-\frac{1}{4}\log_2\left(\frac{1}{4}\right)$$

$$=1.5$$

$=H(Y|X=b)=H(Y|X=c)$ by symmetry

so $H(Y|X) = 1.5 \underbrace{\sum_i P[X=x_i]}_{\text{sums to } 1} = 1.5$

$I(X;Y) = H(Y) - 1.5$

to maximize $I(X;Y)$, need only to max.

$H(Y) \longrightarrow$ a r.v. w/ discrete support, so <u>uniform</u> is max entropy

$H(Y) = -\frac{1}{3}\log\frac{1}{3} - \frac{1}{3}\log\frac{1}{3} - \frac{1}{3}\log\frac{1}{3} = \log 3 \approx 1.585$

$C \approx 1.585 - 1.5 = 0.085$ bits/channel use

each bit of info requires $\lceil \frac{1}{C} \rceil = 12$ channel uses

$\longrightarrow$ If I Huffman code $a_1 \to 010$

then I must transmit 36 bits to ~~reliable~~ reliably send a_1

or $b \to 0$

I need <u>12</u> bits to tx reliably

<u>Important example</u> <u>Gaussian Channel Capacity</u>

I send $x_i \in X$, receive $x_i + z_i \in Y = X + Z$

$\uparrow$ AWGN, $\mu = 0$, $\sigma^2 = P_N$

for n large, $\frac{1}{n} \sum_i x_i^2 \le P$ (placing power restriction on X)

$$y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}, \quad x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}, \quad z = \begin{pmatrix} z_1 \\ \vdots \\ z_n \end{pmatrix}$$

n-xtension: $y = x + z \rightarrow z = y - x$

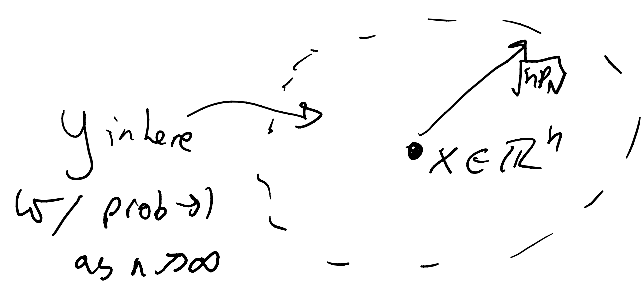
$$\frac{1}{n} \sum_i z_i^2 = \frac{1}{n} \sum_i |y_i - x_i|^2$$

$\downarrow$ sample var (n large)

$$\frac{1}{n} \sum_i z_i^2 \le \sigma^2 = P_N$$

so $\|y - x\|^2 \le n P_N$ w/ prob 1 as $n \nearrow \infty$

y lies w/in an n-dim hypersphere of radius $\sqrt{nP_N}$ centered at x



and $\frac{1}{n}\sum X_i^2 \le P$, so $\frac{1}{n}\sum y_i^2 \le P + P_N$ (triangle)

$$\|y\|^2 \le n(P+P_N)$$

So all likely outputs live in a hypersphere of radius $\sqrt{n(P+P_N)}$,

likely outputs for one input live in hypersphere of radius $\sqrt{P_N}$

n-sphere volume rad. R: $V_n(R) = \underset{\text{constant}}{K_n} R^n$

$$M = \frac{V_n\left(\sqrt{n(P+P_N)}\right)}{V_n\left(\sqrt{n P_N}\right)} = \left(\frac{n(P+P_N)}{n P_N}\right)^{n/2}$$

$$= \left(1+\frac{P}{P_N}\right)^{n/2} = \text{\# messages reliably send}$$

$$C = \frac{\log M}{n} = \boxed{\frac{1}{2}\log\left(1+\frac{P}{P_N}\right) = C}$$