ELEC 4190 – Digital Communications

Introduction to Information Theory

Notes:

Outline

- Measure of information
- Source coding theorem
- Channel coding theorem

- Recommended reading: Proakis and Salehi – Chapter 12
- Extra reading: Lathi and Ding – Chapter 12

Notes:

Outline

- Measure of information
- Source coding theorem
- Channel coding theorem

Notes:

Why Information Theory?

- Study fundamental limits on
  - information sources, and
  - transmission of information over noisy channels
- For example:
  - What is the highest rate at which information can be reliably transmitted over a communication channel?
  - What is the lowest rate at which information can be compressed and still be retrievable with small or no error?
  - What is the complexity of such optimal systems?
- Claude E. Shannon is the father of modern communications due to his contributions to this field

Notes:

Measure of Information

- Announcements!

Notes:

Measure of Information

- Some statements are hardly noticed, some statements may be interesting, and some statements really catch your attention!
- This is equivalent to:
  - almost no information
  - some amount of information
  - large amount of information
- The amount of information carried by a message appears to be related to our ability to anticipate such a message
- This is related to:
  - high probability of occurrence (almost certain event)
  - lower probability of occurrence
  - practically zero probability of occurrence (almost impossible event)

Notes:

Measure of Information (cont.)

- So, surprise (or probability of an event) can be used to measure information
- General rules to an information measure of an output:
  - $I(p_i) \to 0$ as $p_i \to 1$
  - $I(p_i) \to \infty$ as $p_i \to 0$
  - $I(p_i) > I(p_j)$ if $p_i < p_j$
  - $I(p_i) \geq 0$ for $0 \leq p_i \leq 1$
  - $I(p_k) = I(p_i) + I(p_j)0$ if $p_i p_j = p_k$
- The only function that satisfies all these requirements is

$$I(p_i) = \log \frac{1}{p_i} = -\log p_i$$

self-information

The base of the logarithm is not important.
If base 2 is the used, the unit is bits/symbol.

Notes:

sBJksDRPrf96kfrB1XsIog

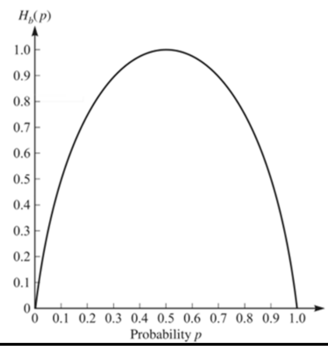
Entropy

- Defines the information content of the source as the weighted average of the self-information $I(p_i)$ of all source outputs

$$H(X) = -\sum_{i=1}^{N} p_i \log p_i$$

- Example: for a memoryless binary source with probabilities of *p* and 1-*p*, we have

$$H(X) = -p \log p - (1-p) \log 1 - p$$



- In fact, the upper bound on the entropy is when $p_i = p = 1/N$

$$0 \le H(X) \le \log N$$

Notes:

sBJksDRPrf96kfrB1XsIog

Example

A source with the bandwidth 4000 Hz is sampled at the Nyquist rate. Assuming that the resulting sequence can be approximately modeled by a discrete memoryless source (DMS) with alphabet A = {-2, -1, 0, 1, 2} and with corresponding probabilities {1/2, 1/4, 1/8, 1/16, 1/16}. Find the average information rate of the source in bit per second.

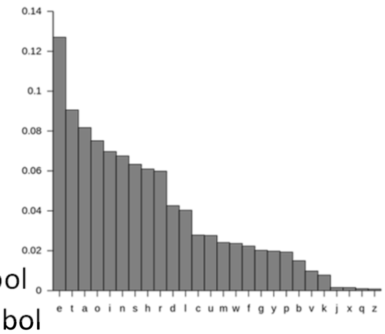
Solution:

- $H(X)$ = 15/8 bits/symbol
- Average information rate = $H(X)\ f_s$

Notes:

Remark: The Intuition of Entropy

- The information content of any message is equal to the minimum number of digits required to encode the message
  - To xllxstxatx, I cxn rxplxce xvexy txirx lextex of x sextexce xitx an x, anx yox stxll xan xanxge xo rxad xt
- Therefore, the entropy is equal to the minimum number of digits per message required, on average, for encoding
- Example: The English language
  - ‘E’ occurs more often than any other letter
  - ‘Z’ is the least frequently letter
  - ‘Q’ is almost always followed by the letter ‘U’
  - If you assume equiprobable, *H* = 4.75 bits/symbol
  - Using actual stats of letters, *H* is between 4.1 bits/symbol
  - Using dependencies, *H* is between 0.6 and 1.3 bits/symbol



Notes:

Outline

- Measure of information
- Source coding theorem
- Channel coding theorem

Notes:

Source Encoding Theorem

- General such that instead of encoding an individual symbol (or message) at a time, we consider successive blocks of *n* symbols
- With entropy $H(X)$, the average compression rate L (bits/source output) for a distortionless source encoding is bounded as follows:

$$L = \frac{L_n}{n} \geq H(X)$$

- That is:
  - n outcomes from a source X can be compressed into roughly $n\,H(X)$ bits
  - The source can be encoded with an arbitrarily small error probability at any rate L as long as $L_n > n\,H(X)$
  - Conversely, if $L_n < n\,H(X)$, the error probability will be bounded away from zero independent of the complexity of the encoder and the decoder
- The ratio of $H(X)/L$ is referred to as the code efficiency

Notes:

Source Encoding Theorem

- This theorem only gives the necessary and sufficient condition for the existence of source codes
- However, it fails to provide an algorithm for the design of source codes that can realize the performance predicted by this theorem

Notes:

Example

Refer to dependencies in the English alphabet

A source generates five equiprobable symbols {A, E, I, O, U}. Find the source entropy when n=1, n=2, and n=3. Find the average codeword length for each case.

Solution:

- $H(X)$ = 2.322 bits/symbol
- n=1: $L_1 > 1 \times 2.322 = 3$ bits → L = 3 bits/1 symbol = 3 bits/symbol → Efficiency = ?
- n=2: $L_2 > 2 \times 2.322 = 5$ bits → L = 5 bits/2 symbols = 2.5 bits/symbol
- n=3: $L_3 > 3 \times 2.322 = 7$ bits → L = 7 bits/3 symbols = 2.333 bits/symbol

- This confirms the Source Encoding Theorem: as n increases, the average length L asymptotically approaches the source entropy (complexity increases as well)
- Next, we will consider to coding algorithms that are close to the entropy bound

Demo

Notes:

Classifications of Source Codes

- Block Codes: map each of the symbols of the source into a fixed sequence of bits
  - May or may not have equal number of bits
- Fixed-length Codes: encodes each symbol of a source into a block of *m* bits, where *m* is the same for all blocks
- Variable-length Codes: codeword length is not the same for all source symbols (e.g., Morse code)
  - Allows mapping more frequent symbols into shorter bit sequences
  - Morse code: E is . and Q is - - . -

Notes:

Classifications of Source Codes (cont.)

- Prefix-free (Instantaneous) Codes: no codeword is a prefix of another codeword
  - Decoding is done as soon as the codeword is fully received
  - For example, 10, 110, 1110, and 11110
- Uniquely Decodable Codes: for each sequence of source symbols, there is a corresponding codeword that is different from a codeword corresponding to any other sequence of source symbols

Notes:

Huffman Source Coding Algorithm

- Fixed length blocks of the source output are mapped to variable length binary blocks
- The idea is:
  - Map the more frequently occurring fixed-length sequences to shorter binary sequences
  - Map the less frequently occurring sequences to longer binary sequences
- Thus, achieving good lossless compression ratios
- Synchronization is an issue because of the variable length

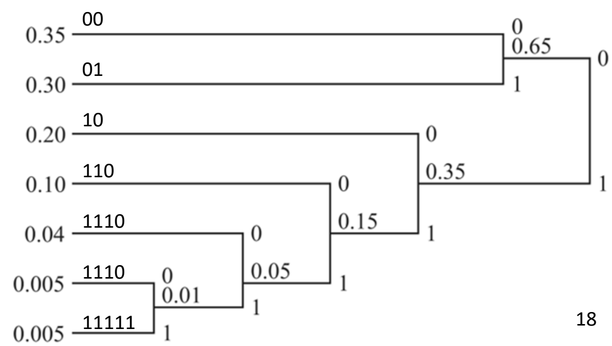
Notes:

Example

A source generates seven messages with probabilities 0.1, 0.35, 0.3, 0.2, 0.04, 0.005, 0.005 respectively. Find the Huffman code.

Procedure:

1. Sort source outputs in decreasing order of their probabilities.
2. Arbitrarily assign 0 and 1 to the two least probable outputs
3. Merge the two outputs into a single output whose probability is the sum of the corresponding probabilities.
4. Repeat until only two remain
5. Append codeword from right to left



Notes:

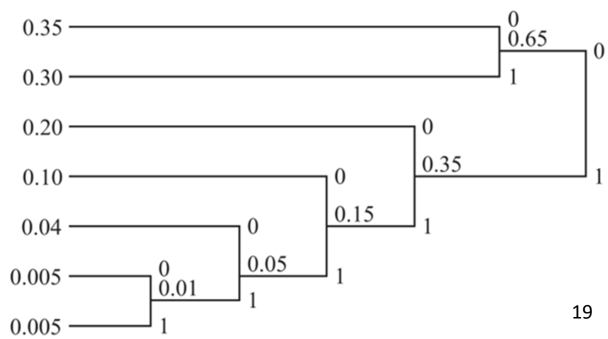
sBJksDRPrf96kfrB1XsIog

Example (cont.)

- Entropy is 2.11
- Average length is 2.21

$$\bar{L} = \sum_{i=1}^{N} p_i l_i$$

Letter	Probability	Self-information	Code
x_1	0.35	1.5146	00
x_2	0.30	1.7370	01
x_3	0.20	2.3219	10
x_4	0.10	3.3219	110
x_5	0.04	4.6439	1110
x_6	0.005	7.6439	11110
x_7	0.005	7.6439	11111



Notes:

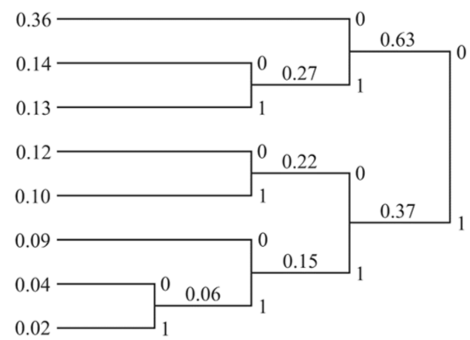
sBJksDRPrf96kfrB1XsIog

Example

A source generates eight messages with probabilities 0.36, 0.14, 0.13, 0.12, 0.1, 0.09, 0.04, 0.02 respectively. Find the Huffman code.

- $H(X)$ = 2.63
- $\overline{L}$ = 2.7

Letter	Code
x_1	00
x_2	010
x_3	011
x_4	100
x_5	101
x_6	110
x_7	1110
x_8	1111



Notes:

sBJksDRPrf96kfrB1XsIog

Example

Design a ternary Huffman code, using 0, 1, and 2 as letters, for a source with output alphabet probabilities given by {0.05, 0.1, 0.15, 0.17, 0.18, 0.22, 0.13}.

- $H(X)$ = 1.7047 ternary symbol/source output
- $\bar{L}$ = 1.78 ternary symbol/source output
- Note: for a fair comparison, we should use $\log_3$ when comping the average code length to the source entropy

Notes:

sBJksDRPrf96kfrB1XsIog

Example

The output of a DMS consists of letters x_1, x_2, and x_3 with probabilities 0.45, 0.35, and 0.20, respectively. The entropy of this source is $H(X) = 1.513$ bits per symbol. Find the Huffman code for this source. If pairs of symbols are encoded by means of the Huffman algorithm, find the resulting code.

Notes:

Example (cont.)

The output of a DMS consists of letters x_1, x_2, and x_3 with probabilities 0.45, 0.35, and 0.20, respectively. The entropy of this source is H(X) = 1.513 bits per symbol. Find the Huffman code for this source. If pairs of symbols are encoded by means of the Huffman algorithm, find the resulting code.

- $H(X)$ = 1.513 bits/letter
- $\bar{L}_1$ = 1.55 bits/letter
- Efficiency = 97.6%

Letter	Probability	Self-information	Code
x_1	0.45	1.156	1
x_2	0.35	1.520	00
x_3	0.20	2.330	01

- $2H(X)$ = 3.026 bits/letter pair
- $\bar{L}_2$ = 3.0675 bits/letter pair
- $\bar{L}$ = 1.534 bits/letter
- Efficiency = 98.6%

Letter pair	Probability	Self-information	Code
x_1x_1	0.2025	2.312	10
x_1x_2	0.1575	2.676	001
x_2x_1	0.1575	2.676	010
x_2x_2	0.1225	3.039	011
x_1x_3	0.09	3.486	111
x_3x_1	0.09	3.486	0000
x_2x_3	0.07	3.850	0001
x_3x_2	0.07	3.850	1100
x_3x_3	0.04	4.660	1101

Notes:

Recap

- Efficient encoding for a DMS may be done on a symbol-by-symbol basis using a variable-length code based on the Huffman algorithm
  - Encoding efficiency asymptotically increases by encoding blocks of *n* symbols at a time (known as extension)
  - Not restricted to binary, e.g., can be used to generate ternary codes
- However:
  - It depends strongly on the source probabilities (statistics) which need to be known in advance to design the code
  - Increasing the source blocks length increases the size of the tree and the complexity of the algorithm
  - Consequently, the application of the Huffman coding method to source coding for many real sources with memory is generally impractical

Notes:

Lempel-Ziv Source Coding Algorithm

- It is a variable-to-fixed-length lossless algorithm
- Does not need source statistics
- Procedure:
  - The sequence at the output of the discrete source is parsed into variable-length blocks, which are called phrases
  - We parse the sequence into the shortest possible phrases not in the dictionary
  - Each new phrase consists of a previous phrase already in the dictionary and a single new source symbol of 0 or 1
  - In encoding a new phrase, we simply specify the location of the existing phrase in the dictionary and append the new letter

Notes:

Example

Apply the Lempel-Ziv Source Coding Algorithm on the following ASCII sequence:
A A B A B B B A B A A B A B B B A B B A B B B

Solution:

- Phrases: A, AB, ABB, B, ABA, ABAB, BB, ABBA, BB
- Dictionary is numbered starting from 1 to 9 in this case
- Thus, we need 4 bits for each phrase + 7 extra bits to represent the new source output
- 0 is used to encode a phrase that has not appeared previously

Codeword = the location of the match + new source output

Location	Content	Codeword
1	A	0A
2	AB	1B
3	ABB	2B
4	B	0B
5	ABA	2A
6	ABAB	5B
7	BB	4B
8	ABBA	3A
9	BBB	7B

Notes:

Example

Apply the Lempel-Ziv Source Coding Algorithm to decode the following sequence:
0A 1B 2B 0B 2A 5B 4B 3A 7B

Solution:

- Position is 0 → new content
- Position is not 0 → find prefix

- The source decoder for the code can construct an identical copy of the dictionary
  - Hence, no need to explicitly transmit the dictionary

Location	Content	Codeword
1	A	0A
2	AB	1B
3	ABB	2B
4	B	0B
5	ABA	2A
6	ABAB	5B
7	BB	4B
8	ABBA	3A
9	BBB	7B

Notes:

Example

Apply the Lempel-Ziv Source Coding Algorithm on the following sequence:
1 0 1 0 1 1 0 1 0 0 1 0 0 1 1 1 0 1 0 1 0 0 0 0 1 1 0 0 1 1 1 0 1 0 1 1 0 0 0 1 1 0 1 1

Solution:

- Phrases: 1, 0, 10, 11, 01, 00, 100, 111, 010, 1000, 011, 001, 110, 101, 10001, 1011
- Dictionary is numbered starting from 1 to 16 in this case
- Thus, we need 4 bits for each phrase + an extra bit to represent the new source output
- 0000 is used to encode a phrase that has not appeared previously

Codeword = the location of the match + new source output

	Dictionary location	Dictionary contents	Code word
1	0001	1	00001
2	0010	0	00000
3	0011	10	00010
4	0100	11	00011
5	0101	01	00101
6	0110	00	00100
7	0111	100	00110
8	1000	111	01001
9	1001	010	01010
10	1010	1000	01110
11	1011	011	01011
12	1100	001	01101
13	1101	110	01000
14	1110	101	00111
15	1111	10001	10101
16		1011	11101

Notes:

Example (cont.)

Apply the Lempel-Ziv Source Coding Algorithm on the following sequence:
1 0 1 0 1 1 0 1 0 0 1 0 0 1 1 1 0 1 0 1 0 0 0 0 1 1 0 0 1 1 1 0 1 0 1 1 0 0 0 1 1 0 1 1

Solution:

- We encoded 44 source bits into 80 coded bits!
- The inefficiency is due to the fact that the sequence is very short
  - As the sequence is increased in length, the encoding procedure becomes more efficient and results in a compressed sequence
- Issue: We will eventually run out of space no matter how large the dictionary is
  - To solve the overflow problem, the source encoder and source decoder must use an identical procedure to remove phrases from the respective dictionaries that are not useful

Demo

Notes:

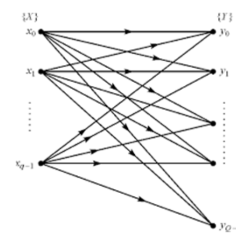
Outline

- Measure of information
- Source coding theorem
- Channel coding theorem

Notes:

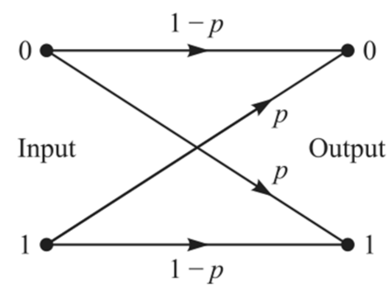
Discrete Memoryless Channels (DMC)

- Now we focus on the channel, not the source
- A discrete channel is a statistical model with an input *X* and an output *Y* where the alphabets of *X* and *Y* are both finite
- A memoryless channel means the current output symbol depends only on the current input symbol and not on any of the previous input symbols
- Note that: $Y = X + Z$ where *Z* is the channel noise (e.g., AWGN)



Notes:

Example: Binary Symmetric Channel (BSC) Model



- Two input symbols $x_1 = 0$ and $x_2 = 1$
- Two out symbols $y_1 = 0$ and $y_2 = 1$
- Conditional probability is used to represent the channel transition probabilities:
  - $P(Y = 1 \mid X = 0) = p$ ← crossover probability
  - $P(Y = 0 \mid X = 0) = 1 - p$
- For example, we already know p for BPSK in the presence of AWGN

$$p = P_b = Q\left(\sqrt{\frac{E_s}{N_0/2}}\right)$$

- In general, we can have more than 2 inputs or outputs

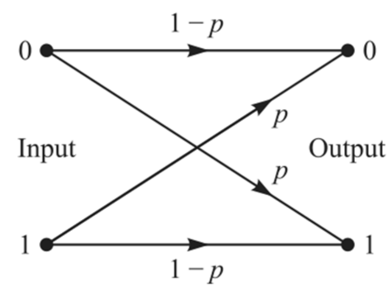
Notes:

sBJksDRPrf96kfrB1XsIog

Example: Binary Symmetric Channel (BSC) Model (cont.)

- The channel can also be discrete-input continuous output
- In this case, for AWGN, we know that

$$f(y|x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y-x)^2}{2\sigma^2}}$$



Notes:

sBJksDRPrf96kfrB1XsIog

Conditional Entropy

- Entropy *H*(*X*) is a measure of the prior uncertainty about the channel input *X* before observing the channel
- Conditional entropy *H*(*X*|*Y*) is a measure of the uncertainty remaining about the channel input *X* after observing the channel

$$H(X|Y) = -\sum_i \sum_j p(x_i, y_j) \log p(x_i|y_j)$$

- Similarly,

$$H(Y|X) = -\sum_i \sum_j p(x_i, y_j) \log p(y_j|x_i)$$

Notes:

Mutual Information

- The difference between entropy and conditional entropy is known as the mutual information
  - In other words, it is the amount by which the uncertainty of *X* is reduced due to the knowledge of *Y*

$$\begin{aligned} I(X;Y) &= H(X) - H(X|Y) \\ &= H(Y) - H(Y|X) \\ &= I(Y;X) \end{aligned}$$

Notes:

sBJksDRPrf96kfrB1XsIog

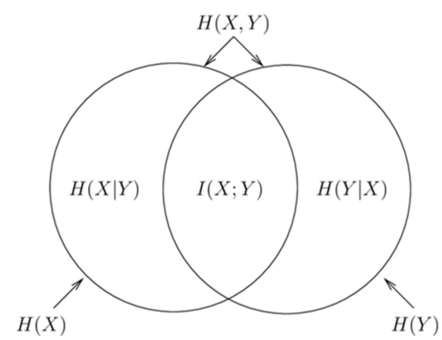
Joint Entropy

- Joint entropy *H*(*X,Y)* is a measure of the uncertainty when we observe both channel input *X* and output *Y* at the same time

$$H(X,Y) = -\sum_i \sum_j p(x_i, y_j) \log p(x_i, y_j)$$

- They are all related, e.g.,

$$\begin{aligned} H(X,Y) &= H(Y) + H(X|Y) \\ &= H(X) + H(Y|X) \\ &= H(X) + H(Y) - I(X;Y) \end{aligned}$$



Notes:

sBJksDRPrf96kfrB1XsIog

Recall

- Law of total probability:

$$p(y_j) = \sum_i p(x_i, y_j) = \sum_i p(x_i)p(y_j|x_i)$$

- Conditional probability:

$$p(x_i, y_j) = p(y_j|x_i)p(x_i)$$

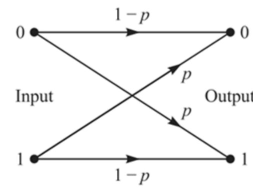
- Bayes' rule:

$$p(x_i|y_j) = \frac{p(y_j|x_i)p(x_i)}{p(y_j)}$$

Notes:

Example

Let X and Y be the input and output of a BSC with crossover probability p=0.1 and input symbol probability of a and 1-a, for 0 and 1. Find I(X; Y).



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

$$H(Y|X) = -\sum_i \sum_j p(x_i, y_j) \log p(y_j|x_i)$$

$$H(Y) = -\sum_j p(y_j) \log p(y_j)$$

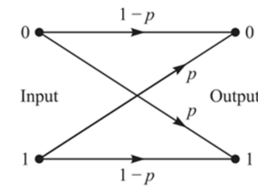
Solution:

- Given:
  - $P(Y=1|X=0) = P(Y=0|X=1) = 0.1$
  - $P(Y=0|X=0) = P(Y=1|X=1) = 0.9$
  - $P(X=0) = a$
  - $P(X=1) = 1-a$
- Marginal probabilities:
  - $P(Y=0) = P(Y=0|X=0)\ P(X=0) + P(Y=0|X=1)\ P(X=1) = 0.8a+0.1$
  - $P(Y=1) = 0.9-0.8a$

Notes:

Example (cont.)

Let X and Y be the input and output of a BSC with crossover probability p=0.1 and input symbol probability of a and 1-a, for 0 and 1. Find I(X; Y).



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

$$H(Y|X) = -\sum_i \sum_j p(x_i, y_j) \log p(y_j|x_i)$$

$$H(Y) = -\sum_j p(y_j) \log p(y_j)$$

Solution:

- Joint probabilities:

  $P(X=0, Y=0) = P(Y=0|X=0)\, P(X=0) = 0.9a$

  $P(X=1, Y=0) = P(Y=0|X=1)\, P(X=1) = 0.1\text{-}0.1a$

  $P(X=0, Y=1) = P(Y=1|X=0)\, P(X=0) = 0.1a$

  $P(X=1, Y=1) = P(Y=1|X=1)\, P(X=1) = 0.9\text{-}0.9a$

- Entropy:

  $H(Y) = -(0.8a+0.1) \log (0.8a+0.1) - (0.9-0.8a) \log (0.9-0.8a)$

  $H(Y|X) = -0.9a \log 0.9 - (0.1-0.1a) \log 0.1 - 0.1a \log 0.1 - (0.9-0.9a) \log 0.9$

  $= -0.1 \log 0.1 - 0.9 \log 0.9 = 0.468996$ bits/source output

- Mutual information:

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

Notes:

sBJksDRPrf96kfrB1XsIog

Channel Coding Theorem

- The capacity of a discrete memoryless channel is given by

$$C = \max_{P(x)} I(X;Y)$$ bits/symbol (or bits/channel use)

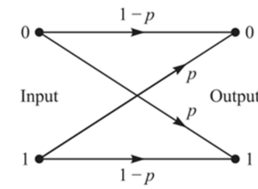
- If the information rate $R = (\log_2 M)/n$ from the source is less than C, then it is theoretically possible to achieve reliable (error-free) transmission through the channel
- One of the most important results in information theory and gives a fundamental limit on the possibility of reliable communication over a noisy channel
- This is an upper limit, there are many other impairments in real channels

Notes:

sBJksDRPrf96kfrB1XsIog

Example

Let X and Y be the input and output of a BSC with crossover probability p=0.1 and input symbol probability of a and 1-a, for 0 and 1. Find C.



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

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

Solution:

- Entropy:
  $H(Y) = -(0.8a+0.1) \log (0.8a+0.1) - (0.9-0.8a) \log (0.9-0.8a)$
  $H(Y|X) = -0.1 \log 0.1 - 0.9 \log 0.9$
- Mutual information:
  $I(X;Y) = H(Y) - H(Y|X)$
- $H(Y|X)$ is constant, so mutual information is maximum when $H(Y)$ is maximum → using the derivative, $a = 0.5$
- Then, $C = \log 2 - H(Y|X) = 0.531004$ bits/source output

Notes:

sBJksDRPrf96kfrB1XsIog

Gaussian Channel Capacity Theorem

- Under AWGN, the capacity of a discrete memoryless channel is

$$C = \frac{1}{2}\log_2\left(1 + \frac{P}{N_0 W}\right) \quad \text{bits/symbol}$$

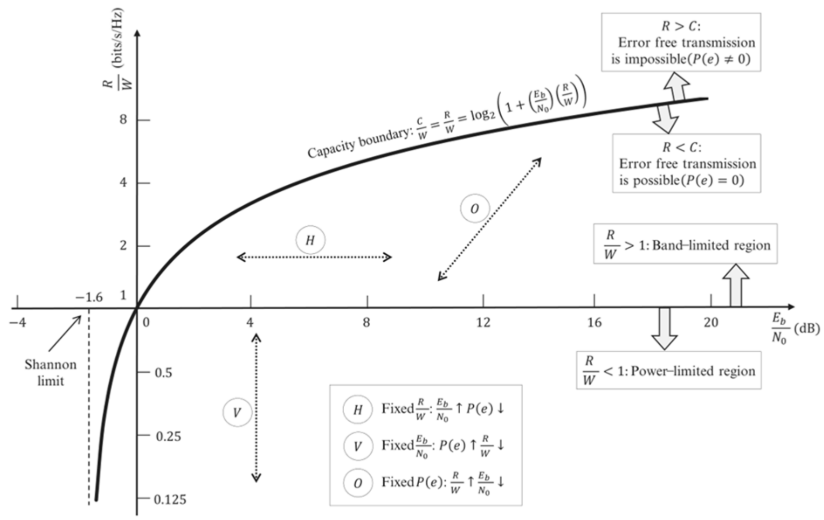
- Equivalently,

$$C = W\log_2\left(1 + \frac{P}{N_0 W}\right) \quad \text{bits/sec}$$

SNR (ratio not dB)

Notes:

Recall



Grami

Notes:

sBJksDRPrf96kfrB1XsIog

Summary

- By now you should know:
  - Source coding theorem:
    Average number of bits per source symbol can be made as small as possible, but not smaller than the entropy of the source measured in bits
  - Channel coding theorem:
    If the entropy of a DMS is less than the capacity of a discrete memoryless channel, then there exists a channel coding scheme for which the source output can be transmitted over the channel with an arbitrarily small probability of bit error

Notes: