Sirindhorn International Institute of Technology

Thammasat University

School of Information, Computer and Communication Technology

Digital Communications

Asst. Prof. Dr. Prapun Suksompong
prapun@siit.tu.ac.th

September 24, 2014

The subject of digital communications involves the transmission of information in digital form from a source that generates the information to one or more destinations. This course extends the knowledge gained from ECS332 (Principles of Communications) and ECS315 (Probability and Random Processes). Basic principles that underlie the analysis and design of digital communication systems are covered. This semester, the main focus includes performance analysis (symbol error probability), optimal receivers, and limits (information theoretic capacity). These topics are challenging but the presented material are carefully selected to keep the difficulty level appropriate for undergraduate students.

Contents

Sirindhorn International Institute of Technology

Thammasat University

School of Information, Computer and Communication Technology

ECS452 2014/1 Part I.1 Dr.Prapun

1 Elements of a Digital Communication System

1.1. Figure 1 illustrates the functional diagram and the basic elements of a digital communication system.

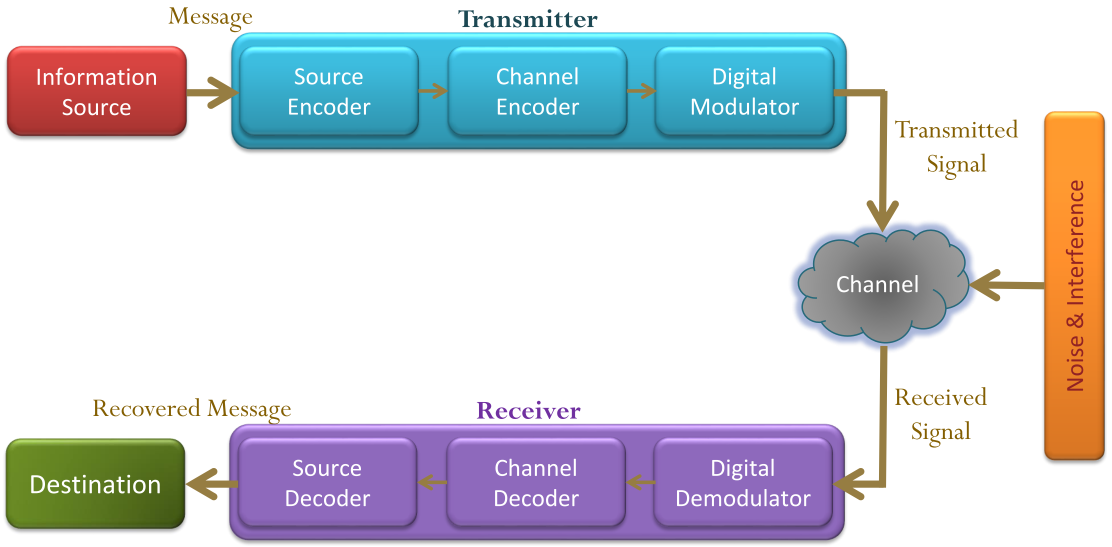

Figure 1: Basic elements of a digital communication system

1.2. The source output may be either

- an **analog** signal, such as an audio or video signal,

or

- a **digital** signal, such as the output of a computer, that is discrete in time and has a finite number of output characters.

1.3. Source Coding: The messages produced by the source are converted into a sequence of binary digits.

- Want to represent the source output (message) by as few binary digits as possible.
    - ◦ In other words, we seek an efficient representation of the source output that results in little or no redundancy.
- The process of efficiently converting the output of either an analog or digital source into a sequence of binary digits is called **source encoding** or **data compression**.

1.4. Channel Coding:

- Introduce, in a controlled manner, some *redundancy* in the binary information sequence that can be used at the receiver to overcome the effects of noise and interference encountered in the transmission of the signal through the channel.
    - ◦ The added redundancy serves to increase the reliability of the received data and improves the fidelity of the received signal.
- See Examples 1.5 and 1.6.

Example 1.5. Trivial channel coding: Repeat each binary digit n times, where n is some positive integer.

Example 1.6. More sophisticated channel coding: Taking k information bits at a time and mapping each k-bit sequence into a unique n-bit sequence, called a **codeword**.

- The amount of redundancy introduced by encoding the data in this manner is measured by the ratio n/k. The reciprocal of this ratio, namely k/n, is called the rate of the code or, simply, the **code rate**.

1.7. The binary sequence at the output of the channel encoder is passed to the **digital modulator**, which serves as the interface to the physical (analog) communication channel.

- Since nearly all the communication channels encountered in practice are capable of transmitting electrical signals (waveforms), the primary purpose of the digital modulator is to map the binary information sequence into signal waveforms.

- The digital modulator may simply map the binary digit 0 into a waveform $s_0(t)$ and the binary digit 1 into a waveform $s_1(t)$. In this manner, each bit from the channel encoder is transmitted separately.
  We call this **binary modulation**.
- The modulator may transmit b coded information bits at a time by using $M = 2^b$ distinct waveforms $s_i(t), i = 0, 1, \ldots, M-1$, one waveform for each of the 2^b possible b-bit sequences.
  We call this **M-ary modulation** ($M > 2$).

1.8. The **communication channel** is the physical medium that is used to send the signal from the transmitter to the receiver.

- The physical channel may be
  - a pair of wires that carry the electrical signal, or
  - an optical fiber that carries the information on a modulated light beam, or
  - an underwater ocean channel in which the information is transmitted acoustically, or
  - free space over which the information-bearing signal is radiated by use of an antenna.

  Other media that can be characterized as communication channels are data storage media, such as magnetic tape, magnetic disks, and optical disks.
- Whatever the physical medium used for transmission of the information, the essential feature is that the transmitted signal is corrupted in a random manner by a variety of possible mechanisms, such as additive **thermal noise** generated by electronic devices; man-made noise, e.g., automobile ignition noise; and **atmospheric noise**, e.g., electrical lightning discharges during thunderstorms.
- Other channel impairments including noise, attenuation, distortion, fading, and interference (such as interference from other users of the channel).

1.9. At the receiving end of a digital communication system, the digital demodulator processes the channel-corrupted transmitted waveform and reduces the waveforms to a sequence of numbers that represent estimates of the transmitted data symbols (binary or M-ary).

This sequence of numbers is passed to the channel decoder, which attempts to reconstruct the original information sequence from knowledge of the code used by the channel encoder and the redundancy contained in the received data.

- A measure of how well the demodulator and decoder perform is the frequency with which errors occur in the decoded sequence. More precisely, the average probability of a bit-error at the output of the decoder is a measure of the performance of the demodulator-decoder combination.
- In general, the probability of error is a function of the code characteristics, the types of waveforms used to transmit the information over the channel, the transmitter power, the characteristics of the channel (i.e., the amount of noise, the nature of the interference), and the method of demodulation and decoding.

1.10. As a final step, when an analog output is desired, the source decoder accepts the output sequence from the channel decoder and, from knowledge of the source encoding method used, attempts to reconstruct the original signal from the source.

- Because of channel decoding errors and possible distortion introduced by the source encoder, and perhaps, the source decoder, the signal at the output of the source decoder is an approximation to the original source output.
- The difference or some function of the difference between the original signal and the reconstructed signal is a measure of the distortion introduced by the digital communication system.

2 Source Coding

In this section, we look at the "source encoder" part of the system. This part removes redundancy from the message stream or sequence. We will focus only on <u>binary</u> source coding.

2.1. The material in this section is based on [C & T Ch 2, 4, and 5].

2.1 General Concepts

Example 2.2. Suppose your message is a paragraph of (written natural) text in English.

- Approximately 100 possibilities for characters/symbols.
    - For example, a character-encoding scheme called (ASCII) (American Standard Code for Information Interchange) originally[1] had 128 specified characters – the numbers 0–9, the letters a–z and A–Z, some basic punctuation symbols[2], and a blank space.
- Do we need 7 bits per characters?

2.3. A sentence of English–or of any other language–always has more information than you need to decipher it. The meaning of a message can remain unchanged even though parts of it are removed.

Example 2.4.

- "J-st tr- t- r–d th-s s-nt-nc-." [3]
- "Thanks to the redundancy of language, yxx cxn xndxrstxnd whxt x xm wrxtxng xvxn xf x rxplxcx xll thx vxwxls wxth xn 'x' (t gts lttl hrdr f y dn't vn kn whr th vwls r)." [4]

[1]Being American, it didn't originally support accented letters, nor any currency symbols other than the dollar. More advanced Unicode system was established in 1991.

[2]There are also some control codes that originated with Teletype machines. In fact, among the 128 characters, 33 are non-printing control characters (many now obsolete) that affect how text and space are processed and 95 printable characters, including the space.

[3]Charles Seife, Decoding the Universe. Penguin, 2007

[4]Steven Pinker, The Language Instinct: How the Mind Creates Language. William Morrow, 1994

2.5. It is estimated that we may only need about 1 bits per character in English text.

Definition 2.6. Discrete Memoryless Sources (DMS): Let us be more specific about the information source.

- The message that the information source produces can be represented by a **vector** of characters $X_1, X_2, \ldots, X_n$.
  - A perpetual message source would produce a never-ending **sequence** of characters $X_1, X_2, \ldots$.
- These X_k's are random variables (at least from the perspective of the decoder; otherwise, these is no need for communication).
- For simplicity, we will assume our source to be discrete and memoryless.
  - Assuming a discrete source means that the random variables are all discrete; that is, they have supports which are countable. Recall that "countable" means "finite" or "countably infinite".
    - We will further assume that they all share the same support and that the support is finite. This support is called the source **alphabet**.
  - Assuming a memoryless source means that there is no dependency among the characters in the sequence.
    - More specifically,

$$p_{X_1,X_2,\ldots,X_n}(x_1, x_2, \ldots, x_n) = p_{X_1}(x_1) \times p_{X_2}(x_2) \times \cdots \times p_{X_n}(x_n). \tag{1}$$

    - This means the current model of the source is far from the source of normal English text. English text has dependency among the characters. However, this simple model provide a good starting point.
    - We will further assume that all of the random variables share the same probability mass function (pmf). We denote this shared pmf by $p_X(x)$. In which case, (1) becomes

$$p_{X_1,X_2,\ldots,X_n}(x_1, x_2, \ldots, x_n) = p_X(x_1) \times p_X(x_2) \times \cdots \times p_X(x_n). \tag{2}$$

    - We will also assume that the pmf $p_X(x)$ is known. In practice, there is an extra step of estimating this $p_X(x)$.
    - To save space, we may see the pmf $p_X(x)$ written simply as $p(x)$, i.e. without the subscript part.
  - The shared support of X which is usually denoted by S_X becomes the source alphabet. Note that we also often see the use of $\mathcal{X}$ to denote the support of X.

- In conclusion, our simplified source code can be characterized by a random variable X. So, we only need to specify its pmf $p_X(x)$.

Example 2.7. See slides.

Definition 2.8. An **encoder** $c(\cdot)$ is a function that maps each of the character in the (source) alphabet into a corresponding (binary) codeword.

- In particular, the codeword corresponding to a source character x is denoted by $c(x)$.

- Each codeword is constructed from a code alphabet.

  - A binary codeword is constructed from a two-symbol alphabet, wherein the two symbols are usually taken as 0 and 1.
  - It is possible to consider non-binary codeword. Morse code discussed in Example 2.13 is one such example.

- Mathematically, we write

$$\text{Encoder } c : S_X \to \{0,1\}^*$$

where

$$\{0,1\}^* = \{\varepsilon, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, \ldots\}$$

is the set of all finite-length binary strings.

- The length of the codeword associated with source character x is denoted by $\ell(x)$.

- ∘ In fact, writing this as $\ell\left(c(x)\right)$ may be clearer because we can see that the length depends on the choice of the encoder. However, we shall follow the notation above[5].

Example 2.9. $c(\text{red}) = 00$, $c(\text{blue}) = 11$ is a source code for $S_X = \{\text{red}, \text{blue}\}$.

Example 2.10. Suppose the message is a sequence of basic English words which happen according to the probabilities provided in the table below.

x	$p(x)$	Codeword $c(x)$	$\ell(x)$
Yes	4%		
No	3%		
OK	90%		
Thank You	3%		

Definition 2.11. The **expected length** of a code $c(\cdot)$ for a source which is characterized by a random variable X with probability mass function $p_X(x)$ is given by

$$\mathbb{E}\left[\ell(X)\right] = \sum_{x \in S_X} p_X(x)\ell(x).$$

Example 2.12. Back to Example 2.10. Consider a new encoder:

x	$p(x)$	Codeword $c(x)$	$\ell(x)$
Yes	4%	01	
No	3%	001	
OK	90%	1	
Thank You	3%	0001	

Observe the following:

- Data compression can be achieved by assigning short descriptions to the most frequent outcomes of the data source, and necessarily longer descriptions to the less frequent outcomes.
- When we calculate the expected length, we don't really use the fact that the source alphabet is the set $\{\text{Yes}, \text{No}, \text{OK}, \text{Thank You}\}$. We would get the same answer if it is replaced by the set $\{1, 2, 3, 4\}$, or the

[5]which is used by the standard textbooks in information theory.

set $\{a, b, c, d\}$. All that matters is that the alphabet size is 4, and the corresponding probabilities are $\{0.04, 0.03, 0.9, 0.03\}$.

Therefore, for brevity, we often find DMS source defined only by its alphabet size and the list of probabilities.

Example 2.13. The Morse code is a reasonably efficient code for the English alphabet using an alphabet of four symbols: a dot, a dash, a letter space, and a word space. [See Slides]

- Short sequences represent frequent letters (e.g., a single dot represents E) and long sequences represent infrequent letters (e.g., Q is represented by "dash,dash,dot,dash").

Example 2.14. Thought experiment: Let's consider the following code

x	$p(x)$	Codeword $c(x)$	$\ell(x)$
1	4%	0	
2	3%	1	
3	90%	0	
4	3%	1	

This code is bad because we have ambiguity at the decoder. When a codeword "0" is received, we don't know whether to decode it as source symbol "1" or source symbol "3". If we want to have lossless source coding, this ambiguity is not allowed.

Definition 2.15. A code is **nonsingular** if every source symbol in the source alphabet has different codeword.

As seen from Example 2.14, nonsingularity is an important concept. However, it turns out that this property is not enough.

Example 2.16. Another thought experiment: Let's consider the following code

x	$p(x)$	Codeword $c(x)$	$\ell(x)$
1	4%	01	
2	3%	010	
3	90%	0	
4	3%	10	

2.17. We usually wish to convey a sequence (string) of source symbols. So, we will need to consider **concatenation** of codewords; that is, if our source string is

$$X_1, X_2, X_3, \ldots$$

then the corresponding encoded string is

$$c(X_1)c(X_2)c(X_3)\cdots.$$

In such cases, to ensure decodability, we may

(a) use fixed-length code, or

(b) use variable-length code and

  (i) add a special symbol (a "comma" or a "space") between any two codewords

  or

  (ii) use uniquely decodable codes.

Definition 2.18. A code is called **uniquely decodable** if any encoded string has $\underline{\text{only one possible}}$ source string producing it.

Example 2.19. The code used in Example 2.16 is not uniquely decodable because source string "2", source string "34", and source string "13" share the same code string "010".

2.20. It may not be easy to check unique decodability of a code. Also, even when a code is uniquely decodable, one may have to look at the entire string to determine even the first symbol in the corresponding source string. Therefore, we focus on a subset of uniquely decodable codes called prefix code.

Definition 2.21. A code is called a **prefix code** if no codeword is a prefix[6] of any other codeword.

- Equivalently, a code is called a **prefix code** if you can put all the codewords into a binary tree were all of them are $\underline{\text{leaves}}$.

[6]String s_1 is a prefix of string s_2 if there exist a string s_3, possibly empty, such that $s_2 = s_1 s_3$.

- A more appropriate name would be "**prefix-free**" code.

Example 2.22. The code used in Example 2.12 is a prefix code.

x	Codeword $c(x)$
1	01
2	001
3	1
4	0001

Example 2.23.

x	Codeword $c(x)$
1	10
2	110
3	0
4	111

2.24. Any prefix code is uniquely decodable.

- The end of a codeword is immediately recognizable.
- Each source symbol can be decoded as soon as we come to the end of the codeword corresponding to it. In particular, we need not wait to see the codewords that come later.
- It is also commonly called an **instantaneous code**

Example 2.25. The code used in Example 2.12 (and Example 2.22) is a prefix code and hence it is uniquely decodable.

2.26. The nesting relationship among all the types of source codes is shown in Figure 2.

Example 2.27.

x	Codeword $c(x)$
1	001
2	01
3	1
4	0001

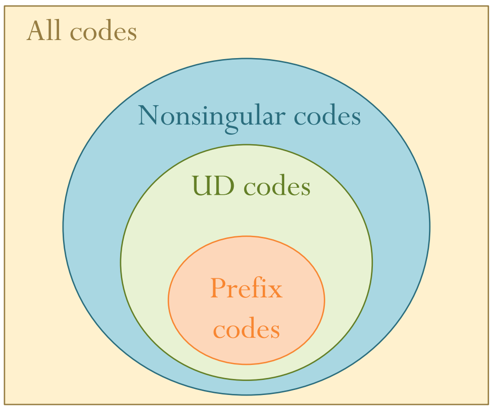

Figure 2: Classes of codes

Example 2.28. [2, p 106–107]

x	Codeword $c(x)$
1	10
2	00
3	11
4	110

This code is not a prefix code because codeword "11" is a prefix of codeword "110".

This code is uniquely decodable. To see that it is uniquely decodable, take any code string and start from the beginning.

- If the first two bits are 00 or 10, they can be decoded immediately.
- If the first two bits are 11, we must look at the following bit(s).
    - If the next bit is a 1, the first source symbol is a 3.
    - If the length of the string of 0's immediately following the 11 is even, the first source symbol is a 3.
    - If the length of the string of 0's immediately following the 11 is odd, the first codeword must be 110 and the first source symbol must be 4.

By repeating this argument, we can see that this code is uniquely decodable.

ECS452 2014/1 Part I.2 Dr.Prapun

2.29. For our present purposes, a better code is one that is uniquely decodable and has a shorter expected length than other uniquely decodable codes. We do not consider other issues of encoding/decoding complexity or of the relative advantages of block codes or variable length codes. [3, p 57]

2.2 Optimal Source Coding: Huffman Coding

In this section we describe a very popular source coding algorithm called the Huffman coding.

Definition 2.30. Given a source with known probabilities of occurrence for symbols in its alphabet, to construct a binary Huffman code, create a binary tree by repeatedly combining[7] the probabilities of the two least likely symbols.

- Developed by David Huffman as part of a class assignment[8].

[7]The Huffman algorithm performs *repeated source reduction* [3, p 63]:

- At each step, two source symbols are combined into a new symbol, having a probability that is the sum of the probabilities of the two symbols being replaced, and the new reduced source now has one fewer symbol.
- At each step, the two symbols to combine into a new symbol have the two lowest probabilities.
    - ◦ If there are more than two such symbols, select any two.

[8]The class was the first ever in the area of information theory and was taught by Robert Fano at MIT in 1951.

- ◦ Huffman wrote a term paper in lieu of taking a final examination.
- ◦ It should be noted that in the late 1940s, Fano himself (and independently, also Claude Shannon) had developed a similar, but suboptimal, algorithm known today as the ShannonFano method. The difference between the two algorithms is that the ShannonFano code tree is built from the top down, while the Huffman code tree is constructed from the bottom up.

- By construction, Huffman code is a prefix code.

Example 2.31.

x	$p_X(x)$	Codeword $c(x)$	$\ell(x)$
1	0.5		
2	0.25		
3	0.125		
4	0.125		

$\mathbb{E}\left[\ell(X)\right] =$

Note that for this particular example, the values of $2^{\ell(x)}$ from the Huffman encoding is inversely proportional to $p_X(x)$:

$$p_X(x) = \frac{1}{2^{\ell(x)}}.$$

In other words,

$$\ell(x) = \log_2 \frac{1}{p_X(x)} = -\log_2(p_X(x)).$$

Therefore,

$$\mathbb{E}\left[\ell(X)\right] = \sum_x p_X(x)\ell(x) =$$

Example 2.32.

x	$p_X(x)$	Codeword $c(x)$	$\ell(x)$
‘a’	0.4		
‘b’	0.3		
‘c’	0.1		
‘d’	0.1		
‘e’	0.06		
‘f’	0.04		

$\mathbb{E}\left[\ell(X)\right] =$

Example 2.33.

x	$p_X(x)$	Codeword $c(x)$	$\ell(x)$
1	0.25		
2	0.25		
3	0.2		
4	0.15		
5	0.15		

$$\mathbb{E}\left[\ell(X)\right] =$$

Example 2.34.

x	$p_X(x)$	Codeword $c(x)$	$\ell(x)$
	1/3		
	1/3		
	1/4		
	1/12		

$$\mathbb{E}\left[\ell(X)\right] =$$

x	$p_X(x)$	Codeword $c(x)$	$\ell(x)$
	1/3		
	1/3		
	1/4		
	1/12		

$$\mathbb{E}\left[\ell(X)\right] =$$

2.35. The set of codeword lengths for Huffman encoding is not unique. There may be more than one set of lengths but all of them will give the same value of expected length.

Definition 2.36. A code is **optimal** for a given source (with known pmf) if it is uniquely decodable and its corresponding expected length is the shortest among all possible uniquely decodable codes for that source.

2.37. The Huffman code is optimal.

2.3 Source Extension (Extension Coding)

2.38. One can usually (not always) do better in terms of expected length (per source symbol) by encoding blocks of several source symbols.

Definition 2.39. In, an ***n*-th extension** coding, n successive source symbols are grouped into blocks and the encoder operates on the blocks rather than on individual symbols. [1, p. 777]

Example 2.40.

x	$p_X(x)$	Codeword $c(x)$	$\ell(x)$
Y(es)	0.9		
N(o)	0.1		

(a) First-order extension:

$\mathbb{E}\left[\ell(X)\right] =$

YNNYYYNYYNNN...

(b) Second-order Extension:

x_1x_2	$p_{X_1,X_2}(x_1,x_2)$	$c(x_1,x_2)$	$\ell(x_1,x_2)$
YY			
YN			
NY			
NN			

$\mathbb{E}\left[\ell(X_1,X_2)\right] =$

(c) Third-order Extension:

$x_1x_2x_3$	$p_{X_1,X_2,X_3}(x_1,x_2,x_3)$	$c(x_1,x_2,x_3)$	$\ell(x_1,x_2,x_3)$
YYY			
YYN			
YNY			
$\vdots$			

$\mathbb{E}\left[\ell(X_1,X_2,X_3)\right] =$

Sirindhorn International Institute of Technology

Thammasat University

School of Information, Computer and Communication Technology

ECS452 2014/1 Part I.3 Dr.Prapun

2.4 (Shannon) Entropy for Discrete Random Variables

Entropy is a measure of uncertainty of a random variable [2, p 13].

It arises as the answer to a number of natural questions. One such question that will be important for us is "What is the average length of the shortest *description* of the random variable?"

Definition 2.41. The **entropy** $H(X)$ of a discrete random variable X is defined by

$$H(X) = -\sum_{x \in S_X} p_X(x) \log_2 p_X(x) = -\mathbb{E}\left[\log_2 p_X(X)\right].$$

- The log is to the base 2 and entropy is expressed in bits (per symbol).
  - The base of the logarithm used in defining H can be chosen to be any convenient real number $b > 1$ but if $b \neq 2$ the unit will not be in bits.
  - If the base of the logarithm is e, the entropy is measured in nats.
  - Unless otherwise specified, base 2 is our default base.
- Based on continuity arguments, we shall assume that $0 \ln 0 = 0$.

Example 2.42. The entropy of the random variable X in Example 2.31 is 1.75 bits (per symbol).

Example 2.43. The entropy of a fair coin toss is 1 bit (per toss).

2.44. Note that entropy is a functional of the (unordered) probabilities from the pmf of X. It does not depend on the actual values taken by the random variable X, Therefore, sometimes, we write $H(p_X)$ instead of $H(X)$ to emphasize this fact. Moreover, because we use only the probability values, we can use the row vector representation $\underline{p}$ of the pmf p_X and simply express the entropy as $H(\underline{p})$.

In MATLAB, to calculate $H(X)$, we may define a row vector `pX` from the pmf p_X. Then, the value of the entropy is given by

```
HX = -pX*(log2(pX))'.
```

Example 2.45. The entropy of a uniform (discrete) random variable X on $\{1, 2, 3, \ldots, n\}$:

Example 2.46. The entropy of a Bernoulli random variable X:

Definition 2.47. Binary Entropy Function : We define $h_b(p)$, $h(p)$ or $H(p)$ to be $-p \log p - (1-p) \log(1-p)$, whose plot is shown in Figure 3.

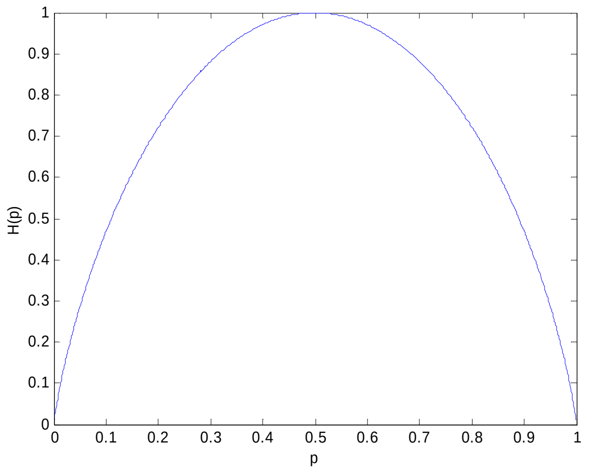

Figure 3: Binary Entropy Function

2.48. Two important facts about entropy:

(a) $H(X) \le \log_2 |S_X|$ with equality if and only if X is a uniform random variable.

(b) $H(X) \ge 0$ with equality if and only if X is not random.

In summary,

$$\underset{\text{deterministic}}{0} \le H(X) \le \underset{\text{uniform}}{\log_2 |S_X|}.$$

Theorem 2.49. The expected length $\mathbb{E}[\ell(X)]$ of any uniquely decodable binary code for a random variable X is greater than or equal to the entropy $H(X)$; that is,

$$\mathbb{E}[\ell(X)] \ge H(X)$$

with equality if and only if $2^{-\ell(x)} = p_X(x)$. [2, Thm. 5.3.1]

Definition 2.50. Let $L(c, X)$ be the expected codeword length when random variable X is encoded by code c.

Let $L^*(X)$ be the minimum possible expected codeword length when random variable X is encoded by a uniquely decodable code c:

$$L^*(X) = \min_{\text{UD } c} L(c, X).$$

2.51. Given a random variable X, let c_{Huffman} be the Huffman code for this X. Then, from the optimality of Huffman code mentioned in 2.37,

$$L^*(X) = L(c_{\text{Huffman}}, X).$$

Theorem 2.52. The optimal code for a random variable X has an expected length less than $H(X) + 1$:

$$L^*(X) < H(X) + 1.$$

2.53. Combining Theorem 2.49 and Theorem 2.52, we have

$$H(X) \le L^*(X) < H(X) + 1. \tag{3}$$

Definition 2.54. Let $L_n^*(X)$ be the minimum expected codeword length per symbol when the random variable X is encoded with n-th extension uniquely decodable coding. Of course, this can be achieve by using n-th extension Huffman coding.

2.55. An extension of (3):

$$H(X) \le L_n^*(X) < H(X) + \frac{1}{n}. \tag{4}$$

In particular,

$$\lim_{n\to\infty} L_n^*(X) = H(X).$$

In otherwords, by using large block length, we can achieve an expected length per source symbol that is arbitrarily close to the value of the entropy.

2.56. Operational meaning of entropy: Entropy of a random variable is the average length of its shortest description.

2.57. References

- Section 16.1 in Carlson and Crilly [1]
- Chapters 2 and 5 in Cover and Thomas [2]
- Chapter 4 in Fine [3]
- Chapter 14 in Johnson, Sethares, and Klein [5]
- Section 11.2 in Ziemer and Tranter [9]

ECS452 2014/1 Part I.4 Dr.Prapun

3 An Introduction to Digital Communication Systems Over Discrete Memoryless Channel (DMC)

In this section, we keep our analysis of the communication system simple by considering purely digital systems. To do this, we assume all non-source-coding parts of the system, including the physical channel, can be combined into an "equivalent channel" which we shall simply refer to in this section as the "channel".

3.1 Discrete Memoryless Channel (DMC) Models

Example 3.1. The **binary symmetric channel (BSC)**, which is the simplest model of a channel with errors, is shown in Figure 4.

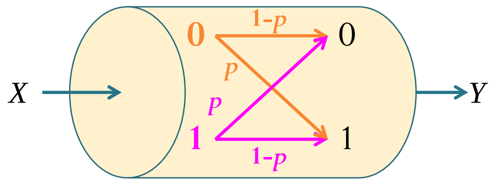

Figure 4: Binary symmetric channel and its channel diagram

- "Binary" means that the there are two possible values for the input and also two possible values for the output. We normally use the symbols 0 and 1 to represent these two values.
- Passing through this channel, the input symbols are complemented with probability p.

- It is simple, yet it captures most of the complexity of the general problem.

Definition 3.2. Our model for discrete memoryless channel (DMC) is shown in Figure 5.

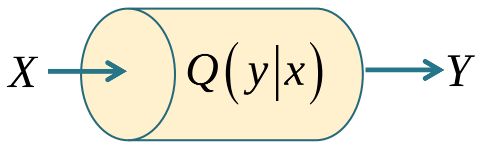

Figure 5: Discrete memoryless channel

- The channel input is denoted by a random variable X.
    - The pmf $p_X(x)$ is usually denoted by simply $p(x)$ and usually expressed in the form of a row vector $\underline{p}$ or $\underline{\pi}$.
    - The support S_X is often denoted by $\mathcal{X}$.
- Similarly, the channel output is denoted by a random variable Y.
    - The pmf $p_Y(y)$ is usually denoted by simply $q(y)$ and usually expressed in the form of a row vector $\underline{q}$.
    - The support S_Y is often denoted by $\mathcal{Y}$.
- The channel corrupts its input X in such a way that when the input is $X = x$, its output Y is randomly selected from the conditional pmf $p_{Y|X}(y|x)$.
    - This conditional pmf $p_{Y|X}(y|x)$ is usually denoted by $Q(y|x)$ and usually expressed in the form of a probability transition matrix $\mathbf{Q}$:

$$\begin{array}{cc} & y \\ x & \begin{bmatrix} \ddots & \vdots & \iddots \\ \cdots & P[Y = y|X = x] & \cdots \\ \iddots & \vdots & \ddots \end{bmatrix} \end{array}$$

- ∘ The channel is called memoryless[9] because its channel output at a given time is a function of the channel input at that time and is not a function of previous channel inputs.
- ∘ Here, the transition probabilities are assumed constant. However, in many commonly encountered situations, the transition probabilities are time varying. An example is the wireless mobile channel in which the transmitter-receiver distance is changing with time.

Example 3.3. For a binary symmetric channel (BSC) defined in 3.1,

Example 3.4. Suppose, for a DMC, we have $\mathcal{X} = \{x_1, x_2\}$ and $\mathcal{Y} = \{y_1, y_2, y_3\}$. Then, its probability transition matrix $\mathbf{Q}$ is of the form

$$\mathbf{Q} = \left[\begin{array}{ccc} Q\left(y_1 | x_1\right) & Q\left(y_2 | x_1\right) & Q\left(y_2 | x_1\right) \\ Q\left(y_1 | x_2\right) & Q\left(y_2 | x_2\right) & Q\left(y_2 | x_2\right) \end{array} \right].$$

You may wonder how this $\mathbf{Q}$ happens in real life. Let's suppose that the input to the channel is binary; hence, $\mathcal{X} = \{0, 1\}$ as in the BSC. However, in this case, after passing through the channel, some bits can be lost[10] (rather than corrupted). In such case, we have three possible outputs of the channel: 0, 1, e where the "e" represents the case in which the bit is erased by the channel.

[9]Mathematically, the condition that the channel is memoryless may be expressed as [6, Eq. 6.5-1 p. 355]

$$p_{X_1^n | Y_1^n}\left(x_1^n | y_1^n\right) = \prod_{k=1}^{n} Q\left(y_k \left| x_k \right.\right).$$

[10]The receiver knows which bits have been erased.

3.5. Knowing the input probabilities $\underline{p}$ and the channel probability transition matrix $\mathbf{Q}$, we can calculate the output probabilities $\underline{q}$ from

$$\underline{q} = \underline{p}\mathbf{Q}.$$

To see this, recall the **total probability theorem**: If a (finite or infinitely) countable collection of events $\{B_1, B_2, \ldots\}$ is a partition of Ω, then

$$P(A) = \sum_i P(A \cap B_i) = \sum_i P(A|B_i)P(B_i). \tag{5}$$

For us, event A is the event $[Y = y]$. Applying this theorem to our variables, we get

$$\begin{aligned} q(y) = P\left[Y = y\right] &= \sum_x P\left[X = x, Y = y\right] \\ &= \sum_x P\left[Y = y|X = x\right] P\left[X = x\right] = \sum_x Q(y|x)p(x). \end{aligned}$$

This is exactly the same as the matrix multiplication calculation performed to find each element of $\underline{q}$.

Example 3.6. For a binary symmetric channel (BSC) defined in 3.1,

3.7. Recall, from ECS315, that there is another matrix called the **joint probability matrix P**. This is the matrix whose elements give the joint probabilities $P_{X,Y}(x,y) = P[X = x, Y = y]$:

$$\begin{array}{cc} & y \\ x & \left[\begin{array}{ccc} \ddots & \vdots & \iddots \\ \cdots & P[X = x, Y = y] & \cdots \\ \iddots & \vdots & \ddots \end{array}\right] \end{array}.$$

Recall also that we can get $p(x)$ by adding the elements of $\mathbf{P}$ in the row corresponding to x. Similarly, we can get $q(y)$ by adding the elements of $\mathbf{P}$ in the column corresponding to y.

By definition, the relationship between the conditional probability $Q(y|x)$ and the joint probability $P_{X,Y}(x,y)$ is

$$Q(y|x) = \frac{P_{X,Y}(x,y)}{p(x)}.$$

Equivalently,

$$P_{X,Y}(x,y) = p(x)Q(y|x).$$

Therefore, to get the matrix $\mathbf{P}$ from matrix $\mathbf{Q}$, we need to multiply each row of $\mathbf{Q}$ by the corresponding $p(x)$. This could be done easily in MATLAB by first constructing a diagonal matrix from the elements in $\underline{p}$ and then multiply this to the matrix $\mathbf{Q}$:

$$\mathbf{P} = \left(\operatorname{diag}\left(\underline{p}\right)\right)\mathbf{Q}.$$

Example 3.8. Binary Assymmetric Channel (BAC): Consider a binary input-output channel whose matrix of transition probabilities is

$$\mathbf{Q} = \left[\begin{array}{cc} 0.7 & 0.3 \\ 0.4 & 0.6 \end{array}\right].$$

(a) Draw the channel diagram.

(b) If the two inputs are equally likely, find the corresponding output probabilities and the joint probability matrix $\mathbf{P}$ for this channel.

[9, Ex. 11.3]

3.2 Decoder and Symbol Error Probability

3.9. Knowing the characteristics of the channel and its input, on the receiver side, we can use this information to build a "good" receiver.

We now consider a part of the receiver called the **(channel) decoder**. Its job is to guess the value of the channel input[11] X from the value of the received channel output Y. We denote this guessed value by $\hat{X}$. A "good" receiver is the one that (often) guesses correctly.

Quantitatively, to measure the performance of a decoder, we define a quantity called the (symbol) error probability.

Definition 3.10. The **(symbol) error probability**, denoted by $P(\mathcal{E})$, can be calculated from

$$P(\mathcal{E}) = P\left[\hat{X} \neq X\right].$$

3.11. A "good" detector should guess based on all the information it has. Here, the only information it receives is the value of Y. So, a detector is a function of Y, say, $g(Y)$. Therefore, $\hat{X} = g(Y)$. Sometimes, we also write $\hat{X}$ as $\hat{X}(Y)$ to emphasize that the decoded value $\hat{X}$ depends on the received value Y.

Definition 3.12. A "naive" decoder is a decoder that simply sets $\hat{X} = Y$.

Example 3.13. Consider the BAC channel and input probabilities specified in Example 3.8. Find $P(\mathcal{E})$ when $\hat{X} = Y$.

[11]To simplify the analysis, we still haven't considered the channel encoder. (It may be there but is included in the equivalent channel or it may not be in the system at all.)

3.14. For general DMC, the error probability of the naive decoder is

Example 3.15. With the derived formula, let's revisit Example 3.8 and Example 3.13

Example 3.16. Find the error probability $P(\mathcal{E})$ when a naive decoder is used with a DMC channel in which $\mathcal{X} = \{0, 1\}$, $\mathcal{Y} = \{1, 2, 3\}$, $\mathbf{Q} = \begin{bmatrix} 0.5 & 0.2 & 0.3 \\ 0.3 & 0.4 & 0.3 \end{bmatrix}$ and $\underline{p} = [0.2, 0.8]$.

3.3 Optimal Decoding for BSC

To introduce the idea of optimal decoding, let's revisit the binary symmetric channel in Example 3.1. Here, we will attempt to find the "best" decoder. Of course, by "best", we mean "having minimum value of error probability".

3.17. It is interesting to first consider the question of how many reasonable decoders we can use.

So, only four reasonable detectors.

Example 3.18. Suppose $p = 0$. Which detector should we use?

Example 3.19. Suppose $p = 1$. Which detector should we use?

Example 3.20. Suppose $p_0 = 0$. Which detector should we use?

Example 3.21. Suppose $p_0 = 1$. Which detector should we use?

Example 3.22. Suppose $p = 0.1$ and $p_0 = 0.8$. Which detector should we use?

3.23. To formally compare the performance of the four detectors, we now derive the formula for the error probability of the four detectors. First, we apply the total probability theorem by using the events $[X = x]$ to partition the sample space.

$$P(\mathcal{C}) = \sum_x P(\mathcal{C} \,|[X = x])\, P[X = x]$$

Of course, event $\mathcal{C}$ is the event $[\hat{X} = X]$. Therefore,

$$P(\mathcal{C}) = \sum_x P\left[\hat{X} = X \,|X = x\right] P[X = x] = \sum_x P\left[\hat{X} = x \,|X = x\right] P[X = x].$$

For binary channel, there are only two possible value of x: 0 or 1. So,

$$\begin{aligned} P(\mathcal{C}) &= \sum_{x \in \{0,1\}} P\left[\hat{X} = x \,|X = x\right] P[X = x] \\ &= P\left[\hat{X} = 0 \,|X = 0\right] P[X = 0] + P\left[\hat{X} = 1 \,|X = 1\right] P[X = 1] \\ &= P\left[\hat{X} = 0 \,|X = 0\right] p_0 + P\left[\hat{X} = 1 \,|X = 1\right] (1 - p_0) \end{aligned}$$

$\hat{X}$	$P\left[\hat{X} = 0 \,\vert X = 0\right]$	$P\left[\hat{X} = 1 \,\vert X = 1\right]$	$P(\mathcal{C})$	$P(\mathcal{E})$
Y	$1 - p$	$1 - p$	$1 - p$	p
$1 - Y$	p	p	p	$1 - p$
1	0	1	$1 - p_0$	p_0
0	1	0	p_0	$1 - p_0$

3.4 Optimal Decoding for DMC

Example 3.24. Let's return to Example 3.16 and find the error probability $P(\mathcal{E})$ when a specific decoder is used. In that example, we have $\mathcal{X} = \{0, 1\}$, $\mathcal{Y} = \{1, 2, 3\}$, $\mathbf{Q} = \begin{bmatrix} 0.5 & 0.2 & 0.3 \\ 0.3 & 0.4 & 0.3 \end{bmatrix}$ and $\underline{p} = [0.2, 0.8]$. The decoder table below specifies the decoder under consideration:

Y	$\hat{X}$
1	0
2	1
3	0

Using MATLAB, we can find the error probability for all possible reasonable detectors in this example.

3.25. For general DMC, it would be tedious to list all possible detectors. In fact, there are $|\mathcal{X}|^{|\mathcal{Y}|}$ reasonable detectors.

It is even more time-consuming to try to calculate the error probability for all of them. Therefore, in this section, we will derive the formula of the "optimal" detector.

3.26. We first note that to minimize $P(\mathcal{E})$, we need to maximize $P(\mathcal{C})$. Here, we apply the total probability theorem by using the events $[Y = y]$ to partition the sample space:

$$P(\mathcal{C}) = \sum_y P(\mathcal{C}\,|[Y = y])\, P[Y = y].$$

Event $\mathcal{C}$ is the event $[\hat{X} = X]$. Therefore,

$$P(\mathcal{C}) = \sum_y P\left[\hat{X} = X\,|Y = y\right] P[Y = y].$$

Now, recall that our detector $\hat{X}$ is a function[12] of Y; that is $\hat{X} = g(Y)$ for some function g. So,

$$\begin{aligned} P(\mathcal{C}) &= \sum_y P[g(Y) = X\,|Y = y]\, P[Y = y] \\ &= \sum_y P[X = g(y)\,|Y = y]\, P[Y = y] \end{aligned}$$

In this form, we see[13] that for each $Y = y$, we should maximize $P[X = g(y)\,|Y = y]$. Therefore, for each y, the decoder $g(y)$ should output the value of x which maximizes $P[X = x|Y = y]$:

$$g_{\text{optimal}}(y) = \arg\max_x P[X = x\,|Y = y].$$

[12]This change of notation allows one to see the dependence of $\hat{X}$ on Y.

[13]We also see that any decoder that produces random results (on the support of X) can not be better than our optimal detector. Outputting the value of x which does not maximize the a posteriori probability reduces the contribution in the sum that gives $P(\mathcal{C})$.

In other words,

$$\hat{x}_{\text{optimal}}(y) = \arg\max_x P[X = x | Y = y]$$

and

$$\hat{X}_{\text{optimal}} = \arg\max_x P[X = x | Y].$$

Definition 3.27. The <u>*optimal*</u> detector is the detector that maximizes the a posteriori probability $P[X = x | Y = y]$. This detector is called the **maximum a posteriori probability (MAP) detector**:

$$\hat{x}_{\text{MAP}}(y) = \hat{x}_{\text{optimal}}(y) = \arg\max_x P[X = x | Y = y].$$

- After the fact, it is quite intuitive that this should be the best detector.

  Recall that the decoder don't have a direct access to the X value.

  - Without knowing the value of Y, to minimize the error probability, it should guess the most likely value of X which is the value of x that maximize $P[X = x]$.
  - Knowing $Y = y$, the decoder can update its probability about x from $P[X = x]$ to $P[X = x|Y = y]$. Therefore, the detector should guess the value of the most likely x value conditioned on the fact that $Y = y$.

3.28. We can "simplify" the formula for the MAP detector even further.

Fist, recall "Form 1" of the Bayes' theorem:

$$P(B|A) = P(A|B)\frac{P(B)}{P(A)}.$$

Here, we have $B = [X = x]$ and $A = [Y = y]$.

Therefore,

$$\boxed{\hat{x}_{\text{MAP}}(y) = \arg\max_{x} Q(y|x)\,p(x)}. \tag{6}$$

3.29. A recipe to find the MAP detector and its corresponding error probability:

(a) Find the **P** matrix by scaling elements in each row of the **Q** matrix by their corresponding prior probability $p(x)$.

(b) Select (by circling) the maximum value in each column (for each value of y) in the **P** matrix.

- If there are multiple max values in a column, select only one.

(i) The corresponding x value is the value of $\hat{x}$ for that y.

(ii) The sum of the selected values from the **P** matrix is $P(\mathcal{C})$.

(c) $P(\mathcal{E}) = 1 - P(\mathcal{C})$.

Example 3.30. Find the MAP detector and its corresponding error probability for the DMC channel in Example 3.24 (and Example 3.16) when the prior probability vector is $\underline{p} = [0.2, 0.8]$.

Example 3.31. Repeat Example 3.30 but with $\underline{p} = [0.6, 0.4]$.

3.32. MAP detector for BSC: For BSC,

$$\hat{x}_{\text{MAP}}(y) = \arg\max_{x\in\{0,1\}} Q(y\,|x)\,p(x).$$

Therefore,

$$\hat{x}_{\text{MAP}}(0) = \begin{cases} 1, & \text{when } Q(0\,|1)\,p(1) > Q(0\,|0)\,p(0), \\ 0, & \text{when } Q(0\,|1)\,p(1) < Q(0\,|0)\,p(0), \end{cases}$$

and

$$\hat{x}_{\text{MAP}}(1) = \begin{cases} 1, & \text{when } Q(1\,|1)\,p(1) > Q(1\,|0)\,p(0), \\ 0, & \text{when } Q(1\,|1)\,p(1) < Q(1\,|0)\,p(0). \end{cases}$$

Definition 3.33. In many scenarios, the MAP detector is too complicated or the prior probabilities are unknown. In such cases, we may consider using a detector that ignores the prior probability term in (6). This detector is called the **maximum likelihood (ML)** detector:

$$\boxed{\hat{x}_{\text{ML}}(y) = \arg\max_{x} Q(y\,|x)}. \tag{7}$$

Observe that

- ML detector is generally sub-optimal
- ML detector is the same as the MAP detector when X is a uniform random variable.
    - In other words, when the prior probabilities $p(x)$ are uniform, the ML detector is optimal.

Example 3.34. Find the ML detector and the corresponding error probability for the system in Example 3.22 in which we have BSC with $p = 0.1$ and $p_0 = 0.8$.

Note that

- the prior probabilities p_0 (and p_1) is not used
- the ML detector and the MAP detector are the same in this example
    - ML detector can be optimal even when the prior probabilities are not uniform.

Recall that for BSC with $\hat{x}(y) = y$, the error probability $P(\mathcal{E}) = p$. So, in this example, because $\hat{x}_{\mathrm{ML}}(y) = y$, we have $P(\mathcal{E}) = p = 0.1$.

3.35. In general, for BSC, it's straightforward to show that

(a) when $p < 0.5$, we have $\hat{x}_{\mathrm{ML}}(y) = y$ with corresponding $P(\mathcal{E}) = p$.

(b) when $p > 0.5$, we have $\hat{x}_{\mathrm{ML}}(y) = 1-y$ with corresponding $P(\mathcal{E}) = 1-p$.

(c) when $p = 0.5$, all four reasonable detectors have the same $P(\mathcal{E}) = 1/2$.

- In fact, when $p = 0.5$, the channel completely destroys any connection between X and Y. In particular, in this scenario, $X \perp\!\!\!\perp Y$. So, the value of the observed y is useless.

3.36. A recipe to find the ML detector and its corresponding error probability:

(a) Select (by circling) the maximum value in each column (for each value of y) in the $\mathbf{Q}$ matrix.

- If there are multiple max values in a column, select only one.
- The corresponding x value is the value of $\hat{x}$ for that y.

(b) Find the $\mathbf{P}$ matrix by scaling elements in each row of the $\mathbf{Q}$ matrix by their corresponding prior probability $p(x)$.

(c) In the $\mathbf{P}$ matrix, select the elements corresponding to the selected positions in the $\mathbf{Q}$ matrix.

(d) The sum of the selected values from the $\mathbf{P}$ matrix is $P(\mathcal{C})$.

(e) $P(\mathcal{E}) = 1 - P(\mathcal{C})$.

Example 3.37. Solve Example 3.34 using the recipe in 3.36.

Example 3.38. Find the ML detector and its corresponding error probability for the DMC channel in Example 3.24 (, Example 3.16, and Example 3.30) in which $\mathcal{X} = \{0,1\}$, $\mathcal{Y} = \{1,2,3\}$, $\mathbf{Q} = \begin{bmatrix} 0.5 & 0.2 & 0.3 \\ 0.3 & 0.4 & 0.3 \end{bmatrix}$ and $\underline{p} = [0.2, 0.8]$.

4 An Introduction to Channel Coding and Decoding over BSC

4.1. Recall that **channel coding** introduces, in a controlled manner, some *redundancy* in the (binary) information sequence that can be used at the receiver to overcome the effects of noise and interference encountered in the transmission of the signal through the channel.

Example 4.2. Repetition Code: Repeat each bit n times, where n is some positive integer.

- Use the channel n times to transmit 1 info-bit
- The (transmission) rate is $\frac{1}{n}$ [bpcu].
  - bpcu = bits per channel use

4.3. Two classes of channel codes

(a) Block codes

- To be discussed here.
- Realized by combinational/combinatorial circuit.

(b) Convolutional codes

- Encoder has memory.
- Realized by sequential circuit. (Recall state diagram, flip-flop, etc.)

Definition 4.4. Block Encoding: Take k (information) bits at a time and map each k-bit sequence into a (unique) n-bit sequence, called a **codeword**.

- The code is called (n, k) code.
- Working with k-info-bit blocks means there are potentially $M = 2^k$ different information blocks.
    - The table that lists all the 2^k mapping from the k-bit info-block $\underline{\mathbf{s}}$ to the n-bit codeword $\underline{\mathbf{x}}$ is called the **codebook**.
    - The M info-blocks are denoted by $\underline{\mathbf{s}}^{(1)}, \underline{\mathbf{s}}^{(2)}, \dots, \underline{\mathbf{s}}^{(M)}$.
      The corresponding M codewords are denoted by $\underline{\mathbf{x}}^{(1)}, \underline{\mathbf{x}}^{(2)}, \dots, \underline{\mathbf{x}}^{(M)}$, respectively.
- To have unique codeword for each information block, we need $n \geq k$. Of course, with some redundancy added to combat the error introduced by the channel, we need $n > k$.
    - The amount of redundancy is measured by the ratio $\frac{n}{k}$.
    - The number of redundant bits is $r = n - k$.
- Here, we use the channel n times to convey k (information) bits.
    - The ratio $\frac{k}{n}$ is called the rate of the code or, simply, the **code rate**.
    - The (transmission) rate is $R = \frac{k}{n} = \frac{\log_2 M}{n}$ [bpcu].

4.5. When the mapping from the information block $\underline{\mathbf{s}}$ to the codeword $\underline{\mathbf{x}}$ is invertible, the task of the decoder can be separated into two steps:

- First, find $\hat{\underline{\mathbf{x}}}$ which is its guess of the $\underline{\mathbf{x}}$ value based on the observed value of $\underline{\mathbf{y}}$.
- Second, map $\hat{\underline{\mathbf{x}}}$ back to the corresponding $\hat{\underline{\mathbf{s}}}$ based on the codebook.

You may notice that it is more important to recover the index of the codeword than the codeword itself. Only its index is enough to indicate which info-block produced it.

4.6. General idea for ML decoding of block codes over BSC: **minimum-distance decoder**

- To recover the value of $\underline{\mathbf{x}}$ from the observed value of $\underline{\mathbf{y}}$, we can apply what we studied about optimal detector in the previous section.
  - The optimal detector is again given by the MAP detector:
  $$\hat{\underline{\mathbf{x}}}_{\mathrm{MAP}}\left(\underline{\mathbf{y}}\right) = \arg\max_{\underline{\mathbf{x}}} Q\left(\underline{\mathbf{y}}\left|\underline{\mathbf{x}}\right.\right) p\left(\underline{\mathbf{x}}\right). \tag{8}$$
  - When the prior probabilities $p\left(\underline{\mathbf{x}}\right)$ is unknown or when we want simpler decoder, we may consider using the ML decoder:
  $$\hat{\underline{\mathbf{x}}}_{\mathrm{ML}}\left(\underline{\mathbf{y}}\right) = \arg\max_{\underline{\mathbf{x}}} Q\left(\underline{\mathbf{y}}\left|\underline{\mathbf{x}}\right.\right). \tag{9}$$

  In this section, we will mainly focus on the ML decoder.

- By the memoryless property of the channel,
$$Q\left(\underline{\mathbf{y}}|\underline{\mathbf{x}}\right) = Q(y_1|x_1) \times Q(y_2|x_2) \times \cdots \times Q(y_n|x_n).$$
Furthermore, for BSC,
$$Q(y_i|x_i) = \begin{cases} p, & y_i \neq x_i, \\ 1-p, & y_i = x_i. \end{cases}$$
Therefore,
$$Q\left(\underline{\mathbf{y}}|\underline{\mathbf{x}}\right) = p^{d\left(\underline{\mathbf{x}},\underline{\mathbf{y}}\right)}(1-p)^{n-d\left(\underline{\mathbf{x}},\underline{\mathbf{y}}\right)} = \left(\frac{p}{1-p}\right)^{d\left(\underline{\mathbf{x}},\underline{\mathbf{y}}\right)}(1-p)^n, \tag{10}$$
where $d\left(\underline{\mathbf{x}},\underline{\mathbf{y}}\right)$ is the number of coordinates in which the two blocks $\underline{\mathbf{x}}$ and $\underline{\mathbf{y}}$ differ.
Note that when $p < 0.5$, which is usually the case for practical systems, we have $p < 1-p$ and hence $0 < \frac{p}{1-p} < 1$. In which case, to maximize $Q\left(\underline{\mathbf{y}}|\underline{\mathbf{x}}\right)$, we need to minimize $d\left(\underline{\mathbf{x}},\underline{\mathbf{y}}\right)$. In other words, $\hat{\underline{\mathbf{x}}}_{\mathrm{ML}}\left(\underline{\mathbf{y}}\right)$ should be the codeword $\underline{\mathbf{x}}$ which has the minimum distance from the observed $\underline{\mathbf{y}}$:
$$\hat{\underline{\mathbf{x}}}_{\mathrm{ML}}\left(\underline{\mathbf{y}}\right) = \arg\min_{\underline{\mathbf{x}}} d\left(\underline{\mathbf{x}},\underline{\mathbf{y}}\right). \tag{11}$$
In conclusion, for block coding over BSC with $p < 0.5$, the ML decoder is the same as the minimum distance decoder.

Example 4.7. Repetition Code and Majority Voting: Back to Example 4.2.

Let $\underline{\mathbf{0}}$ and $\underline{\mathbf{1}}$ denote the n-dimensional row vectors $00\ldots0$ and $11\ldots1$, respectively. Observe that

$$d\left(\underline{\mathbf{x}}, \underline{\mathbf{y}}\right) = \begin{cases} \#1 \text{ in } \underline{\mathbf{y}}, & \text{when } \underline{\mathbf{x}} = \underline{\mathbf{0}}, \\ \#0 \text{ in } \underline{\mathbf{y}}, & \text{when } \underline{\mathbf{x}} = \underline{\mathbf{1}}. \end{cases}$$

Therefore, the minimum distance detector is

$$\hat{\underline{\mathbf{x}}}_{\text{ML}}\left(\underline{\mathbf{y}}\right) = \begin{cases} \underline{\mathbf{0}}, & \text{when } \#1 \text{ in } \underline{\mathbf{y}} < \#0 \text{ in } \underline{\mathbf{y}}, \\ \underline{\mathbf{1}}, & \text{when } \#1 \text{ in } \underline{\mathbf{y}} > \#0 \text{ in } \underline{\mathbf{y}}. \end{cases}$$

Equivalently,

$$\hat{s}_{\text{ML}}\left(\underline{\mathbf{y}}\right) = \begin{cases} 0, & \text{when } \#1 \text{ in } \underline{\mathbf{y}} < \#0 \text{ in } \underline{\mathbf{y}}, \\ 1, & \text{when } \#1 \text{ in } \underline{\mathbf{y}} > \#0 \text{ in } \underline{\mathbf{y}}. \end{cases}$$

This is the same as taking a majority vote among the received bit in the $\underline{\mathbf{y}}$ vector.

The corresponding error probability is

$$P\left(\mathcal{E}\right) = \sum_{c=\lceil \frac{n}{2} \rceil}^{n} \binom{n}{c} p^c (1-p)^{n-c}.$$

For example, when $p = 0.01$, we have $P(\mathcal{E}) \approx 10^{-5}$. Figure 6 compares the error probability when different values of n are used.

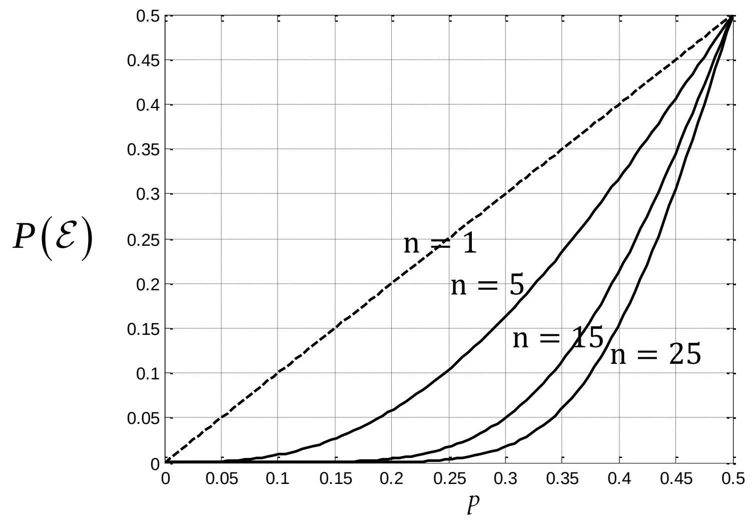

Figure 6: Error probability for a system that uses repetition code at the transmitter (repeat each info-bit n times) and majority voting at the receiver. The channel is assumed to be binary symmetric with crossover probability p.

- Notice that the error probability decreases to 0 when n is increased. It is then possible to transmit with arbitrarily low probability of error using this scheme.
- However, the (transmission) rate $R = \frac{k}{n} = \frac{1}{n}$ is also reduced as n is increased.

So, in the limit, although we can have very small error probability, we suffer tiny (transmission) rate.

We may then ask "what is the maximum (transmission) rate of information that can be *reliably* transmitted over a communications channel?" Here, reliable communication means that the error probability can be made arbitrarily small. Shannon provided the solution to this question in his seminal work. We will revisit this question in the next section.

4.8. General idea for MAP decoding of block codes over BSC: Of course, when the prior probabilities are known, the optimal decoder is given by the MAP decoder:

$$\underline{\hat{\mathbf{x}}}_{\text{MAP}}\left(\underline{\mathbf{y}}\right) = \arg\max_{\underline{\mathbf{x}}} Q\left(\underline{\mathbf{y}}\left|\underline{\mathbf{x}}\right.\right) p\left(\underline{\mathbf{x}}\right). \tag{12}$$

By definition, $p\left(\underline{\mathbf{x}}\right) \equiv P\left[\underline{\mathbf{X}} = \underline{\mathbf{x}}\right]$. Therefore,

$$p\left(\underline{\mathbf{x}}^{(i)}\right) \equiv P\left[\underline{\mathbf{X}} = \underline{\mathbf{x}}^{(i)}\right] = P\left[\underline{\mathbf{S}} = \underline{\mathbf{s}}^{(i)}\right].$$

Again, because the BSC is memoryless, from (10), we have

$$Q\left(\underline{\mathbf{y}}|\underline{\mathbf{x}}\right)=p^{d\left(\underline{\mathbf{x}},\underline{\mathbf{y}}\right)}(1-p)^{n-d\left(\underline{\mathbf{x}},\underline{\mathbf{y}}\right)}=\left(\frac{p}{1-p}\right)^{d\left(\underline{\mathbf{x}},\underline{\mathbf{y}}\right)}(1-p)^{n}, \tag{13}$$

Therefore,

$$\hat{\underline{\mathbf{x}}}_{\text{MAP}}\left(\underline{\mathbf{y}}\right)=\arg\max_{\underline{\mathbf{x}}}\left(\frac{p}{1-p}\right)^{d\left(\underline{\mathbf{x}},\underline{\mathbf{y}}\right)}(1-p)^{n}p\left(\underline{\mathbf{x}}\right) \tag{14}$$

$$=\arg\max_{\underline{\mathbf{x}}}\left(\frac{p}{1-p}\right)^{d\left(\underline{\mathbf{x}},\underline{\mathbf{y}}\right)}p\left(\underline{\mathbf{x}}\right). \tag{15}$$

Example 4.9. Consider a communication over a BSC with $p=\frac{1}{3}$. Suppose repetition code is used with $n=3$. Assume that the info-bit has $P\left[S=0\right]=\frac{3}{4}$. Find the ML and MAP decoders and their error probabilities.

n_1	n_0	n_1-n_0	$\left(\frac{p}{1-p}\right)^{n_1-n_0}$	$\times\frac{p_0}{p_1}$	$\hat{s}_{\text{MAP}}\left(\underline{\mathbf{y}}\right)$	$\hat{s}_{\text{ML}}\left(\underline{\mathbf{y}}\right)$
0						
1						
2						
3						

5 Mutual Information and Channel Capacity

In Section 2, we have seen the use of a quantity called entropy to measure the amount of randomness in a random variable. In this section, we introduce several more information-theoretic quantities. These quantities are important in the study of Shannon's results.

5.1 Information-Theoretic Quantities

Definition 5.1. Recall that, the **entropy** of a discrete random variable X is defined in Definition 2.41 to be

$$H(X) = -\sum_{x \in S_X} p_X(x) \log_2 p_X(x) = -\mathbb{E}\left[\log_2 p_X(X)\right]. \tag{16}$$

Similarly, the entropy of a discrete random variable Y is given by

$$H(Y) = -\sum_{y \in S_Y} p_Y(y) \log_2 p_Y(y) = -\mathbb{E}\left[\log_2 p_Y(Y)\right]. \tag{17}$$

In our context, the X and Y are input and output of a discrete memoryless channel, respectively. In such situation, we have introduced some new notations in Section 3.1:

Under such notations, (16) and (17) become

$$H(X) = -\sum_{x \in \mathcal{X}} p(x) \log_2 p(x) = -\mathbb{E}\left[\log_2 p(X)\right] \tag{18}$$

and

$$H(Y) = -\sum_{y \in \mathcal{Y}} q(y) \log_2 q(y) = -\mathbb{E}\left[\log_2 q(Y)\right]. \tag{19}$$

Definition 5.2. The **joint entropy** for two random variables X and Y is given by

$$H(X,Y) = -\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x,y) \log_2 p(x,y) = -\mathbb{E}\left[\log_2 p(X,Y)\right].$$

Example 5.3. Random variables X and Y have the following joint pmf matrix $\mathbf{P}$:

$$\begin{bmatrix} \frac{1}{8} & \frac{1}{16} & \frac{1}{16} & \frac{1}{4} \\ \frac{1}{16} & \frac{1}{8} & \frac{1}{16} & 0 \\ \frac{1}{32} & \frac{1}{32} & \frac{1}{16} & 0 \\ \frac{1}{32} & \frac{1}{32} & \frac{1}{16} & 0 \end{bmatrix}$$

Find $H(X)$, $H(Y)$ and $H(X,Y)$.

Definition 5.4. Conditional entropy:

(a) The (conditional) entropy of Y when we know $X = x$ is denoted by $H\left(Y\left|X = x\right.\right)$ or simply $H(Y|x)$. It can be calculated from

$$H\left(Y\left|x\right.\right) = -\sum_{y\in\mathcal{Y}} Q\left(y\left|x\right.\right)\log_2 Q\left(y\left|x\right.\right) = -\mathbb{E}\left[\log_2(Q(Y|x))\right| X = x\right].$$

- Note that the above formula is what we should expect it to be. When we want to find the entropy of Y, we use (19):

  $$H\left(Y\right) = -\sum_{y\in\mathcal{Y}} q\left(y\right)\log_2 q\left(y\right).$$

  When we have an extra piece of information that $X = x$, we should update the probability about Y from the unconditional probability $q(y)$ to the conditional probability $Q(y|x)$.

- Note that when we consider $Q(y|x)$ with the value of x fixed and the value of y varied, we simply get the whole x-row from $\mathbf{Q}$ matrix. So, to find $H\left(Y\left|x\right.\right)$, we simply find the "usual" entropy from the probability values in the row corresponding to x in the $\mathbf{Q}$ matrix.

(b) The (average) conditional entropy of Y when we know X is denoted by $H(Y|X)$. It can be calculated from

$$\begin{aligned}
H\left(Y\left|X\right.\right) &= \sum_{x\in\mathcal{X}} p\left(x\right) H\left(Y\left|x\right.\right) \\
&= -\sum_{x\in\mathcal{X}} p\left(x\right) \sum_{y\in\mathcal{Y}} Q\left(y\left|x\right.\right) \log_2 Q\left(y\left|x\right.\right) \\
&= -\sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} p\left(x,y\right) \log_2 Q\left(y\left|x\right.\right) \\
&= -\mathbb{E}\left[\log_2 Q\left(Y\left|X\right.\right)\right]
\end{aligned}$$

- Note that $Q(y|x) = \frac{p(x,y)}{p(x)}$. Therefore,

$$\begin{aligned}
H\left(Y|X\right) &= -\mathbb{E}\left[\log_2 Q\left(Y\left|X\right.\right)\right] = -\mathbb{E}\left[\log_2 \frac{p\left(X,Y\right)}{p\left(X\right)}\right] \\
&= \left(-\mathbb{E}\left[\log_2 p\left(X,Y\right)\right]\right) - \left(-\mathbb{E}\left[\log_2 p\left(X\right)\right]\right) \\
&= H\left(X,Y\right) - H\left(X\right)
\end{aligned}$$

Example 5.5. Continue from Example 5.3. Random variables X and Y have the following joint pmf matrix $\mathbf{P}$:

$$\begin{bmatrix}
\frac{1}{8} & \frac{1}{16} & \frac{1}{16} & \frac{1}{4} \\
\frac{1}{16} & \frac{1}{8} & \frac{1}{16} & 0 \\
\frac{1}{32} & \frac{1}{32} & \frac{1}{16} & 0 \\
\frac{1}{32} & \frac{1}{32} & \frac{1}{16} & 0
\end{bmatrix}$$

Find $H(Y|X)$ and $H(X|Y)$.

Definition 5.6. The **mutual information**[14] $I(X;Y)$ between two random variables X and Y is defined as

$$\begin{aligned}
I(X;Y) &= H(X) - H(X|Y) && (20)\\
&= H(Y) - H(Y|X) && (21)\\
&= H(X) + H(Y) - H(X,Y) && (22)\\
&= \mathbb{E}\left[\log_2 \frac{p(X,Y)}{p(X)\,q(Y)}\right] = \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\log\frac{p(x,y)}{p(x)\,q(y)} && (23)\\
&= \mathbb{E}\left[\log_2 \frac{P_{X|Y}(X|Y)}{p(X)}\right] = \mathbb{E}\left[\log_2 \frac{Q(Y|X)}{q(Y)}\right]. && (24)
\end{aligned}$$

- Mutual information quantifies the reduction in the uncertainty of one random variable due to the knowledge of the other.

- Mutual information is a measure of the amount of information one random variable contains about another [2, p 13].
- It is natural to think of $I(X;Y)$ as a measure of how far X and Y are from being independent.
  - Technically, it is the (Kullback-Leibler) divergence between the joint and product-of-marginal distributions.

5.7. Some important properties

(a) $H(X,Y) = H(Y,X)$ and $I(X;Y) = I(Y;X)$.
However, in general, $H(X|Y) \neq H(Y|X)$.

(b) I and H are always ≥ 0.

(c) There is a one-to-one correspondence between Shannon's information measures and set theory. We may use an **information diagram**, which

[14]The name mutual information and the notation $I(X;Y)$ was introduced by [Fano, 1961, Ch 2].

is a variation of a Venn diagram, to represent relationship between Shannon's information measures. This is similar to the use of the Venn diagram to represent relationship between probability measures. These diagrams are shown in Figure 7.

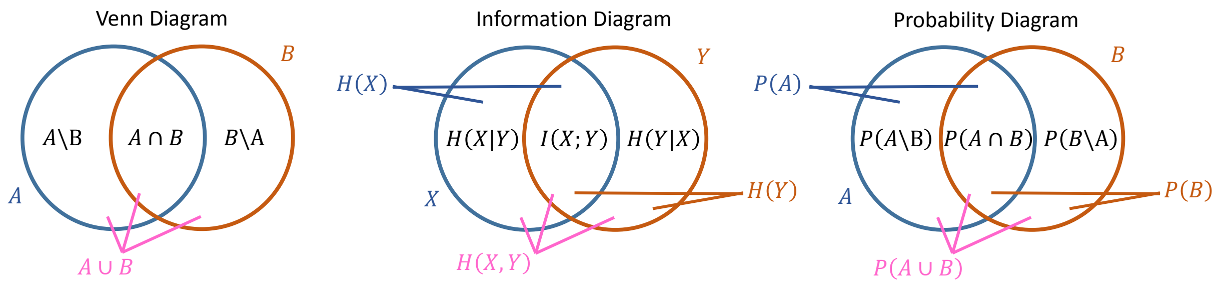

Figure 7: Venn diagram and its use to represent relationship between information measures and relationship between information measures

- Chain rule for information measures:

$$H\left(X,Y\right)=H\left(X\right)+H\left(Y\left|X\right.\right)=H\left(Y\right)+H\left(X\left|Y\right.\right).$$

(d) $I(X;Y)\geq 0$ with equality if and only if X and Y are independent.

- When this property is applied to the information diagram (or definitions (20), (21), and (22) for $I(X,Y)$), we have
  (i) $H(X|Y)\leq H(X)$,
  (ii) $H(Y|X)\leq H(Y)$,
  (iii) $H(X,Y)\leq H(X)+H(Y)$

  Moreover, each of the inequalities above becomes equality if and only $X \perp\!\!\!\perp Y$.

(e) We have seen in Section 2.4 that

$$\underset{\text{deterministic (degenerated)}}{0}\leq H\left(X\right)\leq\underset{\text{uniform}}{\log_2\left|\mathcal{X}\right|}. \tag{25}$$

Similarly,

$$\underset{\text{deterministic (degenerated)}}{0}\leq H\left(Y\right)\leq\underset{\text{uniform}}{\log_2\left|\mathcal{Y}\right|}. \tag{26}$$

For conditional entropy, we have

$$\underset{\exists g\ Y=g(X)}{0} \le H\left(Y\left|X\right.\right) \le \underset{X \perp\!\!\!\perp Y}{H\left(Y\right)} \tag{27}$$

and

$$\underset{\exists g\ X=g(Y)}{0} \le H\left(X\left|Y\right.\right) \le \underset{X \perp\!\!\!\perp Y}{H\left(X\right)}. \tag{28}$$

For mutual information, we have

$$\underset{X \perp\!\!\!\perp Y}{0} \le I\left(X;Y\right) \le \underset{\exists g\ X=g(Y)}{H\left(X\right)} \tag{29}$$

and

$$\underset{X \perp\!\!\!\perp Y}{0} \le I\left(X;Y\right) \le \underset{\exists g\ Y=g(X)}{H\left(Y\right)}. \tag{30}$$

Combining 25, 26, 29, and 30, we have

$$0 \le I\left(X;Y\right) \le \min\left\{H\left(X\right), H\left(Y\right)\right\} \le \min\left\{\log_2\left|\mathcal{X}\right|, \log_2\left|\mathcal{Y}\right|\right\} \tag{31}$$

(f) $H\left(X\left|X\right.\right) = 0$ and $I(X;X) = H(X)$.

Example 5.8. Find the mutual information $I(X;Y)$ between the two random variables X and Y whose joint pmf matrix is given by $\mathbf{P} = \begin{bmatrix} \frac{1}{2} & \frac{1}{4} \\ \frac{1}{4} & 0 \end{bmatrix}$.

Example 5.9. Find the mutual information $I(X;Y)$ between the two random variables X and Y whose $\underline{\mathbf{p}} = \left[\frac{1}{4}, \frac{3}{4}\right]$ and $\mathbf{Q} = \begin{bmatrix} \frac{1}{4} & \frac{3}{4} \\ \frac{3}{4} & \frac{1}{4} \end{bmatrix}$.

ECS452 2014/1 Part A.1 Dr.Prapun

A Multiple Random Variables

In this class, there are many occasions where we have to deal with more than one random variables at a time. Therefore, in this section, we provide a review of some important concepts that are useful for dealing with multiple random variables.

A.1 A Pair of Discrete Random Variables

Definition A.1. ***Joint pmf***: If X and Y are two discrete random variables the function $p_{X,Y}(x,y)$ defined by

$$p_{X,Y}(x,y) = P\left[X = x, Y = y\right]$$

is called the ***joint probability mass function*** of X and Y.

(a) We can visualize the joint pmf via stem plot. See Figure 8.

(b) To evaluate the probability for a statement that involves both X and Y random variables,
we first find all pairs (x,y) that satisfy the condition(s) in the statement, and then add up all the corresponding values from the joint pmf.

Definition A.2. When both X and Y take finitely many values (both have finite supports), say $S_X = \{x_1, \dots, x_m\}$ and $S_Y = \{y_1, \dots, y_n\}$, respectively, we can arrange the probabilities $p_{X,Y}(x_i, y_j)$ in an $m \times n$ matrix

$$\begin{bmatrix} p_{X,Y}(x_1, y_1) & p_{X,Y}(x_1, y_2) & \dots & p_{X,Y}(x_1, y_n) \\ p_{X,Y}(x_2, y_1) & p_{X,Y}(x_2, y_2) & \dots & p_{X,Y}(x_2, y_n) \\ \vdots & \vdots & \ddots & \vdots \\ p_{X,Y}(x_m, y_1) & p_{X,Y}(x_m, y_2) & \dots & p_{X,Y}(x_m, y_n) \end{bmatrix}. \tag{32}$$

- We shall call this matrix the ***joint pmf matrix*** and denote it by $P_{X,Y}$.
- The sum of all the entries in the matrix is one.

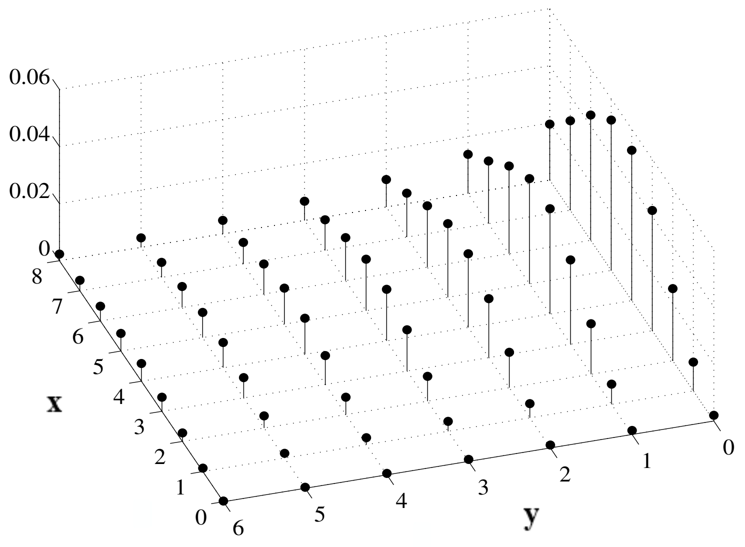

Figure 8: Example of the plot of a joint pmf. [4, Fig. 2.8]

A.3. From the joint pmf, we can find $p_X(x)$ and $p_Y(y)$ by

$$p_X(x) = \sum_y p_{X,Y}(x, y) \tag{33}$$

$$p_Y(y) = \sum_x p_{X,Y}(x, y) \tag{34}$$

In this setting, $p_X(x)$ and $p_Y(y)$ are call the ***marginal pmfs*** (to distinguish them from the joint one).

(a) Suppose we have the joint pmf matrix in (32). Then, the sum of the entries in the ith row is $p_X(x_i)$, and
the sum of the entries in the jth column is $p_Y(y_j)$:

$$p_X(x_i) = \sum_{j=1}^{n} p_{X,Y}(x_i, y_j) \quad \text{and} \quad p_Y(y_j) = \sum_{i=1}^{m} p_{X,Y}(x_i, y_j)$$

(b) In MATLAB, suppose we save the joint pmf matrix as P_XY, then the marginal pmf (row) vectors p_X and p_Y can be found by

```
p_X = (sum(P_XY,2))'
p_Y = (sum(P_XY,1))
```

Definition A.4. The ***conditional pmf*** of X given Y is defined as

$$p_{X|Y}(x|y) = P[X = x|Y = y]$$

which gives

$$p_{X,Y}(x,y) = p_{X|Y}(x|y)p_Y(y) = p_{Y|X}(y|x)p_X(x). \tag{35}$$

Definition A.5. Two random variables X and Y are said to be ***identically distributed*** if, for every B, $P[X \in B] = P[Y \in B]$.

A.6. To check whether two random varaibles X and Y are identically distributed, it is easier to test another equivalent condition: Two random variables X and Y are ***identically distributed*** if and only if

$$p_X(c) = p_Y(c) \text{ for all } c.$$

Definition A.7. Two random variables X and Y are said to be ***independent*** if the events $[X \in B]$ and $[Y \in C]$ are independent for all sets B and C.

A.8. To check whether two random varaibles X and Y are independent, it is easier to test another equivalent condition: Two random variables X and Y are ***independent*** if and only if

$$p_{X,Y}(x,y) = p_X(x) \times p_Y(y) \text{ for all } x, y.$$

Definition A.9. Two random variables X and Y are said to be ***independent and identically distributed (i.i.d.)*** if X and Y are both independent and identically distributed.

A.2 Multiple Discrete Random Variables

Here, we extend the concepts presented in the previous subsection, which deals with only two random variables, into their general form. In particular, we consider n random variables simultaneously.

A.10. Random variables $X_1, X_2, \ldots, X_n$ can be characterized by their **joint pmf**

$$p_{X_1,X_2,\ldots,X_n}(x_1, x_2, \ldots, x_n) = P\left[X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n\right].$$

They are ***identically distributed*** if and only if

$$p_{X_i}(c) = p_{X_j}(c) \text{ for all } c, i, j.$$

They are ***independent*** if and only if

$$p_{X_1,X_2,\ldots,X_n}(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} p_{X_i}(x_i) \text{ for all } x_1, x_2, \ldots, x_n.$$

Definition A.11. A ***pairwise independent*** collection of random variables is a collection of random variables any two of which are independent.

Definition A.12. You may notice that it is tedious to write the n-tuple $(X_1, X_2, \ldots, X_n)$ every time that we want to refer to this collection of random variables. A more convenient notation uses a ***vector*** $\mathbf{X}$ to represent all of them at once, keeping in mind that the ith component of $\mathbf{X}$ is the random variable X_i. This allows us to express $p_{X_1,X_2,\ldots,X_n}(x_1, x_2, \ldots, x_n)$ as $p_{\mathbf{X}}(\mathbf{x})$.

Alternatively, when it is important to show the indices of the random variables, we may write the vector in the form X_1^n or $X_{[n]}$ where $[n] = \{1, 2, \ldots, n\}$.

A.13. The expected value of "any" function g of a discrete random variable X can be calculated from

$$\mathbb{E}\left[g(X)\right] = \sum_x g(x) p_X(x).$$

Similarly[15], the expected value of "any" real-valued function g of multiple discrete random variables can be calculated from

$$\mathbb{E}\left[g(\mathbf{X})\right] = \sum_{\mathbf{x}} g(\mathbf{x}) p_{\mathbf{X}}(\mathbf{x}).$$

Note that $\mathbb{E}\left[\cdot\right]$ is a **linear** operator. In particular,

$$\mathbb{E}\left[\sum_{i=1}^{n} c_i g_i(X_i)\right] = \sum_{i=1}^{n} c_i \mathbb{E}\left[g_i(X_i)\right].$$

[15] Again, these are called the **law/rule of the lazy statistician** (LOTUS) [8, Thm 3.6 p 48],[4, p. 149] because it is so much easier to use the above formula than to first find the pmf of $g(X)$ or $g(X, Y)$. It is also called **substitution rule** [7, p 271].

References

[1] A. Bruce Carlson and Paul B. Crilly. *Communication Systems: An Introduction to Signals and Noise in Electrical Communication*. McGraw-Hill, 5th international edition edition, 2010. 2.39, 2.57

[2] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 2006. 2.28, 2.4, 2.49, 2.57, 5.6

[3] T. L. Fine. *Signals, Systems, and Networks*. 2008. 2.29, 7, 2.57

[4] John A. Gubner. *Probability and Random Processes for Electrical and Computer Engineers*. Cambridge University Press, 2006. 8, 15

[5] C. Richard Johnson Jr, William A. Sethares, and Andrew G. Klein. *Software Receiver Design: Build Your Own Digital Communication System in Five Easy Steps*. Cambridge University Press, 2011. 2.57

[6] John Proakis and Masoud Salehi. *Digital Communications*. McGraw-Hill Science/Engineering/Math, 5th edition edition, 2007. 9

[7] Henk Tijms. *Understanding Probability: Chance Rules in Everyday Life*. Cambridge University Press, 2 edition, August 2007. 15

[8] Larry Wasserman. *All of Statistics: A Concise Course in Statistical Inference*. Springer, 2004. 15

[9] Rodger E. Ziemer and William H. Tranter. *Principles of Communications*. John Wiley & Sons Ltd, 2010. 2.57, 3.8