Sirindhorn International Institute of Technology

Thammasat University

School of Information, Computer and Communication Technology

# ECS452 2016/2 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.

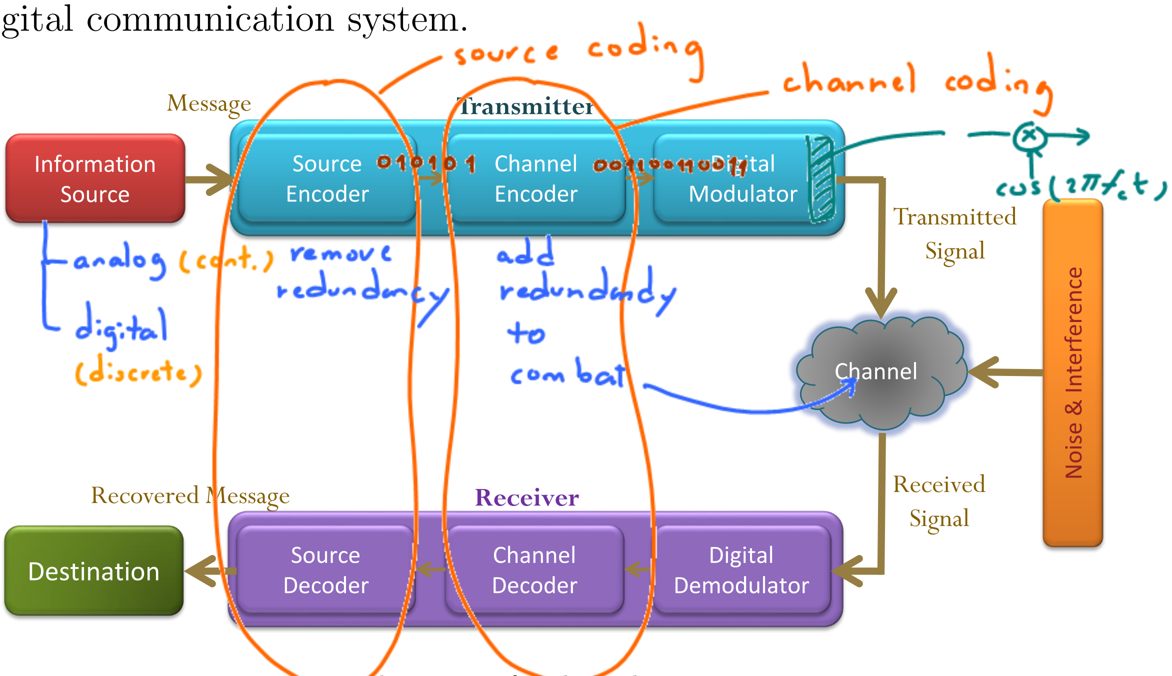


Figure 1: Basic elements of a digital communication system

**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.

In ECS332, we talked about the conversion from analog signal to bits via the process of sampling and quantization < uniform / non-uniform

① PCM → 
② DPCM
③ DM

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

- This process is called source coding or source encoding.
- For this course, we want to also represent the source output (message) by as few bits as possible.
  - In other words, we seek an efficient representation of the source output that results in little or no redundancy.
  - Therefore, source coding may be referred to as **data compression**.

(Huffman coding)

**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.

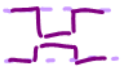
- Since nearly all the communication channels encountered in practice are capable of transmitting electrical signals (waveforms), the primary purpose of the digital modulator is to map the binary information sequence into signal waveforms.

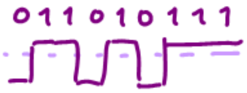
In ECS332, we studied line coding.

- 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**.

$0 \rightarrow s_0(t)$

$1 \rightarrow s_1(t)$





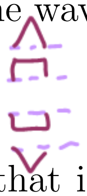
- The modulator may transmit $b$ coded information bits at a time by using $M = 2^b$ distinct waveforms $s_i(t), i = 0, 1, \ldots, M-1$, one waveform for each of the $2^b$ possible $b$-bit sequences.
  We call this ***M*-ary modulation** ($M > 2$).

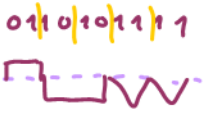
$00 \rightarrow s_0(t)$

$01 \rightarrow s_1(t)$

$10 \rightarrow s_2(t)$

$11 \rightarrow s_3(t)$





quaternary modulation ($M=4$)

**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.

distortion

Lossless

## 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 binary source coding.

so that each transmitted bit (symbol) carries max information.

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

Here, assume that the message is already available in digital format.

### 2.1 General Concepts

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

source symbols

- 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

For Thai text, we need ≈ 1-3 bits/character

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

Simplified model of the message source (to make the analysis easier)

**Definition 2.6. Discrete Memoryless Sources (DMS)**: Let us be more specific about the information source. $\boxed{\text{source}} \mapsto x_1, x_2, x_3, \dots$

- The message that the information source produces can be represented by a **vector** of characters $X_1, X_2, \dots, X_n$.
    - A perpetual message source would produce a never-ending **sequence** of characters $X_1, X_2, \dots$.
- 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,

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

        * 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

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

All RVs are i.i.d.

- 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.

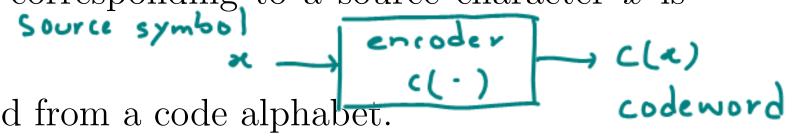
* The shared support of $X$ which is usually denoted by $S_X$ becomes the source alphabet. Note that we also often see the use of $\mathcal{X}$ to denote the support of $X$.

- 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.

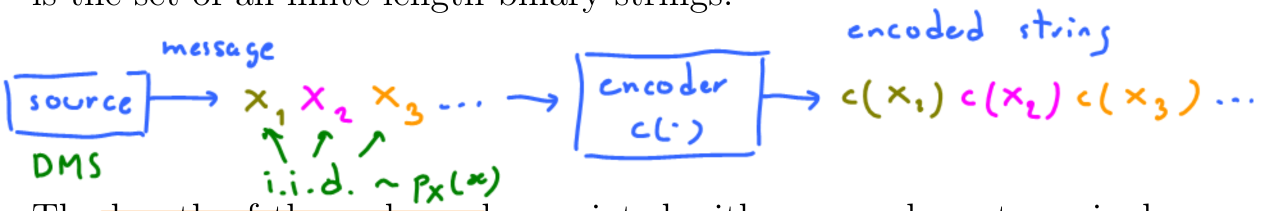
- Mathematically, we write

$$\text{Encoder } c : \overset{\text{source alphabet}}{S_X} \to \underset{\text{code alphabet}}{\{0,1\}^*}$$

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 $\ell(x)$.

codeword length

- In fact, writing this as $\ell(c(x))$ 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.

Naive coding

| $x$ | $p(x)$ | Codeword $c(x)$ | $\ell(x)$ |
|---|---|---|---|
| Yes | 4% | 00 | 2 |
| No | 3% | 01 | 2 |
| OK | 90% | 10 | 2 |
| Thank You | 3% | 11 | 2 |

$$\begin{aligned}\mathbb{E}[\ell(X)] &= 2\times 0.04 + 2\times 0.03 \\ &\quad + 2\times 0.9 + 2\times 0.03 \\ &= 2(0.04+0.03+0.9+0.03) \\ &= 2\times 1 = 2 \text{ bits}\end{aligned}$$

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

Expected code length

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

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

| $x$ | $p(x)$ | Codeword $c(x)$ | $\ell(x)$ |
|---|---|---|---|
| Yes | 4% | 01 | 2 |
| No | 3% | 001 | 3 |
| OK | 90% | 1 | 1 |
| Thank You | 3% | 0001 | 4 |

$$\begin{aligned}\mathbb{E}[\ell(X)] &= 2\times 0.04 + 3\times 0.03 \\ &\quad + 1\times 0.9 + 4\times 0.03 \\ &= \frac{1}{100}(8+9+90+12) \\ &= 1.19 \text{ bits}\end{aligned}$$

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

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

would get the same answer if it is replaced by the set $\{1, 2, 3, 4\}$, or the 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 | 1 |
| 2 | 3% | 1 | 1 |
| 3 | 90% | 0 | 1 |
| 4 | 3% | 1 | 1 |

$\mathbb{E}[\ell(X)] = 1$ bit

Not non-singular

= singular

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 the source symbol "1" or the 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.

Equivalently, for any $x_1, x_2 \in S_X$, if $x_1 \neq x_2$, then $c(x_1) \neq c(x_2)$.

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

Observe that this code is non-singular.

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

$$\mathbb{E}[\ell(X)] = 2\times 0.04 + 3\times 0.03 + 1\times 0.9 + 2\times 0.03$$
$$= 1.13 \text{ bits}$$

Problem:

When "010" is received at the decoder, how can we know whether it is
"010" or
"0" followed by "10" or
"01" followed by "0"

Not prefix-free ⇒ Not prefix code

Not UD

**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 (as in Example 2.10), or

(b) use variable-length code and

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

  0 ★ 10 ★ 010 ★ ...

  or

  (ii) use uniquely decodable codes.

**Definition 2.18.** A code is called **uniquely decodable** (UD) if any encoded string has 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. (See Example 2.28.) 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.

prefix-free 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 where all of them are leaves.

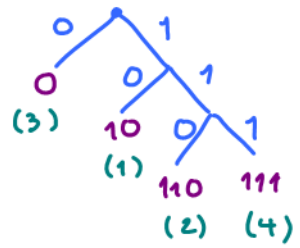
upside-down tree

[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.**

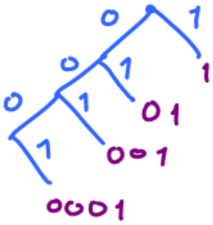
| $x$ | Codeword $c(x)$ |
|---|---|
| 1 | 10 |
| 2 | 110 |
| 3 | 0 |
| 4 | 111 |



1 0 | 0 | 1 1 0 | 1 1 0 1 1 1 1 0 0 1 0
(1) (3) (2)

**Example 2.23.** The code used in Example 2.12 is a prefix code.

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



**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.
- Therefore, another name for "prefix code" is **instantaneous code**.

**Example 2.25.** The code used in Example 2.12 (, Example 2.22, and Example 2.23) 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 | 1 |
| 2 | 10 |
| 3 | 100 |
| 4 | 1000 |

Non-singular: Yes

Prefix-free : No

c(1) is a prefix of c(2)

UD: Yes.

Observe that a "1" signifies the beginning of every codeword.

Problem: Need to look at the first bit of the next codeword to determine the end of the current one } ⇒ not instantaneous.

100/1000/1/10/100/1/1/1

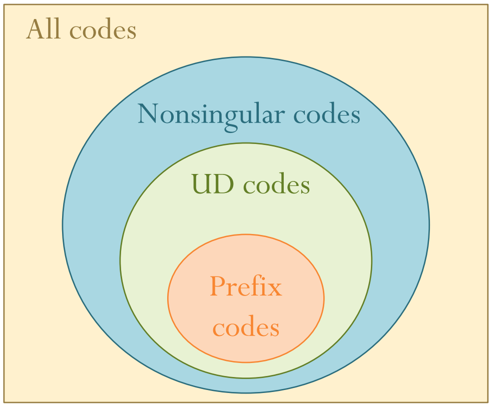


Figure 2: Classes of codes

**Example 2.28.** [5, 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 next bit is a 0, we need to count how many 0s are there before 1 shows up again.
  - 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.

Two famous prefix codes < Shannon-Fano code ← suboptimal
Huffman code ← optimal

**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. [6, 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.

Once we have a binary tree, we know this is a prefix code.

Steps
① Combine the two least-likely (combined) symbols.
② Repeat ① until you cannot repeat more.

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

[7] The Huffman algorithm performs *repeated source reduction* [6, p 63]:

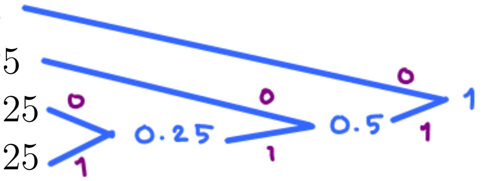
- At each step, two source symbols are combined into a new symbol, having a probability that is the sum of the probabilities of the two symbols being replaced, and the new reduced source now has one fewer symbol.
- At each step, the two symbols to combine into a new symbol have the two lowest probabilities.
  - 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 $c(x)$ | $\ell(x)$ |
|---|---|---|---|---|
| 1/2 | 1 | 0.5  | 0 | 1 |
| 1/4 | 2 | 0.25 | 10 | 2 |
| 1/8 | 3 | 0.125 | 110 | 3 |
| 1/8 | 4 | 0.125 | 111 | 3 |

$\mathbb{E}[\ell(X)] = \underbrace{1\times 0.5 + 2\times 0.25}_{1} + \underbrace{3\times 0.125 + 3\times 0.125}_{3/4} = 1.75$ bits

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,

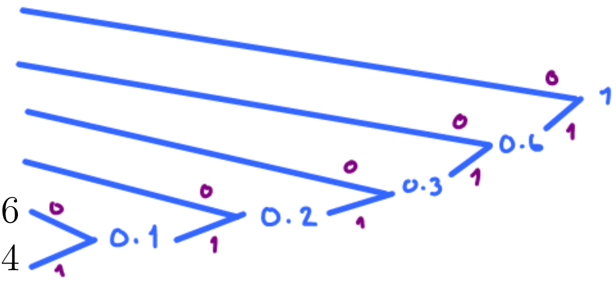
$$\ell(x) = \log_2 \frac{1}{p_X(x)} = -\log_2(p_X(x)).$$

Therefore,

$$\mathbb{E}[\ell(X)] = \sum_x p_X(x)\ell(x) = -\sum_x p_X(x) \log_2 p_X(x) \equiv H(X)$$

$$H(X) = -\sum_x p_X(x) \log_2 p_X(x)$$
$$= -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{4}\log_2\frac{1}{4} - \frac{1}{8}\log_2\frac{1}{8} - \frac{1}{8}\log_2\frac{1}{8}$$
$$= \frac{1}{2} + \frac{1}{2} + \frac{3}{4}$$
$$= 1 + \frac{3}{4} = 1.75$$
[bits]

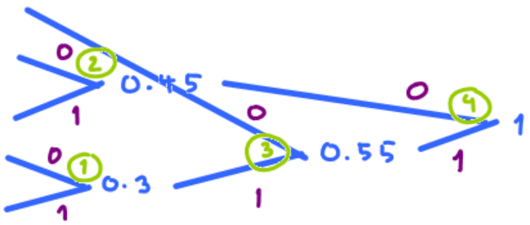
**Example 2.32.**

| $x$ | $p_X(x)$ | Codeword $c(x)$ | $\ell(x)$ |
|---|---|---|---|
| 'a' | 0.4  | 0 | 1 |
| 'b' | 0.3 | 10 | 2 |
| 'c' | 0.1 | 110 | 3 |
| 'd' | 0.1 | 1110 | 4 |
| 'e' | 0.06 | 11110 | 5 |
| 'f' | 0.04 | 11111 | 5 |

$$\mathbb{E}[\ell(X)] = 0.4\times1 + 0.3\times2 + 0.1\times3 + 0.1\times4 + 0.06\times5 + 0.04\times5$$
$$= 0.4 + 0.6 + 0.3 + 0.4 + 0.5 = 2.2 \text{ bits}$$

$H(X) = 2.1435$

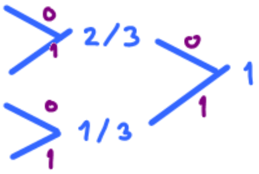
**Example 2.33.**



| $x$ | $p_X(x)$ | Codeword $c(x)$ | $\ell(x)$ |
|---|---|---|---|
| 1 | 0.25 | 10 | 2 |
| 2 | 0.25 | 00 | 2 |
| 3 | 0.2 | 01 | 2 |
| 4 | 0.15 | 110 | 3 |
| 5 | 0.15 | 111 | 3 |

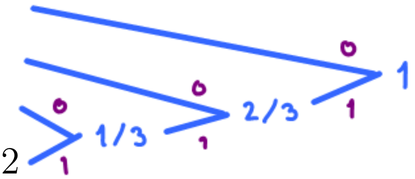
$\mathbb{E}[\ell(X)] =$ 2.3 bits

**Example 2.34.**



| $x$ | $p_X(x)$ | Codeword $c(x)$ | $\ell(x)$ |
|---|---|---|---|
| a | 1/3 | 00 | 2 |
| b | 1/3 | 01 | 2 |
| c | 1/4 | 10 | 2 |
| d | 1/12 | 11 | 2 |

$\mathbb{E}[\ell(X)] = \mathbb{E}[2] = 2$ bits



| $x$ | $p_X(x)$ | Codeword $c(x)$ | $\ell(x)$ |
|---|---|---|---|
| a | 1/3 | 0 | 1 |
| b | 1/3 | 10 | 2 |
| c | 1/4 | 110 | 3 |
| d | 1/12 | 111 | 3 |

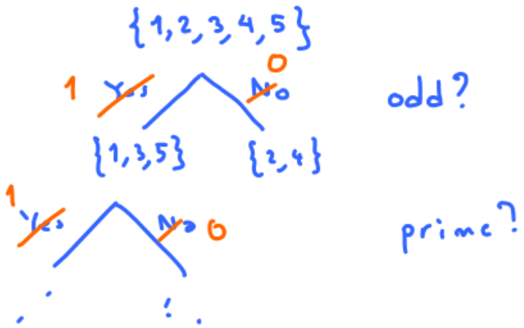
$\mathbb{E}[\ell(X)] = 1 \times \frac{1}{3} + 2 \times \frac{1}{3} + 3 \times \frac{1}{4} + 3 \times \frac{1}{12} = 2$ bits.

**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.

Theorem: **2.37.** The Huffman code is optimal.

It is important to remember that there are many optimal codes: starting from an optimal code, inverting all the bits or exchanging two codewords of the same length will give another optimal code.



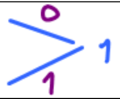
011

## 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. [4, p. 777]

**Example 2.40.**

| $x$ | $p_X(x)$ | Codeword $c(x)$ | $\ell(x)$ |
|---|---|---|---|
| Y(es) | 0.9  | 0 | 1 |
| N(o) | 0.1 | 1 | 1 |

(a) First-order extension:

$H(X) = -0.9\log_2 0.9 - 0.1\log_2 0.1$

$= 0.4685 < 1$

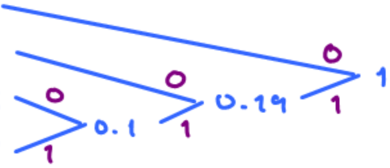
$\mathbb{E}[\ell(X)] = \mathbb{E}[1] = 1$

0 1 1 0 0 0 1 0 0 1 1 1
YNNYYYNYYNNN...
10 110 0 110 10 111

independent

For DMS, $X_1 \perp\!\!\!\perp X_2$

(b) Second-order Extension:

| $x_1x_2$ | $p_{X_1,X_2}(x_1,x_2) = p_{X_1}(x_1)\,p_{X_2}(x_2)$ | $c(x_1,x_2)$ | $\ell(x_1,x_2)$ |
|---|---|---|---|
| YY | $0.9 \times 0.9 = 0.81$  | 0 | 1 |
| YN | $0.9 \times 0.1 = 0.09$ | 10 | 2 |
| NY | $0.1 \times 0.9 = 0.09$ | 110 | 3 |
| NN | $0.1 \times 0.1 = 0.01$ | 111 | 3 |

$\mathbb{E}[\ell(X_1,X_2)] = 0.81 \times 1 + \underbrace{0.09 \times (2+3)}_{0.45} + \underbrace{0.01 \times 3}_{0.03} = 1.29$ bits per 2 source symbols

$L_2 = 0.645$ bits

(c) Third-order Extension: per source symbol

| $x_1x_2x_3$ | $p_{X_1,X_2,X_3}(x_1,x_2,x_3)$ | $c(x_1,x_2,x_3)$ | $\ell(x_1,x_2,x_3)$ |
|---|---|---|---|
| YYY | $0.9 \times 0.9 \times 0.9 = 0.729$ | | |
| YYN | $0.9 \times 0.9 \times 0.1 = 0.081$ | | |
| YNY | | | |
| ⋮ | | | |

[HW]

$\mathbb{E}[\ell(X_1,X_2,X_3)] = 1.5980$ bits per 3 source symbols

$L_3 = 0.5327$ bits per source symbol.

Sirindhorn International Institute of Technology

Thammasat University

School of Information, Computer and Communication Technology

# ECS452 2016/2 Part I.3 Dr.Prapun

## 2.4 (Shannon) Entropy for Discrete Random Variables

Entropy is a measure of uncertainty of a random variable [5, p 13].

Entropy quantifies (measures) the amount of uncertainty (randomness) a RV has.

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

Recall: $\log_2 a = \frac{\ln(a)}{\ln(2)}$

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

negative sign

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

can be omitted

$= \mathbb{E}[i(X)]$

$i(x) = -\log_2 p_X(x)$ = the amount of info. associated with each realized value of $X$.

- 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 $0 \ln 0 = 0$.

$0 \log_2 0 = 0$

L'Hôspital's rule

$$\lim_{x \to 0^+} x \ln x = \lim_{x \to 0^+} \frac{\ln x}{\frac{1}{x}} = \lim_{x \to 0^+} \frac{\frac{1}{x}}{-\frac{1}{x^2}} = \lim_{x \to 0^+} (-x) = 0$$

**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).

The probabilities involved are $\frac{1}{2}, \frac{1}{2}$

$$H([\tfrac{1}{2}\ \tfrac{1}{2}]) = H(X) = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = \frac{1}{2} + \frac{1}{2} = 1 \text{ bit}$$

Three ways for denoting entropy

**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 2 $H(p_X)$ instead of 1 $H(X)$ 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 $H(\underline{p})$. 3

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 $\{1, 2, 3, \dots, n\}$:

$$p_X(x) = \begin{cases} 1/n, & x = 1, 2, \dots, n, \\ 0, & \text{otherwise.} \end{cases}$$

$$\underline{p} = [\underbrace{\tfrac{1}{n}\ \ \tfrac{1}{n}\ \ \cdots\ \ \tfrac{1}{n}}_{n \text{ values}}]$$

$$H(X) = -\sum_x p_X(x)\log_2 p_X(x)$$

$$= -\sum_{x=1}^{n} \underbrace{\frac{1}{n}\log_2\frac{1}{n}}_{A} = -nA = -\cancel{n}\frac{1}{\cancel{n}}\log_2\frac{1}{n}$$

$$= \log_2 n = \log_2|S_X|$$

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

$$p_X(x) = \begin{cases} p, & x = 1, \\ 1-p, & x = 0, \\ 0, & \text{otherwise} \end{cases}$$

$$H(X) = -\sum_x p_X(x)\log_2 p_X(x)$$

$$= -p\log_2 p - (1-p)\log_2(1-p)$$

$$\underline{p} = [1-p \quad p]$$

Ex. $p = 0.3 \Rightarrow H(X) = 0.8813$

Ex. $p = 0.5 \Rightarrow H(X) = 1$

same $H(X)$

Binary RV

$$p_X(x) = \begin{cases} p, & x = a, \\ 1-p, & x = b, \\ 0, & \text{otherwise.} \end{cases}$$

**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.

When p = 0 or p = 1, the RV X becomes deterministic (degenerated)

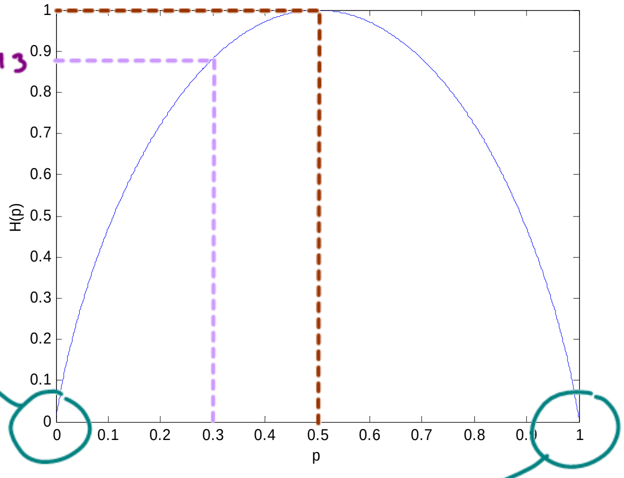


Figure 3: Binary Entropy Function

**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 $2^{-\ell(x)} = p_X(x)$. [5, 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 $c$:

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

Expected length of the optimal UD code

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

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

Expected length of the optimal UD code is the same as the expected length of Huffman code.

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

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

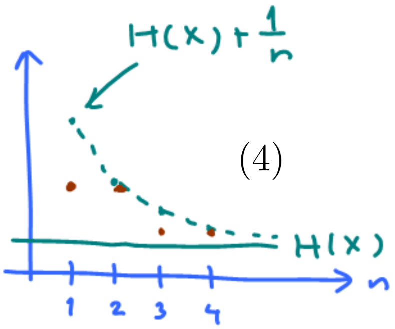
**2.53.** Combining Theorem 2.49 and Theorem 2.52, we have

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

**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 [4]
- Chapters 2 and 5 in Cover and Thomas [5]
- Chapter 4 in Fine [6]
- Chapter 14 in Johnson, Sethares, and Klein [8]
- Section 11.2 in Ziemer and Tranter [17]

Sirindhorn International Institute of Technology

Thammasat University

School of Information, Computer and Communication Technology

ECS452 2016/2 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.

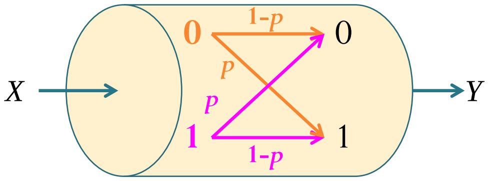


Figure 4: Binary symmetric channel and its channel diagram

- "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 crossover probability $p$.

switchover probability

bit-flip probability

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

**Example 3.2.** Consider a BSC whose samples of input and output are provided below

Time index: 1 2 3 4 5 ... 20

channel input x: 1 0 1 1 1 1 1 1 1 1 1 1 0 1 0 1 1 1 1 1

channel output y: 1 1 1 1 1 1 1 1 1 0 1 1 0 1 0 1 1 1 1 1

Estimate the following (unconditional and conditional) probabilities by their relative frequencies. $\underline{p} = [p(0) \quad p(1)] = [\frac{3}{20} \quad \frac{17}{20}]$ $\quad \mathcal{X} = \{0, 1\}$

$p(0) = P[X = 0] \approx \frac{3}{20}$ $\qquad$ $p(1) = P[X = 1] \approx 1 - P[X=0] = \frac{17}{20}$

$q(0) = P[Y = 0] \approx \frac{3}{20}$ $\qquad$ $q(1) = P[Y = 1] \approx 1 - \frac{3}{20} = \frac{17}{20}$ $\qquad$ $\underline{q} = [q(0) \quad q(1)] = [\frac{3}{20} \quad \frac{17}{20}]$

$Q(0|0) = P[Y = 0|X = 0] \approx \frac{2}{3}$ $\qquad$ $Q(1|0) = P[Y = 1|X = 0] \approx 1 - \frac{2}{3} = \frac{1}{3}$

$Q(0|1) = P[Y = 0|X = 1] \approx \frac{1}{17}$ $\qquad$ $Q(1|1) = P[Y = 1|X = 1] \approx 1 - \frac{1}{17} = \frac{16}{17}$

$$Q = \begin{array}{c|cc} x \backslash y & 0 & 1 \\ \hline 0 & 2/3 & 1/3 \\ 1 & 1/17 & 16/17 \end{array}$$

**Definition 3.3.** Our model for **discrete memoryless channel** (**DMC**) is shown in Figure 5.

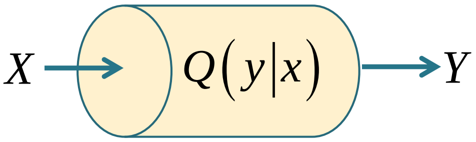


Figure 5: Discrete memoryless channel

- 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 $\underline{\mathbf{p}}$ or $\underline{\pi}$.
  - The support $S_X$ is often denoted by $\mathcal{X}$.

  ↳ channel input alphabet

- 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 $\underline{\mathbf{q}}$.
  - ○ 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 $p_{Y|X}(y|x)$. $= P[Y=y \mid X=x]$

$$Q(y|x) \equiv P[\underbrace{Y=y}_{A} | \underbrace{X=x}_{B}] = P(A|B) = \frac{P(A\cap B)}{P(B)} = \frac{P[Y=y, X=x]}{P[X=x]}$$

channel transition probability

  - ○ This conditional pmf $p_{Y|X}(y|x)$ is denoted by $Q(y|x)$ and usually expressed in the form of a probability transition matrix $\mathbf{Q}$:

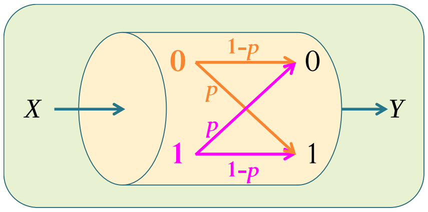
$$\begin{array}{cc} & y \\ x & \left[\begin{array}{ccc} \ddots & \vdots & \iddots \\ \cdots & P[Y=y|X=x] & \cdots \\ \iddots & \vdots & \ddots \end{array}\right] \end{array}$$

  - ○ 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.

[9] Mathematically, the condition that the channel is memoryless may be expressed as [12, Eq. 6.5-1 p. 355]

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

**Example 3.4.** For a binary symmetric channel (BSC) defined in 3.1, we now have three equivalent ways to specify the relevant probabilities:



$$P[Y=0|X=0]=Q(0|0)=1-p$$
$$P[Y=1|X=0]=Q(1|0)=p$$
$$P[Y=0|X=1]=Q(0|1)=p$$
$$P[Y=1|X=1]=Q(1|1)=1-p$$

$$\mathbf{Q}=\begin{array}{c|cc} x \backslash y & 0 & 1 \\ \hline 0 & 1-p & p \\ 1 & p & 1-p \end{array}$$

**Example 3.5.** 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

$$\mathbf{Q} = \left[\begin{array}{ccc} Q\left(y_1|\, x_1\right) & Q\left(y_2|\, x_1\right) & Q\left(y_3|\, x_1\right) \\ Q\left(y_1|\, x_2\right) & Q\left(y_2|\, x_2\right) & Q\left(y_3|\, x_2\right) \end{array}\right].$$

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.

**Example 3.6.** Consider a DMC whose samples of input and output are provided below

x: 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 0 1 1 0 1
y: 1 3 2 2 1 2 1 2 2 3 1 1 1 3 1 3 2 3 1 2

$$\overset{A \,\cap\, B}{P[X=1, Y=2]} = \frac{7}{20}$$

$$= P[\overset{B}{Y=2} \,|\, \overset{A}{X=1}]\, P[\overset{A}{X=1}]$$

$$= \frac{7}{16} \times \frac{16}{20} = \frac{7}{20}$$

Estimate its input probability vector $\underline{\mathbf{p}}$, output probability vector $\underline{\mathbf{q}}$, and $\mathbf{Q}$ matrix.

20 − 4

$$\underline{p} = [p(0) \;\; p(1)] \approx \left[\frac{4}{20} \;\; \frac{16}{20}\right] = [0.2 \;\; 0.8]$$

$$\underline{q} = [q(1) \;\; q(2) \;\; q(3)] \approx \left[\frac{8}{20} \;\; \frac{7}{20} \;\; \frac{5}{20}\right] = [0.4 \;\; 0.35 \;\; 0.25]$$

$$\mathbf{Q} = \begin{array}{c|ccc} x \backslash y & 1 & 2 & 3 \\ \hline 0 & \frac{3}{4} & \frac{0}{4} & \frac{1}{4} \\ 1 & \frac{5}{16} & \frac{7}{16} & \frac{4}{16} \end{array}$$

$$Q(y|x) = P[Y=y \,|\, X=x]$$

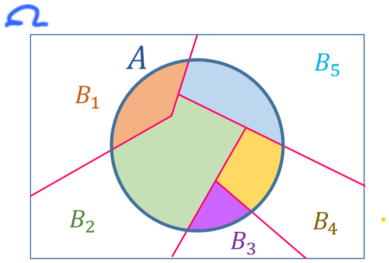
[10] The receiver knows which bits have been erased.

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

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

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 $\Omega$, then

$$P(A) = \sum_{\overset{x}{i}} P(A \cap B_{\overset{x}{i}}) = \sum_{\overset{x}{i}} P(A|B_{\overset{x}{i}})P(B_{\overset{x}{i}}). \qquad (5)$$



$$P(A) = P(A \cap B_1) + P(A \cap B_2)$$
$$+ P(A \cap B_3) + P(A \cap B_4) + P(A \cap B_5)$$

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

$A = [Y=y]$
$B_x = [X=x]$

$$q(y) = P[Y = y] = \sum_x P[X = x, Y = y]$$
$$= \sum_x P[Y = y|X = x]\, P[X = x] = \sum_x Q(y|x)p(x).$$

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

**Example 3.8.** For a binary symmetric channel (BSC) defined in 3.1,

$$q(0) = P[Y = 0] = P[Y = 0, X = 0] + P[Y = 0, X = 1]$$
$$= P[Y = 0|\, X = 0]\, P[X = 0] + P[Y = 0|\, X = 1]\, P[X = 1]$$
$$= Q(0|0)\, p(0) + Q(0|1)\, p(1) = (1 - p) \times p_0 + p \times p_1 \quad = \begin{bmatrix} p(0) & p(1) \end{bmatrix} \begin{bmatrix} Q(0|0) \\ Q(0|1) \end{bmatrix}$$
$$q(1) = P[Y = 1] = P[Y = 1, X = 0] + P[Y = 1, X = 1]$$
$$= P[Y = 1|\, X = 0]\, P[X = 0] + P[Y = 1|\, X = 1]\, P[X = 1]$$
$$= Q(1|0)\, p(0) + Q(1|1)\, p(1) = p \times p_0 + (1 - p) \times p_1 \quad = \begin{bmatrix} p(0) & p(1) \end{bmatrix} \begin{bmatrix} Q(1|0) \\ Q(1|1) \end{bmatrix}$$

first column of Q matrix

second column of Q matrix

$$\underline{q} = \begin{bmatrix} q(0) & q(1) \end{bmatrix} = \begin{bmatrix} \underline{p}\begin{bmatrix} \ \end{bmatrix} & \underline{p}\begin{bmatrix} \ \end{bmatrix} \end{bmatrix} = \underline{p} \overbrace{\begin{bmatrix} \begin{bmatrix} \ \end{bmatrix} & \begin{bmatrix} \ \end{bmatrix} \end{bmatrix}}^{Q}$$

block matrix multiplication (factoring)

$$\underline{q} = \begin{bmatrix} \quad & q(y) & \quad \end{bmatrix} = \underline{p}\, Q = \underbrace{\begin{bmatrix} \quad\quad\quad\quad \end{bmatrix}}_{\underline{p}} \underset{Q}{\begin{bmatrix} \quad & \overset{y}{\boxed{\begin{matrix} \\ \\ \\ \end{matrix}}} & \quad \end{bmatrix}}$$

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

$$\mathbf{P} = \begin{array}{c} \\ x \end{array} \begin{array}{c} y \\ \begin{bmatrix} \ddots & \vdots & \iddots \\ \cdots & P[X = x, Y = y] & \cdots \\ \iddots & \vdots & \ddots \end{bmatrix} \end{array}.$$

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

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \qquad A = [Y = y] \quad B = [X = x]$$

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

$$P[Y = y | X = x] = \frac{P[X = x, Y = y]}{P[X = x]}$$

Equivalently,

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

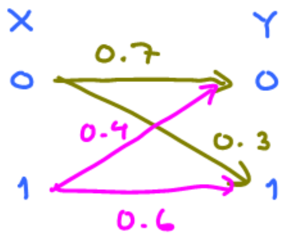
Therefore, to get the matrix **P** from matrix **Q**, we need to multiply each row of **Q** by the corresponding $p(x)$. This could be done easily in MATLAB by first constructing a diagonal matrix from the elements in $\underline{\mathbf{p}}$ and then multiply this to the matrix **Q**:

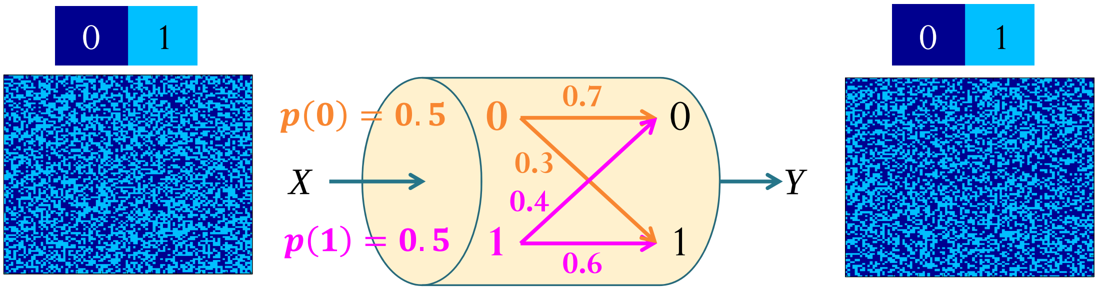
$$\mathbf{P} = \left(\operatorname{diag}\left(\underline{\mathbf{p}}\right)\right)\mathbf{Q}.$$

**Example 3.10. Binary Asymmetric Channel (BAC)**: Consider a binary input-output channel whose matrix of transition probabilities is

$$\mathbf{Q} = \begin{array}{c|cc} x \backslash y & 0 & 1 \\ \hline 0 & 0.7 & 0.3 \\ 1 & 0.4 & 0.6 \end{array}.$$

(a) Draw the channel diagram.





$$\underline{p} = \left[\tfrac{1}{2} \quad \tfrac{1}{2}\right]$$

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

$$\underline{q} = \underline{p}Q = \left[\tfrac{1}{2} \quad \tfrac{1}{2}\right]\begin{bmatrix}0.7 & 0.3\\ 0.4 & 0.6\end{bmatrix} = [0.55 \quad 0.45]$$

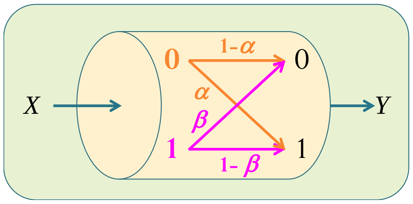
$$Q = \begin{bmatrix}0.7 & 0.3\\ 0.4 & 0.6\end{bmatrix} \begin{array}{c}\xrightarrow{\times p(0)}\\ \xrightarrow[\times p(1)]{}\end{array} \begin{array}{c|cc} x\backslash y & 0 & 1\\ \hline 0 & 0.35 & 0.15\\ 1 & 0.2 & 0.3\end{array} = P$$

$$\downarrow\Sigma \qquad \downarrow\Sigma$$

$$[0.55 \quad 0.45] = \underline{q}$$

[17, Ex. 11.3]

**Example 3.11.** Similar to Example 3.4 where we have three equivalent ways to specify BSC. We also have three different ways to describe BAC:



$P[Y=0|X=0] = Q(0|0) = 1-\alpha$

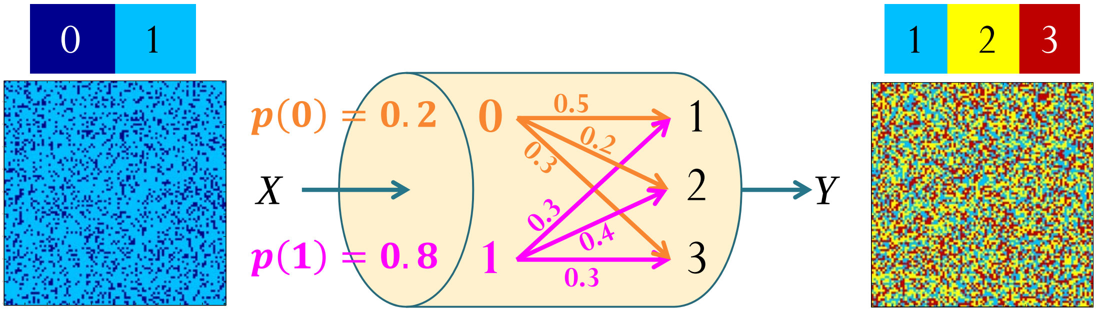
$P[Y=1|X=0] = Q(1|0) = \alpha$

$P[Y=0|X=1] = Q(0|1) = \beta$

$P[Y=1|X=1] = Q(1|1) = 1-\beta$

$$\mathbf{Q} = \begin{array}{c|cc} x\backslash y & 0 & 1\\ \hline 0 & 1-\alpha & \alpha\\ 1 & \beta & 1-\beta\end{array}$$

**Example 3.12.** Find the output probability vector $\underline{\mathbf{q}}$ and the joint probability matrix $\mathbf{P}$ for the following DMC:



$$Q = \begin{array}{c|ccc} x\backslash y & 1 & 2 & 3\\ \hline 0 & 0.5 & 0.2 & 0.3\\ 1 & 0.3 & 0.4 & 0.3\end{array} \begin{array}{c}\xrightarrow{\times 0.2}\\ \xrightarrow[\times 0.8]{}\end{array} \begin{array}{c|ccc} x\backslash y & 1 & 2 & 3\\ \hline 0 & 0.10 & 0.04 & 0.06\\ 1 & 0.24 & 0.32 & 0.24\end{array} = P$$

$$\downarrow\Sigma \qquad \downarrow\Sigma \qquad \downarrow\Sigma$$

$$\underline{q} = [0.34 \quad 0.36 \quad 0.30]$$

# Summary:

DMC: X: Channel Input
Y: Channel Output

Notation used in digital commu. (and info. theory) class is different from probability class

Channel input alphabet

$S_X \to \mathcal{X}$

$S_Y \to \mathcal{Y}$

Channel output alphabet

$P[X=x]$

$p_X(x) \to p(x) \longrightarrow \underline{p}$

$p_Y(y) \to q(y) \longrightarrow \underline{q}$

$P[Y=y]$

$P[Y=y|X=x]$

$p_{Y|X}(y|x) \to Q(y|x) \longrightarrow$ Q matrix

$p_{X,Y}(x,y) \to p(x,y) \longrightarrow$ P matrix

"AND"

$P[X=x, Y=y]$

Put together all values at correrspondng positions in the matrix

When the alphabets are lists of integers,

we usually write $p(x)$ and $q(y)$ as $p_x$ and $q_y$ respectively.

Ex. For BSC, $\mathcal{X} = \{0,1\}$
$p_0 = p(0) = P[X=0]$
$p_1 = p(1) = P[X=1]$

Alternatively, when the members of the alphabet(s) are explicitly indexed,

we often define $p_i \equiv p(x_i)$ and $q_j \equiv q(y_j)$

DMC is defined by its Q matrix:

When $\mathcal{X} = \{x_1, x_2, x_3, \ldots, x_m\}$ and $\mathcal{Y} = \{y_1, y_2, y_3, \ldots, y_n\}$,

$$Q = \begin{array}{c|ccccc} x \backslash y & y_1 & \cdots & y_j & \cdots & y_n \\ \hline x_1 & & & & & \\ \vdots & & & & & \\ x_i & & & Q(y_i|x_i) & & \\ \vdots & & & & & \\ x_m & & & & & \end{array} \quad \begin{array}{c} \xrightarrow{\times p(x_1)} \\ \vdots \\ \xrightarrow{\times p(x_i)} \\ \vdots \\ \xrightarrow{\times p(x_m)} \end{array} \quad \begin{array}{c|ccccc} x \backslash y & y_1 & \cdots & y_j & \cdots & y_n \\ \hline x_1 & & & & & \\ \vdots & & & & & \\ x_i & & & p_{X,Y}(x_i,y_j) & & \\ \vdots & & & & & \\ x_m & & & & & \end{array} = P$$

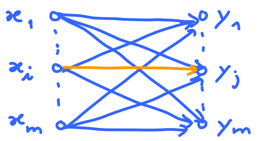
$$\Sigma\downarrow \qquad \Sigma\downarrow \qquad \Sigma\downarrow$$

$$\left[ \qquad\qquad\qquad \right] = \underline{q}$$

The P matrix can be derived from the Q matrix by scaling each row of the Q matrix by the corresponding p(x)

Alternatively, $\underline{q} = \underline{p}Q$

Equivalently, we may define a DMC via its channel diagram (of transition probabilities):

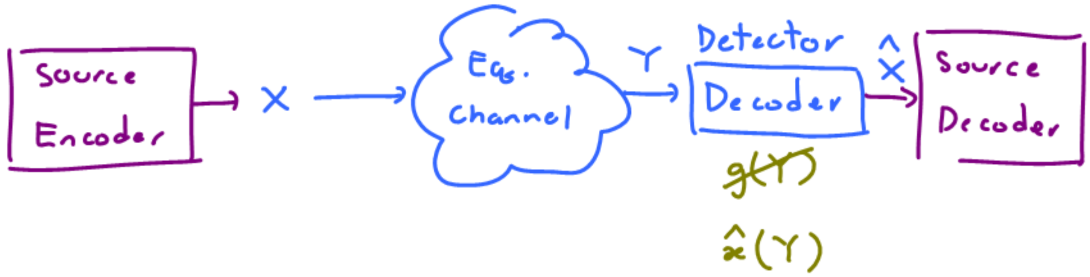


Each arrow should be labeled with $Q(y_j|x_i)$.

## 3.2 Decoder and Symbol Error Probability

**3.13.** 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 $\hat{X}$.



**3.14.** A "good" receiver is the one that (often) guesses correctly. So, our goal here is to

- maximize the probability of correct guessing
- minimize " " " wrong guessing

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

**Definition 3.15.** The **(symbol) error probability**, denoted by $P(\mathcal{E})$, can be calculated from

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

**3.16.** A "good" detector should guess based on all the information it has obtained. Here, the only information it can observe is the value of $Y$. So, a detector is a function of $Y$, say, $g(Y)$. Therefore, $\hat{X} = g(Y)$.

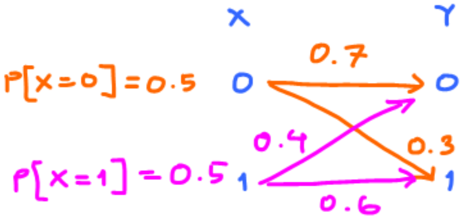
We will write $\hat{X}$ as $\hat{x}(Y)$ to emphasize that the decoded value $\hat{X}$ depends on the observed value $Y$ and that the detector is a deterministic function of the channel output $Y$; the randomness in the decoded value $\hat{X}$ comes from the randomness in $Y$.

**Definition 3.17.** A "naive" decoder is a decoder that simply sets $\hat{X} = Y$.

"pass-through"
"do-nothing"

[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.)

**Example 3.18.** Consider the BAC channel and input probabilities specified in Example 3.10. Find $P(\mathcal{E})$ when $\hat{X} = Y$.



$$P(\mathcal{E}) \equiv P[\hat{X} \neq X] = P[Y \neq X]$$
$$= 0.2 + 0.15 = 0.35$$

$$Q = \begin{bmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{bmatrix} \begin{array}{c} \xrightarrow{\times p(0)} \\ \xrightarrow[\times p(1)]{} \end{array} \begin{array}{c|cc} x\backslash y & 0 & 1 \\ \hline 0 & 0.35 & \underline{0.15} \\ 1 & \underline{0.2} & 0.3 \end{array} = P$$

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

$$\begin{aligned} P(\mathcal{E}) &= P\left[\hat{X} \neq X\right] = P[Y \neq X] = 1 - P[Y = X] \\ &= 1 - \sum_x P[Y = x, X = x] = 1 - \sum_x P[Y = x | X = x] P[X = x] \\ &= 1 - \sum_x Q(x|x)\, p(x) \end{aligned}$$

**Example 3.20.** With the derived formula, let's revisit Example 3.18.

$$P(\mathcal{E}) = 1 - (Q(0|0)\,p(0) + Q(1|1)\,p(1)) = 1 - \left(0.7 \times \frac{1}{2} + 0.6 \times \frac{1}{2}\right) = 0.35.$$

**Example 3.21.** 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 $\underline{\mathbf{p}} = [0.2, 0.8]$.

From Example 3.12,

$$Q = \begin{array}{c|ccc} x\backslash y & 1 & 2 & 3 \\ \hline 0 & 0.5 & 0.2 & 0.3 \\ 1 & 0.3 & 0.4 & 0.3 \end{array} \begin{array}{c} \xrightarrow{\times 0.2} \\ \xrightarrow[\times 0.8]{} \end{array} \begin{array}{c|ccc} x\backslash y & 1 & 2 & 3 \\ \hline 0 & \underline{0.10} & \underline{0.04} & \underline{0.06} \\ 1 & \boxed{0.24} & \underline{0.32} & \underline{0.24} \end{array} = P$$

$$\begin{aligned} P(\mathcal{E}) &\equiv P[\hat{X} \neq X] \\ &= P[Y \neq X] \\ &= 1 - 0.24 = 0.76 \end{aligned}$$

$$\begin{aligned} P(\mathcal{C}) &= P[Y = X] \\ &= 0.24 \end{aligned}$$

**Example 3.22.** DIY Decoder: Consider a different decoder specified in the decoding table below. Find the error probability $P(\mathcal{E})$ when such decoder is used in Example 3.21.

Decoding table →

| $y$ | $\hat{x}(y)$ |
|---|---|
| 1 | 0 |
| 2 | 1 |
| 3 | 0 |

$$\begin{array}{c|ccc} \hat{x} & 0 & 1 & 0 \\ x \backslash y & 1 & 2 & 3 \\ \hline 0 & \boxed{0.10} & 0.04 & \boxed{0.06} \\ 1 & 0.24 & \boxed{0.32} & 0.24 \end{array} = P$$

$$P(\mathcal{C}) = P[\hat{X} = X] = 0.1 + 0.32 + 0.06 = 0.48$$

$$P(\mathcal{E}) = P[\hat{X} \neq X] = 1 - 0.48 = 0.52$$

**Example 3.23.** Repeat Example 3.22 but use the following decoder

| $y$ | $\hat{x}(y)$ |
|---|---|
| 1 | 1 |
| 2 | 1 |
| 3 | 0 |

$$\begin{array}{c|ccc} \hat{x} & 1 & 1 & 0 \\ x \backslash y & 1 & 2 & 3 \\ \hline 0 & 0.10 & 0.04 & \boxed{0.06} \\ 1 & \boxed{0.24} & \boxed{0.32} & 0.24 \end{array} = P$$

$$P(\mathcal{C}) = 0.24 + 0.32 + 0.06 = 0.62$$

$$P(\mathcal{E}) = 1 - P(\mathcal{C}) = 1 - 0.62 = 0.38$$

Observation: For each column of the $\mathbf{P}$ matrix, we circle the probability corresponding to the row of $x$ that has the same value as $\hat{x}(y)$.

**3.24.** A recipe for finding $P(\mathcal{E})$ of any (DIY) decoder:

(a) Find the $\mathbf{P}$ matrix by scaling each row of the $\mathbf{Q}$ matrix by its corresponding $p(x)$.

(b) Write $\hat{x}(y)$ values on top of the $y$ values for the $\mathbf{P}$ matrix.

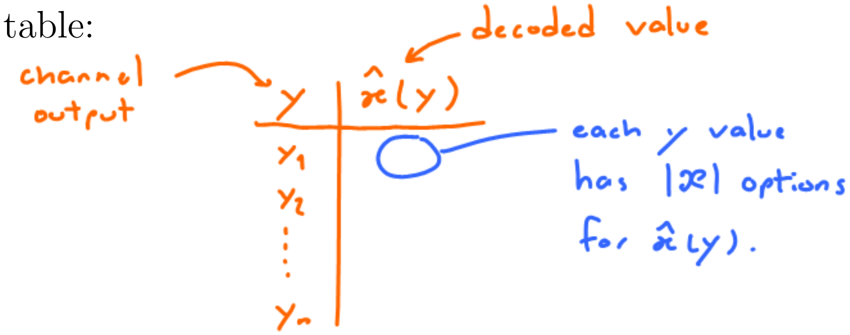
(c) For each $y$ column in the $\mathbf{P}$ matrix, circle the element whose corresponding $x$ value is the same as $\hat{x}(y)$.

(d) $P(\mathcal{C}) =$ the sum of the circled probabilities. $P(\mathcal{E}) = 1 - P(\mathcal{C})$.

## 3.3 Optimal Decoding for DMC

From the previous section, we now know how to compute the error probability for any given decoder. Here, we will attempt to find the "best" decoder. Of course, by "best", we mean "having minimum value of error probability". It is interesting to first consider the question of how many reasonable decoders we can use.

**Definition 3.25. "Reasonable" Decoder**: Recall from 3.16 that a decoder $\hat{x}(\cdot)$ is a function that map each observed valued of the channel output $y$ to the guessed value of the channel input. Therefore we can think of a decoder as a table:



We have already seen this table representation in Example 3.22 and Example 3.23. Such table has $|\mathcal{Y}|$ rows. For each value of $y$, we need to specify what is the value of $\hat{x}(y)$. To have a chance of correct guessing, any "reasonable" decoder would select the value of $\hat{x}(y)$ from $\mathcal{X}$. Therefore, there are $|\mathcal{X}|^{|\mathcal{Y}|}$ reasonable decoders.

**Example 3.26.** The naive decoder in Example 3.21 is not a reasonable decoder. The channel input $X$ is either 0 or 1. So, it does not make sense have a guess value of $\hat{x}(2) = 2$ or $\hat{x}(3) = 3$.

**Example 3.27.** "Reasonable" Decoder for BSC: For BSC in Example 3.1, any decoder has to answer two important questions:

(a) What should be the guess value of $X$ when $Y = 0$ is observed?

$\hat{x}(0) = 0$ or $\hat{x}(0) = 1$

(b) What should be the guess value of $X$ when $Y = 1$ is observed?

$\hat{x}(1) = 0$ or $\hat{x}(1) = 1$

Essentially, any reasonable decoder for the BSC need to complete this table:

| $y$ | $\hat{x}(y)$ |
|---|---|
| 0 | |
| 1 | |

So, only four reasonable decoders for BSC.

①

| $y$ | $\hat{x}(y)$ |
|---|---|
| 0 | 0 |
| 1 | 1 |

Naive

$\hat{x}(y) = y$

②

| $y$ | $\hat{x}(y)$ |
|---|---|
| 0 | 1 |
| 1 | 0 |

$\hat{x}(y) = 1 - y = \bar{y}$

③

| $y$ | $\hat{x}(y)$ |
|---|---|
| 0 | 0 |
| 1 | 0 |

$\hat{x}(y) \equiv 0$

④

| $y$ | $\hat{x}(y)$ |
|---|---|
| 0 | 1 |
| 1 | 1 |

$\hat{x}(y) \equiv 1$

**Example 3.28.** For the DMC defined in Example 3.21, how many reasonable decoders are there?

$$\underbrace{|\mathcal{X}| \times |\mathcal{X}| \times \cdots \times |\mathcal{X}|}_{|\mathcal{Y}| \text{ times}} = |\mathcal{X}|^{|\mathcal{Y}|}$$

| $y$ | $\hat{x}(y)$ |
|---|---|
| 1 | 0 or 1 |
| 2 | 0 or 1 |
| 3 | 0 or 1 |

$2^3 = 8$

We calculate the error probability of three decoders in Example 3.21, Example 3.22, and Example 3.23. There are still many possibilities to evaluate. Using `MATLAB`, we can find the error probability for all possible reasonable decoders.

**3.29.** For general DMC, it would be tedious to list all possible decoders. It is even more time-consuming to try to calculate the error probability for all of them. Therefore, in this section, we will derive a visual construction and a formula of the "optimal" decoder.

**3.30.** From the recipe 3.24 for finding $P(\mathcal{C})$ and $P(\mathcal{E})$, we see that $P(\mathcal{C})$ is the sum of our circled numbers. So, to maximize $P(\mathcal{C})$, we want to circle the largest number. For row $y$ in the decoding table, whatever the value we select for $\hat{x}(y)$ will determine which number will be circled in the column corresponding to $y$ in matrix $\mathbf{P}$. To maximize $P(\mathcal{C})$, we want to circle the largest number in the column. This means $\hat{x}(y)$ should be the same as the $x$ value that maximizes the probability value in the corresponding column.

**Example 3.31.** For the DMC and the input probablities defined in Example 3.21, the joint pmf matrix $\mathbf{P}$ was found to be

②

$$\begin{array}{cc} \hat{x} & \begin{array}{ccc} \underline{1} & \underline{1} & \underline{1} \end{array} \\ x \setminus y & \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \mathbf{P} = \begin{array}{c} 0 \\ 1 \end{array} & \left[\begin{array}{ccc} 0.1 & 0.04 & 0.06 \\ 0.24 & 0.32 & 0.24 \end{array}\right] \end{array}$$

③

Therefore, the optimal decoder is

① Decoder table

④

| $y$ | $\hat{x}(y)$ |
|---|---|
| 1 | 1 |
| 2 | 1 |
| 3 | 1 |

$P(\mathcal{C}) = 0.24 + 0.32 + 0.24 = 0.8$

$P(\mathcal{E}) = 1 - 0.8 = 0.2$

**3.32.** Mathematically, 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 decoder is a function of $Y$; that is $\hat{X}=\hat{x}(Y)$. So,

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

In this form, we see[12] that for each $Y=y$, we should maximize $P[X=\hat{x}(y)\,|Y=y]$. Therefore, for each $y$, the decoder $\hat{x}(y)$ should output the value of $x$ which maximizes $P[X=x|Y=y]$:

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

In other words, the *optimal* decoder is the decoder that maximizes the "a posteriori probability" $P[X=x\,|Y=y]$.

**Definition 3.33.** The optimal decoder derived in 3.32 is called the **maximum a posteriori probability (MAP) decoder**:

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

↑

Ex. $5-x^2=f(x)$

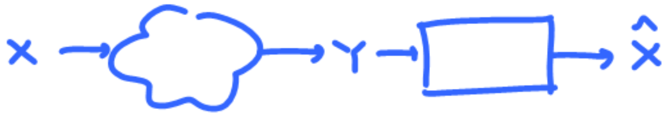
$\max_x f(x) = 5$

$\arg\max_x f(x) = 0$

**3.34.** After the fact, it is quite intuitive that this should be the best decoder. 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 decoder should guess the value of the most likely $x$ value conditioned on the fact that $Y=y$.

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



**3.35.** We can "simplify" the formula for the MAP decoder even further.

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

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

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

$$P[X=x|Y=y] = P[Y=y|X=x]\frac{P[X=x]}{P[Y=y]}$$

Does not depend on $x$ and $>0$. so can ignore.

Therefore,

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

**3.36.** A recipe for finding the MAP decoder and its corresponding error probability:

optimal

(a) Find the $\mathbf{P}$ matrix by scaling elements in each row of the $\mathbf{Q}$ 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 $\mathbf{P}$ matrix.

- If there are multiple max values in a column, select one. This won't affect the optimality of your answer.

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

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

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

**Example 3.37.** We have applied recipe 3.36 back when we try to find the optimal decoder in Example 3.31.

optimal

**Example 3.38.** Find the MAP decoder and its corresponding error probability for the DMC channel whose $\mathbf{Q}$ matrix is given by

$$\mathbf{Q} = \begin{array}{c c} x\setminus y & \begin{matrix} 1 & 2 & 3 \end{matrix} \\ \begin{matrix} 0 \\ 1 \end{matrix} & \begin{bmatrix} 0.5 & 0.2 & 0.3 \\ 0.3 & 0.4 & 0.3 \end{bmatrix} \end{array} \begin{matrix} \xrightarrow{\times 0.6} \\ \xrightarrow[\times 0.4]{} \end{matrix} \begin{array}{c c} \hat{x} & \begin{matrix} 0 & 1 & 0 \end{matrix} \\ x\setminus y & \begin{matrix} 1 & 2 & 3 \end{matrix} \\ \begin{matrix} 0 \\ 1 \end{matrix} & \begin{bmatrix} \boxed{0.3} & 0.12 & \boxed{0.18} \\ 0.12 & \boxed{0.16} & 0.12 \end{bmatrix} \end{array} = \mathbf{P}$$

and $\underline{\mathbf{p}} = [0.6, 0.4]$. Note that the DMC is the same as in Example 3.21 but the input probabilities are different.

$P(0)$ $P(1)$

$P(\mathcal{C}) = 0.3 + 0.16 + 0.18 = 0.64$

$P(\mathcal{E}) = 1 - 0.64 = 0.36$

| $y$ | $\hat{x}(y)$ |
|---|---|
| 1 | 0 |
| 2 | 1 |
| 3 | 0 |

**Definition 3.39.** In many scenarios, the MAP decoder is too complicated or the prior probabilities are unknown. In such cases, we may consider using a *suboptimal* decoder that ignores the prior probability term in (6). This decoder is called the **maximum likelihood (ML)** decoder:

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

**3.40.** ML decoder is the same as the optimal MAP decoder when $X$ is a uniform random variable. In other words, when the prior probabilities $p(x)$ are uniform, the ML decoder is optimal.

**3.41.** A recipe for finding the ML decoder 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 one. Different choices will lead to different $P(\mathcal{E})$. However, if the information about $\underline{\mathbf{p}}$ is not available at the decoder, it can not determine which choice is better anyway.
- 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 **P** matrix, select the elements corresponding to the selected positions in the **Q** matrix.

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

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

**Example 3.42.** Find the ML decoder and its corresponding error probability a the DMC channel in Example 3.21 whose **Q** matrix is

$$Q = \begin{array}{c|ccc} \hat{x}_{ML} & 0 & 1 & 0 \\ x \setminus y & 1 & 2 & 3 \\ 0 & \boxed{0.5} & 0.2 & \boxed{0.3} \\ 1 & 0.3 & \boxed{0.4} & 0.3 \end{array} \xrightarrow[\times 0.8]{\times 0.2} \begin{bmatrix} \boxed{0.1} & 0.04 & \boxed{0.06} \\ 0.24 & \boxed{0.32} & 0.24 \end{bmatrix} = P$$

| $y$ | $\hat{x}_{ML}(y)$ |
|---|---|
| 1 | 0 |
| 2 | 1 |
| 3 | 0 |

and $\underline{\mathbf{p}} = [0.2, 0.8]$.

$P(\mathcal{C}) = 0.1 + 0.32 + 0.06 = 0.48$

$P(\mathcal{E}) = 1 - 0.48 = 0.52$

**Example 3.43.** Find the ML decoder and the corresponding error probability for a communication over BSC with $p = 0.1$ and $\boxed{p_0 = 0.8.}$

$\uparrow$ crossover probability

$$Q = \begin{array}{c|cc} x \setminus y & 0 & 1 \\ 0 & 1-p & p \\ 1 & p & 1-p \end{array} = \begin{array}{c|cc} \hat{x}_{ML}(y) & 0 & 1 \\ x \setminus y & 0 & 1 \\ 0 & \boxed{0.9} & 0.1 \\ 1 & 0.1 & \boxed{0.9} \end{array} \xrightarrow[0.2]{0.8} \begin{array}{c|cc} \hat{x}_{ML}(y) & 0 & 1 \\ x \setminus y & 0 & 1 \\ 0 & \boxed{0.72} & 0.08 \\ 1 & 0.02 & \boxed{0.18} \end{array} = P$$

$P(\mathcal{C}) = 0.72 + 0.18 = 0.9$

Note that

| $y$ | $\hat{x}_{ML}(y)$ |
|---|---|
| 0 | 0 |
| 1 | 1 |

$\hat{x}_{ML}(y) \equiv y$

$P(\mathcal{E}) = 1 - 0.9 = 0.1$

- the prior probabilities $p_0$ (and $p_1$) is not used when finding $\hat{x}_{\text{ML}}$.
- the ML decoder and the MAP decoder are the same in this example
  - ○ ML decoder can be optimal even when the prior probabilities are not uniform.

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

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

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

(c) when $p = 0.5$, all four reasonable decoders 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.

## 3.4 Optimal Block Decoding for Communications Over BSC

**3.45.** The decoding techniques (MAP and ML) discussed in the previous section can be extended to the case in which we simultaneously consider $n$ consecutive channel output symbols resulted from having $n$ input symbols.

Notation-wise, this simply means we consider an input-output vector pair $(\underline{\mathbf{X}}, \underline{\mathbf{Y}})$ instead of an input-output symbol pair $(X, Y)$

$$(X_1, X_2, --, X_n) = \underline{X} \longrightarrow \boxed{\text{DMC}} \longrightarrow \underline{Y} = (Y_1, Y_2, \cdots, Y_n)$$

**3.46.** By the memoryless property of the channel,

$$Q\left(\underline{\mathbf{y}}|\underline{\mathbf{x}}\right) = Q(y_1|x_1) \times Q(y_2|x_2) \times \cdots \times Q(y_n|x_n).$$

**Example 3.47.** For a DMC in which $\mathcal{X} = \{0, 1\}$, $\mathcal{Y} = \{1, 2, 3\}$, $\mathbf{Q} =$

$$\begin{array}{c|ccc} x \backslash y & 1 & 2 & 3 \\ \hline 0 & 0.5 & 0.2 & 0.3 \\ 1 & 0.3 & 0.4 & 0.3 \end{array}$$

, find

(a) $Q(122|100)$ $= 0.2 \times 0.2 \times 0.3 = 0.012$

(b) $Q(333|111)$ $= (0.3)^3 = 0.027$

Extended Q matrix

$$\begin{array}{c|ccccccc} \underline{x} \backslash \underline{y} & 111 & 112 & 121 & \cdots & 122 & \cdots & 333 \\ \hline 000 & & & & & \vdots & & \vdots \\ 001 & & & & & \vdots & & \vdots \\ 010 & & & & & \vdots & & \vdots \\ \vdots & & & & & \vdots & & \vdots \\ 100 & \cdots & \cdots & \cdots & \cdots & 0.012 & & \vdots \\ \vdots & & & & & & & \vdots \\ 111 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0.027 \end{array}$$

**Example 3.48.** For BSC, find

(a) $Q(101|100)$ $= p \times (1-p) \times (1-p) = (1-p)^{\textcircled{2}} p^{\textcircled{1}}$

(b) $Q(111|111)$ $= (1-p)^{\textcircled{3}} p^{\textcircled{0}}$

# positions that $\underline{x}$ and $\underline{y}$ are the same → $n-d$

# " " " different → $d(\underline{x}, \underline{y})$

**3.49.** For BSC,

$$Q(y_i|x_i) = \begin{cases} p, & y_i \neq x_i, \\ 1-p, & y_i = x_i. \end{cases}$$

Therefore,

$$Q\left(\underline{\mathbf{y}}|\underline{\mathbf{x}}\right) = p^{d\left(\underline{\mathbf{x}},\underline{\mathbf{y}}\right)}(1-p)^{n-d\left(\underline{\mathbf{x}},\underline{\mathbf{y}}\right)} = \left(\frac{p}{1-p}\right)^{d\left(\underline{\mathbf{x}},\underline{\mathbf{y}}\right)}(1-p)^n, \tag{8}$$

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.

distance (difference)

Hamming distance

$d(101, 100) = 1$

$d(111, 111) = 0$

$d(00101, 01111) = 2$

**3.50.** To recover the value of $\underline{\mathbf{x}}$ from the observed value of $\underline{\mathbf{y}}$, we can apply the vector version of what we studied about optimal decoder in the previous section.

- The optimal decoder is again given by the MAP decoder:

$$\hat{\underline{\mathbf{x}}}_{\mathrm{MAP}}\left(\underline{\mathbf{y}}\right) = \arg\max_{\underline{\mathbf{x}}} Q\left(\underline{\mathbf{y}}\left|\underline{\mathbf{x}}\right.\right) p\left(\underline{\mathbf{x}}\right). \tag{9}$$

- 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:

$$\hat{\underline{\mathbf{x}}}_{\mathrm{ML}}\left(\underline{\mathbf{y}}\right) = \arg\max_{\underline{\mathbf{x}}} Q\left(\underline{\mathbf{y}}\left|\underline{\mathbf{x}}\right.\right). \tag{10}$$

no p(x) term in MLD

Plugging-in

$$Q\left(\underline{\mathbf{y}}|\underline{\mathbf{x}}\right) = p^{d\left(\underline{\mathbf{x}},\underline{\mathbf{y}}\right)}(1-p)^{n-d\left(\underline{\mathbf{x}},\underline{\mathbf{y}}\right)} = \left(\frac{p}{1-p}\right)^{d\left(\underline{\mathbf{x}},\underline{\mathbf{y}}\right)}(1-p)^n, \tag{11}$$

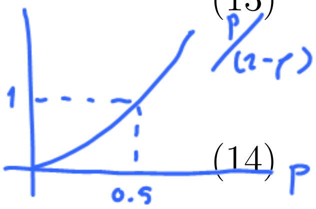
from (8), gives

$$\hat{\underline{\mathbf{x}}}_{\mathrm{MAP}}\left(\underline{\mathbf{y}}\right) = \arg\max_{\underline{\mathbf{x}}} \left(\frac{p}{1-p}\right)^{d\left(\underline{\mathbf{x}},\underline{\mathbf{y}}\right)}(1-p)^n p\left(\underline{\mathbf{x}}\right) \tag{12}$$

$$= \arg\max_{\underline{\mathbf{x}}} \left(\frac{p}{1-p}\right)^{d\left(\underline{\mathbf{x}},\underline{\mathbf{y}}\right)} p\left(\underline{\mathbf{x}}\right). \tag{13}$$

and

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



**3.51. Minimum-distance decoder** as a ML decoding of block codes over BSC:

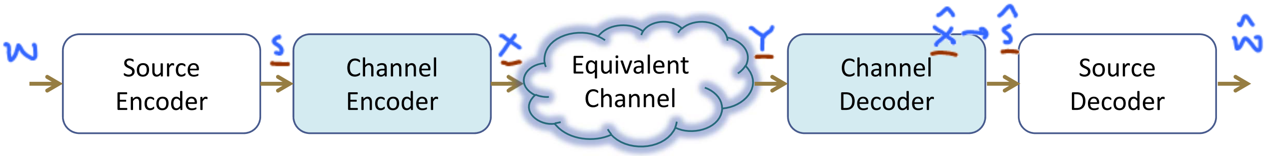
From (14) (or directly from (8)), note that when $p < 0.5$, which is usually the case for practical systems, we have $p < 1-p$ and hence $0 < \frac{p}{1-p} < 1$. In which case, to maximize $Q\left(\underline{\mathbf{y}}|\underline{\mathbf{x}}\right)$, we need to minimize $d\left(\underline{\mathbf{x}},\underline{\mathbf{y}}\right)$. 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 $\underline{\mathbf{y}}$:

$$\hat{\underline{\mathbf{x}}}_{\mathrm{ML}}\left(\underline{\mathbf{y}}\right) = \arg\min_{\underline{\mathbf{x}}} d\left(\underline{\mathbf{x}},\underline{\mathbf{y}}\right). \tag{15}$$

In conclusion, for block coding over BSC with $p < 0.5$, the ML decoder is the same as the minimum distance decoder.

## 3.5 Repetition Code for Channel Coding in Communications Over BSC

**3.52.** 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.



**3.53. 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

**3.54.** Two classes of channel codes

(a) Block codes

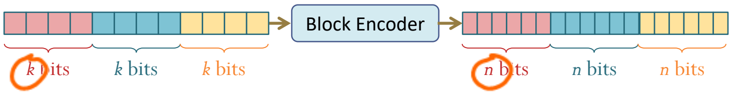
- To be discussed here.
- Realized by combinational/combinatorial circuit.

AND, OR, NOT, NAND, etc gates and wires

(b) Convolutional codes

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

**Definition 3.55. Block Encoding**: Take $k$ (information) bits at a time and map each $k$-bit sequence into a (unique) $n$-bit sequence, called a **codeword**.

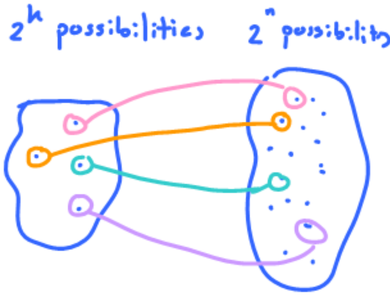


- The code is called $(n, k)$ code.
- Working with $k$-info-bit blocks means there are potentially $M = 2^k$ different information blocks.
  - The table that lists all the $2^k$ mapping from the $k$-bit info-block $\underline{\mathbf{s}}$ to the $n$-bit codeword $\underline{\mathbf{x}}$ is called the **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.

k bits n bits

$2^k$ rows

| index $i$ | info-block $\underline{\mathbf{s}}$ | codeword $\underline{\mathbf{x}}$ |
|---|---|---|
| 1 | $\underline{\mathbf{s}}^{(1)} = 000\dots0$ | $\underline{\mathbf{x}}^{(1)} = \underbrace{\_\ \_\ \dots\ \_}_{n \text{ positions}}$ |
| 2 | $\underline{\mathbf{s}}^{(2)} = 000\dots1$ | $\underline{\mathbf{x}}^{(2)} =$ |
| $\vdots$ | $\vdots$ | $\vdots$ |
| $M$ | $\underline{\mathbf{s}}^{(M)} = 111\dots1$ | $\underline{\mathbf{x}}^{(M)} =$ |



  - By the bijective mapping from $\underline{\mathbf{s}}$ to $\underline{\mathbf{x}}$,

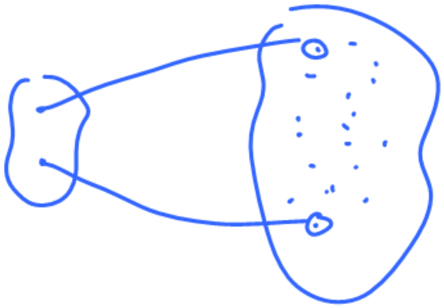
$$p_i \equiv p\left(\underline{\mathbf{x}}^{(i)}\right) \equiv P\left[\underline{\mathbf{X}} = \underline{\mathbf{x}}^{(i)}\right] = P\left[\underline{\mathbf{S}} = \underline{\mathbf{s}}^{(i)}\right].$$

- 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].

**Example 3.56.** Find the codebook and code rate for the encoder which uses repetition code with $n = 5$.

codebook

| $s$ | $\underline{x}$ |
|---|---|
| 0 | $00000 \equiv \underline{0}$ |
| 1 | $11111 \equiv \underline{1}$ |



**Example 3.57.** To get some idea about the difficulty of finding an optimal encoder, we need to consider the size of our search space. For $k = 5$ and $n = 10$, how many encoders are possible?

$$\binom{2^n}{2^k} = \binom{2^{10}}{2^5} = \binom{1024}{32} \approx 5 \times 10^{60}$$

| index | $\underline{s}$ | $\underline{x}$ |
|---|---|---|
| 1 | 00000 | $\underbrace{-\,-\,-\,\cdots\,-}_{n \text{ positions}}$ |
| 2 | 00001 | |
| ⋮ | 00010 | |
| | ⋮ | |
| $2^k = 2^5 = 32$ | 11111 | |

**3.58. Decoding**: When the mapping from the information block $\underline{\mathbf{s}}$ to the codeword $\underline{\mathbf{x}}$ is invertible, the task of any 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.

**Example 3.59.** Repetition Code and Majority Voting: Back to Example 3.53.

Recall: ① MAP decoder is optimal $(\min P(\mathcal{E}))$

② ML decoder is suboptimal. However, it can be optimal (the same as the MAPD) when the codewords are equally-likely.

③ ML decoder is the same as the min-distance decoder when the crossover probability $p$ of BSC is $< 0.5$.

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

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

Ex $\underline{y} = 01010$

$d(\underline{0}, \underline{y}) = 2 = \#1$ in $\underline{y}$

$d(\underline{1}, \underline{y}) = 3 = \#0$ in $\underline{y}$

Therefore, the minimum distance decoder 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

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

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.

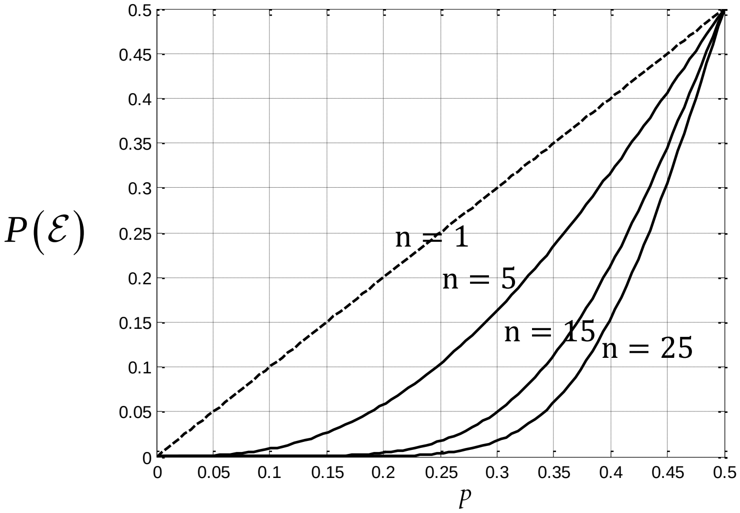


Figure 6: Error probability for a system that uses repetition code at the transmitter (repeat each info-bit $n$ times) and majority voting at the receiver. The channel is assumed to be binary symmetric with crossover probability $p$.

- 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.

**3.60.** 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 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.

## 4.1 Information-Theoretic Quantities

**Definition 4.1.** Recall that, the **entropy** of a discrete random variable $X$ is defined in Definition 2.41 to be

Amount of randomness in RV X

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

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:

$$\begin{array}{r|l|l} S_X \equiv \mathcal{X} & p_X(x) \equiv p(x) \rightsquigarrow \underline{\mathbf{p}} & p_{Y|X}(y|x) \equiv Q(y|x) \rightsquigarrow \mathbf{Q} \text{ matrix} \\ S_Y \equiv \mathcal{Y} & p_Y(y) \equiv q(y) \rightsquigarrow \underline{\mathbf{q}} & p_{X,Y}(x,y) \equiv p(x,y) \rightsquigarrow \mathbf{P} \text{ matrix} \end{array}$$

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 4.2.** The **joint entropy** for two random variables $X$ and $Y$ is given by

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

**Example 4.3.** Random variables $X$ and $Y$ have the following joint pmf matrix $\mathbf{P}$:

$$P = \begin{array}{c|cccc|c} x \backslash y & y_1 & y_2 & y_3 & y_4 & p(x) \\ \hline x_1 & \frac{1}{8} & \frac{1}{16} & \frac{1}{16} & \frac{1}{4} & \xrightarrow{\Sigma} 1/2 \\ x_2 & \frac{1}{16} & \frac{1}{8} & \frac{1}{16} & 0 & \xrightarrow{\Sigma} 1/4 \\ x_3 & \frac{1}{32} & \frac{1}{32} & \frac{1}{16} & 0 & \xrightarrow{\Sigma} 1/8 \\ x_4 & \frac{1}{32} & \frac{1}{32} & \frac{1}{16} & 0 & \xrightarrow{\Sigma} 1/8 \\ \hline & \Sigma\downarrow & \Sigma\downarrow & \Sigma\downarrow & \Sigma\downarrow & \\ & 1/4 & 1/4 & 1/4 & 1/4 & \end{array}$$

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

$$H(X) = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{4}\log_2\frac{1}{4} - 2\times\frac{1}{8}\log_2\frac{1}{8} = \frac{1}{2} + \frac{2}{4} + \frac{6}{8} = \frac{7}{4} \text{ bits (per symbol)}$$

$$H(Y) = -4\times\frac{1}{4}\log_2\frac{1}{4} = 2 \text{ bits (per symbol)}$$

$$H(X,Y) = -\frac{1}{8}\log_2\frac{1}{8} - \frac{1}{16}\log_2\frac{1}{16} - \frac{1}{16}\log_2\frac{1}{16} \cdots\cdot \qquad (16 \text{ terms})$$

$$= \left(-\frac{1}{4}\log_2\frac{1}{4}\right) + 2\left(-\frac{1}{8}\log_2\frac{1}{8}\right) + 6\left(-\frac{1}{16}\log_2\frac{1}{16}\right) + 4\left(-\frac{1}{32}\log_2\frac{1}{32}\right)$$

$$= \frac{27}{8} \text{ bits (per symbol)}$$

**Definition 4.4.** The (conditional) entropy of $Y$ when we know $X = x$ is denoted by $H(Y|X = x)$ or simply $H(Y|x)$. It can be calculated from

The amount of randomness still remained in Y when we know that X = x.

$$\boxed{H(Y|x)} = -\sum_{y\in\mathcal{Y}} Q(y|x)\log_2 Q(y|x)$$

- 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):

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

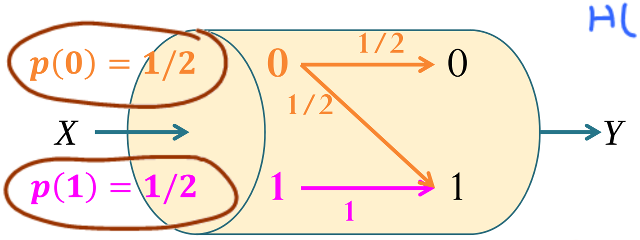
  When we have an extra piece of information that $X = x$, we should update the probability about $Y$ from the unconditional probability $q(y)$ to the conditional probability $Q(y|x)$.

- 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 $H(Y|x)$, we simply find the "usual" entropy from the probability values in the row corresponding to $x$ in the $\mathbf{Q}$ matrix.

**Example 4.5.** Consider the following DMC (actually BAC)

$H(x) = \log_2 2 = 1$

$-\frac{1}{2}\log_2\frac{1}{2} + -\frac{1}{2}\log_2\frac{1}{2}$



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

$= \frac{3}{2}$

$H(Y) = -\frac{1}{4}\log_2\frac{1}{4} + -\frac{3}{4}\log_2\frac{3}{4}$

$$Q = \begin{array}{c|cc} x\backslash y & 0 & 1 \\ \hline 0 & 1/2 & 1/2 \\ 1 & 0 & 1 \end{array} \begin{array}{c} \xrightarrow{\times 1/2} \\ \xrightarrow[\times 1/2]{} \end{array} \begin{array}{c|cc} x\backslash y & 0 & 1 \\ \hline 0 & 1/4 & 1/4 \\ 1 & 0 & 1/2 \\ \hline \Sigma & 1/4 & 3/4 \end{array} = P$$

Originally $P[Y=y] = q(y) = \begin{cases} 1/4, & y=0, \\ 3/4, & y=1, \\ 0, & \text{otherwise.} \end{cases}$

(a) Suppose we know that $X=0$.

The "x = 0" row in the $\mathbf{Q}$ matrix gives $Q(y|0) = \begin{cases} 1/2, & y=0,1, \\ 0, & \text{otherwise;} \end{cases}$

that is, given $x=0$, the RV $Y$ will be uniform.

$H(Y|0) = \log_2 2 = 1$

(b) Suppose we know that $X=1$. The "x = 1" row in the $\mathbf{Q}$ matrix gives $Q(y|1) = \begin{cases} 1, & y=1, \\ 0, & \text{otherwise;} \end{cases}$ that is, given $x=1$, the RV $Y$ is degenerated (deterministic).

$H(Y|1) = 0$

**Definition 4.6. Conditional entropy**: The (average) conditional entropy of $Y$ when we know $X$ is denoted by $H(Y|X)$. It can be calculated from

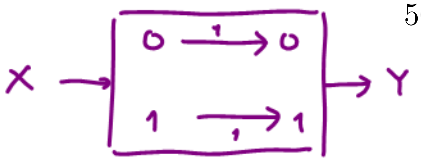
$$H(Y|X) = \sum_{x\in\mathcal{X}} p(x)\,H(Y|x).$$

The average amount of randomness still remained in Y when we know X

**Example 4.7.** In Example 4.5,

$H(Y|X) = p(0)H(Y|0) + p(1)H(Y|1) = \frac{1}{2}\times 1 + \frac{1}{2}\times 0 = \frac{1}{2}$

Easy Example



$\left.\begin{array}{l} H(Y|0) = 0 \\ H(Y|1) = 0 \end{array}\right\} \to H(Y|X) = \sum p(x)\times 0$

$= 0.$

**4.8.** An alternative way to calculate $H(Y|X)$ can be derived by first rewriting it as

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

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

Compare this with

$P(A \setminus B) = P(A \cup B) - P(B)$

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

**Example 4.9.** In Example 4.5,

$H(Y|X) = H(X,Y) - H(X) = \frac{3}{2} - 1 = \frac{1}{2}$.

**Example 4.10.** Continue from Example 4.3. Random variables $X$ and $Y$ have the following joint pmf matrix $\mathbf{P}$:

$$\mathbf{P} = \begin{array}{c} x \backslash y \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{array} \begin{bmatrix} \frac{1}{8} & \frac{1}{16} & \frac{1}{16} & \frac{1}{4} \\ \frac{1}{16} & \frac{1}{8} & \frac{1}{16} & 0 \\ \frac{1}{32} & \frac{1}{32} & \frac{1}{16} & 0 \\ \frac{1}{32} & \frac{1}{32} & \frac{1}{16} & 0 \end{bmatrix} \begin{array}{c} \xrightarrow{\times 2} \\ \xrightarrow{\times 4} \\ \xrightarrow{\times 8} \\ \xrightarrow{\times 8} \end{array} \begin{bmatrix} 1/4 & 1/2 & 1/4 & 0 \\ & & & \\ & & & \end{bmatrix} Q$$

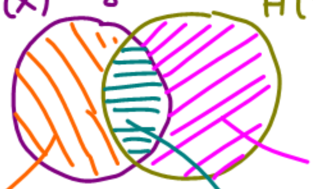
Find $H(Y|X)$ and $H(X|Y)$.

From 4.3 $\begin{cases} H(X) = 7/4 \\ H(Y) = 2 \\ H(X,Y) = \frac{27}{8} \end{cases}$

$H(Y|x_2) = \left(-\frac{1}{4}\log_2\frac{1}{4}\right)\times 2 - \frac{1}{2}\log_2\frac{1}{2}$
$= 3/2$

$H(Y|X) = H(X,Y) - H(X)$
$= \frac{27}{8} - \frac{14}{8} = \frac{13}{8}$

$H(X|Y) = H(Y,X) - H(Y)$
$= \frac{27}{8} - 2 = \frac{11}{8}$



$H(X) = \frac{14}{8}$

$H(Y) = \frac{16}{8}$

$H(Y|X) = \frac{13}{8}$

$H(X|Y) = \frac{11}{8}$

$= H(X) + H(Y) - H(X,Y)$
$= H(X) - H(X|Y)$
$= H(Y) - H(Y|X)$

$I(X;Y) = 3/8$

# ECS 452: In-Class Exercise # _

## Instructions

1. Separate into groups of no more than three persons.
2. The group cannot be the same as your former group.
3. Only one submission is needed for each group.
4. ***Write down all the steps*** that you have done to obtain your answers. You may not get full credit even when your answer is correct without showing how you get your answer.
5. **Do not panic.**

| Date: _ _ / _ _ / 2017 | | | |
|---|---|---|---|
| Name | ID (last 3 digits) | | |
| | | | |
| | | | |
| | | | |

1. Consider two random variables $X$ and $Y$ whose joint pmf matrix is given by

$$\mathbf{P} = \begin{array}{c|cc|cl} {}_x\backslash^y & 3 & 5 & & p(x) \\ \hline 1 & \frac{1}{18} & \frac{5}{18} & \xrightarrow{\Sigma} & \frac{6}{18} = \frac{1}{3} \\ 2 & \frac{4}{9} & \frac{2}{9} & \xrightarrow{\Sigma} & \frac{6}{9} = \frac{2}{3} \\ \hline & \Sigma\downarrow & \Sigma\downarrow & & \\ q(y) & \frac{9}{18} & \frac{9}{18} & & \end{array}.$$

Calculate the following quantities. Except for part (e), all of your answers should be of the form X.XXXX.

a. $H(X,Y)$ $= -\frac{1}{18}\log_2\frac{1}{18} - \frac{5}{18}\log_2\frac{5}{18} - \frac{4}{9}\log_2\frac{4}{9} - \frac{2}{9}\log_2\frac{2}{9}$

$\approx 1.7472$ bits (per symbol)

b. $H(X)$ $= H([\,1/3 \quad 2/3\,]) = -\frac{1}{3}\log_2\frac{1}{3} - \frac{2}{3}\log_2\frac{2}{3} \approx 0.9183$ bits (per symbol)

c. $H(Y)$ $= H([\,1/2 \quad 1/2\,]) = \log_2 2 = 1.000$ bit (per symbol)

For uniform RV, the entropy is simply $\log_2$ of the size of its support.

d. $H(Y|X)$ $= H(X,Y) - H(X) \approx 1.7472 - 0.9183$

$\approx 0.8289$ bits (per symbol)

$H(Y|X) = \underset{1/3}{p(1)}H(Y|1) + \underset{2/3}{p(2)}H(Y|2)$

$\approx 0.8289$ bits (per symbol)

$H(Y|1) = H([\,1/6 \quad 5/6\,]) \approx 0.6500$

e. **Q** matrix

$$P = \begin{bmatrix} 1/18 & 5/18 \\ 4/9 & 2/9 \end{bmatrix} \begin{array}{c} \times\frac{1}{p(x)} \\ \xrightarrow{\times 3} \\ \xrightarrow{\times 3/2} \end{array} \begin{array}{c|cc} {}_x\backslash^y & 3 & 5 \\ \hline 1 & 1/6 & 5/6 \\ 2 & 2/3 & 1/3 \end{array} = Q$$

$H(Y|2) = H([\,2/3 \quad 1/3\,]) \approx 0.9183$

**Definition 4.11.** The **mutual information**[13] $I(X;Y)$ between two random variables $X$ and $Y$ is defined as

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

Interpretation ※1

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

$$I(X;Y) = H(X) - \underbrace{H(X|Y)}$$

Amount of reduction in the randomness of X due to the knowledge of Y.

Amount of randomness in X

Amount of randomnes still remained in X when we know Y.

Interpretation ※2

- Mutual information is a measure of the amount of information one random variable contains about another [5, p 13].

Interpretation ※3

- It is also 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.

**4.12.** 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 $\geq 0$.

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

[13] 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.

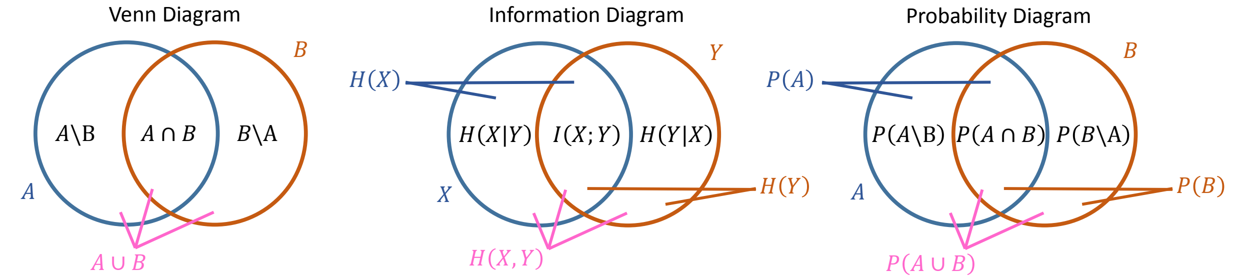


Figure 7: Venn diagram and its use to represent relationship between information measures and relationship between probabilities

- Chain rule for information measures:

$$H(X,Y) = H(X) + H(Y|X) = H(Y) + H(X|Y).$$

(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) $H(Y|X) \leq H(Y)$,

  Conditioning only reduces entropy

  (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

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

Similarly,

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

For conditional entropy, we have

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

and

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

For mutual information, we have

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

and

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

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

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

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

**Example 4.13.** Find the mutual information $I(X;Y)$ between the two random variables $X$ and $Y$ whose joint pmf matrix is given by $\mathbf{P} = \begin{bmatrix} \frac12 & \frac14 \\ \frac14 & 0 \end{bmatrix}$.

$\xrightarrow{\Sigma} 3/4$

$\xrightarrow[\Sigma]{} 1/4$

$\Sigma\downarrow \quad \downarrow\Sigma$

$\frac34 \quad 1/4$

$I(X;Y) = H(X) + H(Y) - \underbrace{H(X,Y)} \approx 0.1226$

$\hookrightarrow -\frac12\log_2\frac12 - \left(\frac14\log_2\frac14\right)\times 2 = 1.5$

$H(X) = H([3/4 \quad 1/4]) = -\frac34\log_2\frac34 \quad -\frac14\log_2\frac14 \approx 0.8113$

**Example 4.14.** Find the mutual information $I(X;Y)$ between the two random variables $X$ and $Y$ whose $\underline{\mathbf{p}} = \left[\frac14, \frac34\right]$ and $\mathbf{Q} = \begin{array}{c|cc} x_1 & \frac14 & \frac34 \\ x_2 & \frac34 & \frac14 \end{array}$.

Method ①  $Q = [\quad] \xrightarrow[\times 3/4]{\times 1/4} [\quad] = P$

$\underline{q} = \underline{p}\,Q$

$= \left[\frac58 \quad \frac38\right]$

Method ②  $H(Y|x_1) = H([1/4 \quad 3/4]) \approx 0.8113$

$H(Y|x_2) = H([3/4 \quad 1/4]) \approx 0.8113$

$H(Y|X) = \underbrace{p(x_1)}_{1/4}H(Y|x_1) + \underbrace{p(x_2)}_{3/4}H(Y|x_2) \approx 0.8113$

$I(X;Y) = \underbrace{H(Y)}_{0.9544} - H(Y|X) \approx 0.1432$

## 4.2 Operational Channel Capacity

**4.15.** In Section 3, we have studied how to compute the error probability $P(\mathcal{E})$ for digital communication systems over DMC. At the end of that section, we studied block encoding where the channel is used $n$ times to transmit a $k$-bit info-block.

In this section, our consideration is "reverse".

**4.16.** In this and the next subsections, we introduce a quantity called channel capacity which is crucial in benchmarking communication system. Recall that, in Section 2 where source coding was discussed, we were interested in the minimum rate (in bits per source symbol) to represent a source. Here, we are interested in the maximum rate (in bits per channel use) that can be sent through a given channel *reliably*.

**4.17.** Here, **reliable communication** means *arbitrarily* small positive error probability *can be achieved.*

- This seems to be an impossible goal.
  - ○ If the channel introduces errors, how can one correct them all?
    - ∗ Any correction process is also subject to error, ad infinitum.

**Definition 4.18.** Given a DMC, its **"operational" channel capacity** is the maximum rate at which *reliable* communication over the channel *is possible*.

by using appropriate block length n, channel encoder, channel decoder.

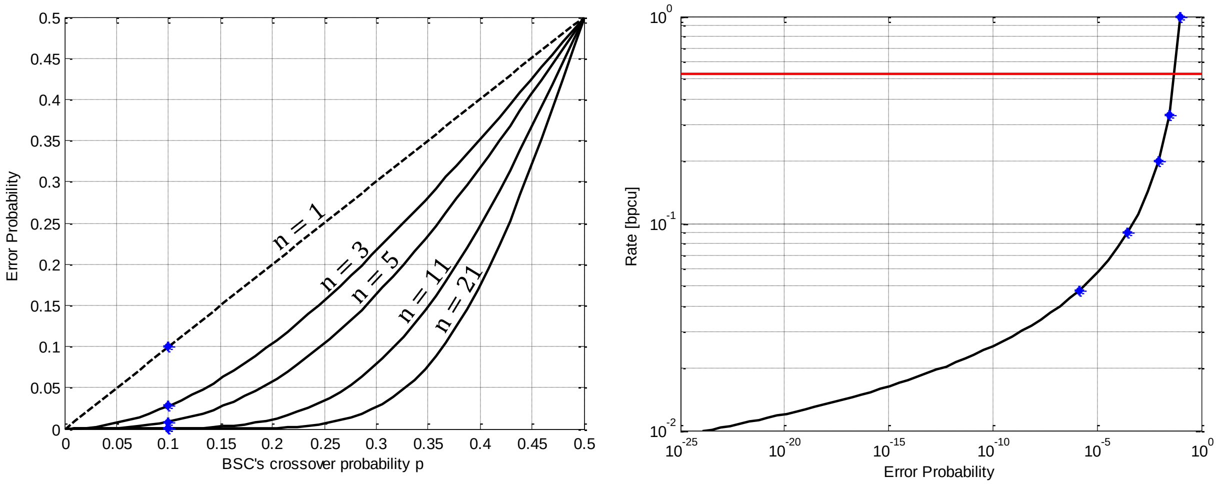
- The channel capacity is the maximum rate in bits per channel use at which information *can be* sent with **arbitrarily low** error probability.

**4.19.** Claude Shannon showed, in his 1948 landmark paper, that this operational channel capacity is the same as the information channel capacity which we will discuss in the next subsection. From this, we can omit the words “operational” and “information” and simply refer to both quantities as the *channel capacity*.

**Example 4.20.** In Example 4.34, we will find that the capacity of a BSC with crossover probability $p = 0.1$ is approximately 0.531 bits per channel use. This means that for any rate $R < 0.531$ and any error probability $P(\mathcal{E})$ that we desire, as long as it is greater than 0, we can find a suitable $n$, a rate $R$ encoder, and a corresponding decoder which will yield an error probability that is at least as low as our set value.

- Usually, for very low desired value of $P(\mathcal{E})$, we may need large value of $n$.

**Example 4.21.** Repetition code is not good enough.



(a)

(b) $p = 0.1$

Figure 8: Performance of repetition coding with majority voting at the decoder

- From Figure 8b, it is clear that repetition coding can not reliably achieve such transmission rate. In fact, when we require error probability to be less than 0.1, the required repetition code needs $n \geq 3$. (For simplicity, let’s assume only odd value of $n$ can be used here.) However, this means the rate is $\leq \frac{1}{3} \approx 0.33$ which is already significantly less than 0.531.
- In fact, for any rate $> 0$, we can see from Figure 8b that communication system based on repetition coding is not “reliable” according

to Definition 4.17. For example, for rate = 0.02 bits per channel use, repetition code can't satisfy the requirement that the error probability must be less than $10^{-15}$. In fact, Figure 8b shows that as we reduce the error probability to 0, the rate also goes to 0 as well. Therefore, there is no positive rate that works for all error probability.

- However, because the channel capacity is 0.531 [bpcu], there must exist other encoding techniques which give better error probability than repetition code.
  - Although Shannon's result gives us the channel capacity, it does not give us any explicit instruction on how to construct codes which can achieve that value.

## 4.3 Information Channel Capacity

**4.22.** In Section 4.1, we have studied how to compute the value of mutual information $I(X;Y)$ between two random variables $X$ and $Y$. Recall that, here, $X$ and $Y$ are the channel input and output, respectively. We have also seen, in Example 4.13, how to compute $I(X;Y)$ when the joint pmf matrix $\mathbf{P}$ is given. Furthermore, we have also worked on Example 4.14 in which the value of mutual information is computed from the prior probability vector $\underline{\mathbf{p}}$ and the channel transition probability matrix $\mathbf{Q}$. This second type of calculation is crucial in the computation of channel capacity. This kind of calculation is so important that we may write the mutual information $I(X;Y)$ as $I(\underline{\mathbf{p}}, \mathbf{Q})$.

**Definition 4.23.** Given a DMC channel, we define its "information" channel capacity as

$$C = \max_{\underline{\mathbf{p}}} I\left(X;Y\right) = \max_{\underline{\mathbf{p}}} I\left(\underline{\mathbf{p}}, \mathbf{Q}\right), \tag{32}$$

where the maximum is taken over all possible input pmfs $\underline{\mathbf{p}}$.

- Again, as mentioned in 4.19, Shannon showed that the "information" channel capacity defined here is equal to the "operational" channel capacity defined in Definition 4.18.
  - Thus, we may drop the word "information" in most discussions of channel capacity.

**Example 4.24.** The capacity of a BAC whose $Q(1|0) = 0.9$ and $Q(0|1) = 0.4$ can be found by first realizing that $I(X;Y)$ here is a function of a single variable: $p_0$. The plot of $I(X;Y)$ as a function of $p_0$ gives some rough estimates of the answers. One can directly solve for the optimal $p_0$ by simply taking derivative of $I(X;Y)$ and set it equal to 0. This gives the capacity value of 0.0918 bpcu which is achieved by $\underline{\mathbf{p}} = [0.5376,\ 0.4624]$.

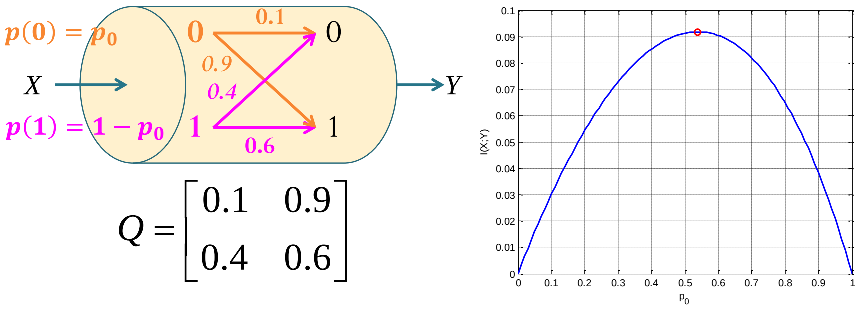


Figure 9: Maximization of mutual information to find capacity of a BAC channel. Capacity of 0.0918 bits is achieved by $\underline{\mathbf{p}} = [0.5376,\ 0.4624]$

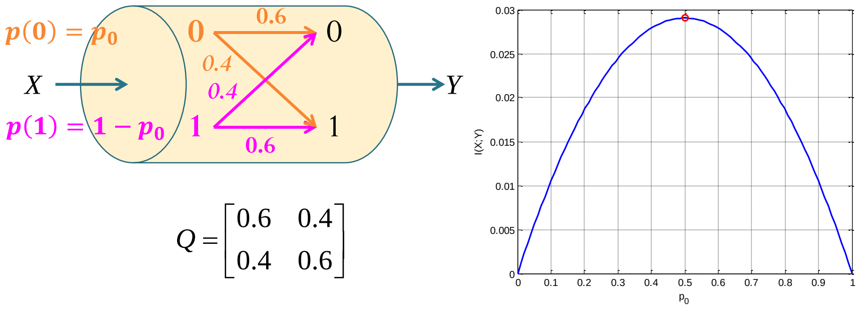


Figure 10: Maximization of mutual information to find capacity of a BSC channel. Capacity of 0.029 bits is achieved by $\underline{\mathbf{p}} = [0.5,\ 0.5]$

**4.25. Blahut-Arimoto Algorithm** [5, Section 10.8]: Alternatively, in 1972, Arimoto [1] and Blahut [2] independently developed an iterative algorithm to help us approximate the pmf $\underline{\mathbf{p}}^*$ which achieves capacity $C$. To do this, start with any (guess) input pmf $p_0(x)$, define a sequence of pmfs $p_r(x)$, $r = 0, 1, \ldots$ according to the following iterative prescription:

(a) $q_r(y) = \sum\limits_x p_r(x)\, Q(y\,|x)$ for all $y \in \mathcal{Y}$.

(b) $c_r(x) = 2^{\left(\sum\limits_y Q(y|x) \log_2 \frac{Q(y|x)}{q_r(y)}\right)}$ for all $x \in \mathcal{X}$.

(c) It can be shown that

$$\log_2\left(\sum_x p_r(x)\,c_r(x)\right) \le C \le \log_2\left(\max_x c_r(x)\right).$$

- If the lower-bound and upper-bound above are close enough. We take $p_r(x)$ as our answer and the corresponding capacity is simply the average of the two bounds.
- Otherwise, we compute the pmf

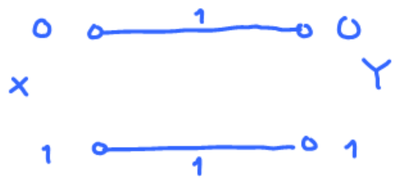
$$p_{r+1}(x) = \frac{p_r(x)\,c_r(x)}{\sum_x p_r(x)\,c_r(x)} \quad \text{for all } x \in \mathcal{X}$$

and repeat the steps above with index $r$ replaced by $r+1$.

## 4.4 Special Cases for Calculation of Channel Capacity

In this section, we study special cases of DMC whose capacity values can be found (relatively) easy.

**Example 4.26.** Find the channel capacity of a noiseless binary channel (a BSC whose crossover probability is $p = 0$).



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

$H(X|Y) = 0$ ← Intuitively, knowing Y, the value of X is completely determined.

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

To maximize $I(X;Y)$, we maximize $H(X)$ by using $\underline{p} = [1/2 \;\; 1/2]$.

$C = \log_2|\mathcal{X}| = \log_2 2$

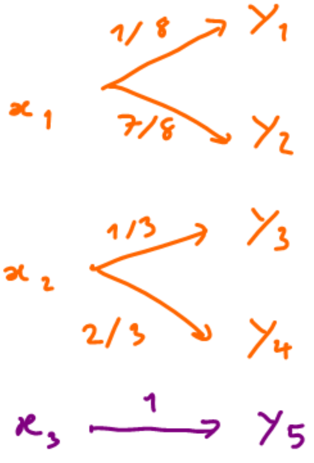
$= 1$ bpcu

bit per channel use

**Example 4.27.** Noisy Channel with Nonoverlapping Outputs: Find the channel capacity of a DMC whose

Ex. 4.27b

$$\mathbf{Q} = \begin{array}{c|ccccc} x \backslash y & y_1 & y_2 & y_3 & y_4 & y_5 \\ \hline x_1 & 1/8 & 7/8 & 0 & 0 & 0 \\ x_2 & 0 & 0 & 1/3 & 2/3 & 0 \\ x_3 & 0 & 0 & 0 & 0 & 1 \end{array}$$



$I(X;Y) = H(X) - \underbrace{H(X|Y)}_{0}$

$\le \log_2|\mathcal{X}| = \log_2 2 = 1$

with equality iff X is uniform $\underline{p} = [1/2 \;\; 1/2]$

$C = 1$ bpcu which happens when $\underline{p} = [1/2 \;\; 1/2]$.

$C = \log_2|\mathcal{X}|$

$= \log_2 3$ bpcu

is achieved by $\underline{p} = [1/3 \;\; 1/3 \;\; 1/3]$.

In this example, the channel appears to be noisy, but really is not. Even though the output of the channel is a random consequence of the input, the input can be determined from the output, and hence every transmitted bit can be recovered without error.

**4.28.** Reminder:

(a) Some definitions involving entropy

(i) Binary entropy function: $h(p) = -p\log_2 p - (1-p)\log_2(1-p)$

(ii) $H(X) = -\sum_x p(x)\log_2 p(x)$

(iii) $H(\underline{\mathbf{p}}) = -\sum_i p_i \log_2(p_i)$

(b) A key entropy property that will be used frequently in this section is that for any random variable $X$,

max

$$H(X) \le \log_2 |\mathcal{X}| \text{ with equality iff } X \text{ is uniform.}$$

[NO²]

**4.29.** A DMC is a **noisy channel with nonoverlapping outputs** when there is only one non-zero element in each column of its $\mathbf{Q}$ matrix. For such channel,

$$C = \log_2 |\mathcal{X}| \text{ is achieved by uniform } p(x).$$

**Definition 4.30.** A DMC is called **symmetric** if (1) all the rows of its probability transition matrix $\mathbf{Q}$ are permutations of each other and (2) so are the columns.

**Example 4.31.** For each of the following $\mathbf{Q}$, is the corresponding DMC symmetric?

$\frac{1}{6}+\frac{3}{6}=\frac{4}{6}$

$$\begin{bmatrix} 0.3 & 0.2 & 0.5 \\ 0.5 & 0.3 & 0.2 \\ 0.2 & 0.5 & 0.3 \end{bmatrix}, \quad \begin{bmatrix} 0.3 & 0.2 & 0.5 \\ 0.5 & 0.2 & 0.3 \\ 0.2 & 0.5 & 0.3 \end{bmatrix}, \quad \begin{bmatrix} 1/3 & 1/6 & 1/2 \\ 1/3 & 1/2 & 1/6 \end{bmatrix}, \quad \begin{bmatrix} 0.1 & 0.9 \\ 0.4 & 0.6 \end{bmatrix}$$

① ✓ ② ✓ symmetric $\Downarrow$ weakly symmetric

① ✓ ② ✗ not symmetric $\Uparrow$ not weakly symmetric

① ✓ ② ✗ not symmetric $\Sigma$ $\Sigma$ $\Sigma$ ② ✓ 2/3 2/3 2/3 weakly symmetric

① ✗ ② ✗ not symmetric not weakly symmetric

**4.32.** Q: Does symmetric DMC always have square $\mathbf{Q}$?

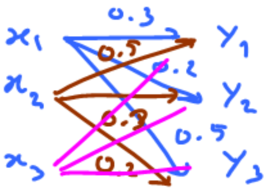
$$\begin{bmatrix} 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \end{bmatrix} \qquad \begin{bmatrix} 1/2 & 0 & 1/2 & 0 \\ 0 & 1/2 & 0 & 1/2 \end{bmatrix}$$

$$\begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix} \qquad \begin{bmatrix} a & a & b & b \\ b & b & a & a \end{bmatrix} \text{ where } a+b=\frac{1}{2}$$

**Example 4.33.** Find the channel capacity of a DMC whose

$$\mathbf{Q} = \begin{array}{c|ccc} & y_1 & y_2 & y_3 \\ \hline x_1 & 0.3 & 0.2 & 0.5 \\ x_2 & 0.5 & 0.3 & 0.2 \\ x_3 & 0.2 & 0.5 & 0.3 \end{array} .$$

$\underline{r}$



Solution: First, recall that the capacity $C$ of a given DMC can be found by (32):

Earlier, we used $I(X;Y) = H(X) - \underbrace{H(X|Y)}_{\text{was 0 now, } \neq 0.}$

$$C = \max_{\underline{\mathbf{p}}} I(X;Y) = \max_{\underline{\mathbf{p}}} I\left(\underline{\mathbf{p}}, \mathbf{Q}\right).$$

$$I(X;Y) = H(Y) - \underbrace{H(Y|X)}$$

$$= \sum_x p(x) H(Y|x) = \sum_x p(x) H(\underline{r}) = H(\underline{r})$$

We see that, to maximize $I(X;Y)$, we need to maximize $H(Y)$. Of course, we know that the maximum value of $H(Y)$ is $\log_2 |\mathcal{Y}|$ which happens when $Y$ is uniform. Therefore, if we can find $\underline{\mathbf{p}}$ which makes $Y$ uniform, then this same $\underline{\mathbf{p}}$ will give the channel capacity.

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

first column sum

$$\left[\tfrac{1}{3}\times 1, \tfrac{1}{3}\times 1, \tfrac{1}{3}\times 1\right] = \left[\tfrac{1}{3} \;\; \tfrac{1}{3} \;\; 1/3\right] \begin{bmatrix} 0.3 & 0.2 & 0.5 \\ 0.5 & 0.3 & 0.2 \\ 0.2 & 0.5 & 0.3 \end{bmatrix}$$

Try uniform $X$

second column sum

$$C = \log_2 |\mathcal{Y}| - H(\underline{r}) = \log_2 3 - H([0.3 \; 0.2 \; 0.5]) \approx 1.5850 - 1.4855$$
$$= 0.0995 \text{ [bpcu]}.$$

Remark: If we can't find $\underline{\mathbf{p}}$ that makes $Y$ uniform, then $C < \log_2 |\mathcal{Y}| - H(\underline{\mathbf{r}})$ and we have to find a different technique to calculate $C$.

**Example 4.34.** Find the channel capacity of a BSC whose crossover probability is $p = 0.1$.

$$Q = \begin{array}{c|cc} x \backslash y & 0 & 1 \\ \hline 0 & 1-p & p \\ 1 & p & 1-p \end{array} = \begin{bmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{bmatrix}$$

$$C = \log_2 |\mathcal{Y}| - H(\underline{r})$$
$$= \log_2 2 - H([0.9 \;\; 0.1])$$
$$= 1 - H(0.1) \approx 1 - 0.469 = 0.531.$$

**Definition 4.35.** A DMC is called **weakly symmetric** if (1) all the rows of its probability transition matrix $\mathbf{Q}$ are permutations of each other and (2) all the column sums are equal.

- It should be clear from the definition that a symmetric channel is automatically weakly symmetric.

Condition (1) makes all $H(Y|x)$ the same. Therefore, $H(Y|X)$ is easy to find and does not depend on $\underline{p}$.

Condition (2) guarantees that uniform $X$ gives uniform $Y$.

**4.36.** For a weakly symmetric channel,

$$C = \log_2 |\mathcal{Y}| - H(\underline{\mathbf{r}}),$$

where $\underline{\mathbf{r}}$ is any row from the $\mathbf{Q}$ matrix. The capacity is achieved by a uniform pmf on the channel input.

- Important special case: For BSC, $C = 1 - H(p)$.

**4.37.** Properties of channel capacity

(a) $C \geq 0$

(b) $C \leq \min\{\log_2 |\mathcal{X}|, \log_2 |\mathcal{Y}|\}$

**Example 4.38.** Find the channel capacity of a DMC whose

$$\mathbf{Q} = \begin{bmatrix} 0.9 & 0.05 & 0.05 \\ 0.05 & 0.9 & 0.05 \\ 0.025 & 0.025 & 0.95 \end{bmatrix}$$

Suppose four choices are provided:
(a) 1.0944 (b) 1.5944 (c) 2.0944 (d) 2.5944

**Example 4.39.** Another case where capacity can be easily calculated: Find the channel capacity of a DMC of which all the rows of its $\mathbf{Q}$ matrix are the same.

$$Q = \begin{bmatrix} \overbrace{0.5 \quad 0.2 \quad 0.3}^{\underline{r}} \\ 0.5 \quad 0.2 \quad 0.3 \end{bmatrix}$$

$$\underline{q} = \underline{p}Q = [p_1 \quad p_2]\begin{bmatrix} 0.5 & 0.2 & 0.3 \\ 0.5 & 0.2 & 0.3 \end{bmatrix}$$

$$= [\underset{\sim}{0.5} \quad \underset{\sim}{0.2} \quad \underset{\sim}{0.3}] = \underline{r}$$

$$p_1 \times 0.5 + p_2 \times 0.5 = 0.5(p_1 + p_2)$$

$$H(Y|x) = H(\underline{r})$$

$$H(Y) = H(\underline{r})$$

$$I(X;Y) = H(Y) - H(Y|X) = H(\underline{r}) - H(\underline{r}) = 0.$$

$$\Rightarrow C = \max_{\underline{p}} \overbrace{I(X;Y)}^{0} = 0 \quad [\text{bpcu}]$$

(can use any $\underline{p}$)

**4.40.** In this section, we worked with "toy" examples in which finding capacity is relatively easy. In general, there is no closed-form solution for computing capacity. When we have to deal with cases that do not fit in any special family of $\mathbf{Q}$ described in the examples above, the maximum may be found by standard nonlinear optimization techniques or the Blahut-Arimoto Algorithm discussed in 4.25.

## 4.5 Shannon's Coding theorem

**4.41. Shannon's (Noisy Channel) Coding theorem** [Shannon, 1948]

(a) Reliable communication over a (discrete memoryless) channel is possible if the communication rate $R$ satisfies $R < C$, where $C$ is the channel capacity.

In particular, for any $R < C$, there exist codes (encoders and decoders) with sufficiently large $n$ such that

$$P(\mathcal{E}) \leq 2^{-n \times E(R)},$$

where $E(R)$ is

- a positive function of $R$ for $R < C$ and
- completely determined by the channel characteristics

(b) At rates higher than capacity, reliable communication is impossible.

**4.42.** Significance of Shannon's (noisy channel) coding theorem:

(a) Express the limit to reliable communication

(b) Provides a yardstick to measure the performance of communication systems.

- A system performing near capacity is a near optimal system and does not have much room for improvement.
- On the other hand a system operating far from this fundamental bound can be improved (mainly through coding techniques).

**4.43.** Shannon's nonconstructive proof for his coding theorem

- Shannon introduces a method of proof called **random coding**.
- Instead of looking for the best possible coding scheme and analyzing its performance, which is a difficult task,
  - ∘ all possible coding schemes are considered
    - ∗ by generating the code randomly with appropriate distribution
  - ∘ and the performance of the system is averaged over them.

- Then it is proved that if $R < C$, the average error probability tends to zero.

- Again, Shannon proved that
  - as long as $R < C$,
  - at any arbitrarily small (but still positive) probability of error,
  - one can find (there exist) at least one code (with sufficiently long block length $n$) that performs better than the specified probability of error.

- If we used the scheme suggested and generate a code at random, the code constructed is likely to be good for long block lengths.

- No structure in the code. Very difficult to decode

**4.44.** Practical codes:

- In addition to achieving low probabilities of error, useful codes should be "simple", so that they can be encoded and decoded efficiently.

- Shannon's theorem does not provide a practical coding scheme.

- Since Shannon's paper, a variety of techniques have been used to construct good error correcting codes.
  - The entire field of coding theory has been developed during this search.

- Turbo codes have come close to achieving capacity for Gaussian channels.