Sirindhorn International Institute of Technology

Thammasat University

School of Information, Computer and Communication Technology

ECS452 2019/2 Part I.4 Dr.Prapun

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

3.1 Discrete Memoryless Channel (DMC) Models

In this section, we keep our analysis of the communication system simple by considering purely digital systems. (Recall that the transmitted signal from an antenna is an analog waveform.) To do this, as shown in Figure 5, we assume all non-source-coding parts of the system, including the physical (analog) channel, can be combined into an "equivalent channel" which we shall simply refer to in this section as the "channel".

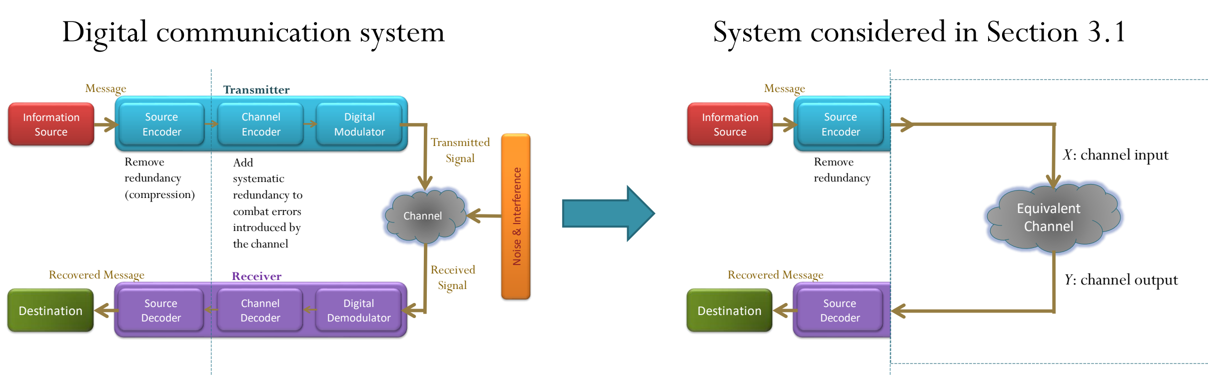


Figure 4: Equivalent Channel Considered in Section 3.1.

Example 3.1. In Chapter 2, the (equivalent) channel does not change (corrupt) its input. The channel output is assumed to be the same as the channel input.

Example 3.2. The **binary symmetric channel (BSC)**, which is the simplest model of a channel with errors, is shown in Figure 5.

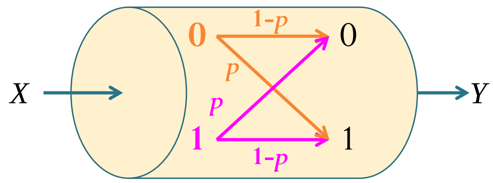


Figure 5: Binary symmetric channel and its channel diagram

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

Example 3.3. Consider a BSC whose samples of input and output are provided below

x: 1 0 1 1 1 1 1 1 1 1 1 1 0 1 0 1 1 1 1 1
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.

$P[X=0]$	$P[X=1]$
$P[Y=0]$	$P[Y=1]$
$P[Y=0 \mid X=0]$	$P[Y=1 \mid X=0]$
$P[Y=0 \mid X=1]$	$P[Y=1 \mid X=1]$

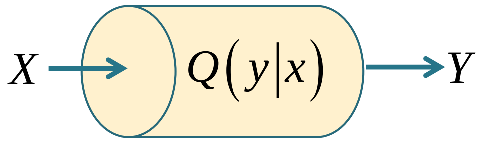


Figure 6: Discrete memoryless channel

Definition 3.4. Our general model for **discrete memoryless channel** (**DMC**) is shown in Figure 6.

- The **channel input** is denoted by a random variable X.
  - The pmf $p_X(x)$ is usually denoted by $p(x)$.
  - The support S_X is often denoted by $\mathcal{X}$.
    - $\mathcal{X}$ may be referred to as the **channel input alphabet**.
    - In many DMC, $|\mathcal{X}|$ is a power of two.
  - For finite $|\mathcal{X}|$, the whole pmf $p(x)$ is usually expressed in the form of a row vector $\underline{\mathbf{p}}$ or $\underline{\pi}$.
- Similarly, the **channel output** is denoted by a random variable Y.
  - The pmf $p_Y(y)$ is usually denoted by $q(y)$ and usually expressed in the form of a row vector $\underline{\mathbf{q}}$.
  - The support S_Y is often denoted by $\mathcal{Y}$ and referred to as the **channel output alphabet**.
- 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 $p_{Y|X}(y|x)$.

  - In this context, each conditional probability $p_{Y|X}(y|x)$ is usually referred to as the **channel transition probability**.
  - The conditional pmf $p_{Y|X}(y|x)$ is usually denoted by $Q(y|x)$.

and usually expressed in the form of a (probability) **transition matrix Q**:

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

This matrix is often referred to as the "matrix of transition probabilities" or simply the "**channel matrix**".

- The channel is called **memoryless**[10] 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.

- When the alphabets are collections of integers, we usually write $p(x)$ and $q(y)$ as p_x and q_y respectively.

  Alternatively, when the members of the alphabets are explicitly indexed (as $x_1, x_2, \ldots$ and $y_1, y_2, \ldots$), we often define

$$p_i \equiv p(x_i) \quad \text{and} \quad q_j = q(y_j).$$

3.5. The channel matrix **Q** is often defined or visualized in the form of the **channel diagram** as shown in Figure 8. Note that each arrow should be labeled with the transition probability $Q(y|x)$. See also Example 3.12.

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

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

where x_1^n denotes the vector $(x_1, x_2, \ldots, x_n)$.

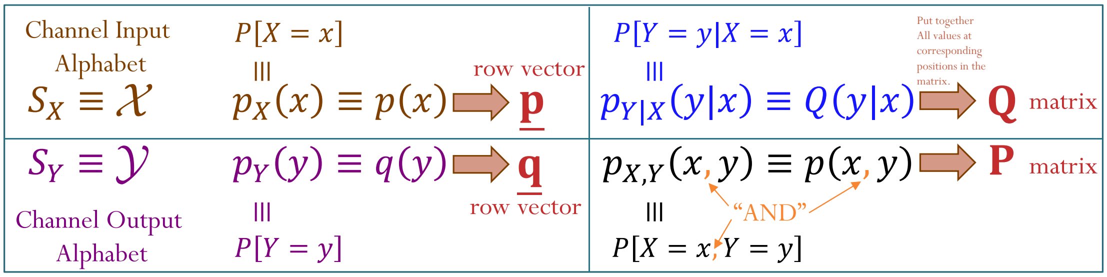


Figure 7: Notation involved in defining and describing characteristics of digital communication channels

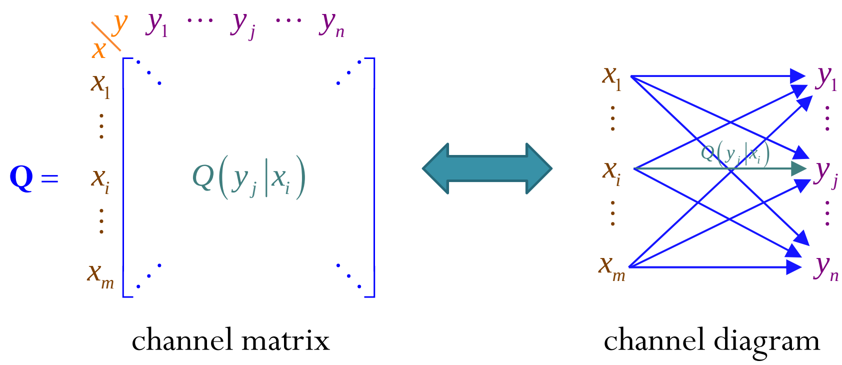
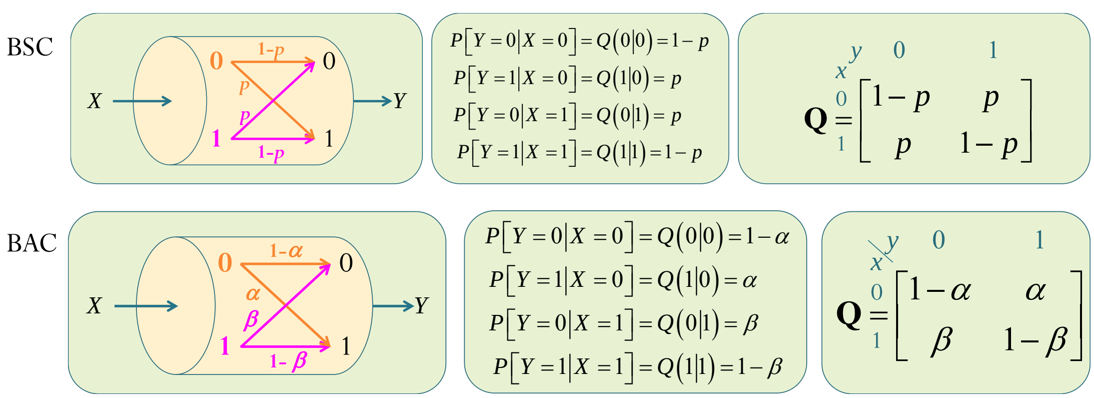
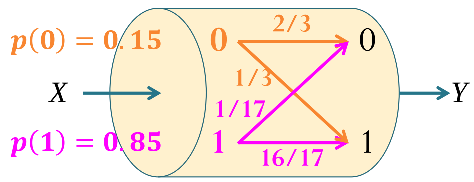


Figure 8: Conversion between the channel matrix and the channel diagram.

3.6. We now have three equivalent ways to specify a binary symmetric channel (BSC) defined in Example 3.2. A more general binary channel that may not be symmetric is called **binary asymmetric channel (BAC)**.



Example 3.7. For the binary channel estimated in Example 3.3,



Channel Input Alphabet

$\mathcal{X} \equiv S_X$

Input Probabilities

pmf $p_X(x) \equiv P[X = x]$

$p_X(0)$

$p_X(1)$

Input Probability Vector

$\underline{\mathbf{p}}$

Channel Transition Probabilities

Conditional pmf

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

$p_{Y|X}(0|0)$

$p_{Y|X}(1|0)$

$p_{Y|X}(0|1)$

$p_{Y|X}(1|1)$

Channel Matrix

$\mathbf{Q}$

Channel Output Alphabet

$\mathcal{Y} \equiv S_Y$

Output Probabilities

pmf $p_Y(y) \equiv P[Y = y]$

$p_Y(0)$

$p_Y(1)$

Output Probability Vector

$\underline{\mathbf{q}}$

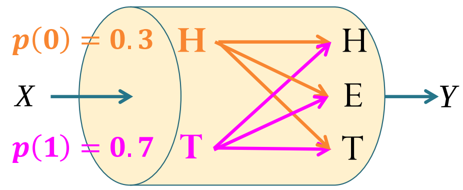
Example 3.8. Suppose, for a DMC, we have $\mathcal{X} = \{x_1, x_2\}$ and $\mathcal{Y} = \{y_1, y_2, y_3\}$. Then, its probability transition matrix $\mathbf{Q}$ is of the form

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

You may wonder how this $\mathbf{Q}$ happens in physical system. 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[11] (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.

[11]We assume the receiver knows which bits have been erased.

Example 3.9.



Channel Input Alphabet
$\mathcal{X}$
Input Probabilities
pmf $p(x) \equiv P[X = x]$
$p(H) \equiv P[X = H]$
$p(T) \equiv P[X = T]$
Input Probability Vector
$\underline{\mathbf{p}}$

Channel Transition Probabilities

Channel Matrix

$\mathbf{Q}$

Channel Output Alphabet
$\mathcal{Y}$
Output Probabilities
pmf $q(y) \equiv P[Y = y]$
$q(H) \equiv P[Y = H]$
$q(E) \equiv P[Y = E]$
$q(T) \equiv P[Y = T]$
Output Probability Vector
$\underline{\mathbf{q}}$

Example 3.10. 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

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

3.11. Observe that the sum along any row of the $\mathbf{Q}$ matrix is 1.

- This is different from the $\mathbf{P}$ matrix (the joint probability matrix) that was the main focus in basic probability class. Recall that, for $\mathbf{P}$ matrix, the sum of *all* elements in the matrix is 1.

∘ See 3.15 for more discussion about the $\mathbf{P}$ matrix.

Example 3.12. The channel diagram for a channel whose

$$\mathcal{X} = \{0, 1\}, \quad \mathcal{Y} = \{1, 2, 3\}, \quad \underline{\mathbf{p}} = [0.2 \ 0.8], \quad \text{and} \quad \mathbf{Q} = \begin{bmatrix} 0.5 & 0.2 & 0.3 \\ 0.3 & 0.4 & 0.3 \end{bmatrix}$$

is shown in Figure 9.

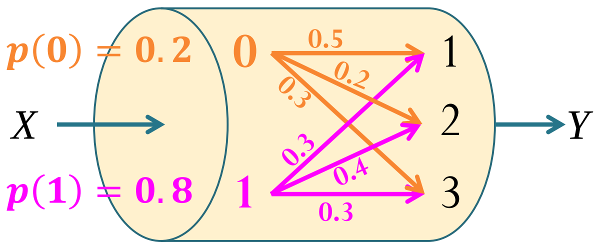


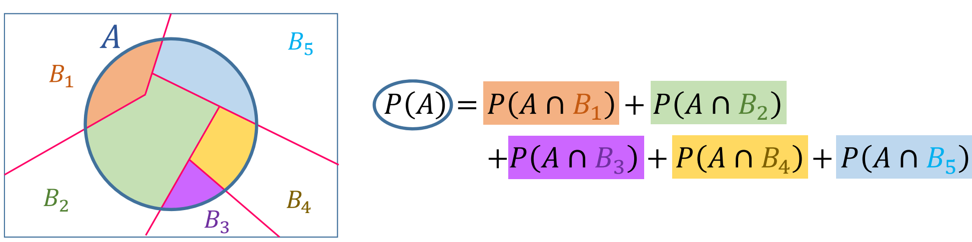
Figure 9: Channel diagram for Example 3.12.

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

$$\underline{\mathbf{q}} = \underline{\mathbf{p}}\mathbf{Q}. \tag{5}$$

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

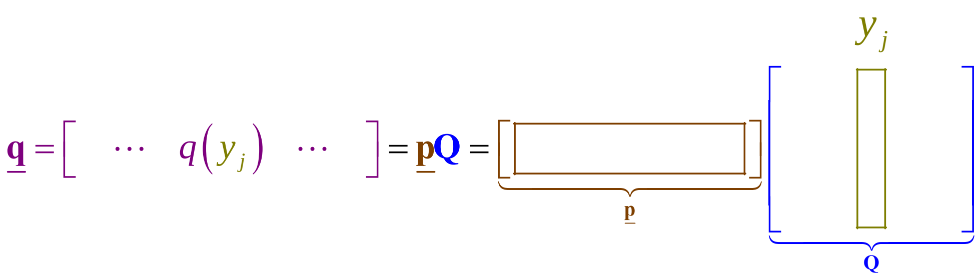
$$P(A) = \sum_x P(A \cap B_i) = \sum_x P(A|B_x)P(B_x). \tag{6}$$



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

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

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



Example 3.14. For a binary symmetric channel (BSC) defined in 3.2,

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

Therefore,

$$
\begin{aligned}
\underline{\mathbf{q}} &= \begin{bmatrix} q(0) & q(1) \end{bmatrix} = \left[\underline{\mathbf{p}} \begin{bmatrix} Q(0\,|0) \\ Q(0\,|1) \end{bmatrix} \;\; \underline{\mathbf{p}} \begin{bmatrix} Q(1\,|0) \\ Q(1\,|1) \end{bmatrix} \right] \\
&= \underline{\mathbf{p}} \begin{bmatrix} Q(0\,|0) & Q(1\,|0) \\ Q(0\,|1) & Q(1\,|1) \end{bmatrix} = \underline{\mathbf{p}}\mathbf{Q}
\end{aligned}
$$

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

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

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

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

Equivalently,

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

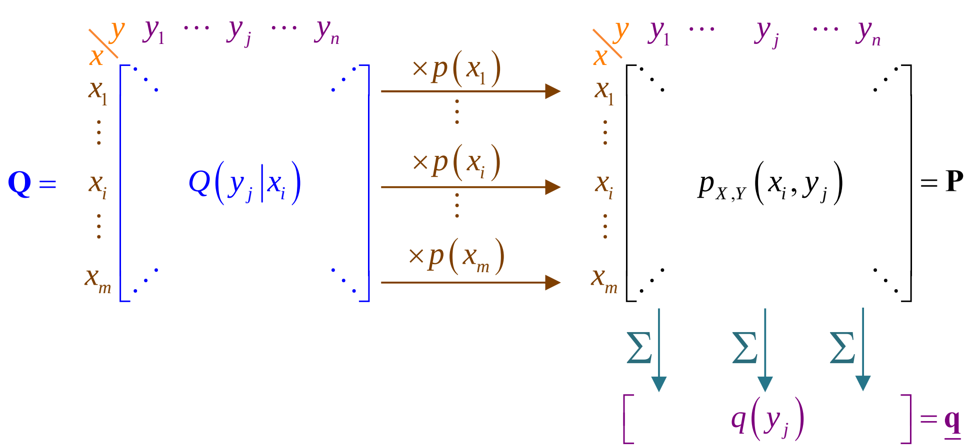


Figure 10: Conversion from the **Q** matrix to the **P** matrix and the output probability vector $\underline{\mathbf{q}}$.

Therefore, to get the matrix $\mathbf{P}$ from matrix $\mathbf{Q}$, we need to multiply each row of $\mathbf{Q}$ by the corresponding $p(x)$. This is illustrated in Figure 10. The same calculation 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 $\mathbf{Q}$:

$$\mathbf{P} = \left(\text{diag}\left(\underline{\mathbf{p}}\right)\right)\mathbf{Q}. \tag{7}$$

$$\underbrace{\begin{bmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & c & 0 \\ 0 & 0 & 0 & d \end{bmatrix}}_{\text{diag}\left(\left[\begin{array}{cccc} a & b & c & d \end{array}\right]\right)} \begin{bmatrix} 5 & 4 & 6 & 6 \\ 6 & 1 & 6 & 3 \\ 1 & 2 & 1 & 5 \\ 6 & 4 & 6 & 1 \end{bmatrix} = \begin{bmatrix} 5a & 4a & 6a & 6a \\ 6b & b & 6b & 3b \\ c & 2c & c & 5c \\ 6d & 4d & 6d & d \end{bmatrix}$$

Remarks:

(a) Both $\mathbf{P}$ and $\mathbf{Q}$ give the statistical relationship between the two random variables X and Y.

(b) The $\mathbf{P}$ matrix gives complete information about X and Y. Any probability calculation involving X and Y can be found from the $\mathbf{P}$ matrix.

(c) However, from (7) above, we see that knowing the $\mathbf{p}$ vector and the $\mathbf{Q}$ matrix also gives the complete information as well.

Once the $\mathbf{P}$ matrix is obtained, we can calculate the output probability vector $\underline{\mathbf{q}}$ by adding the elements of $\mathbf{P}$ along each column; this gives

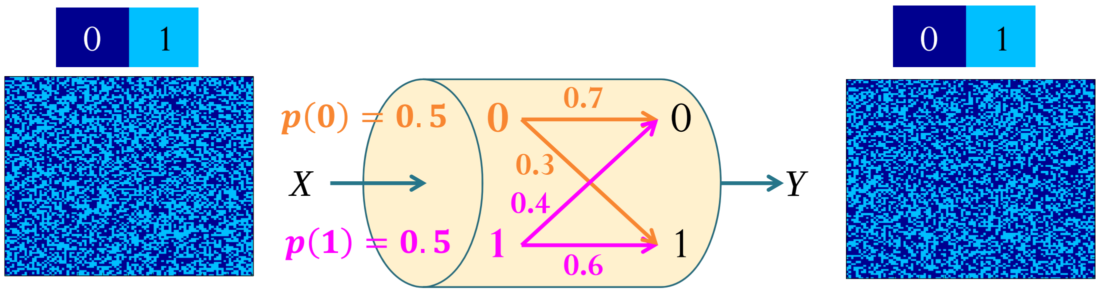
$$\underline{\mathbf{q}} = \left[\begin{array}{cccc} 1 & 1 & \cdots & 1 \end{array}\right]\mathbf{P} = \left[\begin{array}{cccc} 1 & 1 & \cdots & 1 \end{array}\right]\text{diag}\left(\underline{\mathbf{p}}\right)\mathbf{Q} = \underline{\mathbf{p}}\mathbf{Q}.$$

$$\left[\begin{array}{cccc} 1 & 1 & 1 & 1 \end{array}\right] \begin{bmatrix} 5 & 4 & 6 & 6 \\ 6 & 1 & 6 & 3 \\ 1 & 2 & 1 & 5 \\ 6 & 4 & 6 & 1 \end{bmatrix} = \left[\begin{array}{cccc} 18 & 11 & 19 & 15 \end{array}\right]$$

$$\left[\begin{array}{cccc} 1 & 1 & 1 & 1 \end{array}\right] \underbrace{\begin{bmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & c & 0 \\ 0 & 0 & 0 & d \end{bmatrix}}_{\text{diag}\left(\left[\begin{array}{cccc} a & b & c & d \end{array}\right]\right)} = \left[\begin{array}{cccc} a & b & c & d \end{array}\right]$$

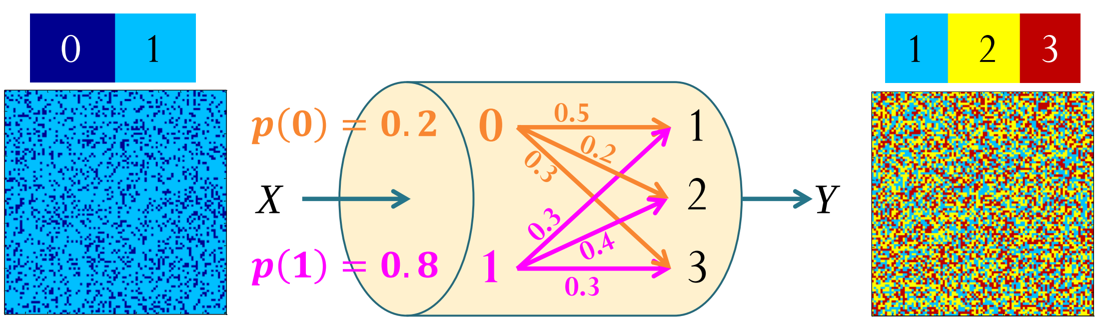
Example 3.16. Binary Asymmetric Channel (BAC): Consider a binary input-output channel whose matrix of transition probabilities is

$$\mathbf{Q} = \begin{bmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{bmatrix}.$$



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. [18, Ex. 11.3]

Example 3.17. Find the output probability vector $\underline{\mathbf{q}}$ and the joint probability matrix $\mathbf{P}$ for the DMC defined in Example 3.12:



3.2 Decoder and Symbol Error Probability

3.18. 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[12] X from the value of the received channel output Y. We denote this guessed value by $\hat{X}$.

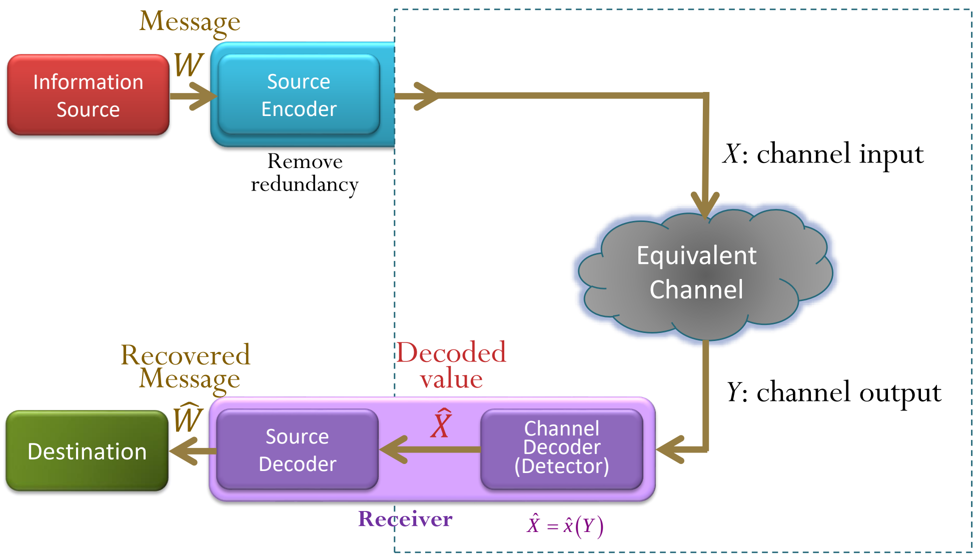


Figure 11: Adding a (channel) decoder to improve the performance of the system in Figure 4 considered in Section 3.1.

3.19. A "good" decoder is the one that (often) guesses correctly. So, our goal here is to

- ____________ the probability of ____________ guessing
- ____________ the probability of ____________ guessing

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

[12] 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.)

Definition 3.20. The **(symbol) error probability**, denoted by $P(\mathcal{E})$, can be calculated from

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

- $\mathcal{E}$ is the (decoding) error event.
- A dual (complementary) quantity is the probability of correct decoding:

$$P(\mathcal{C}) = P\left[\hat{X} = X\right] = 1 - P(\mathcal{E}).$$

- For channel with binary input, the error probability is the same as bit error rate (BER).

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

We use $\hat{x}(\cdot)$ to denote this function $g(\cdot)$; So, $\hat{X} = \hat{x}(Y)$. It is a deterministic function operating on the random channel output Y ; the randomness in the decoded value $\hat{X}$ comes from the randomness in Y.

Definition 3.22. A "**naive**" decoder is a decoder that simply sets $\hat{X} = Y$. Equivalently,

$$\hat{x}_{\text{naive}}(y) = y.$$

- We may call this a "pass-through" decoder or a "do-nothing" decoder.
- It usually has poor performance. However, we shall see that, in some cases, this simple decoder can be optimal[13].

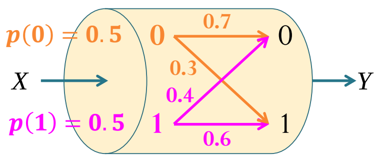
3.23. For general DMC, the error probability of the naive decoder is

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

[13] "Optimal" means nothing has better error probability in that particular situation.

Example 3.24. Consider the BAC channel and input probabilities specified in Example 3.16. Find $P(\mathcal{E})$ when $\hat{X} = Y$.

(a) Method 1:



$$\mathbf{Q} = \begin{bmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{bmatrix}$$

(b) Method 2: To calculate the error probability directly, we write

$$\begin{aligned} P(\mathcal{E}) &= P\left[\hat{X} \neq X\right] = P[Y \neq X] \\ &= P[Y = 0, X = 1] + P[Y = 1, X = 0] \\ &= P[Y = 0 \,|X = 1]\, P[X = 1] + P[Y = 1 \,|X = 0]\, P[X = 0] \\ &= 0.4 \times \frac{1}{2} + 0.3 \times \frac{1}{2} = 0.35. \end{aligned}$$

(c) Method 3: Using the derived formula from 3.23,

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

Example 3.25. Continue from Examples 3.12 and 3.17. 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 $\underline{\mathbf{p}} = [0.2, 0.8]$.

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

Example 3.26. 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 3.25.

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

Example 3.27. Repeat Example 3.26 but use the following decoder

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

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

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

(a) Find the **P** matrix by scaling each row of the **Q** matrix by its corresponding $p(x)$.

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

(c) For column y in the **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. $P(\mathcal{E}) = 1 - P(\mathcal{C})$.

Sirindhorn International Institute of Technology

Thammasat University

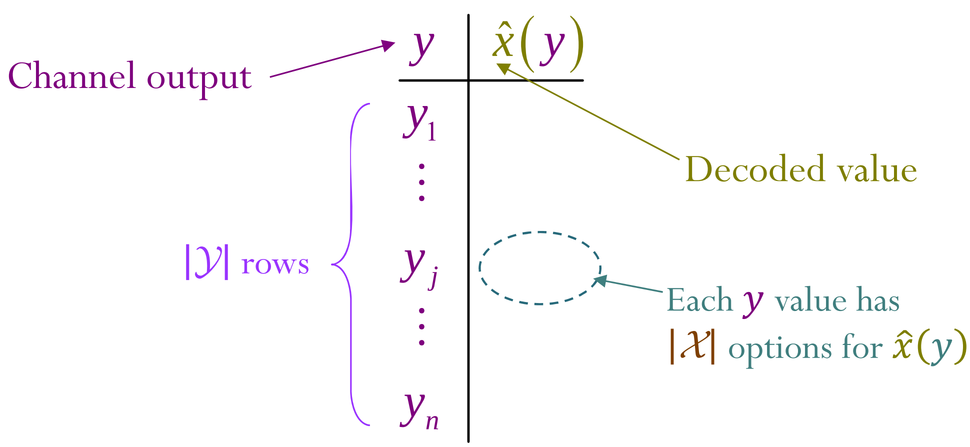
School of Information, Computer and Communication Technology

ECS452 2019/2 Part I.5 Dr.Prapun

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.

3.29. How many "reasonable" decoders are there?: Recall from 3.21 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 3.26 and Example 3.27. 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 3.30. The naive decoder in Example 3.25 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 3.31. "Reasonable" Decoder for BSC: For BSC in Example 3.2, any decoder has to answer two important questions:

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

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

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

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

So, only four reasonable decoders for BSC:

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

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

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

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

Example 3.32. For the DMC defined in Example 3.25, how many reasonable decoders are there?

We calculate the error probability of three decoders in Example 3.25, Example 3.26, and Example 3.27. There are still many reasonable possibilities to evaluate. Using `MATLAB`, we can find the error probability for all possible reasonable decoders:

y	$\hat{x}_{D1}(y)$	$\hat{x}_{D2}(y)$	$\hat{x}_{D3}(y)$	$\hat{x}_{D4}(y)$	$\hat{x}_{D5}(y)$	$\hat{x}_{D6}(y)$	$\hat{x}_{D7}(y)$	$\hat{x}_{D8}(y)$
1	0	0	0	0	1	1	1	1
2	0	0	1	1	0	0	1	1
3	0	1	0	1	0	1	0	1
$P(\mathcal{E})$	0.80	0.62	0.52	0.34	0.66	0.48	0.38	0.20

3.33. 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.

3.34. From the recipe 3.28 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 3.35. For the DMC and the input probablities defined in Example 3.25, the joint pmf matrix $\mathbf{P}$ was found to be

$$
\begin{array}{ll}
x \setminus y & \begin{array}{ccc} \;\;1\;\; & \;\;2\;\; & \;\;3\;\; \end{array} \\
\begin{array}{l} 0 \\ 1 \end{array} & \begin{bmatrix} 0.1 & 0.04 & 0.06 \\ 0.24 & 0.32 & 0.24 \end{bmatrix}
\end{array}
$$

Therefore, the optimal decoder is

3.36. Deriving the optimal decoder: 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[14] 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[15] $P[X = x|Y = y]$:

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

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

Definition 3.37. The optimal decoder derived in 3.36 is called the **maximum a posteriori probability (MAP) decoder**:

$$\hat{x}_{\text{MAP}}(y) = \hat{x}_{\text{optimal}}(y) = \arg\max_x P[X = x \,|Y = y]. \tag{8}$$

3.38. 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.

[14] 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 $P(\mathcal{C})$.

[15] For those who are not familiar with the "arg max" (arguments of the maximum) function,

$$\arg\max_x f(x) = \text{the } x \text{ value that maximizes } f(x).$$

The corresponding maximum value of $f(x)$ is $\max_x f(x)$. In other words, in contrast to global maximum, referring to the largest outputs of a function, $\arg\max$ refers to the inputs, or arguments, at which the function outputs are as large as possible. For example, for $f(x) = 5 - x^2$, we have $\arg\max_x f(x) = 0$ and $\max_x f(x) = 5$.

- 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 $P\left[X = x\right]$.
- Knowing $Y = y$, the decoder can update its probability about x from $P[X = x]$ to $P\left[X = x|Y = y\right]$. Therefore, the decoder should guess the value of the most likely x value conditioned on the fact that $Y = y$.

3.39. We should manipulate Formula (3.37) for the MAP decoder a bit further because, in practice, we usually only know $p(x)$ and $Q(y|x)$. To connect these terms to $P\left[X = x|Y = y\right]$ required in (3.37), 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 $A = [Y = y]$.

Therefore,

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

Note that the term $P\left[Y = y\right]$ does not depend on x and it is positive; therefore, it does not change the the answer of arg max and hence can be ignored.

3.40. A recipe for finding the MAP decoder (optimal decoder) and its corresponding error probability:

(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 3.41. We have applied recipe 3.40 back when we try to find the optimal decoder in Example 3.35.

Example 3.42. Find the MAP decoder and its corresponding error probability for the DMC channel whose $\mathbf{Q}$ matrix is given by

$$\begin{array}{cc} x \setminus y & \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{array}{c} 0 \\ 1 \end{array} & \begin{bmatrix} 0.5 & 0.2 & 0.3 \\ 0.3 & 0.4 & 0.3 \end{bmatrix} \end{array}$$

and $\underline{\mathbf{p}} = [0.6, 0.4]$. Note that the DMC is the same as in Example 3.25 but the input probabilities are different.

Definition 3.43. 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 (9). This decoder is called the **maximum likelihood (ML)** decoder:

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

3.44. ML decoder is the same as the 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.

3.45. 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 $\mathbf{P}$ matrix, select the elements corresponding to the selected positions in the $\mathbf{Q}$ matrix.

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

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

Example 3.46. Find the ML decoder and its corresponding error probability for the DMC channel in Example 3.25 whose $\mathbf{Q}$ matrix is

$$\begin{array}{cc} x \setminus y & \begin{array}{ccc} 1 & \ \ 2 & \ \ 3 \end{array} \\ \begin{array}{c} 0 \\ 1 \end{array} & \begin{bmatrix} 0.5 & 0.2 & 0.3 \\ 0.3 & 0.4 & 0.3 \end{bmatrix} \end{array}$$

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

Example 3.47. Find the ML decoder and the corresponding error probability for a communication over BSC with $p = 0.1$ and $p_0 = 0.8$.

Note that

- the prior probabilities p_0 (and p_1) is not used when finding $\hat{x}_{\mathrm{ML}}$,
- 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.

3.48. MAP vs. ML:

$$\hat{x}_{\text{MAP}}(y) = \arg\max_x p_{X,Y}(x,y)$$
$$= \hat{x}_{\text{optimal}}(y)$$
$$= \arg\max_x \overbrace{P\left[X = x \middle| Y = y\right]}^{\text{a posteriori probability}}$$
$$= \arg\max_x Q(y|x)\overbrace{p(x)}^{\text{prior probability}}$$

- Decoder is derived from the **P** matrix
- Select (by circling) the maximum value in each column (for each value of y) in the **P** matrix.
- The corresponding x value is the value of $\hat{x}_{\text{MAP}}(y)$.

$$\hat{x}_{\text{ML}}(y) = \arg\max_x \overbrace{Q(y|x)}^{\text{likelihood funtion}}$$

Optimal at least when $p(x)$ is uniform (the channel inputs are equally likely)

Can be derived without knowing the channel input probabilities.

- Decoder is derived from the **Q** matrix
- Select (by circling) the maximum value in each column (for each value of y) in the **Q** matrix.
- The corresponding x value is the value of $\hat{x}_{\text{ML}}(y)$.

- Once the decoder (the decoding table) is derived $P(\mathcal{C})$ and $P(\mathcal{E})$ are calculated by adding the corresponding probabilities in the **P** matrix.

3.49. 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

3.50. 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)

3.51. By the memoryless property of the channel,

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

Example 3.52. For a DMC in which $\mathcal{X} = \{0,1\}$, $\mathcal{Y} = \{1,2,3\}$, $\mathbf{Q} = \begin{bmatrix} 0.5 & 0.2 & 0.3 \\ 0.3 & 0.4 & 0.3 \end{bmatrix}$, find

(a) $Q(122|100)$

(b) $Q(333|111)$

Example 3.53. For BSC, find

(a) $Q(101|100)$

(b) $Q(111|111)$

3.54. For BSC,

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

Therefore,

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

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.

Example 3.55. $d\,(101, 100) =$ ___, $d\,(111, 111) =$ ___,
$d\,(00101, 01111) =$ ___

3.56. 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:

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

- 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:

$$\hat{\underline{\mathbf{x}}}_{\text{ML}}\left(\underline{\mathbf{y}}\right) = \arg\max_{\underline{\mathbf{x}}} Q\left(\underline{\mathbf{y}}\,|\underline{\mathbf{x}}\right). \tag{13}$$

Plugging-in

$$Q\left(\underline{\mathbf{y}}|\underline{\mathbf{x}}\right) = p^{d\left(\underline{\mathbf{x}},\underline{\mathbf{y}}\right)}(1-p)^{n-d\left(\underline{\mathbf{x}},\underline{\mathbf{y}}\right)} = \left(\frac{p}{1-p}\right)^{d\left(\underline{\mathbf{x}},\underline{\mathbf{y}}\right)}(1-p)^n, \tag{14}$$

from (11), gives

$$\begin{aligned}
\hat{\underline{\mathbf{x}}}_{\text{MAP}}\left(\underline{\mathbf{y}}\right) &= \arg\max_{\underline{\mathbf{x}}} \left(\frac{p}{1-p}\right)^{d\left(\underline{\mathbf{x}},\underline{\mathbf{y}}\right)}(1-p)^n p\left(\underline{\mathbf{x}}\right) &(15)\\
&= \arg\max_{\underline{\mathbf{x}}} \left(\frac{p}{1-p}\right)^{d\left(\underline{\mathbf{x}},\underline{\mathbf{y}}\right)} p\left(\underline{\mathbf{x}}\right). &(16)
\end{aligned}$$

and

$$\hat{\underline{\mathbf{x}}}_{\text{ML}}\left(\underline{\mathbf{y}}\right) = \arg\max_{\underline{\mathbf{x}}} \left(\frac{p}{1-p}\right)^{d\left(\underline{\mathbf{x}},\underline{\mathbf{y}}\right)}. \tag{17}$$

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

From (17) (or directly from (11)), 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, $\hat{\underline{\mathbf{x}}}_{\text{ML}}\left(\underline{\mathbf{y}}\right)$ should be the codeword $\underline{\mathbf{x}}$ which has the minimum distance from the observed $\underline{\mathbf{y}}$:

$$\hat{\underline{\mathbf{x}}}_{\text{ML}}\left(\underline{\mathbf{y}}\right) = \arg\min_{\underline{\mathbf{x}}} d\left(\underline{\mathbf{x}},\underline{\mathbf{y}}\right). \tag{18}$$

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

3.58. 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.

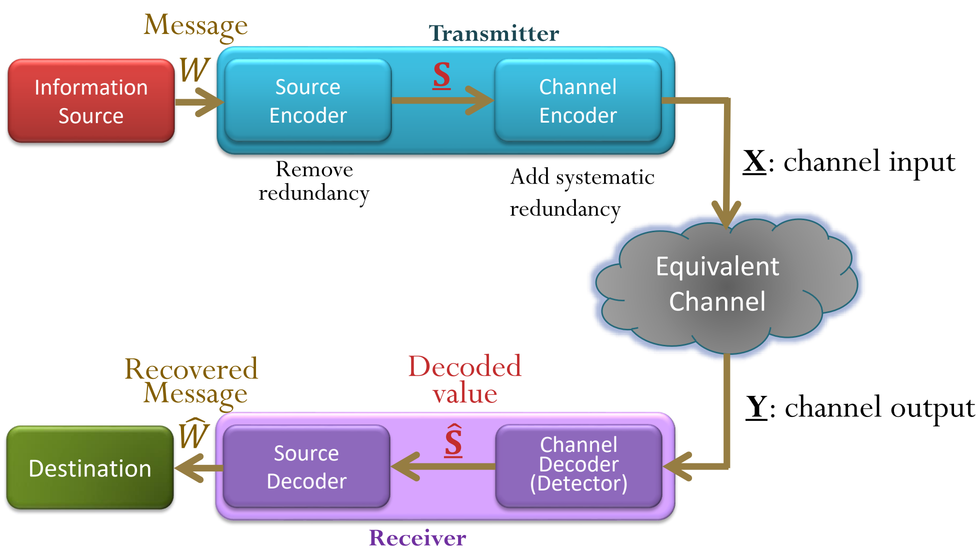


Figure 12: System model for Section 3.5. A channel encoder is added to improve the performance of the system considered in Figure 11 in Section 3.2.

- Note that variables X and Y are still used for the channel input and channel output, respectively. However, as in Section 3.4, we consider blocks (vectors) of them. Therefore, the variables used are $\underline{\mathbf{X}}$ and $\underline{\mathbf{Y}}$.
- Because we introduce another box between the source encoder and the (equivalent) channel, the output of the source encoder is not the same as the channel input anymore. Therefore, we rename the output of the source encoder as S. Again, when we consider a block of output from the source encoder, we denote it by $\underline{\mathbf{S}}$.
- The job of the decoder is now to (correctly) guess the value of $\underline{\mathbf{S}}$. Its output is now denoted by $\hat{\underline{\mathbf{S}}}$.
    - Usually, the mapping (by the channel encoder) from $\underline{\mathbf{S}}$ to $\underline{\mathbf{X}}$ is bijective[16]; so is the mapping from W to $\underline{\mathbf{S}}$ by the source encoder.

[16] A bijection, bijective function, or one-to-one correspondence is a function between the elements of two sets, where each element of one set is paired with exactly one element of the other set, and each element of the other set is paired with exactly one element of the first set.

Therefore, one can also say that, as before, the job of the decoder is still to (correctly) guess the value of $\underline{\mathbf{X}}$. Once we have the value of $\underline{\mathbf{X}}$, we can directly map it back to $\underline{\mathbf{S}}$ and then the original message W.

3.59. Repetition Code: A simple example of channel encoding is to 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

3.60. Two classes of channel codes

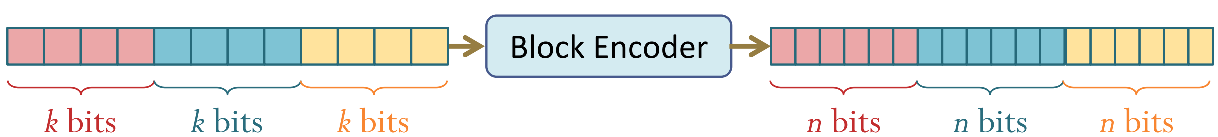
(a) Block codes

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

(b) Convolutional codes

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

Definition 3.61. Block Encoding: Take k (information) bits at a time and map each k-bit sequence into a (unique) n-bit sequence, called a **codeword**[17].



- The code is called (n, k) code.
- Working with k-info-bit blocks means there are potentially $M = 2^k$ different information blocks.

[17]Yes, we used this term already in Chapter 2. Both uses of the term "codeword" denote the outputs of the encoding processes.

- 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 $\underline{\mathbf{s}}^{(1)}, \underline{\mathbf{s}}^{(2)}, \ldots, \underline{\mathbf{s}}^{(M)}$.
  The corresponding M codewords are denoted by $\underline{\mathbf{x}}^{(1)}, \underline{\mathbf{x}}^{(2)}, \ldots, \underline{\mathbf{x}}^{(M)}$, respectively.

index i	info-block $\underline{\mathbf{s}}$	codeword $\underline{\mathbf{x}}$
1	$\underline{\mathbf{s}}^{(1)} = 000\ldots0$	$\underline{\mathbf{x}}^{(1)} =$
2	$\underline{\mathbf{s}}^{(2)} = 000\ldots1$	$\underline{\mathbf{x}}^{(2)} =$
$\vdots$	$\vdots$	$\vdots$
M	$\underline{\mathbf{s}}^{(M)} = 111\ldots1$	$\underline{\mathbf{x}}^{(M)} =$

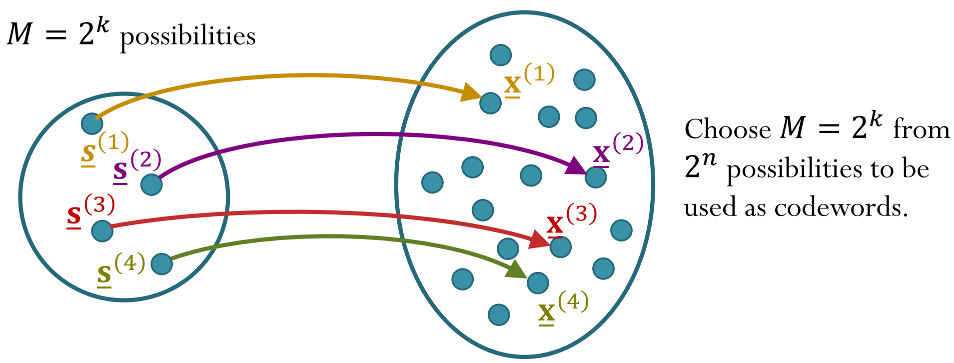


Figure 13: The mapping for block encoding.

- 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 3.62. Find the codebook and code rate for the encoder which uses repetition code with $n = 5$.

Example 3.63. 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 over BSC are possible?

3.64. 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. Knowing its index is enough to indicate which info-block produced it.

Example 3.65. Repetition Code and Majority Voting: Back to Example 3.59.

First recall that

(1) MAP decoder is optimal. (It minimizes $P(\mathcal{E})$).

(2) ML decoder is suboptimal. However, it can be optimal (same $P(\mathcal{E})$ as the MAP decoder), for example, when the codewords are equally-likely.

(3) ML decoder is the same as the minimum distance decoder when the crossover probability of the BSC p is < 0.5 (which is usually the case).

Therefore, minimum distance decoder can be optimal in many situations.

In this example, assume $p < 0.5$. Let $\underline{\mathbf{0}}$ and $\underline{\mathbf{1}}$ denote the n-dimensional row vectors $00\ldots0$ and $11\ldots1$, respectively. Observe that

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

Therefore, the minimum distance decoder is

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

Equivalently,

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

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

The corresponding error probability is

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

For example, when $p = 0.01$, we have $P(\mathcal{E}) \approx 10^{-5}$. Figure 14 shows how we can view this as having the original BSC channel replaced by a new one with better crossover probability.

Figure 15 compares the error probability when different values of n are used.

- 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.

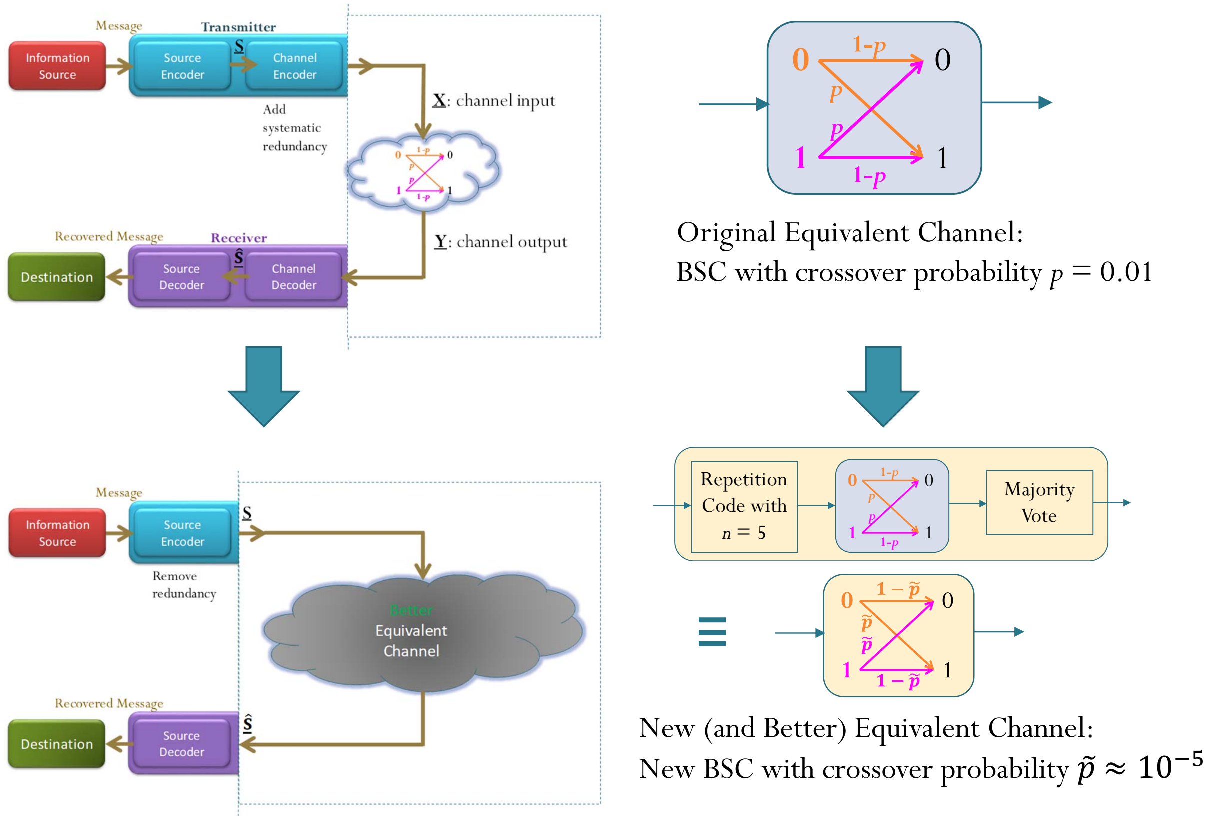


Figure 14: With the addition of channel encoder and channel decoder, the performance of the system is improved. The original BSC, when combined with the channel encoder and channel decoder, can be viewed as a new equivalent BSC with better crossover probability.

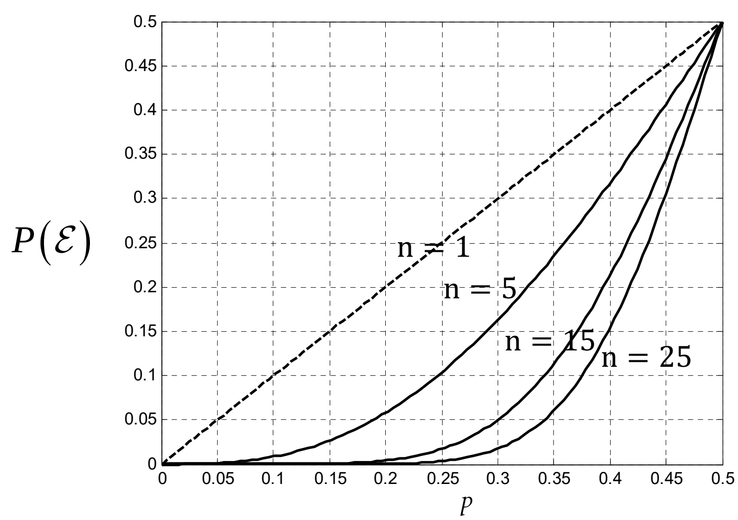


Figure 15: Error probability for a system that uses repetition code at the transmitter (repeat each info-bit n times) and majority voting at the receiver. The channel is assumed to be binary symmetric with crossover probability p.

Example 3.66. Consider a BSC whose crossover probability is $p = 0.2$. A channel code use the following codebook:

s	$\underline{\mathbf{x}}$
0	011
1	100

(a) Suppose the codeword 011 was transmitted. What is the probability that the channel output is 101?

(b) Assume that the two possibilities of the info-bit S are equally likely. Suppose we observed 101 at the output of this channel.

(i) What is the probability that the codeword 011 was transmitted?

(ii) What is the probability that the codeword 100 was transmitted?

(iii) At the receiver, if a MAP decoder is used, find the decoded codeword and the corresponding decoded info-bit.

3.67. 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 chapter.