

# Information Theory

Po-Ning Chen, Professor

Department of Electrical and Computer Engineering

National Chiao Tung University

Hsin Chu, Taiwan 30010, R.O.C.

# Chapter 0

## Overview : Philosophy behind Information Theory

Po-Ning Chen, Professor

Department of Electrical and Computer Engineering

National Chiao Tung University

Hsin Chu, Taiwan 30010, R.O.C.

# Philosophy behind Information Transmission

I: 0-i

- Work Philosophy of General Researchers
  - Design a code or transmission scheme
  - Show either by mathematics or by simulations that it works well.
- Work Philosophy of Information Theorists
  - Design a code or transmission scheme (perhaps done by some other researchers)
  - Show either by mathematics or by simulations that it works well (perhaps done by some other researchers)
  - Find the cause why the scheme works well. I.e., find the significant feature that makes a scheme work well by showing:
    - \* all the schemes with such feature works well;
    - \* all the schemes without such feature cannot work well.
- After showing the above two points (by either mathematics or simulations), we can claim that we find the key factor that determines the system design.
- One can then base on the knowledge (about the key factor) to improve the old design (or knowing that the old design is optimal, and there is no room for improvement).

# A Philosophy behind Information Theory

I: 0-ii

- Information Theory is a philosophy behind *information manipulation*, such as *information transmission*. It answers the *fundamental questions* not only qualitatively but quantitatively.
- This *philosophy* can be applied to many other areas that involve *information manipulation* in other forms, such as:
  - data compression
  - statistical physics (thermodynamics, quantum information theory);
  - computer science (algorithmic complexity, resolvability);
  - probability theories (large deviations, limit theorems);
  - statistical theories (hypothesis testing, multi-user detection, Fisher information, estimation);
  - economics (gambling and investment theories);
  - biology (biological information theory);
  - cryptography (theories behind security and watermarking, etc.);
  - networking (self-similarity, traffic regulation theory);
  - and more...etc.

# Example

I: 0-iii

- Work Philosophy of Information Theorists
  - Design a switch
  - Show either by mathematics or by simulations that it works well.
  - Find the cause why it works well. I.e., find the significant feature that makes it work well.
    - \* All the switches with such feature works well.
    - \* All the switches without such feature cannot work well.
  - After showing the above two points (either by mathematics or by simulations), we can claim that the key factor that determines the system design has been located.
  - One can then base on the knowledge (about the key factor) to refine the old design (or knowing that the old design is optimal, and there is no room for improvement).
- Quantitative interests may rely on:
  - What is the underlying limit that a good switch could achieve?
  - How can this limit be achieved?

# Syllabus

I: 0-iv

## **Instructor information :**

Po-Ning Chen

Engineering Building 4, Room 831

Phone : 03-5731670

email: poning@faculty.nctu.edu.tw

## **Text :**

There is no *required* text. A set of copyrighted class notes will be provided in installments. You may obtain the latest version of lecture notes, titled *Information Theory for Single-User Systems*, from <http://shannon.cm.nctu.edu.tw>.

The following is a list of *recommended* references:

1. *A Student's Guide to Coding and Information Theory*, Stefan M. Moser and Po-Ning Chen, Cambridge University Press, January 2012.
2. *Elements of Information Theory*, Thomas M. Cover and Joy A. Thomas, 2nd edition, John Wiley & Sons, Inc., July 2006.
3. *A First Course in Information Theory (Information Technology: Transmission, Processing, and Storage)*, Raymond W. Yueng, Plenum Pub Corp., May 2002.

# Syllabus

I: 0-v

4. *Principles and Practices of Information Theory*, Richard E. Blahut, Addison Wesley, 1988.
5. *Information Theory and Reliable Communication*, Robert G. Gallager, 1985.
6. *Information Theory*, Robert B. Ash, Dover Publications, Inc., 1965.
7. *Mathematical Foundations of Information Theory*, A. I. Khinchin, Dover Publications, Inc., 1957.

# Syllabus

I: 0-vi

## **Grading System :**

- Your semester grade will be contributed equally by the *midterm exam*, the *final exam*, and a *written report* on a self-selected paper on *IEEE transactions on Information Theory*.

A list of the key factors that affect your grade of the report is provided below.

- 50% will be emphasized on your answers to the following two questions:
  - (1) What is the problem the paper tried to solve (Problem formation)?
  - (2) What is the achievement or the conclusion of the paper?
- 50% will be contributed from:
  - (3) Technical background dated back to the time the paper was written. (Some students may select a very old paper, so some of the problems may already have a better solution than the one proposed in the paper. Hence, the technical background dated back to the time the paper was written is important.)
  - (4) The key idea (on which the solution is based) proposed in the paper.
  - (5) Something, in your opinion, needs to be further explored.
  - (6) Personal comments on the paper.
  - (7) Any potential applications.

# Syllabus

I: 0-vii

(8) Any potential future extensions.

- Please note that a written report that selectively copies the sentences from the paper is not preferred. I will appreciate more to a written report that is full of “your own words.” Hence, it is suggested that you read the paper several times, dig into its details, collect the thoughts, and then put the paper aside before you start to write the report.

Please be noted that the selected paper has to be one that is published on IEEE transactions on Information Theory. Papers published in other journals are not acceptable in this course.

# Syllabus

I: 0-viii

## **Lecture Schedule :**

- The first lecture will be given on **September 26**.
- You need to submit the *title, authors, page numbers, month* and *year* of the paper you select for reporting by the end of **November 7** lecture. I will then post the list of papers being selected on my website two days after. Students who fail to submit his/her choice in time will be deducted **50%** from his/her *paper study* grade.
- Some students may not like to select a paper that others have selected, so you can switch to other paper at any time before the end of **November 28** lecture. I will constantly update the list on my website so that the most recent choices are always posted. Note that **November 28** will be a strict deadline for the submission of the paper title that you wish to study.

# Syllabus

I: 0-ix

- There will be no lecture on **September 19** and **October 10** because they are holidays.
- **Midterm** will be held on November 14. The coverage of midterm will be decided later.
- The last lecture will be given on **January 9, 2014**.
- **Final exam** will be held on **January 16, 2014**.
- The written report will be due on **January 9, 2014**. You shall submit it before the lecture ends.

# Chapter 1

## Introduction

Po-Ning Chen, Professor

Department of Electrical and Computer Engineering

National Chiao Tung University

Hsin Chu, Taiwan 30010, R.O.C.

# Introduction

I: 1-1

- What is information?
  - Uncertainty
    - \* Information is a message that is previously *uncertain* to receivers.
- Representation of Information
  - After obtaining the information, one may wish to store it or convey it to others; this raises the question that:
    - how to represent information for ease of storage or conveying it to others?*

# Representation of Information

I: 1-2

- How to represent information for ease of storage or conveying it to others?

An answer from an engineer:

– Reality:

\* 26 English letters and their concatenations  $\implies$  Language

– Computer and Digital Communications:

\* 0-1 symbols and their concatenations  $\implies$  code.

After the information is symbolized, storage or conveyance of these symbols become straightforward.

# Dictionary and Codebook

I: 1-3

- Assumption made by the transmitter and also the receiver of symbolized information
  - All “possible symbols” of the conveyed information are priori known.
  - Receiver is only *uncertain* about which “possibility” is going to be received.
- Example. In a conversation using English,
  - it is priori known that one of the vocabularies in an English *dictionary* is going to be spoken.
  - Just cannot tell which before its reception.
- Example. In coded digital communications,
  - the *codebook* (or simply *code*)—the collection of all possible concatenations of pre-defined symbols—is always priori known (to the receiver).
  - Only uncertain about which is going to be received.

# Compactness of Codes

I: 1-4

- What is the “impact” upon
  - “describe the same information in terms of different dictionaries”or
  - “describe the same information in terms of different codebooks”
- Answer: different degree of compactness!
  - Some codebook may yield a more *lengthy* description than the other.
  - E.g., with event probabilities  $\{1/2, 1/4, 1/8, 1/8\}$ ,

$$\text{code 1} \left\{ \begin{array}{l} \text{event one} : 00 \\ \text{event two} : 01 \\ \text{event three} : 10 \\ \text{event four} : 11 \end{array} \right.$$

$$\text{code 2} \left\{ \begin{array}{l} \text{event one} : 0 \\ \text{event two} : 10 \\ \text{event three} : 110 \\ \text{event four} : 111 \end{array} \right.$$

$$\begin{aligned} & \text{Average codeword length} \\ &= (1/2) \times 2 \text{ bits} + (1/4) \times 2 \text{ bits} \\ &+ (1/8) \times 2 \text{ bits} + (1/8) \times 2 \text{ bits} \\ &= 2 \text{ bits per event} \end{aligned}$$

$$\begin{aligned} & \text{Average codeword length} \\ &= (1/2) \times 1 \text{ bits} + (1/4) \times 2 \text{ bits} \\ &+ (1/8) \times 3 \text{ bits} + (1/8) \times 3 \text{ bits} \\ &= 7/4 \text{ bits per event (**more compact**)} \end{aligned}$$

## How to find the most compact code?

I: 1-5

- Straightforward Approach
  - To exhaust the average codeword lengths of **all** possible code designs and pick the one with the smallest average codeword length
  - *A tedious work if the number of events is large.*
- Alternative Approach
  - Derive the minimum average codeword length among all possible codes, and construct a code that achieves this minimum
  - *Is it possible to derive such minimum without exhausting all possible code designs?* (“Yes.” answered by Shannon. We can do this without performing a true code design, simply by means of **measuring** the **information** we are going to transmit.)

## How to measure information

I: 1-6

- Quantitative Definition of Information Content (Engineering view)
  - The average codeword length (usually, in bits) of the most compact code representing this information
- Under the above definition, engineers can directly determine the minimum space required to store the information based on the *information measure quantity*, namely, how many bits this information consists of.
- Question: This definition leads us to nowhere, since it may not be easy to find the most compact code directly.
  - It may be *possible* to exhaust all possible 4-event descriptive codes (two of them are illustrated in slide I: 1-4)
  - but as the number of events grows, the work becomes tedious and time-consuming.

# How to measure information

I: 1-7

- Quantitative Definition of Information Content (Probabilistic view)

- Axioms:

- \* **Monotonicity in event probability:** If an event is less likely to happen, it should carry more information when it occurs, because it is more uncertain that the event would happen.

- \* **Additivity:** It is reasonable to have “additivity” for information measure, i.e., the degree-of-uncertainty of a joint event should equal the sum of the degree-of-uncertainty of each individual event.

- \* **Continuity:** A small change in event probability should only yield a small variation in event uncertainty. For example, two events respectively with probabilities 0.20001 and 0.19999 should reasonably possess comparable information content.

- The only “measure” satisfying these axioms is:

$$\text{self-information of an event} = \log_2 \frac{1}{\text{event probability}} \text{ bits.}$$

(We will prove this claim later.)

- It is then legitimate to adopt the *entropy*—the expected value of the self-information—as a measure of information.

# Example of Entropy Computation

I: 1-8

**E.g.**, with event probabilities  $\{1/2, 1/4, 1/8, 1/8\}$ ,

$$\text{code 1} \left\{ \begin{array}{l} \text{event one} : 00 \\ \text{event two} : 01 \\ \text{event three} : 10 \\ \text{event four} : 11 \end{array} \right. \quad \text{code 2} \left\{ \begin{array}{l} \text{event one} : 0 \\ \text{event two} : 10 \\ \text{event three} : 110 \\ \text{event four} : 111 \end{array} \right.$$

$$\begin{aligned} & \text{Average codeword length} \\ & = 2 \text{ bits per event} \end{aligned}$$

$$\begin{aligned} & \text{Average codeword length} \\ & = \underline{7/4 \text{ bits per event}} \quad (\mathbf{more \ compact}) \end{aligned}$$

$$\left\{ \begin{array}{l} \text{self-information of event one} = \log_2 \frac{1}{1/2} = 1 \text{ bit} \\ \text{self-information of event two} = \log_2 \frac{1}{1/4} = 2 \text{ bits} \\ \text{self-information of event three} = \log_2 \frac{1}{1/8} = 3 \text{ bits} \\ \text{self-information of event four} = \log_2 \frac{1}{1/8} = 3 \text{ bits} \end{array} \right.$$

$$\text{Entropy} = \frac{1}{2} \times 1 \text{ bit} + \frac{1}{4} \times 2 \text{ bits} + \frac{1}{8} \times 3 \text{ bits} + \frac{1}{8} \times 3 \text{ bits} = \underline{\underline{\frac{7}{4} \text{ bits per event}}}$$

## Lessen from the previous example

I: 1-9

- The previous example seems to hint that code 2 is the most compact code among all possible code designs in the sense of having the *smallest average codeword length*.
- If this statement is true, then the two definitions on information content are equivalent:
  - (Engineering view) The average codeword length of the most compact code representing the information
  - (Probabilistic view) Entropy of the information
- In 1948, Shannon proved that the above two views are actually equivalent (under some constraints). I.e., the minimum average code length for a source descriptive code is indeed equal to the entropy of the source.
- One can then compute the entropy of a source, and assures that if the average codeword length of a code equals the source entropy, the code is optimal.

## Contribution of Shannon

I: 1-10

- Shannon's work laid the foundation for the field of information theory.
- His work indicates that the mathematical results of information theory can serve as a guide for the development of information manipulation systems.

# Measure of Compactness for a Code

I: 1-11

## **A few notes on the compactness of a code:**

- The *measure of information* is defined based on the definition of compactness.
  - The average codeword length of the most compact code representing the information
  - Here, “the most compact code” = “the code with the smallest average codeword length.”
  - Shannon shows “the smallest average codeword length” = entropy.
- Yet, the definition of *measure of code compactness* may be application-dependent. Some examples are:
  - the average codeword length (with respect to event probability) of a code (if the average codeword length is crucial for the application).
  - the maximum codeword length of a code (if the maximum codeword length is crucial for the application).
  - the average function values (cost or penalty) of codeword lengths of a code (e.g., if a larger penalty should apply to a longer codeword).

## Measure of Compactness for a Code

I: 1-12

$$\text{code 1} \left\{ \begin{array}{l} \text{event one} : 00 \\ \text{event two} : 01 \\ \text{event three} : 10 \\ \text{event four} : 11 \end{array} \right. \quad \text{code 2} \left\{ \begin{array}{l} \text{event one} : 0 \\ \text{event two} : 10 \\ \text{event three} : 110 \\ \text{event four} : 111 \end{array} \right.$$

$$\begin{array}{l} \text{Average codeword length} \\ = 2 \text{ bits per event} \end{array}$$

$$\begin{array}{l} \text{Average codeword length} \\ = 7/4 \text{ bits per event} \end{array}$$

$$\begin{array}{l} \text{Maximal codeword length} \\ = 2 \text{ bits} \end{array}$$

$$\begin{array}{l} \text{Maximal codeword length} \\ = 3 \text{ bits} \end{array}$$

- Code 1 is more compact in the sense of *shorter maximum codeword length*.
- Code 2 is more compact in the sense of *smaller average codeword length*.

# Measure of Compactness for a Code

Event probabilities:  $\{1/2, 1/4, 1/8, 1/8\}$

$$\text{code 1} \left\{ \begin{array}{l} \text{event one} : 00 \\ \text{event two} : 01 \\ \text{event three} : 10 \\ \text{event four} : 11 \end{array} \right. \quad \text{code 2} \left\{ \begin{array}{l} \text{event one} : 0 \\ \text{event two} : 10 \\ \text{event three} : 110 \\ \text{event four} : 111 \end{array} \right.$$

**E.g. Minimization of average function values of codeword length.**

- For a fixed  $t > 0$ , to minimize

$$\sum_{z \in \text{event space}} \Pr(z)2^{t \cdot \ell(z)}, \quad \left( \text{or equivalently, } L(t) \triangleq \frac{1}{t} \log_2 \sum_{z \in \text{event space}} \Pr(z)2^{t \cdot \ell(z)} \right)$$

where  $\ell(z)$  represents the codeword length for event  $z$ .

- The average function values of codeword length equals:

$$\sum_{z \in \text{event space}} \Pr(z)2^{t \cdot \ell(z)} = \frac{1}{2}2^{2t} + \frac{1}{4}2^{2t} + \frac{1}{8}2^{2t} + \frac{1}{8}2^{2t} = 2^{2t} \quad \text{for code 1;}$$

$$\sum_{z \in \text{event space}} \Pr(z)2^{t \cdot \ell(z)} = \frac{1}{2}2^t + \frac{1}{4}2^{2t} + \frac{1}{8}2^{3t} + \frac{1}{8}2^{3t} = \frac{2^t}{4}(2^{2t} + 2^t + 2) \quad \text{for code 2.}$$

## Measure of Compactness for a Code

I: 1-14

- $L(t) = \frac{1}{t} \log_2 \sum_{z \in \text{event space}} \Pr(z) 2^{t \cdot \ell(z)} = 2$  for code 1;

$$L(t) = \frac{1}{t} \log_2 \sum_{z \in \text{event space}} \Pr(z) 2^{t \cdot \ell(z)} = 1 + \frac{1}{t} \log_2 \frac{(2^{2t} + 2^t + 2)}{4} \quad \text{for code 2.}$$

– **Observation 1:** Code 1 is more compact when  $t > 1$ , and code 2 is more compact when  $0 < t < 1$ .

– **Observation 2:**

$$\begin{aligned} \lim_{t \downarrow 0} \frac{1}{t} \log_2 \sum_{z \in \text{event space}} \Pr(z) 2^{t \cdot \ell(z)} &= \sum_{z \in \text{event space}} \Pr(z) \ell(z) \\ &= \text{Average codeword length.} \end{aligned}$$

$$\begin{aligned} \lim_{t \uparrow \infty} \frac{1}{t} \log_2 \sum_{z \in \text{event space}} \Pr(z) 2^{t \cdot \ell(z)} &= \max_{z \in \text{event space}} \ell(z) \\ &= \text{Maximum codeword length.} \end{aligned}$$

## Lessen from the previous extension

I: 1-15

- Extension definition of measure of information content
  - (Engineering view) The minimum cost, i.e.,  $L(t)$ , of the most compact code representing the information
  - (Probabilistic view) Rényi Entropy of the information

$$H\left(Z; \frac{1}{1+t}\right) \triangleq \frac{1+t}{t} \log_2 \sum_{z \in \text{event space}} [\text{Pr}(z)]^{1/(1+t)}.$$

- In 1965, Cambell proved that the above two views are actually equivalent.  
[CAM65] L. L. Cambell, “A coding theorem and Rényi’s entropy,” *Informat. Contr.*, vol. 8, pp. 423–429, 1965.

$$\lim_{t \downarrow 0} H\left(Z; \frac{1}{1+t}\right) = \sum_{z \in \text{event space}} \text{Pr}(z) \log_2 \frac{1}{\text{Pr}(z)}$$

$$\lim_{t \uparrow \infty} H\left(Z; \frac{1}{1+t}\right) = \log_2(\text{number of events})$$

Hence, code 1 is optimal in the sense of “maximal-codeword-length” measure. Is it an interesting extension of code compactness?

# Data Transmission over Noisy Channel

I: 1-16

- In the case of data transmission over *noisy* channel, the concern is different from that for data storage (or error-free transmission).
  - The sender wishes to transmit to the receiver a sequence of pre-defined information symbols under an acceptable information-symbol error rate.
  - Code redundancies are therefore added to combat the *noise*.

For example, one may employ the three-times repetition code:

\* 1 → 111

\* 0 → 000

and apply the majority law at the receiver end so that one-bit error can be recovered.

- The three-times repetition code transmits **one information bit** per **three channel bits**. Hence, the information transmission efficiency (or channel code rate) is termed  $1/3$  (*zero-one*) *information symbol per channel usage*.

## Concern on Channel Code Design

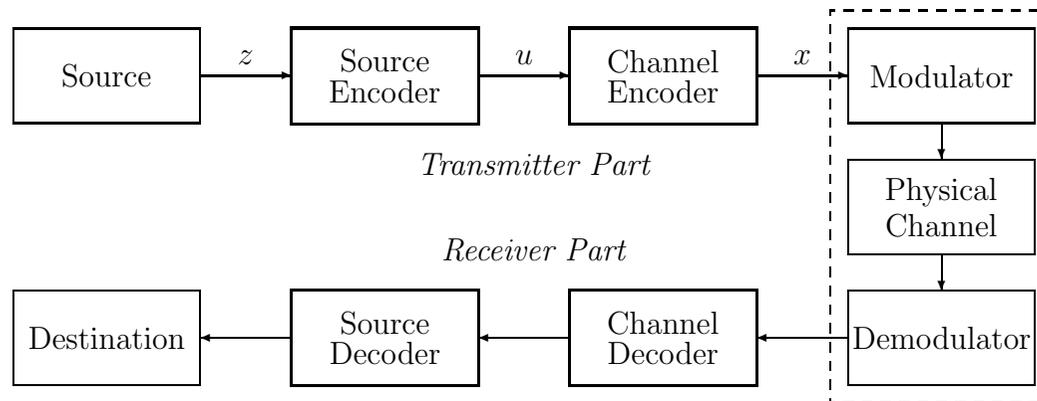
I: 1-17

- Fix a noisy channel. What is the maximum transmission efficiency attainable for channel code designs, subject to an *arbitrarily small* error probability for information symbols?
- Before we explore the query, it is better to clarify the relation between source coder and channel coder. This will help deciphering the condition of *arbitrarily small information-transmission error probability*.

# Information Transmission

I: 1-18

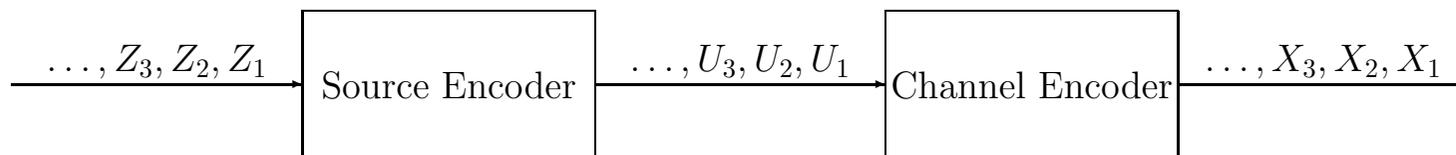
- Source coder maps information symbols (representing events) to source codewords (e.g.,  $u = f(z)$ ).
- Channel coder maps source codewords to channel codewords (e.g.,  $x = g(u)$ ).
- These two coders can be jointly treated as one mapping directly from information symbols to channel codewords (e.g.,  $x = g(f(z)) = h(z)$ ).
- It is nature to foresee that a joint-design of source-channel code (i.e., to find the best  $h(\cdot)$  mapping) is advantageous, but hard.



# Separate Design of Source and Channel Coders

I: 1-19

- Source encoder
  - Find the most compact representation of the informative message.
- Channel encoder
  - According to the noise pattern, add the redundancy so that the source code bits can be *reliably* transmitted.



# Source Encoder Design

I: 1-20



- For source encoder, the system designer wishes to minimize the number of  $U$ 's required to represent one  $Z$ 's, i.e,

Compression rate = number of  $U$ 's per number of  $Z$ 's.

- Shannon tells us that (for i.i.d.  $Z$ 's)

Minimum compression rate = entropy of  $Z$  (or entropy rate of  $Z_1, Z_2, Z_3, \dots$ )

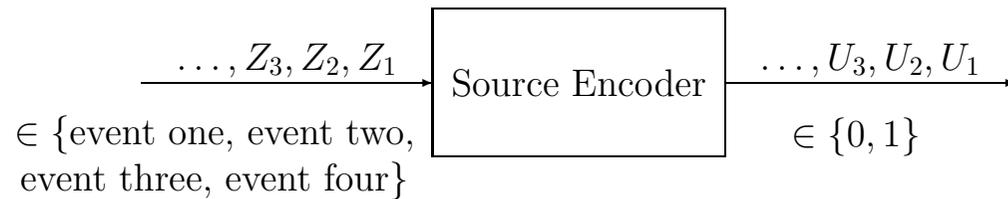
$$= \sum_{z \in \mathcal{Z}} P_Z(z) \log_{|U|} \frac{1}{P_Z(z)} \text{ code symbol per source symbol}$$

\* *entropy rate = entropy per  $Z$  symbol.*

\* *For i.i.d. process, entropy of  $Z$  = entropy rate of  $Z_1, Z_2, Z_3, \dots$*

# Source Encoder Design

I: 1-21



- $\mathcal{Z} = \{\text{event one, event two, event three, event four}\}$ .
- $\mathcal{U} = \{0, 1\}$ ; hence,  $|\mathcal{U}| = 2$ .
- Shannon tells us that (for i.i.d.  $Z$ 's)

$$\begin{aligned} \text{Minimum compression rate} &= \text{entropy of } Z \\ &= \sum_{z \in \mathcal{Z}} P_Z(z) \log_2 \frac{1}{P_Z(z)} \text{ code bit per source symbol} \end{aligned}$$

**Claim:** If the source encoder is *optimal*, its output  $\dots, U_3, U_2, U_1$  is (asymptotically) uniformly distributed over  $\mathcal{U}$ .

## Source Encoder Design

I: 1-22

**E.g.**,  $\dots, Z_3, Z_2, Z_1 \in \{\text{event one, event two, event three, event four}\} = \{e_1, e_2, e_3, e_4\}$  with probabilities  $(1/2, 1/4, 1/8, 1/8)$ . We already know that

$$\text{code 2} \left\{ \begin{array}{l} \text{event one} : 0 \\ \text{event two} : 10 \\ \text{event three} : 110 \\ \text{event four} : 111 \end{array} \right.$$

has the minimum average codeword length equal to the entropy. (No further compression is possible; so code 2 completely compresses the event information.)

- Then

$$\Pr\{U_1 = 0\} = \Pr\{Z_1 = e_1\} = 1/2,$$

So the first source code bit is **uniformly** distributed.

- 

$$\begin{aligned} \Pr\{U_2 = 0\} &= \Pr(Z_1 = e_1 \wedge Z_2 = e_1) + \Pr(Z_1 = e_2) \\ &= \Pr(Z_1 = e_1) \Pr(Z_2 = e_1) + \Pr(Z_1 = e_2) = \frac{1}{2} \times \frac{1}{2} + \frac{1}{4} = \frac{1}{2}. \end{aligned}$$

So the second source code bit is **uniformly** distributed.

## Source Encoder Design

I: 1-23

- 

$$\begin{aligned}\Pr\{U_3 = 0\} &= \Pr\{Z_1 = e_1 \wedge Z_2 = e_1 \wedge Z_3 = e_1\} + \Pr\{Z_1 = e_1 \wedge Z_2 = e_2\} \\ &\quad + \Pr\{Z_1 = e_2 \wedge Z_2 = e_1\} + \Pr\{Z_1 = e_3\} \\ &= \frac{1}{8} + \frac{1}{8} + \frac{1}{8} + \frac{1}{8} = \frac{1}{2}.\end{aligned}$$

So the third source code bit is **uniformly** distributed.

- .....

Consequently,  $U_1, U_2, U_3, \dots$  is uniformly distributed over  $\{0, 1\}$ .  
(It can be shown that  $U_1, U_2, U_3, \dots$  is i.i.d.)

# Source Encoder Design

I: 1-24



**An alternative interpretation:** If  $U \in \{0, 1\}$  is not uniformly distributed, then its entropy

$$R = p \log_2 \frac{1}{p} + (1 - p) \log_2 \frac{1}{1 - p} < 1 \text{ number of } \bar{U}\text{'s/number of } U\text{'s,}$$

where  $\Pr\{U = 0\} = p$ , and  $\bar{U} \in \{0, 1\}$ .

Hence, from Shannon, there exists *another source encoder* such that the resultant bits required become

$$m = nR < n.$$

Further compression to code 2 is obtained, a contradiction!

# Source Encoder Design

I: 1-25



**Summary:** The output of an *optimal* source encoder in the sense of minimizing the average per-letter codeword length (i.e., the number of  $U$  divided by the number of  $Z$ ), which asymptotically achieves the per-letter source entropy (i.e., the overall entropy of  $Z_1, Z_2, \dots$  divided by the number of  $Z$ ), should be asymptotically i.i.d. with uniform marginal distribution.

In case the average per-letter codeword length of the optimal source code equals the per-letter source entropy, its output becomes exactly i.i.d. with equally probable marginal.

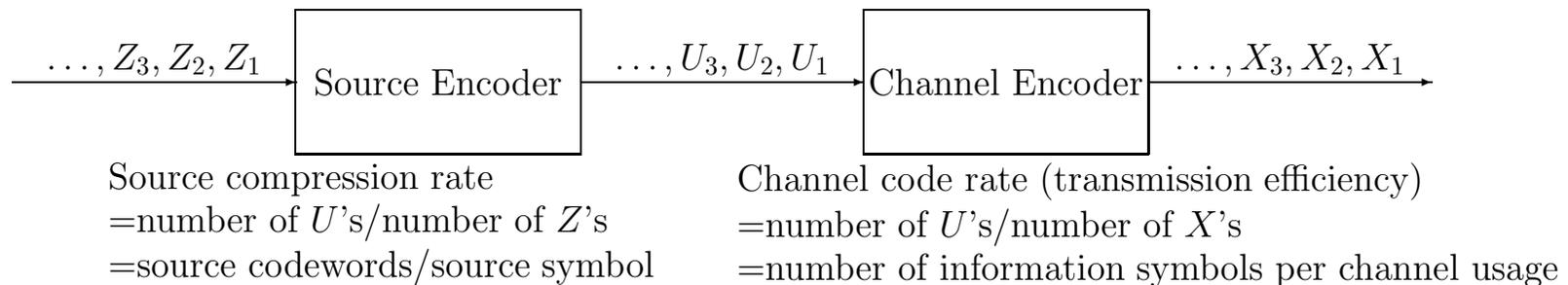
Note that an optimal code means that for  $U$  and  $\bar{U}$  in  $\mathcal{U}$ , the entropy of  $U$  equals

$$1 \text{ (number of } \bar{U}\text{'s/number of } U\text{'s)} = \sum_{u \in \mathcal{U}} P_U(u) \log_{|\mathcal{U}|} \frac{1}{P_U(u)},$$

which only holds when  $U$  is uniformly distributed over  $\mathcal{U}$ .

# Separate Design of Source and Channel Codes

I: 1-26



- The one who designs the channel code may assume that the one who designs the source code does a *good* (i.e., optimal) job in compression.
- So he assumes that the channel inputs are uniformly distributed; hence,  $\dots, U_3, U_2, U_1$  are completely *information* symbols without redundancy.
- What he concerns now becomes *the number of information symbols per channel usage*, subject to an acceptable transmission error.
- Since  $\{U_j\}_{j=1}^m$  is uniformly distributed, the error rate is computed by:

$$\text{error} = \frac{1}{|\mathcal{U}|^m} \sum_{(u_1, u_2, \dots, u_m) \in \mathcal{U}^m} \Pr\{\text{error} | (u_1, u_2, \dots, u_m) \text{ is transmitted}\},$$

which is often referred to as *average error criterion*.

# Reliable = Arbitrarily Small Error Probability

I: 1-27

- Now back to the question that:
  - Fix a noisy channel. What is the maximum transmission efficiency (i.e., channel code rate) attainable for channel code designs, subject to an arbitrarily small error probability for information symbols?
- What is *arbitrarily small error probability*?
  - *Manager*: Fix a noisy channel. Can we find a channel code that satisfies a criterion that the information transmission error  $< \mathbf{0.1}$ , and the channel code rate =  $1/3$  (number of  $U$ 's/number of  $X$ 's)?  
*Engineer*: Yes, I am capable to construct such a code.
  - *Manager*: For the same noisy channel, can we find a channel code that satisfies a criterion that the information transmission error  $< \mathbf{0.01}$ , and the channel code rate =  $1/3$  (number of  $U$ 's/number of  $X$ 's)?  
*Engineer*: Yes, I can achieve this by modifying the previous code.
  - *Manager*: How about information transmission error  $< \mathbf{0.001}$  with the same code rate?  
*Engineer*: No problem at all. In fact, for  $1/3$  code rate, I can find a code to fulfill *arbitrary* small error demand.

# Reliable = Arbitrarily Small Error Probability

I: 1-28

- *Shannon*: Then  $1/3$  code rate is a *reliable* transmission code rate for this noisy channel.
- Note that *arbitrary small* is not equivalent to *exact zero*. In other words, the existence of codes for the demand of arbitrarily small error does not necessarily indicate the existence of zero-error codes.
- Definition of Channel Capacity
  - *Channel capacity* is the maximum *reliable* transmission code rate for a noisy channel.
- Question
  - Can one determine the maximum reliable transmission code rate without exhausting all possible channel code designs?
  - Shannon said, “Yes.”

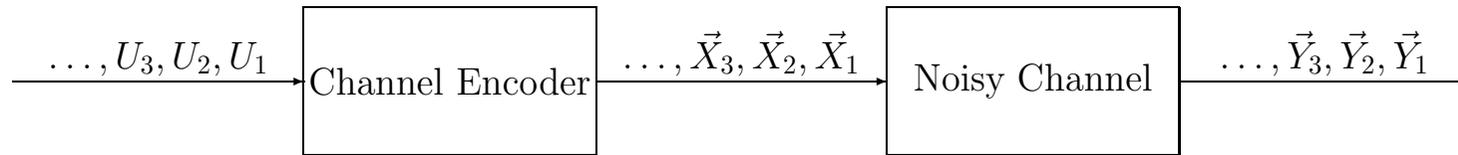
## Mutual information

I: 1-29

- Observe that a good channel code basically increases the *certainty* of channel outputs to channel inputs, although both the channel inputs and channel outputs are *uncertain* before the transmission begins (where channel inputs are decided by the information transmitted, and channel outputs are the joint results of the channel inputs and noise).
- So the design of a good channel code should consider more the statistically “shared information” between the channel inputs and outputs so that once a channel output is observed, the receiver is more *certain* about which channel input is transmitted.

## Example

I: 1-30



Channel code rate (transmission efficiency)  
= number of  $U$ 's / number of  $\vec{X}$ 's  
= number of information symbols per channel usage

### **Channel Model**

*Channel Input* –  $\vec{X} = (V_1, V_2)$  in  $\{(a, a), (a, b), (b, a), (b, b)\}$ .

*Channel Output* – Only  $V_1$  survives at the channel output due to channel noise. I.e., if  $\vec{Y} = (\Lambda_1, \Lambda_2)$  represents the channel output, then  $\Lambda_1 = V_1$  and  $\Lambda_2 = b$ .

### **Common Uncertainty Between Channel Input and Output**

*Input Uncertainty* – The channel input has two uncertainties,  $V_1$  and  $V_2$ , since each of them could be one of  $a$  and  $b$  (prior to the transmission begins).

*Output Uncertainty* – The channel output only possess one uncertainty,  $\Lambda_1$ , because  $\Lambda_2$  is deterministically known to be  $b$ .

*Shared Uncertainty* – So the common uncertainty between channel input and output (prior to the transmission begins) is  $\Lambda_1 = V_1$ .

## Example

I: 1-31

### Channel Code

- Suppose that Jack and Mary wish to use this noisy channel to reliably convey a 4-event information.
- Code design.

$$\begin{aligned}\text{event 1} & : \vec{X}_1, \vec{X}_2 = (a, d) \quad (a, d), \\ \text{event 2} & : \vec{X}_1, \vec{X}_2 = (a, d) \quad (b, d), \\ \text{event 3} & : \vec{X}_1, \vec{X}_2 = (b, d) \quad (a, d), \\ \text{event 4} & : \vec{X}_1, \vec{X}_2 = (b, d) \quad (b, d),\end{aligned}$$

where “ $d$ ” = “don’t-care”.

The resultant transmission rate is

$$\frac{\log_2(4 \text{ events})}{2 \text{ channel usages}} = 1 \text{ information bit per channel usage.}$$

It is noted that the above transmission code only uses uncertainty  $V_1$ . This is simply because uncertainty  $V_2$  is useless to the information exchange between Jack and Mary.

Also note that the events are uniformly distributed since data compressor is assumed to do an optimal job; so the entropy is  $4 \times \left( \frac{1}{4} \log_2 \frac{1}{(1/4)} \right) = 2$  bits.

# Channel Capacity

I: 1-32

- From the above example, one may conclude that the design of a good transmission code should relate to the *common uncertainty* (or more formally, the *mutual information*) between channel inputs and channel outputs.
- It is then natural to wonder whether or not this “relation” can be expressed mathematically.
- Indeed, it was established by Shannon that the bound on the reliable transmission rate (information bits per channel usage) is the maximum channel mutual information (i.e., common uncertainty prior to the transmission begins) attainable.
- With his ingenious work, once again, both engineering and probabilistic viewpoints coincide.

# Key Notes

I: 1-33

- Information measure
  - Equivalence between *engineering standpoint* based on code design and *mathematical standpoint* based on information statistics.
  - Interpretation of a good data compression code is then obtained.
- Channel capacity
  - Equivalence between:
    - \* *engineering standpoint* based on code design = maximum reliable code rate under uniformly distributed information input
    - \* *mathematical standpoint* based on channel statistics = maximum mutual information between channel input and output
  - Interpretation of a good channel code or error correcting code is then obtained.
- These equivalences form the basis of Information theory so that a computable statistically defined expression, such as entropy and mutual information, can be used to determine the optimality of a practical system.