Information Theory

Po-Ning Chen, Professor
Institute of Communications Engineering
National Chiao Tung University
Hsin Chu, Taiwan 30010, R.O.C.

Overview : The philosophy behind information theory

Po-Ning Chen, Professor

Institute of Communications Engineering

National Chiao Tung University

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

1.1: Overview

- *Information theory*: A mathematical framework for the theory of communication by establishing the *fundamental limits* on the performance of various communication systems.
- Claude Elwood Shannon (30 April 1916 – 24 February 2001)

The Bell System Technical Journal

Vol. XXVII *July, 1948* *No. 3*

A Mathematical Theory of Communication

By C. E. SHANNON

Introduction

THE recent development of various methods of modulation such as PCM and PPM which exchange bandwidth for signal-to-noise ratio has intensified the interest in a general theory of communication. A basis for such a theory is contained in the important papers of Nyquist[1] and Hartley[2] on this subject. In the present paper we will extend the theory to include a number of new factors, in particular the effect of noise in the channel, and the savings possible due to the statistical structure of the original message and due to the nature of the final destination of the information.

1.1: Overview

- It is possible to send information-bearing signals at a *fixed positive rate* R through a noisy communication channel with an arbitrarily small probability of error as long as the transmission rate R is below a certain fixed quantity C that depends on the channel statistical characteristics; he "baptized" this quantity with the name of *channel capacity*.

In the more general case with different lengths of symbols and constraints on the allowed sequences, we make the following definition:

Definition: The capacity C of a discrete channel is given by

$$C = \lim_{T \to \infty} \frac{\log N(T)}{T}$$

where $N(T)$ is the number of allowed signals of duration T.

1.1: Overview

- He further proclaimed that random sources can be compressed distortion-free at a minimal rate given by the source's intrinsic amount of information, which he called *source entropy*.

6. Choice, Uncertainty and Entropy

We have represented a discrete information source as a Markoff process. Can we define a quantity which will measure, in some sense, how much information is "produced" by such a process, or better, at what rate information is produced?

⋮

Theorem 2: The only H satisfying the three above assumptions is of the form:

$$H = -K \sum_{i=1}^{n} p_i \log p_i$$

where K is a positive constant.

1.1: Overview

- Shannon went on proving that

> **13. The Fundamental Theorem for a Discrete Channel with Noise**
>
> ⋮
>
> *Theorem 11.* Let a discrete channel have the capacity C and a discrete source the entropy per second H. If $H \leq C$ there exists a coding system such that the output of the source can be transmitted over the channel with an arbitrarily small frequency of errors (or an arbitrarily small equivocation). If $H > C$ it is possible to encode the source so that the equivocation is less than $H - C + \epsilon$ where ϵ is arbitrarily small. There is no method of encoding which gives an equivocation less than $H - C$.

1.1: Overview

- Information theorists gradually expanded their interests beyond communication theory, and investigated fundamental questions in several other related fields. Among them we cite:
  - statistical physics (thermodynamics, quantum information theory);
  - computing and information sciences (distributed processing, compression, algorithmic complexity, resolvability);
  - probability theory (large deviations, limit theorems, Markov decision processes);
  - statistics (hypothesis testing, multi-user detection, Fisher information, estimation);
  - stochastic control (control under communication constraints, stochastic optimization);
  - economics (game theory, team decision theory, gambling theory, investment theory);
  - mathematical biology (biological information theory, bioinformatics);
  - information hiding, data security and privacy;
  - data networks (network epidemics, self-similarity, traffic regulation theory);
  - machine learning (deep neural networks, data analytics).

Syllabus

Instructor information :

Po-Ning Chen
Engineering Building 4, Room 831
Phone : 03-5731670
email: poning@faculty.nctu.edu.tw

Textbook : US$1=NT$30

Fady Alajaji and Po-Ning Chen, *An Introduction to Single-User Information Theory*, Springer Singapore, July 6, 2018.

(NT$1900×0.9/333 pages≈NT$5.14/page)

Additionally, a set of copyrighted class notes for advanced topics will be provided. You can obtain the latest version of the lecture notes from

http://shannon.cm.nctu.edu.tw/it18.htm

References :

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.
   (US$29.32/206 pages ≈NT$4.27/page)
2. *Elements of Information Theory*, Thomas M. Cover and Joy A. Thomas, 2nd edition, John Wiley & Sons, Inc., July 2006.
   (US$93.86/776 pages,≈NT$3.63/page)
3. *Information-Spectrum Method in Information Theory*, Te Sun Han, Springer-Verlag Berlin Heidelberg, 2003.
   (US$109.00/538 pages ≈NT$6.08/page)
4. *A First Course in Information Theory (Information Technology: Transmission, Processing, and Storage)*, Raymond W. Yueng, Plenum Pub Corp., May 2002.
   (US$199.99/412 pages ≈NT$14.56/page)
5. *Principles and Practices of Information Theory*, Richard E. Blahut, Addison Wesley, 1988.
   (Used US$39.99/458 pages ≈NT$2.62/page)

6. *Information Theory: Coding Theorems for Discrete Memoryless Systems*, Imre Csiszár and Z. W. Birnbaun, Academic Press, 1981.
   (US$72.95/464 pages ≈NT$4.72/page)
7. *Information Theory and Reliable Communication*, Robert G. Gallager, Wiley, 1968.
   (US$179.55/608 pages ≈NT$8.86/page)

Grading System :

- Your semester grade will be contributed equally by the *midterm exam* and the *final exam*.

Syllabus

Lecture Schedule :

- The first lecture will be given on **February 22**.
- There will be no lecture on **March 1**, **April 5** and **June 7** because these are holidays.
  - Since we have lost three 3-hour lectures due to holidays, we shall shorten our second 20-minute break by 10 minutes in order to compensate for the lost of coverage.
- **Midterm** will be held on **April 26**. The coverage of midterm will be decided later.
- The last lecture will be given on **June 14, 2019**.
- **Final exam** will be held on **June 21, 2019.**

Chapter 1

Introduction

Po-Ning Chen, Professor

Institute of Communications Engineering

National Chiao Tung University

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

Introduction (Not in text)

- 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; this raises the question that:

> *how to represent information for ease of storing it*
> *or for ease of conveying it?*

Representation of information

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

  An answer from an engineer:

  - Reality:
    - 26 English letters and their concatenations $\Longrightarrow$ Language
  - Computer and Digital Communications:
    - 0-1 symbols and their concatenations $\Longrightarrow$ Code.

  After the information is symbolized, the "storing" or "conveying" operation of these symbols becomes straightforward.

Dictionary and codebook

- Assumption made by both transmitter and receiver about symbolized information
  - All "possible symbols" of the conveyed information are *a priori* known.
  - The receiver is only *uncertain* about which symbol 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 *a priori* known (to the receiver).
  - Only uncertain about which is going to be received.

Compactness of codes

- 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 $\{1/2, 1/4, 1/8, 1/8\}$,

$$\text{code 1} \begin{cases} \text{event one} : 00 \\ \text{event two} : 01 \\ \text{event three} : 10 \\ \text{event four} : 11 \end{cases} \qquad \text{code 2} \begin{cases} \text{event one} : 0 \\ \text{event two} : 10 \\ \text{event three} : 110 \\ \text{event four} : 111 \end{cases}$$

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

How to find the most compact code?

- 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?

- 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?

- 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 (but disjoint) 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:

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

  (This claim will be proven in Theorem 2.1.)
- It is thus legitimate to adopt the *entropy*—the expected value of the self-information—as a (averaged) measure of information.

Example of computation of entropy

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

$$\text{code 1} \begin{cases} \text{event one} : 00 \\ \text{event two} : 01 \\ \text{event three} : 10 \\ \text{event four} : 11 \end{cases} \qquad \text{code 2} \begin{cases} \text{event one} : 0 \\ \text{event two} : 10 \\ \text{event three} : 110 \\ \text{event four} : 111 \end{cases}$$

$$\begin{array}{ll} & \text{Average codeword length} \\ = & \text{2 bits per event} \end{array} \qquad \begin{array}{ll} & \text{Average codeword length} \\ = & \underline{\text{7/4 bits per event}} \ \textbf{(more compact)} \end{array}$$

$$\begin{cases} \text{self-informaiton of event one} = \log_2 \dfrac{1}{1/2} = 1 \text{ bit} \\ \text{self-informaiton of event two} = \log_2 \dfrac{1}{1/4} = 2 \text{ bits} \\ \text{self-informaiton of event three} = \log_2 \dfrac{1}{1/8} = 3 \text{ bits} \\ \text{self-informaiton of event four} = \log_2 \dfrac{1}{1/8} = 3 \text{ bits} \end{cases}$$

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

Lessen from the previous example

- The previous example hints 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 below 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

- 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

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

$$\text{code 1} \begin{cases} \text{event one} : 00 \\ \text{event two} : 01 \\ \text{event three} : 10 \\ \text{event four} : 11 \end{cases} \qquad \text{code 2} \begin{cases} \text{event one} : 0 \\ \text{event two} : 10 \\ \text{event three} : 110 \\ \text{event four} : 111 \end{cases}$$

Average codeword length
= 2 bits per event

Average codeword length
= 7/4 bits per event

Maximal codeword length
= 2 bits

Maximal codeword length
= 3 bits

- 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: $\{1/2, 1/4, 1/8, 1/8\}$

$$\text{code 1} \begin{cases} \text{event one} : 00 \\ \text{event two} : 01 \\ \text{event three} : 10 \\ \text{event four} : 11 \end{cases} \qquad \text{code 2} \begin{cases} \text{event one} : 0 \\ \text{event two} : 10 \\ \text{event three} : 110 \\ \text{event four} : 111 \end{cases}$$

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

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

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

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

- The average function values of codeword length equals:

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

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

Measure of compactness for a code

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

  $L(t) = \dfrac{1}{t}\log_2 \displaystyle\sum_{z\,\in\,\text{event space}} \Pr(z)2^{t\cdot\ell(z)} = 1 + \dfrac{1}{t}\log_2 \dfrac{(2^{2t}+2^t+2)}{4}$ 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**:

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

Lessen from the previous extension

- 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

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

- In 1965, Cambell proved that the above two views are equivalent.

  [CAM65] L. L. Cambell, "A coding theorem and Rényi's entropy," *Informat. Contr.*, vol. 8, pp. 423–429, 1965.

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

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

Data transmission over noisy channel

- 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:

    * $1 \rightarrow 111$
    * $0 \rightarrow 000$

    and apply the majority law at the receiver 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*.

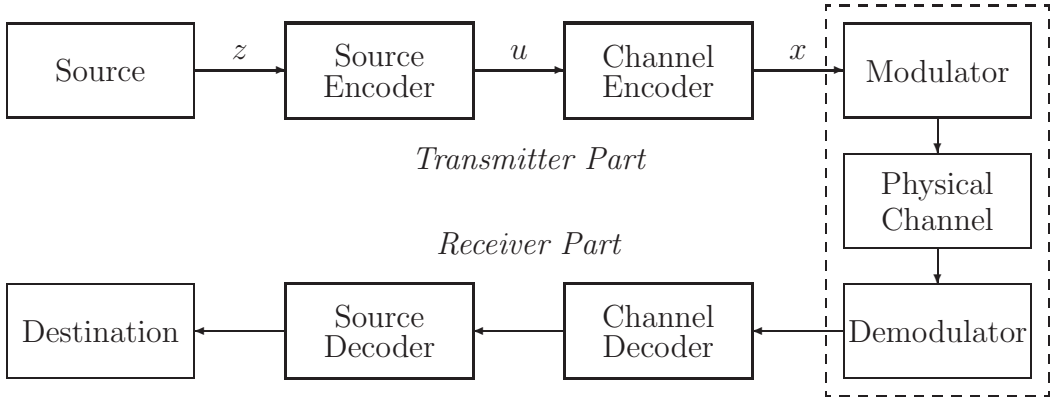
Concern on channel code design

- 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*.

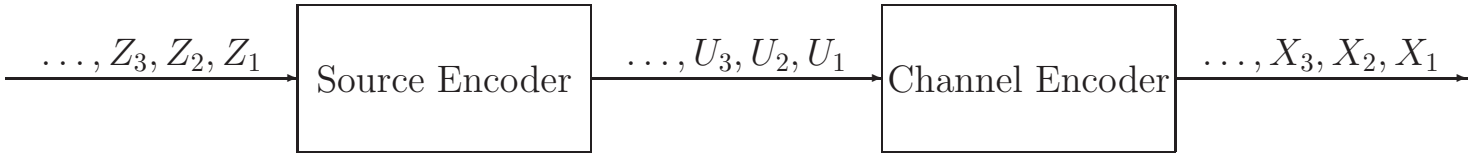
Information transmission

- 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.

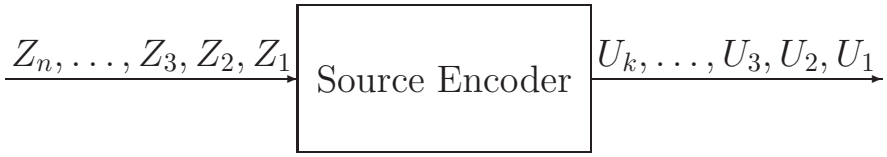


Separate design of source and channel coders

- 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



- 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.

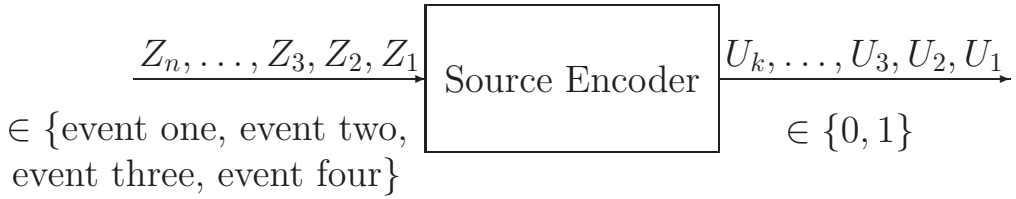
- Shannon tells us that (for i.i.d. Z's)

$$\begin{aligned}\text{Minimum compression rate} &= \text{entropy of } Z \text{ (or entropy rate of } Z_1, Z_2, Z_3, \ldots) \\ &= \sum_{z \in \mathcal{Z}} P_Z(z) \log_{|\mathcal{U}|} \frac{1}{P_Z(z)} \text{ code symbol per source symbol}\end{aligned}$$

**entropy rate = entropy per Z symbol.*

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

Source encoder design



- $\mathcal{Z} = \{\text{event one, event two, event three, event four}\}$.
- $\mathcal{U} = \{0, 1\}$; 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 $\ldots, U_3, U_2, U_1$ is (asymptotically) uniformly distributed over $\mathcal{U}$.

Source encoder design

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

$$\text{code 2} \begin{cases} \text{event one} & : \ 0 \\ \text{event two} & : \ 10 \\ \text{event three} & : \ 110 \\ \text{event four} & : \ 111 \end{cases}$$

has the minimum average codeword length equal to the source 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 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 code bit is **uniformly** distributed.

Source encoder design

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

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

- . . . . . .

Consequently, each of $U_1, U_2, U_3, \ldots$ is uniformly distributed over $\{0, 1\}$.

(It can be shown that $U_1, U_2, U_3, \ldots$ is i.i.d. via $\Pr(U_k|U_1, \ldots, U_{k-1}) = \Pr(U_k)$.)

Source encoder design

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

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

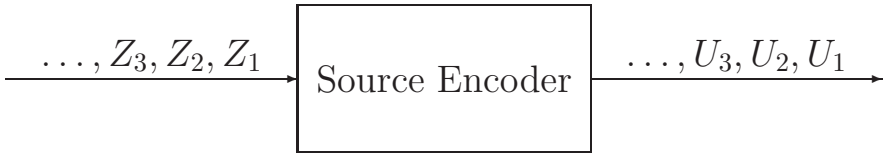
where $\Pr\{U = 0\} = p$, and $\bar{U} \in \{0, 1\}$.

Hence, from Shannon, there exists *another source encoder* such that

$$m = kH(U) < k.$$

Further compression to code 2 is obtained, a contradiction!

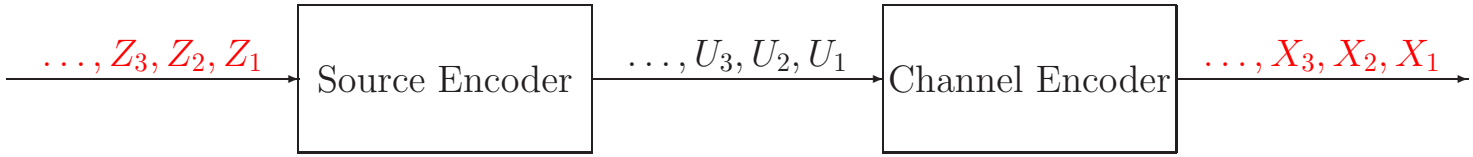
Source encoder design



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 $Z_1, Z_2, \ldots$ 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.

Separate design of source and channel codes



Source compression rate
=number of U's/number of Z's
=source codewords/source symbol

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

- 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 data compression.
- So he assumes that the channel inputs are uniformly distributed; hence, $\ldots, U_3, U_2, U_1$ are completely *information* symbols without redundancy.
- What a channel encoder 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:

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

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

Reliable = Arbitrarily small error probability

- Now back to the question:
  - 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 the new criterion 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

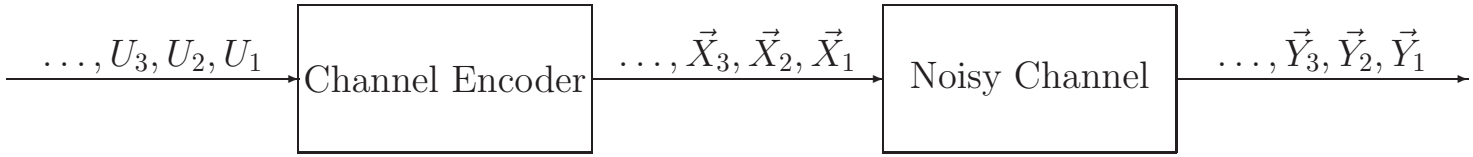
- *Shannon*: 1/3 code rate is a *reliable* transmission code rate for this noisy channel.
- Note that *arbitrary small* is not equivalent to *exact zero*. In other words, the existence of codes for the demand of arbitrarily small error does not necessarily indicate the existence of zero-error codes.
- 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

- 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



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, Λ_1, because Λ_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

Channel Code

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

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

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 from the information transmission perspective.

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

Channel capacity

- 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

- 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.