Information Theory

Lecture notes 2015-2016

Introduction

Organization

Professors:

- Lectures: Nicolae Cleju
- Laboratories: Daniel Matasaru

Grades

Final grade $= 0.75$ Exam $+ 0.25$ Lab

Time schedule

- 14 weeks of lectures (3h each)
- 14 weeks of laboratories (2h each)
- Office hours: by appointment

Course structure

1. Chapter I: Discrete Information Sources
2. Chapter II: Discrete Transmission Channels
3. Chapter III: Source Coding
4. Chapter IV: Channel Coding

Bibliography

1. ***Elements of Information Theory*, Valeriu Munteanu, Daniela Tarniceriu, Ed. CERMI 2007**
2. *Elements of Information Theory*, Thomas M. Cover, Joy A. Thomas, 2nd Edition, Wiley 2006
3. *Information and Coding Theory*, Gareth A. Jones, J. Mary Jones, Springer 2000
4. *Transmisia si codarea informatiei*, lectures at ETTI (Romanian)

Introduction to probabilities

Basic notions of probability

- Random variable = the outcome of an experiment
- Distribution (probability mass function)
- Discrete distribution
- Alphabet
- Logarithm function
- Exponential function
- Average of some values

Basic properties

- Two independent events:

$$p(A \cap B) = p(A) \cdot p(B)$$

Chapter I: Discrete information sources

Block diagram of a communication system

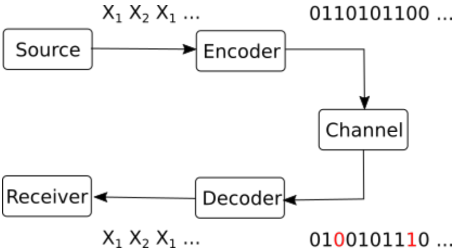

Figure 1: Block diagram of a communication system

What is information?

Example:

- I tell you the following sentence: “your favorite football team lost the last match”.
- Does this message carry information? How, why, how much?
- Consider the following facts:
  - the message carries information only because you didn’t already know the result.
  - if you already known the result, the message is useless (brings no information)
  - since you didn’t know the result, there were multiple results possible (win, equal or lose)
  - the actual information in the message is that *lost* happened, and not *win* or *equal*
  - if the result was to be expected, there is little information. If the result is highly unusual, there is more information in this message

Information source

- We will always consider information in a context similar to the above example.
- We will use terminology from probability theory to define information:
  - there is a *probabilistic source* that can produce a number of different *events*.
  - each event has a certain probability. We know all the probabilities beforehand.
  - at one time, an event is randomly selected according to its probability.
  - afterwards, a new message can be selected, and so on ==> a stream of messages is produced.
- The source is called an *information source* and the selected event is a *message*.
- A message carries the information that **it** happened, and not the other possible message events that could have been selected.
- The quantity of information is dependent in its probability.

Discrete memoryless source

- A discrete memoryless source (DMS) is an information source where the messages are **independent** , i.e. the choice of a message at one time does not depend on what were the previous messages
- Each message has a fixed probability. The set of probabilities is the *distribution* of the source.

$$S : \begin{pmatrix} s_1 & s_2 & s_3 \\ \frac{1}{2} & \frac{1}{4} & \frac{1}{4} \end{pmatrix}$$

- Properties:
  - Discrete: it can take a value from a discrete set (alphabet)
  - Complete: $\sum p(s_i) = 1$
  - Memoryless: succesive values are independent of previous values (e.g. successive throws of a coin)
- A message from a DMS is also called a *random variable* in probabilistics.

Examples

- A coin is a discrete memoryless source (DMS) with two messages:

$$S : \begin{pmatrix} heads & tails \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix}$$

- A dice is a discrete memoryless source (DMS) with six messages:

$$S : \begin{pmatrix} s_1 & s_2 & s_3 & s_4 & s_5 & s_6 \\ \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} \end{pmatrix}$$

- Playing the lottery can be modeled as DMS:

$$S : \begin{pmatrix} s_1 & s_2 \\ 0.9999 & 0.0001 \end{pmatrix}$$

Examples

- An extreme type of DMS containing the certain event:

$$S : \begin{pmatrix} s_1 & s_2 \\ 1 & 0 \end{pmatrix}$$

- Receiving an unknown *bit* (0 or 1) with equal probabilities:

$$S : \begin{pmatrix} 0 & 1 \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix}$$

Information

- When a DMS provides a new message, it gives out some new information, i.e. the information that a particular message took place.
- The information attached to a particular event (message) is rigorously defined as:

$$i(s_i) = -\log_2(p(s_i))$$

- Properties:
  - $i(s_i) \geq 0$
  - lower probability (rare events) means higher information
  - higher probability (frequent events) means lower information
  - a certain event brings no information: $-\log(1) = 0$
  - an event with probability 0 brings infinite information (but it never happens..)

Entropy of a DMS

- We usually don't care about a single message. We are interested in a large number of them (think millions of bits of data).
- We are interested in the *average* information of a message from a DMS.
- Definition: the entropy of a DMS source S is **the average information of a message**:

$$H(S) = \sum_k p_k i(s_k) = -\sum_k p_k \log_2(p_k)$$

where $p_k = p(s_k)$ is the probability of message k.

The choice of logarithm

- Any base of logarithm can be used in the definition.
- Usual convention: use binary logarithm $\log_2()$. H(S) measured in *bits* (*bits / message*)
- If using natural logarithm *ln*(), H(S) is measured in *nats*.
- Logarithm bases can be converted to/from one another:

$$\log_b(x) = \frac{\log_a(x)}{\log_a(b)}$$

- Entropies using different logarithms differ only in scaling:

$$H_b(S) = \frac{H_a(S)}{\log_a(b)}$$

Examples

- Coin: $H(S) = 1 bit/message$
- Dice: $H(S) = \log(6) bits/message$
- Lottery: $H(S) = -0.9999 \log(0.9999) - 0.0001 \log(0.0001)$
- Receiving 1 bit: $H(S) = 1 bit/message$ (hence the name!)

Interpretation of the entropy

All the following interpretations of entropy are true:

- H(S) is the *average uncertainty* of the source S
- H(S) is the *average information* of messages from source S
- A long sequence of N messages from S has total information $\approx N \cdot H(S)$
- H(S) is the minimum number of bits (0,1) required to uniquely represent an average message from source S

Properties of entropy

We prove the following **properties of entropy**:

1. $H(S) \geq 0$ (non-negative)
2. $H(S)$ is maximum when all n messages have equal probability $\frac{1}{n}$. The maximum value is $\max H(S) = \log(n)$.
3. *Diversfication* of the source always increases the entropy

The entropy of a binary source

- Consider a general DMS with two messages:

$$S : \begin{pmatrix} s_1 & s_2 \\ p & 1-p \end{pmatrix}$$

- It's entropy is:

$$H(S) = -p \cdot \log(p) - (1-p) \cdot \log(1-p)$$

- Graphical plot...

Example - Game

Game: I think of a number between 1 and 8. You have to guess it by asking yes/no questions.

- How much uncertainty does the problem have?
- How is the best way to ask questions? Why?
- What if the questions are not asked in the best way?
- On average, what is the number of questions required to find the number?

Example - Game v2

- Suppose I choose a number according to the following distribution:

$$S : \begin{pmatrix} s_1 & s_2 & s_3 & s_4 \\ \frac{1}{2} & \frac{1}{4} & \frac{1}{8} & \frac{1}{8} \end{pmatrix}$$

  - On average, what is the number of questions required to find the number?
  - What questions would you ask?

- What if the distribution is:

$$S : \begin{pmatrix} s_1 & s_2 & s_3 & s_4 \\ 0.14 & 0.29 & 0.4 & 0.17 \end{pmatrix}$$

- In general:
  - What distribution makes guessing the number the most difficult?
  - What distribution makes guessing the number the easiest?

Information flow of a DMS

- Suppose that message s_i takes time t_i to be transmitted via some channel.
- Definition: the information flow of a DMS S is **the average information transmitted per unit of time**:

$$H_\tau(S) = \frac{H(S)}{\overline{t}}$$

where $\overline{t}$ is the average duration of transmitting a message:

$$\overline{t} = \sum_i p_i t_i$$

Extended DMS

- Definition: the n-th order extension of a DMS S, S^n is the source with messages has as messages all the combinations of n messages of S:

$$\sigma_i = \underbrace{s_j s_k \ldots s_l}_{n}$$

- If S has k messages, S^n has k^n messages
- Since S is DMS

$$p(\sigma_i) = p(s_j) \cdot p(s_k) \cdot \ldots \cdot p(s_l)$$

Extended DMS - Example

- Examples:

$$S : \begin{pmatrix} s_1 & s_2 \\ \frac{1}{4} & \frac{3}{4} \end{pmatrix}$$

$$S^2 : \begin{pmatrix} \sigma_1 = s_1 s_1 & \sigma_2 = s_1 s_2 & \sigma_3 = s_2 s_1 & \sigma_4 = s_2 s_2 \\ \frac{1}{16} & \frac{3}{16} & \frac{3}{16} & \frac{9}{16} \end{pmatrix}$$

$$S^3 : \begin{pmatrix} s_1 s_1 s_1 & s_1 s_1 s_2 & s_1 s_2 s_1 & s_1 s_2 s_2 & s_2 s_1 s_1 & s_2 s_1 s_2 & s_2 s_2 s_1 & s_2 s_2 s_2 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \end{pmatrix}$$

Extended DMS - Another example

- Long sequence of binary messages:

$$0100110011100101 00...$$

  - Can be grouped in bits, half-bytes, bytes, 16-bit words, 32-bit long words, and so on.

Property of DMS

- Theorem: The entropy of a n-th order extension is n times larger than the entropy of the original DMS

$$H(S^n) = nH(S)$$

- Interpretation: grouping messages from a long sequence in blocks of n does not change total information (e.g. groups of 8 bits = 1 byte)

An example [memoryless is not enough]

- The distribution (frequencies) of letters in English:

letter	probability	letter	probability
A	.082	N	.067
B	.015	O	.075
C	.028	P	.019
D	.043	Q	.001
E	.127	R	.060
F	.022	S	.063
G	.020	T	.091
H	.061	U	.028
I	.070	V	.010
J	.002	W	.023
K	.008	X	.001
L	.040	Y	.020
M	.024	Z	.001

- Text from a memoryless source with these probabilities:

OCRO HLI RGWR NMIELWIS EU LL NBNESEBYA TH EEI
ALHENHTTPA OOBTTVA NAH BRL

(taken from Elements of Information Theory, Cover, Thomas)

- What's wrong? **Memoryless**

Sources with memory

- **Definition**: A source has memory of order m if the probability of a message depends on the last m messages.
- The last m messages = the **state** of the source (S_i).
- A source with n messages and memory m => n^m states in all.
- For every state, messages can have a different set of probabilities. Notation: $p(s_i|S_k)$ = *"probability of s_i in state S_k"*.
- Also known as *Markov sources*.

Example

- A source with $n = 4$ messages and memory $m = 1$
  - if last message was s_1, choose next message with distribution
$$S_1 : \begin{pmatrix} s_1 & s_2 & s_3 & s_4 \\ 0.4 & 0.3 & 0.2 & 0.1 \end{pmatrix}$$
  - if last message was s_2, choose next message with distribution
$$S_2 : \begin{pmatrix} s_1 & s_2 & s_3 & s_4 \\ 0.33 & 0.37 & 0.15 & 0.15 \end{pmatrix}$$
  - if last message was s_3, choose next message with distribution
$$S_3 : \begin{pmatrix} s_1 & s_2 & s_3 & s_4 \\ 0.2 & 0.35 & 0.41 & 0.04 \end{pmatrix}$$
  - if last message was s_4, choose next message with distribution
$$S_4 : \begin{pmatrix} s_1 & s_2 & s_3 & s_4 \\ 0.1 & 0.2 & 0.3 & 0.4 \end{pmatrix}$$

Transitions

- When a new message is provided, the source **transitions** to a new state:

$$\underbrace{s_i s_j s_k}_{\text{old state}} \; s_l$$

$$s_i \; \underbrace{s_j s_k s_l}_{\text{new state}}$$

- The message probabilities = the probabilities of transitions from some state S_u to another state S_v

Transition matrix

- The transition probabilities are organized in a **transition matrix** $[T]$

$$[T] = \begin{bmatrix} p_{11} & p_{12} & \dots & p_{1N} \\ p_{21} & p_{22} & \dots & p_{2N} \\ \dots & \dots & \dots & \dots \\ p_{N1} & p_{N2} & \dots & p_{NN} \end{bmatrix}$$

- p_{ij} is the transition probability from state S_i to state S_j
- N is the total number of states

Graphical representation

Example here

Entropy of sources with memory

- Each state S_k has a different distribution –> each state has a different entropy $H(S_k)$

$$H(S_k) = -\sum_i p(s_i|S_k) \cdot \log(p(s_i|S_k))$$

- Global entropy = average entropy

$$H(S) = \sum_k p_k H(S_k)$$

where p_k = probability that the source is in state S_i (i.e. after a very long sequence of messages, how many times the source was in state S_k)

Ergodic sources

- Let $p_i^{(t)}$ = the probability that source S is in state S_i at time t.
- In what state will it be at time $t+1$? (after one more message) (probabilities)

$$[p_1^{(t)}, p_2^{(t)}, ...p_N^{(t)}] \cdot [T] = [p_1^{(t+1)}, p_2^{(t+1)}, ...p_N^{(t+1)}]$$

- After one more message:

$$[p_1^{(t)}, p_2^{(t)}, ...p_N^{(t)}] \cdot [T] \cdot [T] = [p_1^{(t+2)}, p_2^{(t+2)}, ...p_N^{(t+2)}]$$

- In general, after n messages the probabilities that the source is in a certain state are:

$$[p_1^{(0)}, p_2^{(0)}, ...p_N^{(0)}] \cdot [T]^n = [p_1^{(n)}, p_2^{(n)}, ...p_N^{(n)}]$$

Ergodicity

- A source is called **ergodic** if every state can be reached from every state, in a finite number of steps.

Property of ergodic sources:

- After many messages, the probabilities of the states *become stationary* (converge to some fixed values), irrespective of the initial probabilities.

$$\lim_{n\to\infty}[p_1^{(n)}, p_2^{(n)}, ...p_N^{(n)}] = [p_1, p_2, ...p_N]$$

Finding the stationary probabilties

- After n messages and after $n+1$ messages, the probabilties are the same:

$$[p_1, p_2, ...p_N] \cdot [T] = [p_1, p_2, ...p_N]$$

- Also $p_1 + p_2 + ... + p_N = 1$.

$=>$ solve system of equations, find values.

Entropy of ergodic sources with memory

- The entropy of an ergodic source with memory is

$$H(S) = \sum_k p_k H(S_k) = -\sum_k p_k \sum_i p(s_i|S_k) \cdot \log(p(s_i|S_k)$$

Example English text as sources with memory

(taken from Elements of Information Theory, Cover, Thomas)

- Memoryless source, equal probabilities:

  XFOML RXKHRJFFJUJ ZLPWCFWKCYJ
  FFJEYVKCQSGXYD QPAAMKBZAACIBZLHJQD

- Memoryless source, probabilities of each letter as in English:

  OCRO HLI RGWR NMIELWIS EU LL NBNESEBYA TH EEI
  ALHENHTTPA OOBTTVA NAH BRL

- Source with memory $m = 1$, frequency of pairs as in English:

  ON IE ANTSOUTINYS ARE T INCTORE ST BE S DEAMY
  ACHIN D ILONASIVE TUCOOWE AT TEASONARE FUSO
  TIZIN ANDY TOBE SEACE CTISBE

- Source with memory $m = 2$, frequency of triplets as in English:

  IN NO IST LAT WHEY CRATICT FROURE BERS GROCID
  PONDENOME OF DEMONSTURES OF THE REPTAGIN IS
  REGOACTIONA OF CRE

- Source with memory $m = 3$, frequency of 4-plets as in English:

  THE GENERATED JOB PROVIDUAL BETTER TRAND THE DISPLAYED
  CODE, ABOVERY UPONDULTS WELL THE CODERST IN THESTICAL
  IT DO HOCK BOTHE MERG. (INSTATES CONS ERATION. NEVER
  ANY OF PUBLE AND TO THEORY. EVENTIAL CALLEGAND TO ELAST
  BENERATED IN WITH PIES AS IS WITH THE)

Chapter summary

- Information of a message: $i(s_k) = -\log_2(p(s_k))$
- Entropy of a memoryless source: $H(S) = \sum_k p_k i(s_k) = -\sum_k p_k \log_2(p_k)$
- Properties of entropy:
  1. $H(S) \geq 0$
  2. Is maximum when all messages have equal probability ($H_{max}(S) = \log(n)$)
  3. *Diversfication* of the source always increases the entropy
- Sources with memory: definition, transitions
- Stationary probabilities of ergodic sources with memory: $[p_1, p_2, ...p_N] \cdot [T] = [p_1, p_2, ...p_N]$, $\sum_i p_i = 1$.
- Entropy of sources with memory:

$$H(S) = \sum_k p_k H(S_k) = -\sum_k p_k \sum_i p(s_i|S_k) \cdot \log(p(s_i|S_k)$$

Chapter II: Discrete Transmission Channels

What are they?

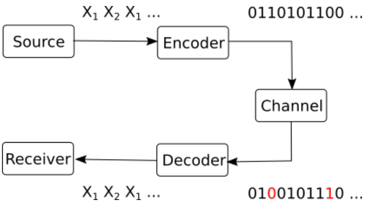

Figure 2: Communication system

- A system of two related random variables
- Input random variable $X = x_1, x_2, \ldots$, output random variable $Y = y_1, y_2, \ldots$
- X and Y are *related*, but still *random* (usually because of *noise*)

What do we want

- Successful communication: receive Y, deduce what was sent X
- We are interested in **deducing X when knowing Y**
- How much does knowing Y tell us about X?
  - Depends on the relation between them
  - Is the same as how much X tells us about Y (symmetrical)

Nomenclature

- Discrete: the input alhabet and the output alphabet are finite
- Memoryless: the output symbol depends only on the current input symbol
- Stationary: the noise arising on the channel is time invariant (i.e. its statistics do not vary in time)

Systems of two random variables

- Two random variables: $X = x_1, x_2, \dots,\ Y = y_1, y_2, \dots$.
- Example: throw a dice (X) and a coin (Y) simultaneously
- How to describe this system?

A single joint information source:

$$X \cap Y : \begin{pmatrix} x_1 \cap y_1 & x_1 \cap y_2 & \dots & x_i \cap y_j \\ p(x_1 \cap y_1) & p(x_1 \cap y_2) & \dots & p(x_i \cap y_j) \end{pmatrix}$$

Arrange in a nicer form (table):

	y_1	y_2	y_3
x_1	...	...	...
x_2	...	...	...
x_3	...	...	...

- Elements of the table: $p(x_i \cap y_j)$

Joint probability matrix

The table constitutes the **joint probability matrix**:

$$P(X, Y) = \begin{bmatrix} p(x_1 \cap y_1) & p(x_1 \cap y_2) & \cdots & p(x_1 \cap y_M) \\ p(x_2 \cap y_1) & p(x_2 \cap y_2) & \cdots & p(x_2 \cap y_M) \\ \vdots & \vdots & \cdots & \vdots \\ p(x_N \cap y_1) & p(x_N \cap y_2) & \cdots & p(x_N \cap y_M) \end{bmatrix}$$

$$\sum_i \sum_j p(x_i \cap y_j) = 1$$

- This matrix completely defines the two-variable system
- This matrix completely defines the communication process

Joint entropy

- The distribution $X \cap Y$ determines the **joint entropy**:

$$H(X,Y) = -\sum_i \sum_j p(x_i \cap y_j) \cdot \log(p(x_i \cap y_j))$$

- This is the global entropy of the system (knowing the input and the output)

Marginal distributions

- $p(x_i) = \sum_j p(x_i \cap y_j)$ = sum of row i from P(X,Y)
- $p(y_j) = \sum_i p(x_i \cap y_j)$ = sum of column j from P(X,Y)
- The distributions $p(x)$ and $p(y)$ are called **marginal distributions** ("summed along the margins")

Examples [marginal distributions not enough]

- Example 1:

$$P(X,Y) = \begin{bmatrix} 0.3 & 0 & \\ 0 & 0.7 & \end{bmatrix}$$

- Example 2:

$$P(X,Y) = \begin{bmatrix} 0.15 & 15 \\ 0.15 & 0.55 \end{bmatrix}$$

- Both have identical $p(x)$ and $p(y)$, but are completely different
- Which one is better for a transmission?
- Marginal distribution are useful, but not enough. Essential is the *relation* between X and Y.

Bayes formula

$$p(A \cap B) = p(A) \cdot p(B|A)$$

$$p(B|A) = \frac{p(A \cap B)}{p(A)}$$

- "The conditional probability of B **given A**" (i.e. given that event A happened)
- Examples. . .
- Independence:

$$p(A \cap B) = p(A)p(B)$$

$$p(B|A) = p(B)$$

Channel matrix

Noise (or channel) matrix:

$$P(Y|X) = \begin{bmatrix} p(y_1|x_1) & p(y_2|x_1) & \cdots & p(y_M|x_1) \\ p(y_1|x_2) & p(y_2|x_2) & \cdots & p(y_M|x_2) \\ \vdots & \vdots & \cdots & \vdots \\ p(y_1|x_N) & p(y_2|x_N) & \cdots & p(y_M|x_N) \end{bmatrix}$$

- Defines the probability of an output **given an input**
- Each row = a separate distribution that indicates the probability of the outputs **if the input is** x_i)
- The sum of each row is 1 (there must be some output if the input is x_i

Relation of channel matrix and joint probability matrix

- $P(Y|X)$ is obtained from $P(X, Y)$ by dividing every row to its sum ($p(x_i)$)
- This is known as *normalization* of rows
- $P(X, Y)$ can be obtained back from $P(Y|X)$ by multiplying each row with $p(x_i)$
- $P(Y|X)$ contains less information than $P(X, Y)$

Definition of a discrete transmission channel

Definition: A discrete transmission channel is defined by three items:

1. The input alphabet $X = \{x_1, x_2, \ldots\}$
2. The output alphabet $Y = \{y_1, y_2, \ldots\}$
3. The noise (channel) matrix $P(Y|X)$ which defines the conditional probabilities of the outputs y_j for every possible input x_i

Graphical representation of a channel

- Nice picture with arrows :)

Three examples

Three examples to help you remember conditional probabilities

- Play and win the lottery
- Gambler's paradox
- CNN: Crippled cruise ship returns; passengers happy to be back

Conditional entropy H(Y|X) (mean error)

- Since each row is a distribution, each row has an entropy
- Entropy of row x_i:

$$H(Y|x_i) = -\sum_j p(y_j|x_i) \log(p(y_j|x_i))$$

- *"The uncertainty of the output message when the input message is x_i"*
- Example: lottery

Conditional entropy H(Y|X) (mean error)

- A different $H(Y|x_i)$ for every x_i
- Compute the average over all x_i:

$$\begin{aligned} H(Y|X) &= \sum_i p(x_i) H(Y|x_i) \\ &= -\sum_i \sum_j p(x_i) p(y_j|x_i) \log(p(y_j|x_i)) \\ &= -\sum_i \sum_j p(x_i \cap y_j) \log(p(y_j|x_i)) \end{aligned}$$

- **"The uncertainty of the output message when we know the input message"** (any input, in general)

Equivocation matrix

Equivocation matrix:

$$P(X|Y) = \begin{bmatrix} p(x_1|y_1) & p(x_1|y_2) & \cdots & p(x_1|y_M) \\ p(x_2|y_1) & p(x_2|y_2) & \cdots & p(x_2|y_M) \\ \vdots & \vdots & \ldots & \vdots \\ p(x_N|y_1) & p(x_N|y_2) & \cdots & p(x_N|y_M) \end{bmatrix}$$

- Defines the probability of an input **given an output**
- Each column = a separate distribution that indicates the probability of the inputs **if the output is** y_j)
- The sum of each column is 1 (there must be some input if the output is y_j

Relation of equivocation matrix and joint probability matrix

- $P(X|Y)$ is obtained from $P(X,Y)$ by dividing every column to its sum ($p(y_j)$)
- This is known as *normalization* of columns
- $P(X,Y)$ can be obtained back from $P(X|Y)$ by multiplying each column with $p(y_j)$
- $P(X|Y)$ contains less information than $P(X,Y)$

Conditional entropy H(X|Y) (equivocation)

- Since each column is a distribution, each column has an entropy
- Entropy of column y_j:

$$H(X|y_j) = -\sum_i p(x_i|y_j)\log(p(x_i|y_j))$$

- *"The uncertainty of the input message when the output message is y_j"*
- Example: ...

Conditional entropy H(X|Y) (equivocation)

- A different $H(X|y_j)$ for every y_j
- Compute the average over all y_j:

$$\begin{aligned} H(X|Y) &= \sum_j p(y_j) H(X|y_j) \\ &= -\sum_i \sum_j p(y_j) p(x_i|y_j) \log(p(x_i|y_j)) \\ &= -\sum_i \sum_j p(x_i \cap y_j) \log(p(x_i|y_j)) \end{aligned}$$

- **"The uncertainty of the input message when we know the output message"** (any output, in general)
- Should be small for a good communication

Properties of conditional entropies

For a general system with two random variables X and Y:

- Conditioning always reduces entropy:

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

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

  (knowing something cannot harm)

- If the variables are independent:

$$H(X|Y) = H(X)$$

$$H(Y|X) = H(Y)$$

  (knowing the second variable does not help at all)

Mutual information I(X,Y)

- Mutual information I(X,Y) = the average information that one variable has about the other
- Mutual information I(X,Y) = the average information that is transmitted on the channel
- Consider a communication channel with X as input and Y as output:
  - We are the receiver and we want to find out the X
  - When we don't know the output: H(X)
  - When we know the output: H(X|Y)
- How much information was transmitted?
  - Reduction of uncertainty:

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

Mutual information I(X,Y)

$$
\begin{aligned}
I(X, Y) &= H(X) - H(X|Y) \\
&= -\sum_i p(x_i)\log(p(x_i)) + \sum_i \sum_j p(x_i \cap y_j)\log(p(x_i|y_j)) \\
&= -\sum_i \sum_j p(x_i \cap y_j)\log(p(x_i)) + \sum_i \sum_j p(x_i \cap y_j)\log(p(x_i|y_j)) \\
&= \sum_i \sum_j p(x_i \cap y_j)\log(\frac{p(x_i|y_j)}{p(x_i)}) \\
&= \sum_i \sum_j p(x_i \cap y_j)\log(\frac{p(x_i \cap y_j)}{p(x_i)p(y_j)})
\end{aligned}
$$

Properties of mutual information

Mutual information $I(X, Y)$ is:

- commutative: $I(X, Y) = I(Y, X)$
- non-negative: $I(X, Y) \geq 0$
- a special case of the Kullback–Leibler distance (relative entropy distance)

Definition: the Kullback–Leibler distance of two distributions is

$$D_{KL}(P, Q) = \sum_i P(i) \log(\frac{P(i)}{Q(i)})$$

- In our case, the distributions are:
  - $P = p(x_i \cap y_j)$ (distribution of our system)
  - $Q = p(x_i) \cdot p(y_j)$ (distribution of two independent variables)

$$I(X, Y) = D(p(x_i \cap y_j), p(x_i) \cdot p(y_j))$$

Relations between the informational measures

- Nice picture with two circles :)
- All six: $H(X)$, $H(Y)$, $H(X, Y)$, $H(X|Y)$, $H(Y|X)$, $I(X, Y)$
- All relations on the picture are valid relations:

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

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

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

...

- If know three, can find the other three
- Simplest to find first H(X), H(Y), H(X,Y) —> then find others

Types of communication channels

1. Channels with zero equivocation

$$H(X|Y) = 0$$

   - Each column of the noise (channel) matrix contains only one non-zero value
   - No doubts on the input messages when the output messages are known
   - All input information is transmitted

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

- Example: codewords...

Types of communication channels

2. Channels with zero mean error

$$H(Y|X) = 0$$

    - Each row of the noise (channel) matrix contains only one non-zero value
    - No doubts on the output messages when the input messages are known
    - *The converse is not necessary true!*
    - All input information is transmitted

- Example: AND gate

Types of communication channels

3. Channels uniform with respect to the input

$$H(Y|x_i) = same$$

- Each row of noise matrix contains the same values, possibly in different order
- $H(Y|x_i) = same = H(Y|X)$
- $H(Y|X)$ does not depend on the actual probabilities $p(x_i)$

Types of communication channels

4. Channels uniform with respect to the output

$$H(X|y_j) = same$$

- Each column of noise matrix contains the same values, possibly in different order
- If the input messages are equiprobable, the output messages are also equiprobable

Types of communication channels

5. Symmetric channels
   - Uniform with respect to the input and to the output
   - Example: binary symmetric channel

Channel capacity

- What is the maximum information we can transmit on a certain channel?
- **Definition:** the information capacity of a channel is the maximum value of the mutual information, where the maximization is done over the input probabilities $p(x_i)$

$$C = \max_{p(x_i)} \; I(X, Y)$$

- i.e. the maximum mutual information we can obtain if we are allowed to choose $p(x_i)$ as we want
- Useful alternative expression:

$$C = \max_{p(x_i)} \left(H(X) - H(X|Y) \right)$$

What channel capacity means

- Channel capacity is the maximum information we can transmit on a channel, on average, with one message
- One of the most important notions in information theory
- Its importance comes from Shannon's second theorem (noisy channel theorem)

Preview of the channel coding theorem

- Even though some information $I(X, Y)$ is transmitted on the channel, there still is the $H(X|Y)$ uncertainty on the input
- We want error-free transmission, with no uncertainty
- Solution: use error coding (see chapter IV)
- How coding works:
  - coder receives k symbols (bits, usually) that we want to transmit
  - coder appends additional m symbols computed via some coding algorithm
  - the total $k + m$ bits are transmitted over a noisy channel
  - the decoding algorithm tries to detect and correct errors, based on the additional m bits that were appended
- Coding rate:

$$R = \frac{k}{k+m}$$

  - stronger protection = bigger m = less efficient
  - weaker protection = smaller m = more efficient

Preview of the channel coding theorem

- A rate is called **achievable** for a channel if, for that rate, there exists a coding and decoding algorithm guaranteed to correct all possible errors on the channel

Shannon's noisy channel coding theorem (second theorem)

For a given channel, all rates below capacity $R < C$ are achievable. All rates above capacity, $R > C$, are not achievable.

Channel coding theorem explained

In layman terms:

- For all coding rates $R < C$, there is a way to recover the transmitted data perfectly (de/coding algorithm will detect and correct all errors)
- For all coding rates $R > C$, there is no way to recover the transmitted data perfectly

Example:

- Send binary digits (0,1) on a channel with capacity 0.7 bits/message
- There exists coding schemes with R $<$ 0.7 that allow perfect recovery
  - i.e. for every 7 bits of data coding adds 3 or more bits, on average $=>$ $R = \frac{7}{7+3}$
- With less than 3 bits for every 7 bits of data $=>$ impossible to recover all the data

Efficiency and redundancy

- Efficiency of a channel:

$$\eta_C = \frac{I(X,Y)}{C}$$

- Absolute redundancy of a channel:

$$R_C = C - I(X,Y)$$

- Relative redundancy of a channel:

$$\rho_C = \frac{R_C}{C} = 1 - \frac{I(X,Y)}{C} = 1 - \eta_C$$

Computing the capacity

- Tricks for easier computation of the capacity
- Channel is uniform with respect to the input:
  - $H(Y|X)$ does not depend on the actual probabilities $p(x_i)$
  - $C = \max_{p(x_i)} I(X,Y) = \max_{p(x_i)} (H(Y) - H(Y|X)) = \max_{p(x_i)} (H(Y)) - H(Y|X)$
  - Should maximize $H(Y)$
- If channel is also uniform with respect to the output:
  - same values on columns of P(Y|X)
  - $p(y_j) = \sum_i p(y_j|x_i)p(x_i)$
  - if $p(x_i) = \text{uniform} = \frac{1}{n}$, then $p(y_j) = \frac{1}{n}\sum_i p(y_j|x_i) = \text{uniform}$
  - therefore $p(y_j)$ are constant = uniform = H(Y) is maximized

Examples of channels and their capacity

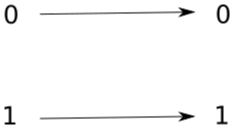

Figure 3: Noiseless binary channel

- Capacity $= 1$ bit/message, when $p(x_1) = p(x_2) = \frac{1}{2}$

Noisy binary non-overlapping channel

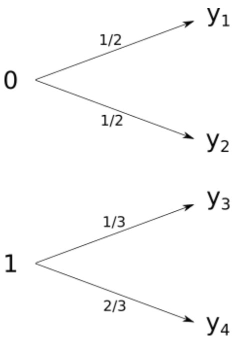

Figure 4: Noisy binary non-overlapping

- There is noise ($H(Y|X) > 0$), but can deduce the input ($H(X|Y) = 0$)
- Capacity $= 1$ bit/message, when $p(x_1) = p(x_2) = \frac{1}{2}$

Noisy typewriter

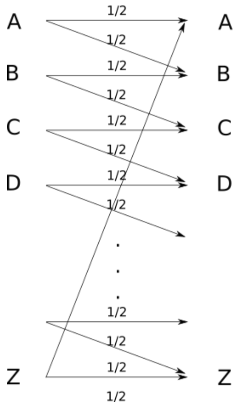

Figure 5: Noisy typewriter

$$\max I(X, Y) = \max \left(H(Y) - H(Y|X)\right) = \max H(Y) - 1$$
$$= \log(26) - 1 = \log(13)$$

Noisy typewriter

- Capacity $= \log(13)$ bit/message, when input probabilities are uniform
- Can transmit 13 letters with no errors (A, C, E, G, $\ldots$)

Binary symmetric channel

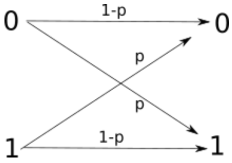

Figure 6: Binary symmetric channel (BSC)

- Capacity $= 1 - H_p = 1 + p\log(p) + (1-p)\log(1-p)$
- Capacity is reached when input distribution is uniform

Binary erasure channel

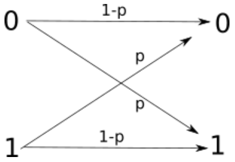

Figure 7: Binary erasure channel

- Different from BSC: here we know when errors happened
- Capacity $= 1 - p$
- Intuitive meaning: lose p bits, remaining bits $=$ capacity $= 1 - p$

Symmetric channel of n-th order

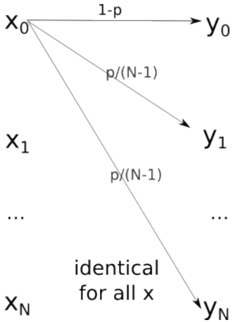

Figure 8: N-th order symmetric channel

- Extension of binary symmetric channel for n messages
- $1 - p$ chances that symbol has no error
- p chances that symbol is changed, uniformly to any other (N-1) symbols ($\frac{p}{N-1}$ each)

Symmetric channel of n-th order

- Channel is symmetric $=>$

$$C = \max_{p(x_i)} I(X,Y) = \max_{p(x_i)} (H(Y) - H(Y|X)) = \max_{p(x_i)} (H(Y)) - H(Y|X)$$

- $\max_{p(x_i)} (H(Y)) = \log(N)$
- $H(Y|X) = H(Y|x_i) =$ entropy of any row (same values)

$==>$

$$C = \log(N) + (1-p)\log(1-p) + p\log(\frac{p}{N-1})$$

- Capacity is reached when input probabilities are uniform

Chapter summary

- Channel = Probabilistic system with two random variables X and Y
- Characterization of transmission:
  - P(X,Y) => H(X,Y) *joint entropy*
  - $p(x_i)$, $p(y_j)$ *marginal distributions* => H(X), H(Y)
  - P(Y|X) *channel matrix* => H(Y|X) *average noise*
  - P(X|Y) => H(X|Y) *equivocation*
  - I(X,Y) *mutual information*
- Channel capacity: $C = \max_{p(x_i)} I(X, Y)$
- Examples:
  - Binary symmetric channel: $C = 1 - H_p$
  - Binary erasure channel: $C = 1 - p$
  - N-th symmetric channel: $C = \log(N) - H(of\ a\ row\ of\ channel\ matrix)$

History

Figure 9: Claude Shannon (1916 - 2001)

- *A mathematical theory of communications*, 1948

Exercises and problems

- At blackboard only

Chapter III: Source coding

What does coding do?

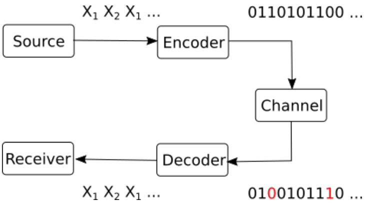

Figure 10: Communication system

- Why coding?

1. Source coding:
    - Convert source messages to channel symbols (for example 0,1)
    - Minimize number of symbols needed
    - Adapt probabilities of symbols to maximize mutual information
2. Error control
    - Protection against channel errors

Source-channel separation theorem

Source-channel separation theorem (informal):

- It is possible to obtain the best reliable communication by performing the two tasks separately:
    1. Source coding: to minimize number of symbols needed
    2. Error control coding (channel coding): to provide protection against noise

Source coding

- Assume we code for transmission over ideal channels with no noise
- Transmitted symbols are perfectly recovered at the receiver
- Main concerns:
  - minimize the number of symbols needed to represent the messages
  - make sure we can decode the messages
- Advantages:
  - Efficiency
  - Short communication times
  - Can decode easily

Definitions

- Let $S = s_1, s_2, ...s_N$ = an input discrete memoryless source
- Let $X = x_1, x_2, ...x_M$ = the alphabet of the code
  - Example: binary: $\{0,1\}$
- A **code** is a mapping from S to the set of all codewords:

$$C = c_1, c_2, ...c_N$$

Message	Codeword
s_1	$c_1 = x_1x_2x_1...$
s_2	$c_2 = x_1x_2x_2...$
...	....
s_N	$c_3 = x_2x_2x_2...$

- Decoding: given a sequence of symbols, deduce the original sequence of messages
- Codeword length l_i = the number of symbols in c_i

Example: ASCII code

Letter	ASCII Code	Binary	Letter	ASCII Code	Binary
a	097	01100001	A	065	01000001
b	098	01100010	B	066	01000010
c	099	01100011	C	067	01000011
d	100	01100100	D	068	01000100
e	101	01100101	E	069	01000101
f	102	01100110	F	070	01000110
g	103	01100111	G	071	01000111
h	104	01101000	H	072	01001000
i	105	01101001	I	073	01001001
j	106	01101010	J	074	01001010
k	107	01101011	K	075	01001011
l	108	01101100	L	076	01001100
m	109	01101101	M	077	01001101
n	110	01101110	N	078	01001110
o	111	01101111	O	079	01001111
p	112	01110000	P	080	01010000
q	113	01110001	Q	081	01010001
r	114	01110010	R	082	01010010
s	115	01110011	S	083	01010011
t	116	01110100	T	084	01010100
u	117	01110101	U	085	01010101
v	118	01110110	V	086	01010110
w	119	01110111	W	087	01010111
x	120	01111000	X	088	01011000
y	121	01111001	Y	089	01011001
z	122	01111010	Z	090	01011010

Figure 11: ASCII code (partial)

Average code length

- How to measure representation efficiency of a code?
- **Average code length** = average of the codeword lengths:

$$\bar{l} = \sum_i p(s_i) l_i$$

- The probability of a codeword = the probability of the corresponding message
- Smaller average length: code more efficient (better)
- How small can the average length be?

Definitions

A code can be:

- **non-singular**: all codewords are different
- **uniquely decodable**: for any received sequence of symbols, there is only one corresponding sequence of messages
  - i.e. no sequence of messages produces the same sequence of symbols
  - i.e. there is never a confusion at decoding
- **instantaneous** (also known as **prefix-free**): no codeword is prefix to another code
  - A *prefix* = a codeword which is the beginning of another codeword

Examples: at the blackboard

The graph of a code

Example at blackboard

Instantaneous codes are uniquely decodable

Theorem

An instantaneous code is uniquely decodable
(The converse is not necessary true; there exist uniquely decodable codes which are not instantaneous)

Proof

- blackboard
- How to decode an instantaneous code: graph-based decoding
- Advantage on instantaneous code over uniquely decodable: simple decoding

Existence of instantaneous codes

- When can there an instantaneous code exist?

Kraft inequality theorem

There exists an instantaneous code with D symbols and codeword lengths $l_1, l_2, \ldots l_n$ if and only if the lengths satisfy the following inequality:

$$\sum_i D^{-l_i} \geq 1.$$

Proof

At blackboard

Comments:

- If lengths do not satisfy this, no instantaneous code exists
- If the lengths of a code satisfy this, that code can be instantaneous or not (there exists an instantaneous code, but not necessarily that one)
- Kraft inequality means that the codewords lengths cannot be all very small

Kraft inequality for uniquely decodable codes

- Instantaneous codes must obey Kraft inequality
- How about uniquely decodable codes?

McMillan theorem

An uniquely decodable code satisfies the Kraft inequality:

$$\sum_i D^{-l_i} \geq 1.$$

Sal

Consequence

- For every uniquely decodable code, there exists in instantaneous code with the same lengths.
- Even though the class of uniquely decodable codes is larger than that of instantaneous codes, we have no benefit.
- We can always use just instantaneous codes.

Optimal codes

Non-optimal codes