Information Theory

Chapter I: Discrete information sources

Block diagram of a communication system

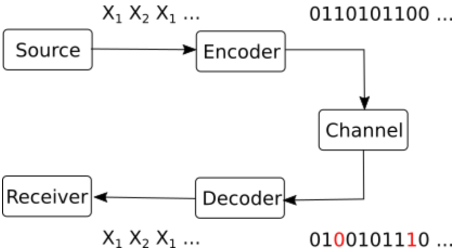

Figure 1: Block diagram of a communication system

- Source: creates information messages
- Encoder: converts messages into symbols for transmission (i.e bits)
- Channel: delivers the symbols, introduces errors
- Decoder: detects/corrects the errors, rebuilds the information messages

What is information?

Example:

- Consider the 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 when you don’t already know the result
  - if you already known the result, the message is useless (brings no information)
  - if the result was to be expected, there is little information. If the result is highly unusual, there is more information in this message (think betting)

Information source

- Information is related to probability theory:
  - there is a *probabilistic source* that can produce a number of different *events*
  - each event has a certain probability. All probabilities are known beforehand
  - at one time, an event is randomly selected according to its probability
- 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 also happened
- The quantity of information is dependent on its probability

Discrete memoryless source

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

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

- Terminology:
  - 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 **creates information**: 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**:

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

where $p(s_k)$ 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 $\log_2()$
- In this case, H(S) is 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 the 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)

Proof: via definition

2. $H(S)$ is maximum when all n messages have equal probability $\frac{1}{n}$. The maximum value is $\max H(S) = \log(n)$

Proof: only for the case of 2 messages, use derivative in definition

3. *Diversification* of the source always increases the entropy

Proof: compare entropies in both cases

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:

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

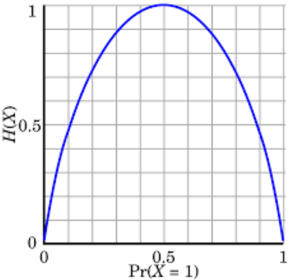

Figure 2: Entropy of a binary source

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?

Efficiency and redundancy

- Efficiency of a DMS:

$$\eta = \frac{H(S)}{H_{max}} = \frac{H(S)}{\log(n)}$$

- Absolute redundancy of a DMS:

$$R = H_{max} - H(S)$$

- Relative redundancy of a DMS:

$$\rho = \frac{H_{max} - H(S)}{H_{max}} = 1 - \eta$$

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:

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

- Measured in **bps** (bits per second)
- Important for data communication

Distance between distributions

- How to measure how similar / how different are two distributions?
  - must have the same number of messages
  - example: $p(s_1), ...p(s_n)$ and $q(s_1), ...q(s_n)$

Definition: the **Kullback–Leibler distance** of two distributions P and Q is

$$D_{KL}(P||Q) = \sum_i p(s_i) \log(\frac{p(s_i)}{q(s_i)})$$

- It is a way to measure the **distance (difference)** between two distributions

- Also known as *relative entropy*, or the Kullback-Leibler *divergence*

Properties of Kullback-Leibler distance

- Properties:
  - $D_{KL}(P||Q)$ is always ≥ 0, and is equal to 0 only when P and Q are the same
  - the higher $D_{KL}(P||Q)$ is, the more different the distributions are
  - it is **not commutative**: $D_{KL}(P||Q) \neq D_{KL}(Q||P)$
- Example: at whiteboard
- Example usage: classification systems (cross-entropy loss)

Extended DMS

- Definition: the **n-th order extension** of a DMS S, S^n is a source which 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, probabilities multiply:

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

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:

$$010011001110010100...$$

  - Can be grouped in bits, half-bytes, bytes, 16-bit words, 32-bit long words, and so on
  - Can be considered:
    - N messages from a binary source (with 1 bit), or
    - N/2 messages from a source with 4 messages (with 2 bits)...
    - etc

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 (notation S_i).
- A source with n messages and memory m => has 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:

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

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

- 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

At whiteboard: draw states and transitions for previous example (source with $n = 4$ messages and memory $m = 1$)

Entropy of sources with memory

- What entropy does s source with memory have?
- 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, the fraction of time when the source was in state S_k)

Ergodic sources

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

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

- After one more message:

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

- In general, starting from time 0, 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.

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

- These are the probabilities to be used in the entropy formula for memory sources

Finding the stationary probabilties

- How to find the stationary probabilities?
- When n is very large, after n messages and after $n+1$ messages the probabilities 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

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