Introduction to source coding

Dr. Yoann Altmann

B39AX – Fall 2023
Heriot-Watt University

Plan

- Types of compression
- Lossless compression
  - Expected code length
  - Prefix codes
  - Optimal codes
  - Shannon source coding theorem (symbol code)
  - Huffman code
- Based on the book:
  “Information Theory,. **Inference**, and **Learning Algorithms”**. David J.C. MacKay (Chap. 4-5)

Revision (DMS)

- X takes values from the alphabet $\mathcal{A} = \{a_1, a_2, \dots, a_N\}$ with probabilities $p_n = \mathbb{p}(X = a_n)$
- Information content of a_n

$$I(a_n) = \log_2\left(\frac{1}{p_n}\right) = -\log_2(p_n)$$

- Entropy

$$H(X) = \mathbb{E}[I(a_n)] = \sum_{n=1}^{N} p_n I(a_n) = -\sum_{n=1}^{N} p_n \log_2 p_n$$

Continuous RV

- Information content of a_n

$$I(x) = \log_2\left(\frac{1}{f(x)}\right) = -\log_2(f(x))$$

- Entropy

$$H(X) = \mathbb{E}[I(x)] = -\int f(x)\log_2\big(f(x)\big)dx$$

Joint entropy

- Let X and Y be two (discrete) RVs defined on $\mathcal{A}_X$ and $\mathcal{A}_Y$

$$H(X,Y) = -\sum_{x\in\mathcal{A}_X, y\in\mathcal{A}_Y} p(x,y)\log_2\big(p(x,y)\big)$$

Important property

- Let X and Y be two RVs. We have

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

if and only if X and Y are independent, i.e.,

$$p(X,Y) = p(X)p(Y)$$

Two main types of compression

- Lossy compression: some original messages are assigned the same code, which makes perfect reconstruction impossible. (not covered in this course)

  Ex: image compression

- Lossless compression: some symbols are shorten, some are made longer. The goal is to shorten the message with high probability

Here we are not considering (channel) noise errors yet

Lossless compression

- In the lossy compression context
  - Compression of blocks of RVs
  - Same code length for each symbol
  - Parts of the alphabet are not coded
- In the lossless compression context
  - Symbol coding
  - Stream coding (not covered in this course)
  - Some codes are short, some are long
- Analysis of variable-length encoding schemes

Notations

- $\mathcal{A}^N$: set of ordered N-tuples of elements from $\mathcal{A}$, i.e., all the strings of lengths N.
- $\mathcal{A}^+$: set of all strings of finite length composed of the elements of $\mathcal{A}$

Examples:

$$\{0,1\}^3 = \{000,001,010,011,100,101,110,111\}$$

$$\{0,1\}^+ = \{0,1,00,01,11,000, \dots\}$$

Binary symbol code

- A binary symbol code C for X is a mapping from $\mathcal{A}$ to $\{0,1\}^+$
- $c(x)$ is the codeword corresponding to x and $l(x)$ is its length, i.e., $l_i = l(a_i)$
- Extended code C^+ is a mapping from $\mathcal{A}^+$ to $\{0,1\}^+$obtained by concatenation, without punctuation of the corresponding codewords
- $c^+(x_1 x_2 \dots x_N) = c(x_1)c(x_2) \dots c(x_N)$

Example

- $\mathcal{A} = \{a, b, c, d\}, \mathcal{P} = \left\{\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}\right\}$
- Let C by defined by

a_i	$c(a_i)$	l_i
a	1000	4
b	0100	4
c	0010	4
d	0001	4

- $c^+(acdb) = 1000001000010100$

Uniquely decodable encoding

- A code C is uniquely decodable if under C^{+}no two distinct strings have the same encoding

$$\forall\, \boldsymbol{x}, \boldsymbol{y} \in \mathcal{A}^{+}, \boldsymbol{x} \neq \boldsymbol{y} \Rightarrow c^{+}(\boldsymbol{x}) \neq c^{+}(\boldsymbol{y})$$

- A valid encoding should be uniquely decodable
- How to ensure this?

Example 1

- $\mathcal{A} = \{a, b, c, d\}, \mathcal{P} = \left\{\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}\right\}$
- Let C by defined by

$\boldsymbol{a_i}$	$\boldsymbol{c(a_i)}$	$\boldsymbol{l_i}$
a	1000	4
b	0100	4
c	0010	4
d	0001	4

- C is uniquely decodable

Example 2

- $\mathcal{A} = \{a, b, c, d\}, \mathcal{P} = \left\{\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}\right\}$
- Let C by defined by

$\boldsymbol{a_i}$	$\boldsymbol{c(a_i)}$	$\boldsymbol{l_i}$
a	1	1
b	110	3
c	0011	4
d	0001	4

- C is not uniquely decodable

$$c^+(\boldsymbol{aada}) = 1100011 = c^+(\boldsymbol{bc})$$

Prefix codes

- A symbol code is called prefix code if no codeword is a prefix of any other codeword

- Prefix code = instantaneous/self-punctuating code
- A prefix code is uniquely decodable
- A uniquely decodable code is not necessary a prefix code

Examples

- $C_1 = \{0,101\}$: prefix code
- $C_2 = \{1,101\}$: not prefix code (but uniquely decodable)
- $C_3 = \{0,10,110,111\}$: prefix code
- $C_4 = \{00,01,10,11\}$: prefix code