COMSATS Institute of Information Technology

ECI750 Multimedia Data Compression

Lecture 2

Information Theory and Coding – 1

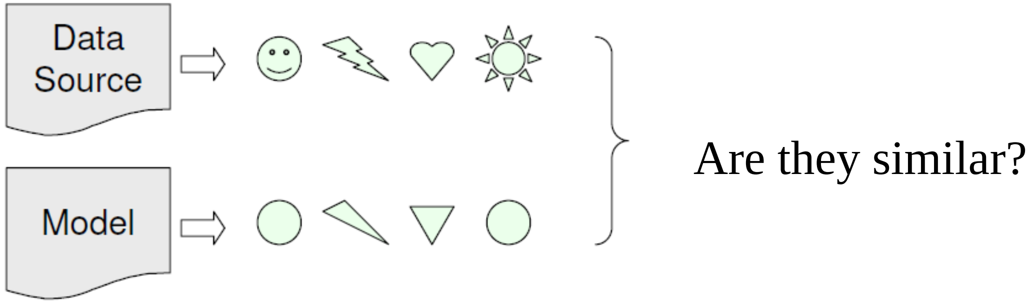
Dr. Shadan Khattak

Department of Electrical Engineering

COMSATS Institute of Information Technology - Abbottabad

Modelling and Coding

- Generally, data compression algorithms go through two phases:
  - Modelling
    - We try to extract information about any redundancy that exists in the data and describe the redundancy in the form of a model.
  - Coding
    - A description of the model and a description of how the data differ from the model are encoded, generally, using a binary alphabet.
    - The difference between the model and the data is often referred to as the residual.



Modelling and Coding

- **Example 1.2.1 (Linear model)**
  - Data sequence x_i,

9	11	11	11	14	13	15	17	16	17	20	21

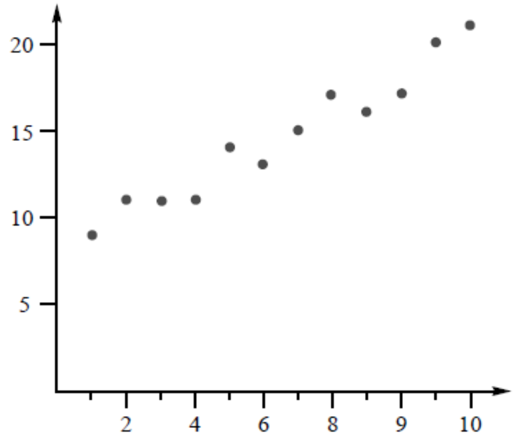
  - Its binary representation will cost: 5 bits $(2^5 = 32)$
  - A model for this data could be a straight line defined by the equation

$$\hat{x}_n = n + 8 \qquad n = 1,2,3 \dots$$

  - Residual:

$$e_n = x_n - \hat{x}_n = 0,1,0,-1,1,-1,0, \dots$$

  - Binary representation of the residual will cost: 2 bits $(2^2 = 4)$



Dr. Shadan Khattak
Department of Electrical Engineering
CIIT - Abbottabad

Modelling and Coding

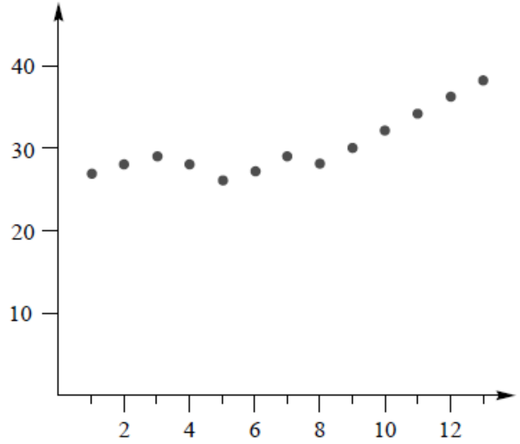
- **<u>Example 1.2.2 (Differential Model)</u>**
  - Data sequence x_i,

27	28	29	28	26	27	29	28	30	32	34	36	38

  - Each value is close to the previous value.
  - Send the first value, then for subsequent values, send difference between it and previous value i.e.,

27	1	1	–1	–2	1	2	–1	2	2	2	2	2

  - Such coding is called predictive coding.
  - The number of distinct values has been reduced (fewer bits are required to represent each number and compression is achieved)



Modelling and Coding

- **<u>Example 1.2.3 (Variable length coding)</u>**
  - Given a sequence of symbols,

    abarayaranbarraybranbfarbfaarbfaaarbaway
  - Unique symbols: 7
  - If fixed length coding (FLC) is used: 3 bits/symbol $(2^3 = 9)$
  - If variable length coding (VLC) is used: 2.58 bits/symbol
  - Compression Ratio = $\frac{3}{2.58} = \frac{1.16}{1} = 1.16{:}1$

TABLE 1.1 **A code with codewords of varying length.**

a	1
n	001
b	01100
f	0100
n	0111
r	000
w	01101
y	0101

Information Theory

- Shannon defines "self-information" as an entity that measures the amount of information associated with an event A of probability P(A) as:

$$i(A) = log_b \frac{1}{P(A)} = -log_b P(A)$$

- If the probability of an event is low, the amount of self-information associated with it is high.
- If the probability of an event is high, the amount of self-information associated with it is low.

$$i(A) = 0 \; for \; P(A) = 1$$
$$i(A) \geq 0 \; for \; 0 \leq P(A) \leq 1$$
$$i(A) > i(B) \; for \; P(A) < P(B)$$
$$i(AB) = i(A) + i(B) \; if \; A \; and \; B \; are \; independent \; events$$

Information Theory

- Preliminaries of the logarithm function
  - $\log_b x = a$ means that $b^a = x$
  - If $\log_b x$ is not available on your calculator, you can calculate it as:
    - $\frac{lnx}{lnb} = a$
    - E.g., $\log_2 8 = 3$
      - This result can also be obtained as:
      - $\frac{ln8}{ln2} = \frac{2.07944154168}{0.69314718056} = 3$

Information Theory

- **Example 2.2.1**

a) For a fair coin:
   - $P(H) = \frac{1}{2},\ i(H) = ?$
   - $P(T) = \frac{1}{2},\ i(T) = ?$

b) If the coin is not fair and the probabilities are:
   - $P(H) = 1/8,\ P(T) = 7/8$
   - $i(H) = ?,\quad i(T) = ?$

Ans:

$a.\ \ i(H) = i(T) = 1\ bit$

$b.\ \ i(H) = 3\ bits,\ i(T) = 0.193\ bits$

Information Theory

- **Entropy**
  - If we have a set of independent events A_i, S is the sample space of all events, then the average self-information is given by:

$$H = \sum P(A_i)i(A_i) = -\sum P(A_i)\log_b P(A_i)$$

  - Here, H is called the ***entropy*** associated with the experiment.
  - Given a data source S, entropy is the minimal average number of bits to represent the output.

Information Theory

- **<u>Estimation of Source Entropy (1)</u>**
  - As entropy depends on probability which may not be known in advance of an event.
  - So what do we do if we don't know the probability of an event? We estimate the entropy!
  - Example:

$$S = 1\ 2\ 3\ 2\ 3\ 4\ 5\ 4\ 5\ 6\ 7\ 8\ 9\ 8\ 9\ 10$$

$$P(1) = P(6) = P(7) = P(10) = \frac{1}{16}$$

$$P(2) = P(3) = P(4) = P(5) = P(8) = P(9) = \frac{2}{16}$$

$$H = -\sum_{i=1}^{10} P(i)\log_2 P(i)$$

$$= 3.25\ bits$$

Dr. Shadan Khattak
Department of Electrical Engineering
CIIT - Abbottabad

Information Theory

- **<u>Estimation of Source Entropy (2)</u>**
  - If we assume a sample-to-sample correlation and remove the correlation by taking differences of neighbouring sample values, we get the residual sequence R i.e.

$$R = 1\ 1\ 1\ -1\ 1\ 1\ 1\ -1\ 1\ 1\ 1\ 1\ 1 - 1\ 1\ 1$$

  - The sequence has only two symbols (1) and (-1).

$$P(1) = \frac{13}{16}, P(-1) = \frac{3}{16}$$

$$H = -\sum P(i) \log_2 P(i)$$

$$= 0.7\ bits$$

Information Theory

- **<u>Estimation of Source Entropy (3)</u>**
  - Source alphabet can be manipulated to reduce its entropy. For example, consider:

$$S = 1\ 2\ 1\ 2\ 3\ 3\ 3\ 3\ 1\ 2\ 3\ 3\ 3\ 3\ 1\ 2\ 3\ 3\ 1\ 2$$

  - What is the entropy if letters 1, 2, and 3 are considered independently?
    - 1.5 bits/symbol.
    - Total bits required = 20 x 1.5 = 30.
  - Does the entropy decrease if we take blocks of two letters instead? E.g., 1 2 and 3 3.
    - Yes. Now entropy is 1 bit/symbol. Total bits required = 10 x 1 = 10 (a reduction of a factor of 3)

Dr. Shadan Khattak
Department of Electrical Engineering
CIIT - Abbottabad

Models for Coding

- Physical Models
- Probability Models
- Markov Models
- Composite Models

Dr. Shadan Khattak
Department of Electrical Engineering
CIIT - Abbottabad

Models for Coding

- Physical Models
  - Speech model
  - Telemetry model
  - Models based on physics are very complicated and are difficult to implement
  - So what do we do? We use statistical (probabilistic) models!

Models for Coding

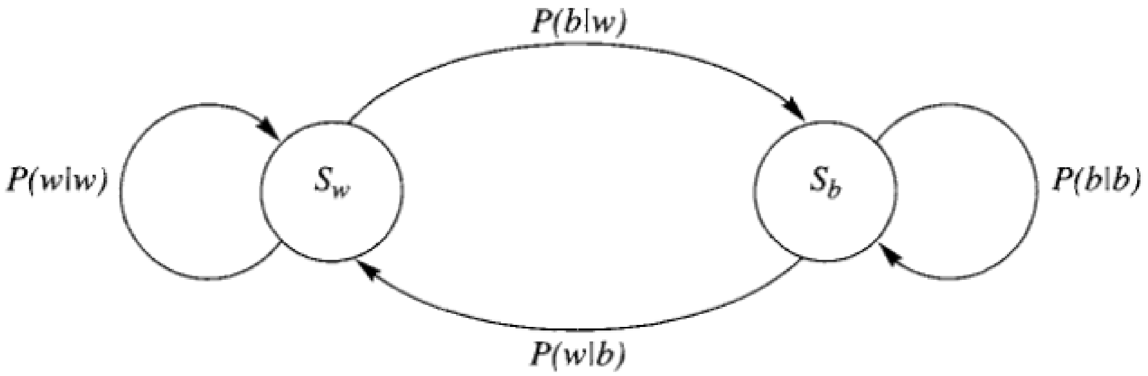
- Probability Models
  - For a source that generates letters from an alphabet $A = \{a_1, a_2, a_3, \dots, a_M\}$, we can have a probability model $P = \{P(a_1), P(a_2), P(a_3), \dots, P(a_M)\}$

Models for Coding

- Markov Models
  - A sequence $\{x_n\}$ fits a k^{th} order Markov model if
    $$P(x_n|x_{n-1},\dots,x_{n-k}) = P(x_n|x_{n-1},\dots,x_{n-k},\dots)$$
    (probability of the next symbols can be determined completely by knowing the past k symbols)
  - Each sequence of $x_{n-1},\dots,x_{n-k}$ is called a state.
  - If the alphabet set has size m, the number of states is m^k.
  - First order Markov model is most commonly used.
  - Markov models are particularly useful in text compression.
  - Also called finite context models
  - The larger the context, the lesser is the entropy but the more complex is the system
    - E.g., an alphabet of 95 letters with the last 4 symbols as context, the total number of contexts is $95^4 \sim 81\ million$

Models for Coding

- Markov Source Model
    - Markov model can also be described using a state transition diagram e.g., a two-state Markov model:



Models for Coding

- Entropy of a Finite-State Process
  - The entropy of a finite state process with state S_i can be computed by:

$$H = \sum_{i=1}^{M} P(S_i)H(S_i)$$

(where $H(S_i)$ is the entropy of a state S_i)

  - For example, for the two-state Markov model:

$$H(S_w) = -P(b|w)logP(b|w) - P(w|w)logP(w|w),$$

(where $\mathrm{P}(w|w) = 1 - P(b|w)$)

Models for Coding

- I.I.D vs Markov Model
  - For the two-state model, assume that

$$P(S_w) = \frac{30}{31}, P(S_b) = \frac{1}{31}$$
$$P(w|w) = 0.99, P(b|w) = 0.01$$
$$P(b|b) = 0.7, P(w|b) = 0.3$$

  - Under the iid assumption, the entropy is 0.206

Models for Coding

- I.I.D vs Markov Model
  - Using Markov Model,

$$H = \sum_{i=1}^{2} P(S_i)H(S_i)$$

$$H(S_w) = -0.01\log(0.01) - 0.99\log(0.99) = 0.0664 + 0.0142 = 0.081$$

$$H(S_B) = -0.7\log(0.7) - 0.3\log(0.3) = 0.881$$

$$H = \frac{30}{31} * 0.081 + \frac{1}{31} * 0.881 = 0.0783 + 0.0284 = 0.1067$$

  - The entropy is 0.1067

Models for Coding

- Composite Source Model
  - A composite source model can be described by n different sources and the probability P_i to select i^{th} source: