DCSP-3: Minimal Length Coding

Jianfeng Feng

Department of Computer Science Warwick Univ., UK

Jianfeng.feng@warwick.ac.uk

http://www.dcs.warwick.ac.uk/~feng/dcsp.html
Automatic Image Caption (better than human)

1. "girl in pink dress is jumping in air."
2. "black and white dog jumps over bar."
3. "young girl in pink shirt is swinging on swing."
4. "man in blue wetsuit is surfing on wave."
5. "little girl is eating piece of cake."
6. "baseball player is throwing ball in game."
7. "woman is holding bunch of bananas."
8. "black cat is sitting on top of suitcase."
This Week’s Summary:

- get familiar with 0 and 1

- Information theory

- Huffman coding: code events as economic as possible
Information sources

- \( X = \{x_1, x_2, \ldots, x_N\} \) with a known probability
- \( P(x_i) = p_i, \ i=1,2,\ldots,N \)

Example 1:
\[
X = (x_1 = \text{lie on bed at 12 noon today}, \quad x_2 = \text{in university at 12 noon today}, \quad x_3 = \text{attend a lecture at 12 noon today})
\]
\[
= (B, U, L)
\]
\[
p = (1/2, 1/4, 1/4),
\]

- \( H(X) = 0.5 \times 1 + 2 \times 1/4 + 2 \times 1/4 = 1.5 \) (Entropy)
- \( B=0, \ U=1, \ L=01 \) (coding)
- \( L_s = 0.5 \times 1 + 0.25 \times 1 + 0.25 \times 2 = 1.25 \) (average coding length)
Information sources

• Example 2. Left: information source \( p(x_i), i = 1, \ldots, 27 \)
  right: codes

To be, or not to be, that is the question— Whether 'tis Nobler in the mind to suffer The Slings and Arrows of outrageous Fortune, Or to take Arms against a Sea of troubles, And by opposing end them? To die, to sleep—

As short as possible

01110 00001111111
11111100000000000
1111000000011000
100010000010000
1011111111100000
Information source coding

- Replacement of the symbols (naked run/office in PM example) with a binary representation is termed **source coding**.

- In any coding operation we replace the symbol with a codeword.

- The purpose of source coding is to reduce the number of bits required to convey the information provided by the information source: 
  
  *minimize the average length of codes.*

- Conjecture: an information source of entropy $H$ needs on average only $H$ binary bits to represent each symbol.
An *instantaneous* code can be found that encodes a source of entropy $H(X)$ with an average number $L_s$ (average length) such that

$$L_s \geq H(X)$$
How does it work?

- Like many theorems of information theory, the theorem tells us nothing of how to find the code.

- However, it is useful results.

- Let us have a look how it works
Example

- Look at the activities of PM in three days with P(O)=0.9
- Calculate probability
- Assign binary codewords to these grouped outcomes.

<table>
<thead>
<tr>
<th>Sequence</th>
<th>OOO</th>
<th>OON</th>
<th>ONO</th>
<th>NOO</th>
<th>NNO</th>
<th>NON</th>
<th>ONN</th>
<th>NNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Probability</td>
<td>0.729</td>
<td>0.081</td>
<td>0.081</td>
<td>0.081</td>
<td>0.009</td>
<td>0.009</td>
<td>0.009</td>
<td>0.001</td>
</tr>
<tr>
<td>Codeword</td>
<td>0</td>
<td>1</td>
<td>01</td>
<td>10</td>
<td>11</td>
<td>00</td>
<td>000</td>
<td>111</td>
</tr>
<tr>
<td>Codeword Length (in bits)</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Weighted code length</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Entropy</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Variable length source coding
Example

• Table 1 shows such a code, and the probability of each code word occurring.

• Entropy is

\[ H(X) = -0.729 \log_2(0.729) - 0.081 \log_2(0.081) \times 3 \]
\[ -0.009 \log_2(0.009) \times 3 - 0.001 \log_2(0.001) \]
\[ = 1.4070 \]

• The average length of coding is given by

\[ L_s = 0.729 \times 1 + 0.081 \times 1 + 2 \times 0.081 \times 2 + 2 \times 0.009 \times 2 \]
\[ + 3 \times 0.009 + 3 \times 0.001 \]
\[ = 1.2 \]
Moreover, without difficulty, we have found a code that has an average bit usage less than the source entropy.

Example
• However, there is a difficulty with the code in Table 1.

• Before a code word can be decoded, it must be parsed.

• Parsing describes that activity of breaking the message string into its component codewords.
After parsing, each codeword can be decoded into its symbol sequence.

An instantaneously parsable code is one that can be parsed as soon as the last bit of a codeword is received.
Instantaneous code

• An instantaneous code must satisfy the prefix condition: that no codeword may be a prefix of any other code.

• For example: in the codeword, we should not use 1 11 to code two events
  When we receive 11, it could be ambiguous

• This condition is not satisfied by the code in Table 1.
Huffman coding

• The code in Table 2, however, is an instantaneously parsable code.

• It satisfies the prefix condition.
Huffman coding

<table>
<thead>
<tr>
<th>Sequence</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>Probability</td>
<td>0.729</td>
<td>0.081</td>
<td>0.081</td>
<td>0.081</td>
<td>0.009</td>
<td>0.009</td>
<td>0.009</td>
<td>0.001</td>
</tr>
<tr>
<td>Codeword</td>
<td>1</td>
<td>011</td>
<td>010</td>
<td>001</td>
<td>00011</td>
<td>00010</td>
<td>00001</td>
<td>00000</td>
</tr>
<tr>
<td>Codeword Length (in bits)</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

Weighted code length

Entropy

Table 2: OOO=A, OON=B, ONO=C, NOO=D, NNO=E, NON=F, ONN=G, NNN=H

- $L_s = 0.729 \times 1 + 0.081 \times 3 \times 3 + 0.009 \times 5 \times 3 + 0.001 \times 5$
  
  $= 1.5980$  (remember entropy is 1.4)
Huffman coding

Decoding

1 1 1 0 1 1 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 1
Huffman coding

• The derivation of the Huffman code tree is shown in the following Figure and the tree itself is shown in the next Figure.

• In both these figures, the letter A to H have been used in replace of the sequence in Table 2 to make them easier to read.
Huffman coding

Creating the tree:

1. Start with as many leaves as there are symbols.

2. Queue all leaf nodes into the first queue (in order).

3. While there is more than one node in the queues:
   - Remove two nodes with the lowest weight from the queues.
   - Create a new internal node, with the two just-removed nodes as children (either node can be either child) and the sum of their weights as the new weight.
   - Update the parent links in the two just-removed nodes to point to the just-created parent node.

4. Queue the new node into the second queue.

5. The remaining node is the root node; the tree has now been generated.
Huffman coding

Figure 22: Example Huffman code tree

- Prefix condition is obviously satisfied since in the tree above, each branch codes one alphabetic.
Huffman coding

- For example, the code in Table 2 uses 1.6 bits/symbol which is only 0.2 bits/symbol more bits per sequence than the theorem tells us is the best we can do.

- We might conclude that there is little point in expending the effort in finding a code less satisfying the inequality above.
Another thought

• How much have we saved in comparison with the most naïve idea?

• i.e. O=1, N=0

• \( L_s = 3 \left[ P(OOO) + \ldots + P(NNN) \right] = 3 \), halving it
My most favourite story (History)

• In 1951, David A Huffman and his MIT information theory classmates were given the choice of a term paper or a final exam.
• The Professor, Robert M Fano, assigned a term paper on the problem of finding the most efficient binary code.
• Huffman, unable to prove any codes were the most efficient, was about to give up when he hit upon the idea of using a frequency-sorted binary tree and quickly proved this method the most efficient.
• In doing so, the student outdid his professor, who had worked with information theory inventor Clude Shannon to develop an optimal code.
• By building the tree from the bottom up instead of the top down, Huffman avoided the major flaw of the suboptimal Shannon-Fano coding.
Coding English: Huffman Coding

Frequency for alphabetics
Turbo coding

- Using Bayesian theorem to code and decode
- Bayesian theorem basically said we should employ priori knowledge as much as possible
- Read yourself
DCSP-4: Fourier Transform

Jianfeng Feng

Department of Computer Science Warwick Univ., UK

Jianfeng.feng@warwick.ac.uk

http://www.dcs.warwick.ac.uk/~feng/dcsp.html
Coding

\[ L_s(X) > H(X) \]

Data transmission

Channel characteristics,
Signalling methods (ADC)
Interference and noise
Fourier transform
Data compression and encryption
Bandwidth

The range of frequencies occupied by the signal is called its **bandwidth**.
Nyquist-Shannon Theorem
Nyquist-Shannon Theorem

(Will be discussed in Chapter 3)

An analogue signal of bandwidth $B$ can be completely recreated from its sampled form provided it is sampled at a rate equal to at least twice its bandwidth.

That is

\[ S > 2B \]
• I will guess that $B = 1$ Hz

• Sample at $2B = 2$ Hz: $x[n] = [0 0 0 0]$

• Intuitively, I would say it will not work
• I will guess that $B = 1$ Hz

• Sample at $2B < 4$ Hz: $x[n] = [1\ 0\ -1\ 0\ 1\ 0\ -1\ 0\ ]$

• According to N-S Thm, we can fully recover the original signal
Example

- I will guess that $B = 1$ Hz

- Sample at $2B < 4$ Hz: $x[n] = [1 \ 0 \ -1 \ 0 \ 1 \ 0 \ -1 \ 0 ]$

- According to N-S Thm, we can fully recover the original signal

- Well, the blue line has the identical frequency, and $x[n]$. **What is wrong?**
Noise in a channel
Noise in a channel

\[ x(t)/G + \sigma(t) \]

Attenuation
Noise in a channel

\[ G[x(t)/G + \sigma(t)] = x(t) + G\sigma(t) \]
Noise in a channel

For a long, long channel we need repeaters

\[ x(t) \xrightarrow{1/G} \xi(t) \rightarrow G \rightarrow \xi_1(t) \xrightarrow{1/G} \xi_1(t) \rightarrow G \rightarrow \cdots \rightarrow G \rightarrow \xi_N(t) \]

\[ \xi_N(t) = x(t) + NG \sigma(t) \]
Noise in a channel
Noise therefore places a limit on the channel at which we can transfer information.

Obviously, what really matters is the signal to noise ratio (SNR).

This is defined by the ratio signal power $S$ to noise power $N$, and is often expressed in deciBels (dB):

$$SNR = 10 \log_{10} \left( \frac{S}{N} \right) \text{ dB}$$
Noise sources

Input noise is common in low frequency circuits and arises from electric fields generated by electrical switching.

It appears as bursts at the receiver, and when present can have a catastrophic effect due to its large power.

Other peoples signals can generate noise: cross-talk is the term give to the pick-up of radiated signals from adjacent cabling.
Noise sources

When radio links are used, interference from other transmitters can be problematic.

Thermal noise is always present. It is due to the random motion of electric charges present in all media. It can be generated externally, or internally at the receiver.

How to tell signal from noise?
Communication Techniques I

Fourier Transform

Time            frequency

bandwidth
noise power
Communication Techniques I

Time

Fourier Transform

frequency

bandwidth

noise power
Communication Techniques II

Time, frequency and bandwidth

We can describe a signal in two ways.

• One way is to describe its evolution in time domain, as we usually do.
• The other way is to describe its frequency content, in frequency domain: what we will learn
Your heartbeat

- Ingredients:
  - a frequency $\omega$ (units: radians)
  - an initial phase $\phi$ (units: radians)
  - an amplitude $A$ (units depending on underlying measurement)

- a trigonometric function
  - e.g. $x[n] = A \cos(\omega n + \phi)$
  - cosine wave, $x(t)$, has a single frequency,
    
    $\omega = 2\pi / T$
    
    where $T$ is the period i.e. $x(t+T) = x(t)$. 
What do we expect?

Time

Power

1 Hz

Fre
What do we expect?

Time

Power

1 Hz

Fre
What do we expect?

- Power
- Time
- 1 Hz
- Fre
What do we expect?

Time

Power

1 Hz

Fre
What do we expect?

![Graph showing a sine wave with time on the x-axis and power on the y-axis. The frequency is labeled as 1 Hz.](image-url)
This representation is quite general.

In fact we have the following theorem due to Fourier.

Any signal $x(t)$ of period $T$ can be represented as the sum of a set of cosinusoidal and sinusoidal waves of different frequencies and phases.
In mathematics, the continuous **Fourier transform** is one of the specific forms of Fourier analysis.

As such, it transforms one function into another, which is called the *frequency domain representation* of the original function (which is often a function in the time-domain).

In this specific case, both domains are continuous and unbounded.

The term **Fourier transform** can refer to either the frequency domain representation of a function or to the process/formula that "transforms" one function into the other.
Fourier Transform III

FIGURE 1. An Electrical Signal f(t)

FIGURE 2. Spectral Composition or Spectrum F(ω) or f(t)

FIGURE 3. Combined Time Domain and Frequency Domain Plots
Fourier Transform IV

- Continuous time (analogous signals): $FT$ (Fourier transform)
  
  it is in theory (in Warwick, we need it)

- Discrete time: $DTFT$ (infinity digital signals)
  
  it is in theory (discrete version)

- $DFT$: Discrete Fourier transform (finite digital signals)
  
  what we can use, one line in Matlab (fft)
History of FT I

- Gauss computes trigonometric series efficiently in 1805

- Fourier invents Fourier series in 1807

- People start computing Fourier series, and develop tricks Good comes up with an algorithm in 1958

- Cooley and Tukey (re)-discover the fast Fourier transform algorithm in 1965 for $N$ a power of a prime

- Winograd combined all methods to give the most efficient FFTs
History of FT II

Gauss
Fourier

History of FT III
Jianfeng Feng

History of FT IV

\[ G(0) = \int g(x)e^{-2\pi iax} \, dx \]
Hi, Prof Feng?

Yeah, uh... I accidentally took the Fourier transform of my cat...

Meow!
Complex Numbers

\[ j = \sqrt{-1} \]
\[ z = x + jy = r \exp(j\theta) \]
\[ x = r \cos(\theta) \]
\[ y = r \sin(\theta) \]
Euler Formular

\[ \exp(j\, a) = \cos a + j \sin a \]
The complex exponential

- the trigonometric function of choice in DSP is the complex exponential:

\[ x[n] = A \exp(j(\omega n + \phi)) \]

\[ = A[\cos(\omega n + \phi) + j \sin(\omega n + \phi)] \]
The complex exponential:

\[
e^x = \lim_{n \to \infty} (1 + x/n)^n
\]

\[
e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}
\]

\[
e^{j\theta} = \cos(\theta) + j \sin(\theta)
\]

\[
e^{jn\theta} = \cos(n\theta) + j \sin(n\theta)
\]

\[
\sin(\theta) = \frac{e^{j\theta} - e^{-j\theta}}{2j}
\]

\[
\cos(\theta) = \frac{e^{j\theta} + e^{-j\theta}}{2}
\]

\[
e = 2.71828182...
\]
Most beautiful Math Formula

$$exp ( j \pi ) + 1 = 0$$

- Where $e$ is Euler's number
- $J$ is the imaginary unit
Fourier's Song

• Integrate your function times a complex exponential
• It's really not so hard you can do it with your pencil
• And when you're done with this calculation You've got a brand new function - the Fourier Transformation
• What a prism does to sunlight, what the ear does to sound
• Fourier does to signals, it's the coolest trick around
• Now filtering is easy, you don't need to convolve All you do is multiply in order to solve.
• From time into frequency - from frequency to time Every operation in the time domain
• Has a Fourier analog - that's what I claim
• Think of a delay, a simple shift in time
• It becomes a phase rotation - now that's truly sublime!
  And to differentiate, here's a simple trick Just multiply by J omega, ain't that slick? Integration is the inverse, what you gonna do? Divide instead of multiply - you can do it too.
• Or make the pulse wide, and the sinc grows dense, The uncertainty principle is just common sense.
• From time into frequency - from frequency to time
• Let's do some examples... consider a sine
• It's mapped to a delta, in frequency - not time
• Now take that same delta as a function of time
• Mapped into frequency - of course - it's a sine!
• Sine x on x is handy, let's call it a sinc.
• Its Fourier Transform is simpler than you think.
• You get a pulse that's shaped just like a top hat...
• Squeeze the pulse thin, and the sinc grows fat.
Example
Fun: Decoding dream (Horikawa et al. Science, 2013)

Subject 2: 118th awakening
book
building
car
character
commodity
computer-screen
covering
dwelling
electronic-equipment
female
food
furniture
male
mercantile-establishmer
point
region
representation
street

Score

Time to awakening (s)
Fun

想了解大脑更多？
微信公众号 neuroscienceme，知乎专栏「神经科学」

首发自优酷「神经科学专栏」： http://i.youku.com/neuroscience
翻译 by 王子舟