COMS3100/7100

Introduction to Communications

Lecture 28: *Source Coding*

This lecture:

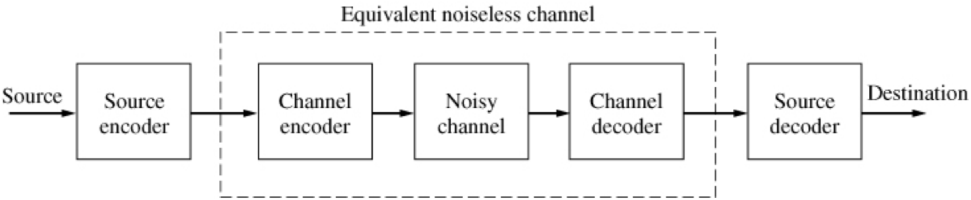
1. Information Theory.
2. Entropy.
3. Source Coding.
4. Huffman Coding.

Ref: CCR pp. 697–709, A Mathematical Theory of Communication.

Information Theory

Claude Shannon's paper, *A Mathematical Theory of Communication*, showed that reliable communication is possible at non-zero rate.

- Shannon proposed the following model for point-to-point communication.



- Shannon showed that there is a very widely applicable, quantitative definition of information.
- The source generates information at a certain *rate*.
- The noisy channel can be shown to have a *capacity* at which it can reliably transmit information.
- Reliable transmission is possible if and only if the source's rate is not greater than the channel's capacity.

The Mathematical Model

The source produces symbols from an *alphabet*.

- The *alphabet* is the (discrete) set of all possible symbols.

E.g., the Roman alphabet, the Unicode character set or the range of possible pixel intensities from a camera sensor.

- The source produces these symbols in a discrete-time sequence.
- The channel has its own alphabet or, rather, *alphabets*: one at the input and one at the output.

E.g., consider a channel including a polar NRZ modulator and demodulator, so the input and output alphabets are $\{-A, +A\}$.

- At each *use* of the channel, the channel output is random (because of noise) but dependent on the input.
- The rate at which the source generates symbols may be different to the rate at which the channel is used.
- Some mechanism is needed to translate between the source and the channel alphabets and back again.

- Shannon found that this can always be broken into two *independent* processes at the transmitter: source and channel coding.
    - Similarly, at the receiver, source and channel decoding.
- Without loss of generality, the common language of the source and channel (de)coder is a *bitstream*.
- Source symbol generation, intermediate bitstream and channel use may all be at different rates.

Information & Entropy

We measure information and entropy in terms of probability and random variables.

Self-Information

- In order to use random variables, map the source alphabet to the numbers $\{0, \ldots, M-1\}$ where M is the size of the alphabet.
- Let the source symbol selected for transmission at a certain time instant be represented as a discrete r.v. X.
- Let i represent one possible value for X and define $p_i = P(X = i)$.
- The amount of information (or surprise) at learning that $X = i$ is $\log(1/p_i) = -\log p_i$.
  - Hence, the rarer the event, *i.e.*, the less probable, the more surprising and the more informative it is.
  - Shannon calls this *self-information*.

  - The base of the logarithm has not been specified, but it is usually taken to be 2, in which case the unit is *bits*.
    - If we take the natural logarithm, the unit is *nats*.
    - (Technically, self-information is dimensionless.)
  - Measurement in bits is natural: if eight symbols are equally likely, it makes sense that they have 3 bits of information each.

Entropy

The expected self-information is called the *entropy* of X:

$$H(X) = E\left[\log \frac{1}{p_i}\right] = -\sum_{i=0}^{M-1} p_i \log p_i.$$

- Shannon used this name because of the similar expression that arises in statistical thermodynamics.
- As in thermodynamics, it can be regarded as measure of randomness or disorder.
- Degenerate r.v.s have zero entropy.

- It can be shown that uniformly distributed r.v.s have the highest entropy for a given M, so that

$$0 \leqslant H(X) \leqslant \log M.$$

- We'll use $H_b(X)$ when we need to be explicit about using a logarithm to the base b.

The Source Coding Theorem

If, each time the source emits a symbol, it is independent of previous symbols and identically distributed, we call it a *discrete memoryless source (DMS)*.

- Suppose the DMS emits r symbols per second.
- Shannon showed that it is possible to use source coding to encode the symbols in a bitstream at $rH_2(X)$ bits per second.
  - Conversely, he showed it is impossible to have a uniquely decodable bitstream at a lower rate.
- This is Shannon's *source coding theorem* (and converse).

Shannon's Source Coding Procedure

In proving the source coding theorem, Shannon devised a simple but impractical source coding scheme.

- We group N symbols together into a *block*.
- All likely sequences (in a certain sense) are identified.
- These sequences are enumerated using binary words of $NH_2(X)$ bits (rounded down; ignoring some sequences if too many).
  - This constitutes the *codebook*.
- To perform source coding, compare a given symbol sequence against those in the codebook & output the code, if there is one.
- The probability that this scheme doesn't work $\to 0$ as $N \to \infty$.
- This scheme is impractical because there is not necessarily much structure in the codebook.

  $\Rightarrow$ The codebook may require massive storage space.
  $\Rightarrow$ The codebook may need to be exhaustively searched.

Variable-Length Codes

An ideal source-coding scheme is theoretically easy but practically difficult.

- Also, for finite N, the scheme is unreliable (not all sequences have codes!).
- How to make the best possible codes for finite block sizes?
- We'll start with codes for a single symbol, *i.e.*, $N = 1$.
- Consider a *variable-length* source code where each symbol maps to a variable number of bits.
- Suppose symbol i is assigned a code of n_i bits.
- It turns out that the code can be made uniquely decodable if and only if the *Kraft inequality* is satisfied:

$$\sum_{i=0}^{M-1} 2^{-n_i} \leqslant 1.$$

Huffman Coding

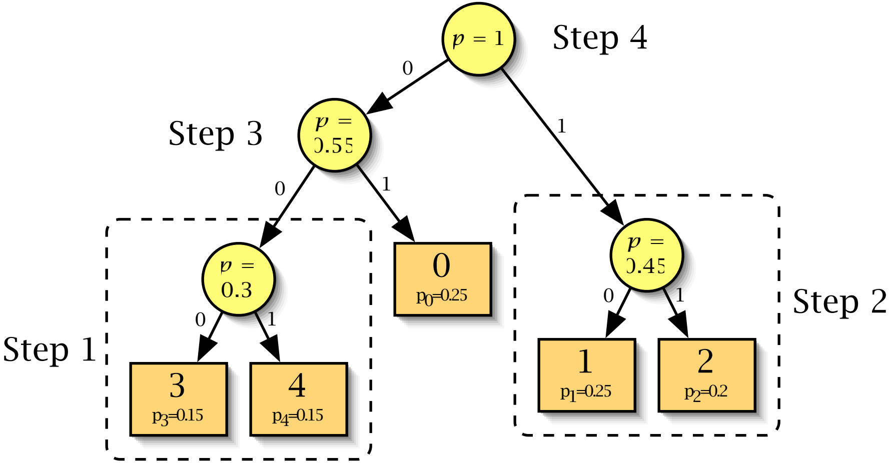
In 1952, David Huffman discovered a simple method of constructing an optimal variable-length, uniquely decodable source code.

- The method constructs a *binary tree*—a tree in which each node has at most two children.
- It proceeds from the bottom up, combining leaves into 'twigs', twigs into 'branches' and so on until the tree is built.
- Let's call any partially assembled portion of the tree a *twig*.
- To start with, there are M twigs which consist only of the leaves themselves, the symbols.
- The probability of a twig is the sum of the probabilities of all of its leaves.
- The code construction algorithm is simply the following step, iterated ($M - 1$ times) until only one twig remains:
  - Choose the two twigs with the least probability and assemble them together to make a larger twig.

- To read off the codes, descend through the tree towards the symbol's leaf.
  - Each time we take a left branch, output a '0', otherwise a '1'.

Example

Consider a source with $M = 5$ for which the probabilities are $p_0 = p_1 = 0.25$, $p_2 = 0.2$, $p_3 = p_4 = 0.15$.



- Average code length is 2.3 bits and the entropy is 2.29 bits.

Developments of Source Coding

Source coding is also known as *data compression*.

- More particularly, *lossless* data compression, since the input and output symbols are identical.
- Our exposition required that the probability distribution is known in advance.
- If we don't, we can use *universal* source coding.
  - Examples: Lempel-Ziv (LZ77) & Lempel-Ziv-Welch (LZW) algorithms & derivatives such as DEFLATE in ZIP & gzip software.
- For sources like English text, lossless coding is very important.
- In other applications, like audio, images and video, we may be able to put up with some distortion for a lower bit rate.
- In 1963, Shannon developed *rate-distortion* theory, the basis of modern *lossy* data compression.
  - Examples: voice coding in mobile phones, MP3 for music, JPEG for images, MPEG for video.