Lecture 1: Shannon's Theorem

Lecturer: Travis Gagie

January 13th, 2015

Welcome to Data Compression! I'm Travis and I'll be your instructor this week. If you haven't registered yet, don't worry, we'll work all the administrative details out later.

Your mark will be based on participation in classes and exercise sessions and on a final exam. We'll let you know the exact marking scheme later.

Any questions before we get started?

Data compression has been around for a long time, but it only got a theoretical foundation when Shannon introduced information theory in 1948.

Proving things about information requires a precise, objective definition of what it is and how to measure it. To sidestep philosophical debates, Shannon considered the following situation:

Suppose Alice and Bob know the probability distribution according to which a random variable X takes on values according to a probability distribution $P = p_1, \ldots, p_n$.

Alice learns X's value and tells Bob. How much information does she convey in the expected case?

Shannon posited three axioms:

- the expected amount of information conveyed should be a continuous function of the probabilities;
- if all possible values are equally likely, then the expected amount of information conveyed is monotonically increasing with how many there are;
- if X is the combination of two random variables Y and Z, then the expected amount information conveyed about X is the expected amount of information conveyed about Y plus the expected amount of information conveyed about Z.

For example, suppose X takes on the values from 0 to 99, Y is the value of X's first digit and Z is the value of X's second digit. When Alice tells Bob X's value, the expected amount of information conveyed about X is the expected amount of information conveyed about Y plus the expected amount of information conveyed about Z.

Shannon showed that the only function that satisfies his axioms is

$$H(P) = \sum_i p_i \log \frac{1}{p_i} .$$

This is called the *entropy* of P (or of X, according to some people).

The base of the logarithm determines the units in which we measure information. Electrical engineers sometimes use ln and work in units called nats. Computer scientists use $\lg = \log_2$ and work in bits.

The quantity $\lg(1/p)$ is sometimes called the self-information of an event with probability p.

"Bit" is short for "binary digit" and 1 bit is the amount of information conveyed when we learn the outcome of flipping a fair coin (because $(1/2)\lg(1/(1/2)) + (1/2)\lg(1/(1/2)) = 1$).

Unfortunately, "bit" has (at least) two meanings in computer science: "Alice sent 10 bits [symbols transmitted] to send Bob 3 bits [units of information]."

More importantly (for us), Shannon showed that the minimum expected message length Alice can achieve with any binary prefix-free code is in $[H(P), H(P) + 1)$.

We consider prefix-free codes because Alice can't relinquish control of the channel to the next transmitter until appending more bits can't change how Bob will interpret her message.

First we'll prove the upper bound.

We can assume without loss of generality that $p_1 \geq \cdots \geq p_n > 0$. Building a binary prefix-free code with expected message length ℓ is equivalent to building a binary tree on n leaves at depths $d_1, \ldots, d_n$ such that $\sum_i p_i d_i = \ell$.

Consider the binary representations of the partial sums

$$0, p_1, p_1 + p_2, \ldots, p_1 + \cdots + p_{n-1} .$$

Since the ith partial sum differs from all the other by at least p_i, the ith binary representation differs from all the others on at least one of its first $\lceil \lg(1/p_i) \rceil$ bits (to the right of the point).

To see why, notice that if two binary fractions agree on their first b bits, then they differ by strictly less than 2^{-b}.

Therefore, the ith binary representation differs agrees with each other representation on fewer than $\lg(1/p_i)$ of its first bits.

All this means we can build a binary prefix-free code with expected message length

$$\sum_i p_i \left\lceil \lg \frac{1}{p_i} \right\rceil < \sum_i p_i \lg \frac{1}{p_i} + 1 = H(P) + 1 \,.$$

Notice we achieve expected message length $H(P)$ if each p_i is an integer power of 2.

Now we'll prove the lower bound.

For any binary prefix-free code with which we can encode X's possible values, consider the corresponding binary tree on n leaves. Without loss of generality, we can assume this tree is strictly binary. Let $d_1, \ldots, d_n$ be the depths of the leaves in order by their corresponding values, and let $Q = q_1, \ldots, q_n = 1/2^{d_1}, \ldots, 1/2^{d_n}$.

The amount by which the expected message length of this code exceeds $H(P)$ is

$$\sum_i p_i d_i - H(P) = -\frac{1}{\ln 2} \sum_i p_i \ln \frac{q_i}{p_i} .$$

Since $\ln x \leq x - 1$ for $x > 0$ with equality if and only if $x = 1$,

$$\sum_i pi \ln \frac{q_i}{p_i} \leq \sum_i p_i \left(\frac{q_i}{p_i} - 1 \right) = \sum_i q_i - \sum_i p_i = 0$$

with equality if and only if $Q = P$.

Therefore, the expected message length exceeds $H(P)$ unless $Q = P$, in which case it equals $H(P)$. Notice we can have

$$Q = 1/2^{d_1}, \ldots, 1/2^{d_n} = P$$

only when each p_i is an integer power of 2. That is, this condition is both necessary and sufficient for us to achieve the entropy (with a binary code).

Theorem (Shannon, 1948)

Suppose Alice and Bob know a random variable X takes on values according to a probability distribution $P = p_1, \ldots, p_n$. If Alice learns the value of X and tells Bob using a binary prefix-free code, then the minimum expected length of her message is at least $H(P)$ and less than $H(P) + 1$, where $H(P) = \sum_i p_i \lg(1/p_i)$ is the entropy of P.

Although Shannon is known as the father of information theory, it wasn't his only important contribution to electrical engineering and computer science. He was also the first to propose modelling circuits with Boolean logic, in his masters thesis.

On Thursday we'll see the result of another masters thesis: Huffman's algorithm for constructing a prefix-free code with minimum expected message length.

It's important to note that Shannon and Huffman considered X to be everything Alice wants to tell Bob. If Alice is sending, say, a 10-page report, then it's unlikely she and Bob have agreed on a distribution over all such reports.

Shannon or Huffman coding can still be applied, e.g., by pretending that each character of the report is chosen independently according to a fixed distribution, then encoding each character according to a code chosen for that distribution. (Alice can start by sending Bob the distribution of characters in the report together with its length.)

We now have the option of using codes that are not prefix-free, as long as they are still uniquely decodable. A code is uniquely decodable if no two strings have same encoding when we apply the code to each of their characters.

For example, reversing each codeword in a prefix-free code produces a code which is is still uniquely decodable, but may no longer be prefix-free.

Fortunately, the following two results imply that we have nothing to lose by continuing to consider only prefix-free codes:

Theorem (Kraft, 1949)

There exists a prefix-free code with codeword lengths $d_1, \ldots, d_n$ if and only if $\sum_i \frac{1}{2^{d_i}} \leq 1$.

Theorem (McMillan, ????)

There exists a uniquely decodable code with codeword lengths $d_1, \ldots, d_n$ if and only if $\sum_i \frac{1}{2^{d_i}} \leq 1$.

Since the characters in a normal news report aren't chosen independently and according to a fixed distribution, Shannon's lower bound doesn't hold. In fact, even pretending they are, we can still avoid using nearly a whole extra bit per character using a technique called arithmetic coding (invented by a Finn!).

This is why you should be very careful of saying data compression schemes are "optimal".

Shannon's and Huffman's results concern lossless compression. We'll avoid discussing lossy compression in this course because in order to do so, first we should agree on a loss function, which can get messy. For example, choosing the right loss function for compressing sound files is more a question of psycho-acoustics than of mathematics.

Exercise:

Modify Shannon's proof to show that even if the encodings of X's possible values must be in a certain lexicographic order, then we can still achieve expected message length less than $H(P) + 2$.