

Lecture 1: Entropy and Source Coding

September 25, 2006

1 Entropy

Let X be a discrete random variable with *alphabet* $\mathcal{X} = \{1, 2, \dots, m\}$. Assume there is a probability mass function $p(x)$ over \mathcal{X} . How many binary questions, on average, does it take to determine the outcome?

Definition 1.1: The *entropy* of a discrete random variable X is defined as:

$$H(X) = \sum_{x \in \mathcal{X}} p(x) \log \frac{1}{p(x)}$$

which can be interpreted as the expected value

$$H(X) = \mathbb{E}_p \left[\log \frac{1}{p(x)} \right].$$

2 Source Coding

Definition 2.1: A (binary) *source code* C for a random variable X is a mapping from \mathcal{X} to a (finite) binary string. Let $C(x)$ be the *codeword* corresponding to x and let $l(x)$ denote the length of $C(x)$.

We focus on codes that are “instantaneous”.

Definition 2.2: A code is called a *prefix code* or an instantaneous code if no codeword is a prefix of any other codeword.

The nice property of a prefix code is that one can transmit multiple outcomes x_1, x_2, \dots, x_n by just concatenating the strings into $C(x_1)C(x_2) \dots C(x_n)$, where the latter denotes the concatenation of $C(x_1), C(x_2)$ up to $C(x_n)$, and this leads to decoding x_i instantly after x_i is received. In this sense, prefix codes are “self punctuating”.

Let the expected length of C be:

$$L(C) = \sum_{x \in \mathcal{X}} p(x) l(x)$$

Theorem 2.3: *The expected length of any (prefix) code is greater than the entropy, i.e.*

$$L(C) \geq H(X)$$

Furthermore, there exists a code such that

$$L(C) \leq H(X) + 1$$

This theorem is actually more general and applies to uniquely extendable codes.

2.1 The Kraft Inequality

Theorem 2.4: (*Kraft Inequality*) *Any prefix code satisfies:*

$$\sum_{x \in \mathcal{X}} 2^{-l(x)} \leq 1$$

Conversely, given a set of codeword lengths which satisfy this inequality, then there exists a prefix code with these lengths.

Proof: Consider flipping (unbiased) coins until either we have a codeword or no codeword is possible. This process will terminate as the codewords are of finite length. Furthermore, as the codewords are a prefix code, the process will terminate instantly upon obtaining a codeword.

Hence

$$\begin{aligned} \Pr[\text{obtaining a codeword}] &= \sum_{x \in \mathcal{X}} \Pr[\text{codeword } x] \\ &= \sum_{x \in \mathcal{X}} 2^{-l(x)} \\ &\leq 1 \end{aligned}$$

where the last step follows since probabilities are bounded by 1. This proves the first statement.

We proved the forward direction with a technique known as the “probabilistic method”. For the converse, order the lengths in ascending order l_1 to l_m . Pick code words in this order subject to the constraint that any previous codeword is not a prefix of the selected code. To prove that this works, consider a full binary tree of depth l_m . Associate each codeword with a path on the tree — from the root to some internal node (the end node of the codeword). The prefix condition states that the path of each codeword must not contain an endpoint of another codeword’s path. With each leaf node, associate a probability mass of 2^{-l_m} . Assign the probability mass of codeword i as 2^{-l_i} . Note that each codeword removes 2^{-l_i} from the remaining mass to be allocated. Furthermore, allocation is always possible, if there is enough remaining mass, by the prefix condition. As the lengths satisfy the Kraft inequality, there is enough initial mass (of 1) to assign all the items to valid codewords.

2.2 The proof of the source coding theorem

We first show that there exists a code within one bit of the entropy. Choose the lengths as:

$$l(x) = \lceil \log \frac{1}{p(x)} \rceil$$

This choice is integer and satisfies the Kraft inequality, hence there exists a code. Also, we can upper bound the average code length as follows:

$$\begin{aligned} \sum_{x \in \mathcal{X}} p(x)l(x) &= \sum_{x \in \mathcal{X}} p(x) \lceil \log \frac{1}{p(x)} \rceil \\ &\leq \sum_{x \in \mathcal{X}} p(x) (\log \frac{1}{p(x)} + 1) \\ &= H(X) + 1 \end{aligned}$$

Now, let us prove the lower bound on $L(C)$. Consider the optimization problem

$$\min_{l(x)} \sum_{x \in \mathcal{X}} p(x)l(x) \quad \text{such that} \quad \sum_{x \in \mathcal{X}} 2^{-l(x)} \leq 1$$

The above finds the shortest possible code length subject to satisfying the Kraft inequality. If we relax the the codelengths to be non-integer, then we can obtain a lower bound.

To do this, the Lagrangian is:

$$\mathcal{L} = \sum_{x \in \mathcal{X}} p(x)l(x) + \lambda (\sum_{x \in \mathcal{X}} 2^{-l(x)} - 1)$$

Taking derivatives with respect to $l(x)$ and λ and setting to 0, leads to:

$$\begin{aligned} p(x) + \ln 2 \lambda 2^{-l(x)} &= 0 \\ \sum_{x \in \mathcal{X}} 2^{-l(x)} - 1 &= 0 \end{aligned}$$

Solving this for $l(x)$ leads to $l(x) = \log \frac{1}{p(x)}$, which can be verified by direct substitution. This proves the lower bound.