Information and Coding Theory **Autumn 2022**

Lecture 2: September 29, 2022

Lecturer: Madhur Tulsiani

1 Source Coding

We will now attempt to make precise the intuition that a random variable X takes $H(X)$ bits to describe on average. We shall need the notion of prefix-free codes as defined below.

Definition 1.1. *A code for a set $\mathcal{X}$ over an alphabet Σ is a map $C : \mathcal{X} \to \Sigma^*$ which maps each element of $\mathcal{X}$ to a finite string over the alphabet Σ. We say that a code is* prefix-free *if for any $x, y \in \mathcal{X}$ such that $x \neq y$, $C(x)$ is not a prefix of $C(y)$ i.e., $C(y) \neq C(x) \circ \sigma$ for any $\sigma \in \Sigma^*$.*

For now, we will just use $\Sigma = \{0, 1\}$. For the rest of lecture, we will use prefix-free code to mean prefix-free code over $\{0, 1\}$. The image $C(x)$ for an image x is also referred to as the *codeword* for x.

Note that a prefix-free code has the convenient property that if we are receiving a stream of coded symbols, we can decode them online. As soon as we see $C(x)$ for some $x \in U$, we know what we have received so far cannot be a prefix for $C(y)$, for any $y \neq x$. The following inequality gives a characterization of the lengths of codewords in a prefix-free code. This will help prove both upper and lower bounds on the expected length of a codeword in a prefix-free code, in terms of entropy.

Proposition 1.2 (Kraft's inequality)**.** *Let $|\mathcal{X}| = n$. There exists a prefix-free code for $\mathcal{X}$ over $\{0, 1\}$ with codeword lengths $\ell_1, \ldots, \ell_n$ if and only if*

$$\sum_{i=1}^{n} \frac{1}{2^{\ell_i}} \leq 1 .$$

For codes over a larger alphabet Σ, we replace 2^{ℓ_i} above by $|\Sigma|^{\ell_i}$.

Proof: Let us prove the "if" part first. Given $\ell_1, \ldots, \ell_n$ satisfying $\sum_i 2^{-\ell_i} \leq 1$, we will construct a prefix-free code C with these codeword lengths. Without loss of generality, we can assume that $\ell_1 \leq \ell_2 \leq \cdots \leq \ell_n = \ell^*$.

It will be useful here to think of all binary strings of length at most ℓ as a complete binary tree. The root corresponds to the empty string and each node at depth d corresponds to a string of length d. For a node corresponding to a string s, its left and right children correspond respectively to the strings $s0$ and $s1$. The tree has 2^{ℓ^*} leaves corresponding to all strings in $\{0,1\}^{\ell^*}$.

We will now construct our code by choosing nodes at depth $\ell_1, \ldots, \ell_n$ in this tree. When we select a node, we will delete the entire tree below it. This will maintain the prefix-free property of the code. We first chose an arbitrary node s_1 at depth ℓ_1 as a codeword of length ℓ_1 and delete the subtree below it. This deletes $1/2^{\ell_1}$ fraction of the leaves. Since there are still more leaves left in the tree, there exists a node (say s_2) at depth ℓ_2. Also, s_1 cannot be a prefix of s_2, since s_2 does not lie in the subtree below s_1. We choose s_2 as the second codeword in our code C. We can similarly proceed to choose other codewords. At each step, we have some leaves left in the tree since $\sum_i 2^{-\ell_i} \leq 1$.

Note that we need to carry out this argument in increasing order of lengths. Otherwise, if we choose longer codewords first, we may have to choose a shorter codeword later which does not lie on the path from the root to any of the longer codewords, and this may not always possible e.g., there exists a code with lengths $1, 2, 2$ but if we choose the strings 01 and 10 first then there is no way to choose a codeword of length 1 which is not a prefix.

For the "only if" part, we can simply reverse the above proof. Let C be a given prefix-free code with codeword lengths $\ell_1, \ldots, \ell_n$ and let $\ell^* = \max\{\ell_1, \ldots, \ell_n\}$. Considering again the complete binary tree of depth ℓ^*, we can now locate the codewords (say) $C(x_1), \ldots, C(x_n)$ as nodes in the tree. We say that a codeword $C(x)$ *dominates* a leaf L if L occurs in the subtree rooted at $C(x)$. Note that the out of the total 2^{ℓ^*} fraction of leaves dominated by a codeword of length ℓ_i is $2^{-\ell_i}$. Also, note that if $C(x)$ and $C(y)$ dominate the same leaf L, then either $C(x)$ appears in the subtree rooted at $C(y)$ or vice-versa. Since the code is prefix-free, this cannot happen and the sets of leaves dominated by codewords must be disjoint. Thus, we have $\sum_i 2^{-\ell_i} \leq 1$.

This part of the proof also has a probabilitic interpretation. Consider an experiment where we generate ℓ^* random bits. For $x \in \mathcal{X}$, let E_x denote the event that the *first* $|C(x)|$ bits we generate are equal to $C(x)$. Note that since C is a prefix-free code, E_x and E_y are mutually exclusive for $x \neq y$. Moreover, the probability that E_x happens is exactly $1/2^{|C(x)|}$. This gives

$$1 \geq \sum_{x \in \mathcal{X}} \mathbb{P}[E_x] = \sum_{x \in \mathcal{X}} \frac{1}{2^{|C(x)|}} = \sum_{i=1}^{n} \frac{1}{2^{\ell_i}}.$$

■

We will show that the concept of entropy, defined in the previous lecture, provides a lower bound on the expected length of any prefix free code. In particular, we will now show that *any* prefix-free code for communicating the value of a random variable X must use at least $H(X)$ on average.

Claim 1.3. *Let X be a random variable taking values in $\mathcal{X}$ and let $C : \mathcal{X} \to \{0,1\}$ be a prefix-free code. Then the expected number of bits used by C to communicate the value of X is at least $H(X)$.*

Proof: The expected number of bits used is $\sum_{x \in \mathcal{X}} p(x) \cdot |C(x)|$. We consider the quantity

$$\begin{aligned} H(X) - \sum_{x \in \mathcal{X}} p(x) \cdot |C(x)| &= \sum_{x \in \mathcal{X}} p(x) \cdot \left(\log\left(\frac{1}{p(x)} \right) - |C(x)| \right) \\ &= \sum_{x \in \mathcal{X}} p(x) \cdot \log\left(\frac{1}{p(x) \cdot 2^{|C(x)|}} \right) . \end{aligned}$$

We consider a random variable Y with takes the value $\frac{1}{p(x) \cdot 2^{|C(x)|}}$ with probability $p(x)$. The above expression then becomes $\mathbb{E}[\log(Y)]$. Using Jensen's inequality gives

$$\mathbb{E}[\log(Y)] \leq \log(\mathbb{E}[Y]) = \log\left(\sum_{x \in \mathcal{X}} p(x) \cdot \frac{1}{p(x) \cdot 2^{|C(x)|}} \right) = \log\left(\sum_{x \in \mathcal{X}} \frac{1}{2^{|C(x)|}} \right)$$

which is non-positive since $\sum_{x \in U} \frac{1}{2^{|C(x)|}} \leq 1$ by Kraft's inequality. ■

The Shannon code: We now construct a (prefix-free) code for conveying the value of X, using at most $H(X) + 1$ bits on average (over the distribution of X). For an element $x \in \mathcal{X}$ which occurs with probability $p(x)$, we will use a codeword of length $\lceil \log(1/p(x)) \rceil$. By Kraft's inequality, there exists a prefix-free code with these codeword lengths, since

$$\sum_{x \in \mathcal{X}} \frac{1}{2^{|C(x)|}} = \sum_{x \in \mathcal{X}} \frac{1}{2^{\lceil \log(1/p(x)) \rceil}} \leq \sum_{x \in \mathcal{X}} \frac{1}{2^{\log(1/p(x))}} = \sum_{x \in \mathcal{X}} p(x) = 1 .$$

Also, the expected number of bits used is

$$\sum_{x \in \mathcal{X}} p(x) \cdot \lceil \log(1/p(x)) \rceil \leq \sum_{x \in \mathcal{X}} p(x) \cdot (\log(1/p(x)) + 1) = H(X) + 1 .$$

This code is known as the Shannon code.

2 Joint Entropy

We have two random variables X and Y. The joint distribution of the two random variables (X, Y) takes values (x, y) with probability $p(x, y)$. Merely by using the definition, we can write down the entropy of $Z = (X, Y)$ trivially. However what we are more interested in is seeing how the entropy of (X, Y), the joint entropy, relates to the individual entropies, which we work out below:

$$\begin{aligned}
H(X,Y) &= \sum_{x,y} p(x,y) \log \frac{1}{p(x,y)} \\
&= \sum_{x,y} p(x)p(y|x) \log \frac{1}{p(x)} + \sum_{x,y} p(x)p(y|x) \log \frac{1}{p(y|x)} \\
&= \sum_{x} p(x) \log \frac{1}{p(x)} \sum_{y} p(y|x) + \sum_{x,y} p(x)p(y|x) \log \frac{1}{p(y|x)} \\
&= H(X) + \sum_{x} p(x) H(Y|X=x) \\
&= H(X) + \mathop{\mathbb{E}}_{x} \left[H(Y|X=x) \right]
\end{aligned}$$

Denoting $\mathbb{E}_x \left[H(Y|X=x) \right]$ as $H(Y|X)$, this can simply be written as

$$H(X,Y) = H(X) + H(Y|X)$$

If we were to redo the calculations, we could similarly obtain:

$$H(X,Y) = H(Y) + H(X|Y)$$

This is called the *Chain Rule* for Entropy. Note that in the calculations above, we treat $(Y|X=x)$ as a random variable, with distribution given by $\mathbb{P}\left[Y=y \mid X=x\right] = p(y|x)$. Also note that $H(Y|X)$ is a simply a shorthand for the *expected* entropy of $(Y|X=x)$, with the expectation taken over the values for X.

Example 2.1. *Consider the random variable (X,Y) with $X \vee Y = 1$ and $X \in \{0,1\}$ and $Y = \{0,1\}$ such that:*

$$(X,Y) = \begin{cases} 01 & \textit{with probability 1/3} \\ 10 & \textit{with probability 1/3} \\ 11 & \textit{with probability 1/3} \end{cases}$$

Now, let us calculate the following:

1. $H(X) = H(Y) = \frac{1}{3}\log 3 + \frac{2}{3}\log\frac{3}{2}$
2. $H(Y|X=0) = 0$
3. $H(Y|X=1) = \frac{1}{2}\log\frac{1}{\frac{1}{2}} + \frac{1}{2}\log\frac{1}{\frac{1}{2}} = 1$
4. $H(Y|X) = \frac{1}{3}\cdot 0 + \frac{2}{3}\cdot 1 = \frac{2}{3}$
5. $H(X,Y) = \frac{1}{3}\log 3 + \frac{1}{3}\log 3 + \frac{1}{3}\log 3 = \log 3$

From the above we see that:

$$H(Y) \geq H(Y|X)$$

this is actually *always* true and we prove this fact below.

Proposition 2.2. $H(Y) \geq H(Y|X)$

Proof: We want to show that $H(Y|X) - H(Y) \leq 0$. Consider the quantity on the left hand side.

$$\begin{aligned}
H(Y|X) - H(Y) &= \sum_x p(x) \sum_y p(y|x) \log \frac{1}{p(y|x)} - \sum_y p(y) \log \frac{1}{p(y)} \\
&= \sum_x p(x) \sum_y p(y|x) \log \frac{1}{p(y|x)} - \sum_y p(y) \log \frac{1}{p(y)} \sum_x p(x|y) \\
&= \sum_{x,y} p(x,y) \left(\log \frac{1}{p(y|x)} - \log \frac{1}{p(y)} \right) \\
&= \sum_{x,y} p(x,y) \left(\log \frac{p(x)p(y)}{p(x,y)} \right)
\end{aligned}$$

Now consider a random variable Z that takes value $\frac{p(x)p(y)}{p(x,y)}$ with probability $p(x,y)$. Then we can use Jensen's inequality to get:

$$\sum_{x,y} p(x,y) \left(\log \frac{p(x)p(y)}{p(x,y)} \right) \leq \log \left(\sum_{x,y} \frac{p(x)p(y)}{p(x,y)} p(x,y) \right) = \log(1) = 0\,.$$

■

Note however the fact that conditioning on X reduces the entropy of Y is only true *on average over all fixings of* X. In particular, in the above example we have $H(Y|X=1) = 1 > H(Y)$. But $H(Y|X)$, which is an average over all fixings of X, is indeed smaller than $H(Y)$. Also, check that above inequality is tight only when X and Y are independent.

Exercise 2.3. *Show that* $H(Y) = H(Y|X)$ *if and only if* X *and* Y *are independent.*

Using induction, we can use the chain rule to show that the following also holds for a tuple of random variables $(X_1, \ldots, X_m)$.

$$H(X_1, X_2, \ldots, X_m) \;=\; H(X_1) + H(X_2|X_1) + H(X_3|X_1, X_2) \ldots H(X_m|X_1, \ldots, X_{m-1})\,.$$

Combining this with the fact that conditioning (on average) reduces the entropy, we get the following inequality which is referred to the sub-additivity property of entropy.

$$H(X_1, X_2, \ldots, X_m) \;\leq\; H(X_1) + H(X_2) + H(X_3) + \cdots + H(X_m)\,.$$