EE376A Information Theory Lecture 8 - 2/3/2015

Lecture 8

Lecturer: Tsachy Weissman *Scribe: Jeremy Kim, Michael P. Kim, Neha Nayak*

1 Wrapping up the Optimality of Huffman Codes

Recap:
Suppose we have a random object $U \sim P$ with alphabet $\mathcal{U} = \{1, 2, \ldots, r\}$. We let $c(u), \ell(u)$ be the codeword and length associated with some $u \in \mathcal{U}$, respectively. Recall, the average length is denoted as $\overline{\ell} = E\ell(U)$. Assume, WLOG, that the symbols in $\mathcal{U}$ are ordered in decreasing order of probability according to p, i.e. $p(1) \geq p(2) \geq \cdots \geq p(r)$. Let U_{r-1} denote a random variable over $\mathcal{U}_{r-1} = \{1, 2, \ldots, r-1\}$ and

$$p(U_{r-1} = i) = \begin{cases} p(i) & 1 \leq i \leq r-2 \\ p(r-1) + p(r) & i = r-1 \end{cases}$$

and, as before, $c_{r-1}(u_{r-1}), \ell_{r-1}(u_{r-1})$ are the codeword and length of $u_{r-1} \in \mathcal{U}_{r-1}$, respectively. Again, $\overline{\ell}_{r-1} = E\ell_{r-1}(U_{r-1})$.

"Splitting a prefix code c_{r-1}": creating a prefix code for U by

$$\begin{cases} c(i) = c_{r-1}(i) & 1 \leq i \leq r-2 \\ c(r-1) = c_{r-1}(r-1)0 & \\ c(r) = c_{r-1}(r-1)1 & \end{cases}$$

We will use the following lemma to justify the optimality of Huffman Codes. Intuitively, we will show that if we start with an optimal code on $r-1$ symbols, splitting gives us an optimal code over r symbols. We can use an inductive argument, starting with a binary object to prove that Huffman Codes are optimal for alphabets with any number of symbols.

Lemma 1. *Let $c_{opt,r-1}$ be an optimal prefix code for U_{r-1}. Let c be the code obtained from $c_{opt,r-1}$ by splitting. Then c is an optimal prefix code for U.*

Proof Note there is an optimal prefix code for U satisfying:

1. $\ell(1) \leq \ell(2) \leq \ldots \leq \ell(r)$
2. $\ell(r-1) = \ell(r) = \ell_{\max}$ [1]
3. $c(r-1)$ and $c(r)$ differ only in the last bit [2]

[1] Suppose in an optimal code the two longest codewords were not of the same length. Since the prefix property holds, no codeword is a prefix of the longest codeword. The longest codeword can be truncated, preserving the prefix property but achieving lower expected codeword length. Since the code was optimal, this leads to a contradiction, so the two longest codewords must be of the same length.

[2] Given any optimal code, rearranging the code can result in a code with this property. Suppose there is a codeword of maximal length such that it does not have a 'sibling'. The last bit of this codeword can be deleted, preserving the prefix property but achieving lower expeted codeword length. This leads to a contradiction, so every codeword of maximal length must have a 'sibling' in an optimal code. By rearranging the assignments of the maximal length codewords, we can ensure that the two least likely symbols are assigned a pair of sibling codewords. This rearrangement maintains the expected codeword length, and achieves property 3.

Note, by the second and third properties, there is an optimal prefix code for U that is the result of splitting a prefix code for U_{r-1} (where the code for U_{r-1} can be seen as our original code with the the codeword $c(r-1)$ is truncated before the final bit). Let ℓ_{r-1} be the length function of a prefix code for U_{r-1} and ℓ be the length function of a prefix code for U obtained by splitting U_{r-1}. Then

$$\begin{aligned}
\overline{\ell} &= \sum_{i=1}^{r} p(i)l(i) \\
&= \sum_{i=1}^{r-2} p(i)l(i) + p(r-1)l(r-1) + p(r)l(r) \\
&= \sum_{i=1}^{r-2} p(U_{r-1}=i)l_{r-1}(i) + (p(r-1)+p(r))l(r-1) \\
&= \sum_{i=1}^{r-2} p(U_{r-1}=i)l_{r-1}(i) + (p(r-1)+p(r))(l_{r-1}(r-1)+1) \\
&= \sum_{i=1}^{r-2} p(U_{r-1}=i)l_{r-1}(i) + p(U_{r-1}=r-1)l_{r-1}(r-1) + (p(r-1)+p(r)) \\
&= \sum_{i=1}^{r-1} p(U_{r-1}=i)l_{r-1}(i) + (p(r-1)+p(r)) \\
&= \overline{\ell}_{r-1} + p(r-1) + p(r)
\end{aligned}$$

$$\overline{\ell} = \overline{\ell}_{r-1} + p(r-1) + p(r)$$

because the expectation sums differ only in the final two terms, where an additional bit is added for symbols $r-1$ and r. We can see, by this simple relationship, that if we want to optimize $\overline{\ell}$ for some fixed probability distribution, it suffices to optimize $\overline{\ell}_{r-1}$. So if $\overline{\ell}_{r-1}$ is optimal, then so is $\overline{\ell}$. Thus, we have that an optimal prefix code for U is obtained by splitting an optimal prefix code for U_{r-1}.

□

Theorem 2 (Kraft-McMillan Inequality)**.** *For all uniquely decodable (UD) codes,*

$$\sum_{u \in \mathcal{U}} 2^{-\ell(u)} \leq 1 \tag{1}$$

Conversely, any integer-valued function satisfying (1) is the length function of some UD code.

To see the "conversely" statement, note that we know how to generate a UD code (in fact, a prefix code) with length function satisfying (1), using Huffman Codes. Here, we prove the first claim of the Kraft-McMillan Inequality.

Proof Take any UD code and let $\ell_{\max} = \max_u \ell(u)$. Fix any integer k and note the following.

$$\begin{aligned}
\left(\sum_{u\in\mathcal{U}} 2^{-\ell(u)}\right)^k &= (\sum_{u_1} 2^{-\ell(u_1)}) \cdot (\sum_{u_2} 2^{-\ell(u_2)}) \cdot \ldots \cdot (\sum_{u_k} 2^{-\ell(u_k)}) \\
&= \sum_{u_1}\sum_{u_2}\cdots\sum_{u_k}\prod_{i=1}^{k} 2^{-\ell(u_i)} \\
&= \sum_{(u_1,\ldots,u_k)} 2^{-\sum_{i=1}^k \ell(u_i)} \\
&= \sum_{u^k} 2^{-\ell(u^k)} \\
&= \sum_{i=1}^{k\cdot\ell_{\max}} \left|\left\{u^k \;\middle|\; \ell(u^k) = i\right\}\right| \cdot 2^{-i} \\
&\leq \sum_{i=1}^{k\cdot\ell_{\max}} 2^i \cdot 2^{-i} \\
&= k \cdot \ell_{\max}
\end{aligned}$$

Note that the inequality in the second to last line arises because we know that our code is one-to-one so there can be at most 2^i symbols whose codewords have length i. Finally, we can see the theorem through the following inequality.

$$\sum_u 2^{-\ell(u)} \leq \lim_{k\to+\infty} (k\ell_{\max})^{1/k} = 1$$

□

Now, we can prove the important theorem relating UD codes to the binary entropy.

Theorem 3. *For all UD codes,* $\overline{\ell} \geq H(U)$

Proof

$$\begin{aligned}
H(U) - \overline{\ell} &= H(U) - E\ell(U) \\
&= E\left[\log\frac{1}{p(U)} - \ell(U)\right] \\
&= E\left[\log\frac{1}{p(U)} + \log 2^{-\ell(U)}\right] \\
&= E\left[\log\frac{2^{-\ell(U)}}{p(U)}\right] \\
&\leq \log E\left[\frac{2^{-\ell(U)}}{p(U)}\right] \\
&= \log\sum_u p(u)\cdot\frac{2^{-\ell(u)}}{p(u)} \\
&= \log\sum_u 2^{-\ell(u)} \\
&\leq \log 1 = 0
\end{aligned}$$

□

Note: suppose $\ell(u) = -\log q(u)$ for some pmf q. Then,

$$\begin{aligned}\overline{\ell} - H &= E[\ell(U) - \log \frac{1}{p(u)}] \\ &= \sum_u p(u) \log \frac{p(u)}{q(u)} \\ &= D(p\|q)\end{aligned}$$

Thus, $D(p\|q)$ can be thought of as the "cost of mismatch", in designing a code for a distribution q, when the actual distribution is p.