EE376A/STATS376A Information Theory Lecture 7 - 1/30/18

Lecture 7: Huffman Codes, Communication and Channel Capacity

Lecturer: Tsachy Weissman *Scribe: Noah Gamboa, Zhaolin Ren*

In this lecture, we will first prove the optimality of Huffman encoding. Then, we will move into the new topic of Communication and Channel Capacity, where we introduce the notion of communication, define terms that enable us to model the problem, and lay out connections that will inform how to approach communication problems.

1 Optimality of Huffman Encoding

Theorem 1. *Suppose $U \sim P$, $\mathcal{U} = \{1, 2, \ldots, r\}$. The Huffman code for U is an optimal prefix code.*

We restate some setup from Lecture 6. Without loss of generality, we may assume that

$$p(1) \geq p(2) \geq \cdots \geq p(r),$$

since we can always relabel the symbols such that this is satisfied.

In addition, let V denote the random variable with source $\mathcal{V} = \{1, 2, \ldots, r-1\}$, which is obtained from U by merging the symbols $r-1$ and r, and let $\{c(i)\}_{i=1}^{r-1}$ be a prefix code for V. As we saw in Lecture 6, we can obtain a prefix code $\{\tilde{c}(i)\}_{i=1}^{r}$ for U by splitting the last codeword, where

$$\tilde{c}(i) = \begin{cases} c(i) & i = 1, 2, \ldots, r-2 \\ c(i)0 & i = r-1 \\ c(i)1 & i = r \end{cases}$$

Now, let $\ell(\cdot)$ denote the length function of $\{c(i)\}_{i=1}^{r-1}$, and $\ell_{split}(\cdot)$ denote the length function of $\{\tilde{c}(i)\}_{i=1}^{r}$. As we proved at the end of Lecture 6, we have that

$$E[\ell_{split}(U)] = E[\ell(V)] + p(r-1) + p(r) \tag{1}$$

We will make use of Equation 1 later in proving Theorem 1.

To prove the theorem, we first begin with an observation and the statement of a key lemma.

Observation 2. *The Huffman code for U is obtained by splitting the Huffman code for V.*

Lemma 3. *Suppose $\{c(i)\}_{i=1}^{r-1}$ is an optimal prefix code for V. If $\{\tilde{c}(i)\}_{i=1}^{r}$ is obtained from $\{c(i)\}_{i=1}^{r-1}$ by splitting, then $\{\tilde{c}(i)\}_{i=1}^{r}$ is optimal for U.*

We note that Observation 2 and Lemma 3 reduce the problem of showing optimality of the Huffman code for U to establishing optimality of the Huffman code construction for V. Thus, by iterating this argument, we merely need to establish optimality of Huffman code in the binary source case, where $r = 2$, which is trivially true. Hence,

Observation 2 + Lemma 3 $\Rightarrow$ Theorem 1

Therefore, in order to prove Theorem 1, it suffices to prove Lemma 3.

Proof of Lemma

To prove the lemma, we first argue that there is an optimal prefix code for U satisfying the following three properties:

1. $\tilde{l}(1) \leq \tilde{l}(2) \leq \cdots \leq \tilde{l}(r-1) \leq \tilde{l}(r) \triangleq \tilde{l}_{max}$.

   Otherwise, since $p(1) \geq p(2) \geq \cdots \geq p(r)$, we can always switch the ordering to achieve a lower expected code length.

2. $\tilde{l}(r-1) = \tilde{l}(r) = \tilde{l}_{max}$.

   To see why this must be the case, suppose for contradiction that $\tilde{l}(r) > \tilde{l}(r-1)$. Since the code satisfies the prefix property, no other codeword is a prefix of the codeword for r, $c(r)$, and so we can truncate the last bits of $c(r)$ such that $\tilde{l}(r-1) = \tilde{l}(r)$ and still preserve the prefix property of the code. However, this modified code will have lower expected code length, contradicting optimality of the unmodified prefix code. Therefore the two least likely codewords must have the same length.

3. **Without loss of generality, we may assume that the two least likely codewords differ only in the last bit.**

   Given any optimal prefix code, we can modify the code to result in a optimal prefix code with this property. We say that two codewords are *siblings* if they differ only in their last bit. If no codeword is the sibling of the codeword for r, then we can assign the sibling to $r-1$ and we are done. If some codeword other than $c(r-1)$ is the sibling of $c(r)$, then we can swap the codewords to ensure that the property holds.

Summarizing, we showed that there exists an optimal prefix code for U that satisfies the three properties above. Combining properties 2 and 3, we see that there exists an optimal prefix code for U that is obtained by splitting some prefix code for V.

As we saw in Equation (1) earlier, if we let $\ell(\cdot)$ denote the length function of some arbitrary prefix code $\{d(i)\}_{i=1}^{r-1}$ for U, and $\ell_{split}(\cdot)$ denote the length function of $\left\{\tilde{d}(i)\right\}_{i=1}^{r}$ which is the code for U as a result of splitting $\{d(i)\}_{i=1}^{r-1}$, we have that

$$E[\ell_{split}(U)] = E[\ell(V)] + p(r-1) + p(r).$$

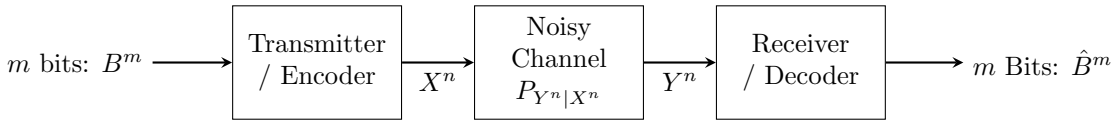
Since $p(r-1)$ and $p(r)$ are constants given a fixed probablity mass function for the source symbols, the above relationship implies that in order to optimize $\mathrm{E}[\ell_{split}(U)]$, it suffices to optimize $\mathrm{E}[\ell(V)]$. Thus, letting $\{c(i)\}_{i=1}^{r-1}$ denote an optimal prefix code for V, if $\{\tilde{c}(i)\}_{i=1}^{r}$ is obtained from $\{c(i)\}_{i=1}^{r-1}$ by splitting, then out of all possible prefix codes for U which are a result of splitting some prefix code for V, $\{\tilde{c}(i)\}_{i=1}^{r}$ has minimal expected code length. Since we know that there exists an optimal prefix code for U that is obtained by splitting some prefix code for V, this implies that $\{\tilde{c}(i)\}_{i=1}^{r}$ must be an optimal prefix code. This completes our proof of Lemma 3, and by extension, proves the optimality of Huffman encoding. □

2 Further Reading on Lossless Compression

1. **Shannon-Fano-Elias Coding** Section 5.9 of Cover & Thomas
2. **Arithmetic Coding** Section 13.3 of Cover & Thomas
3. **Lempel-Ziv Coding** section 13.4 of Cover & Thomas - used for gzip!

3 Communication and Channel Capacity

We want to communicate bits from some source to some target. We use the following model to represent how to transmit this data across a channel that can have noise that corrupts the data. In order to ensure good transmission quality, we will try to encode and decode the data in such a way that reduces the probability of error in the received signal.



We have $B^m = (B_1, B_2, \ldots, B_m)$, and $\hat{B}^m = (\hat{B}_1, \hat{B}_2, \ldots, \hat{B}_m)$. Notice that m is not necessarily equal to n.

Here, let $B_1, B_2, ..., B_m$ be i.i.d. bits $\sim Ber(\frac{1}{2})$. The conditional distribution of the signal which the noisy channel emits given the transmitted signal, $P_{Y^n|X^n}(y^n|x^n)$, is given. In order to transmit B^m across the channel, we first encode these bits into a new vector, $X_1, X_2, ..., X_n$. This is the information that is sent across the noisy channel. Then, the receiver will receive the transformed vector, $Y_1, Y_2, ..., Y_n$. The decoder's job is now to to transform the received bits into a vector $\hat{B}^m$ that closely resembles B^m.

To work with this problem, we first need to make some definitions.

Definition 4. *Scheme:*

$$Scheme \triangleq (m, n, encoder, decoder) \tag{2}$$

A scheme is first characterized by the number of bits we are trying to send (m) and the number of channel uses (n). Once we have picked these, we pick an encoder, that changes the (m) bits into an encoded (n)-tuple. Then, we pick the decoder that converts encoded (n)-tuple into, hopefully, the same (m) bits that were sent.

Definition 5. *Rate:*

$$rate \triangleq \frac{m}{n} \frac{bits}{channel\ use} \tag{3}$$

The higher the rate, the better we are at communicating our information.

Definition 6. *Probability of Error:*

$$P_e^{(n)} \triangleq P(\hat{B}^m \neq B^m) \tag{4}$$

Definition 7. *Achievable Rate: R is an achievable rate if there exists a sequence of schemes $\{Scheme_n\}_{n\geq 1}$ with rate equal to or greater than R that transmits with vanishing probability of error, such that*

$$P_{e,Scheme_n}^{(n)} \xrightarrow{n\to\infty} 0 \tag{5}$$

Note that m has to be scaling and growing with n as it goes to infinity.

With the notion of achievable rate, we can talk about channel capacity.

Definition 8. *Channel Capacity: this is the maximal rate of reliable communication:*

$$C \triangleq \sup\{R | R\ is\ achievable\} \tag{6}$$

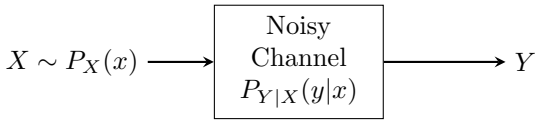
All these definitions are valid for any kind of channel. However, to get started, we choose to work with simpler channels. This brings us to the notion of a memoryless channel, which is a surprisingly common assumption to make in communication.

Definition 9. *Memoryless Channel: The conditional distribution of the output given the input is the product of the same conditional distribution of an output symbol given the input symbol, for every bit sent or received. In other words,*

$$P_{Y^n|X^n}(y^n|x^n) \triangleq \prod_{i=1}^{n} P_{Y|X}(y_i|x_i) \tag{7}$$

A memoryless channel corresponds to n independent uses of a channel that transmits a single symbol.

Now let's consider the "single-letter" channel.



Given the input distribution on X and the conditional distribution of Y given X, we can compute the joint distribution and quantities like the mutual information between the input and the output. In fact, since we can modify P_X, we can find a way to maximize the mutual information between the input and the output. Intuitively, maximizing the mutual information $I(X;Y)$ minimizes the error in the channel because it increases the amount by which the input informs the output. Consider then the following quantity, obtained by taking a maximum over the probability distributions of X:

$$C^{(I)} \triangleq \max_{P_X} I(X;Y) \tag{8}$$

Theorem 10. *The maximum mutual information is the channel capacity*

$$C = C^{(I)} \tag{9}$$

This is profound because it relates how much we can physically transmit over a channel reliably to the mutual information between input and output. This makes the complicated problem of finding channel capacity a clean optimization problem which involves finding the input distribution that maximizes the mutual information between the input and the output.