EE376A Information Theory Lecture 6 - 01/27/2015

Lecture 6: Lossless Compression

Lecturer: Tsachy Weissman *Scribe: Seungmin Lee, Brian Do, Cody Peterson*

1 Variable Length Lossless Compression (Ch. 5)

Last lecture, we talked about how using the AEP, entropy emerges when you want to describe source symbols in fixed length at nearly lossless compression. In fixed-length compression, you map source sequences to representations 1:1. We also said that if you use variable length coding, there is a way to achieve H bits/source symbol with perfect lossless compression, where H is the entropy. How can we achieve such a code? The next few lectures will be devoted to that question.

Let us start with a simple code. Let $l(u)$ represent the length of a binary codeword representing u, $u \in \mathcal{U}$. We can then write $\bar{l} = El(u) = \sum_{u \in \mathcal{U}} p(u)l(u)$ where $\bar{l}$ is the expected length of a codeword.

Example 1. Let $\mathcal{U} = \{a, b, c, d\}$ and let us try to come up with a simple code for this alphabet.

u	p(u)	Codeword	l(u)
a	1/2	0	1
b	1/4	10	2
c	1/8	110	3
d	1/8	111	3

Figure 1: Code I

Note: here $l(u) = -\log p(u)$
$\Rightarrow \bar{l} = \mathrm{E}[l(u)] = \mathrm{E}[-\log p(u)] = H(u)$

This code satisfies the prefix condition since no codeword is the prefix for another codeword. It also looks like the expected code length is equal to the entropy. Is the entropy the limit for variable length coding? Can we do better? Let us try a better code.

Here is a "better" code, where $\bar{l} < H$

u	Codeword	l(u)
a	0	1
b	1	1
c	1	2
d	11	2

Figure 2: Better code with regard to $l(u)$

However, the code in Figure 2 is not uniquely decodable. For instance, both 'abd' and 'cbb' can be represented by the code 0111. These codes are not useful. This motivates the notion of uniquely decodable schemes.

Definition 2. *A code is uniquely decodable(UD) if every sequence of source symbols is mapped to a distinct binary codeword.*

Definition 3.

Prefix Condition: When no codeword is the prefix of any other.

Prefix Code: A code satisfying the prefix condition.

Codes that satisfy the prefix condition are decodable on the fly. Codes that do not satisfy the prefix condition can also be uniquely decodable, but they are less useful.

Exercise 4. Consider Code II in Figure 3

u	Codeword
a	10
b	00
c	11
d	110

Figure 3: Code II

Prove that this code is UD.
Let us construct binary trees to represent codes. Here, the terminal nodes represent source symbols, and the path from the root to each terminal node represents the codeword for that source symbol. We can construct binary trees for all UD codes, as we will see later.

Here are Binary trees for Code I and Code II:

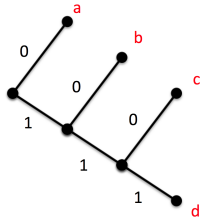

Figure 4: Binary tree for Code I

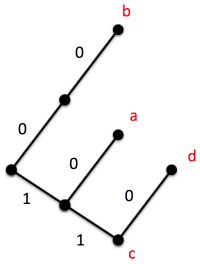

Figure 5: Binary tree for Code II

From here on, let us restrict our attention to prefix codes. In fact, we will see that for any non-prefix code with a given expected code length, we can always find a prefix code with at least as small of a code length.

2 Dyadic Distributions

We would like to systematically construct uniquely decodable prefix codes for any alphabet with arbitrary probability mass functions. We will start with dyadic distributions.

Definition 5. *A dyadic distribution has* $p(u) = 2^{-n_u}$, $\forall u \in \mathcal{U}$, *where* n_u *are integers.* $(*)$

Note: If we find a UD code with $l(u) = n_u = -\log p(u)$, then $l = H$.
We claim that we can always find a UD code for a dyadic distribution.

Lemma 6. *Assume* $(*)$ *and* $n_{max} = \max_{u\in\mathcal{U}} n_u$, *considering that we have a nontrivial distribution (where* $p(u)$ *is not 1 at one value and 0 everywhere else). The number of symbols* u *with* $n_u = n_{max}$ *is even.*

Proof

$$\begin{aligned} 1 &= \sum_{u\in\mathcal{U}} p(u) \\ &= \sum_{u\in\mathcal{U}} 2^{-n_u} \\ &= \sum_{n=1}^{n_{max}} (\#\text{ of symbols } u \text{ with } n_u = n)\cdot 2^{-n} \\ \Rightarrow 2^{n_{max}} &= \sum_{n=1}^{n_{max}} (\#\text{ of symbols } u \text{ with } n_u = n)\cdot 2^{n_{max}-n} \\ &= \sum_{n=1}^{n_{max}-1} (\#\text{ of symbols } u \text{ with } n_u = n)\cdot 2^{n_{max}-n} + (\#\text{ of symbols } u \text{ with } n_u = n_{max}) \end{aligned} \tag{1}$$

Since all terms except (# of symbols u with $n_u = n_{max}$) are even, the number of elements in the alphabet with the smallest probability has to be even. □

Now with this lemma in hand, we can prove our claim that we can find a UD code for a dyadic distribution. Consider the following procedure:

- Choose 2 symbols with $n_u = n_{max}$ and merge them into one symbol with (twice the) probability $2^{-n_{max}+1}$
- The new source distribution is also dyadic.
- Repeat the procedure until left with one symbol.

Note: This procedure induces a binary tree.
E.g.:

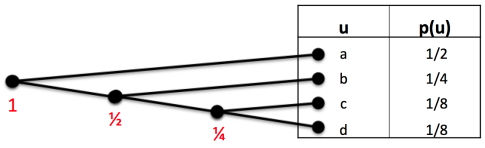

Figure 6: Induced binary tree using the procedure

Also note that the symbol with $p(u) = 2^{-n_u}$ has distance n_u from root. This means that the induced prefix code satisfies $l(u) = n_u = -\log p(u)$

3 General Distribution

How can we get a good code for a non-dyadic distribution? We can attempt to use the above principles. Let us look for a code with

$$l(u) = \lceil -\log p(u) \rceil = n_u^* \quad \forall u \in \mathcal{U}$$

Here, we take the ceiling of $-\log p(u)$ as the length of the codeword for the source symbol u. Because the ceiling of a number is always within 1 of the actual number, the expected code length $\bar{l} = \mathrm{E}[-\log p(u)]$ is within 1 of $H(u)$.

Let us consider the "PMF" $p^*(u) = 2^{-n_u^*}$. This "PMF" is a dyadic distribution because all probabilities are a power of 2. We put PMF in quotes because for a non-dyadic source, $\sum_{u \in \mathcal{U}}^{p^*(u)}$ is less than 1, and so the PMF is not a true PMF. See the following:

$$\begin{aligned} \sum_{u \in \mathcal{U}} p^*(u) &= \sum_{u \in \mathcal{U}} 2^{-n_u^*} = \sum_{u \in \mathcal{U}} 2^{-\lceil -\log p(u) \rceil} \\ &< \sum_{u \in \mathcal{U}} 2^{-(-\log p(u))} = 1 \end{aligned} \tag{2}$$

To make it a true PMF, let us add fictitious source symbols so that $\mathcal{U}^* \supseteq \mathcal{U}$ and $\sum_{u \in \mathcal{U}^*} p^*(u) = 1$. We can thus construct a prefix code for $\mathcal{U}^*$. Because this is a dyadic distribution, we can use the binary tree principle above to construct a code for all $u \in \mathcal{U}^*$, and we can consider only the source symbols u in the original alphabet U. The lengths of each codeword will satisfy

$$l(u) = -\log p^*(u) = n_u^* = \lceil -\log p(u) \rceil \quad \forall u \in \mathcal{U}$$

This procedure will yield an expected code length within 1 of the entropy. In the next lecture, we will see if we can do better. In particular, we will apply the procedure of taking the two smallest probabilities, merging them, and repeating, but generalize it further to non-dyadic distributions.

References

[1] D. J. C. MacKay, *Information Theory, Inference, and Learning Algorithms*, Cambridge University Press, UK, 2003.

[2] S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 6, pp. 721–741, 1984.