6.441 Transmission of Information March 2, 2006

Lecture 7

Lecturer: Madhu Sudan *Scribe: Xiaomeng Shi*

1 Administrative Issues

- Pset 2 due next Wednesday, March 8, 2006.
- Midterm in three weeks (March 23). Project selection due on same day.

2 Today: Data Compression Continued

- Review L06
- Non-singular code
- Uniquely decodable code
- Entropy lower bound

3 Review of Lecture 6

AEP enables us to do fixed length typical sequence encoding. As the length of the sequence approaches infinity, the expected number of bits required per symbol approaches the entropy of the source. The next question we would like to ask is, would variable length encoding help us encode below entropy?

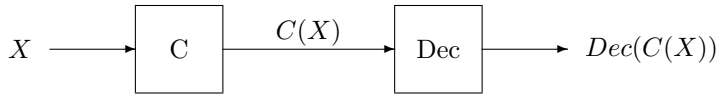


Figure 1: $(\forall X)\ Dec(C(X)) = X \implies C(X)$ is non-singular.

Non-Singular Codes Every elements of the range of X maps into a different string, thus X can be completely recovered from the code. In many situations, however, we don't need such stringent constraint on the code. For example, when compressing videos or pictures, as long as the errors are far apart and not clustered at one point, the effect is often negligible. Distortion theory gives theoretical bounds on the amount of compression that can be achieved under a given amount of distortion.

Non-singular extended codebook A more stringent constraint is to have the extended codebook also non-singular.

Definition 1 *An extended codebook from code C is C^* defined by $C^{(k)} : \mathscr{X}^{(k)} \to \mathscr{D}^{(k)}$, where $\mathscr{X} = \{1, \ldots, m\}$ is the set of source symbols, $\mathscr{D} = \{0, 1, \ldots, D-1\}$ is a D-ary alphabet of codeword symbols, $\mathscr{D}^* = \bigcup_{n \in \mathbb{Z}^+} \mathscr{D}^n$, and $C^{(k)}(x_1, \ldots, x_k) = C(x_1)C(x_2)\ldots C(x_k)$ is a concatenation of the individual codewords.*

Why extended codebooks? We could have simply concatenated sequences and assigned each a probability measure, but the resulting codebook size would be exponentially in sequence length. Through extension, the codebook is smaller in size. An equally important reason is that $C^{(k)}$ needs to be non-singular $(\forall k)$ for the concatenated sequence to be recoverable (uniquely decodable) at the receiver.

4 Best Non-Singular Code

What's the best non-singular code we can achieve, measured in terms of expected code length? Intuitively we would map the shortest codeword to symbols with the highest probability.

For the code to be non-singular, each possible code word must correspond to one unique node on the code tree, extended breath-first. Excluding the root node:

$$\underbrace{D + D^2 + \ldots + D^{l_i - 1}}_{\frac{D^{l_i-1}-1}{D-1}D} < i \leq \underbrace{D + D^2 + \ldots + D^{l_i}}_{\frac{D^{l_i}-1}{D-1}D} \implies l_i = \left\lceil \log_D \left(\frac{i(D-1)}{D} + 1 \right) \right\rceil$$

The expected length of this non-singular code is therefore:

$$L^*_{NS} = \sum_i p_i l_i \neq H(X)$$

This direct comparison with the source entropy is very complex. We will try solving the expected length inexactly to find a bound.

4.1 Bounds on the optimal codelength

How do we impose the condition of non-singularity mathematically?

Define $a(l)$ = number of distinct code words with length l, then

$$a(l) \leq D^l \,, \qquad l_1 \leq l_2 \leq \ldots \leq l_m \,.$$

The sum of probability of all possible code words is therefore

$$\sum_{i=1}^{m} D^{-l_i} = \sum_{l=1}^{l_m} D^{-l} a(l) \;\leq l_m \,.$$

The expected length of non-singular codes is ($*$ means optimal)

$$\begin{aligned} L^*_{NS} &= \sum_{i=1}^{m} p_i l_i = \sum_{i=1}^{m} -p_i \log_D D^{-l_i} = \sum_{i=1}^{m} -p_i \log_D \frac{D^{-l_i}}{p_i} + H(X) \\ (Jensen's\ \ ineq) \quad &\geq -\log_D \sum_{i=1}^{m} D^{-l_i} + H(X) \\ &\geq -\log_D l_m + H(X) \end{aligned}$$

Rather than using l_m, is there a bound that is independent of the codebook?

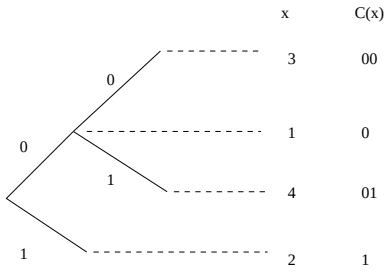
$$\begin{aligned} l_i = \left\lceil \log_D \left(\frac{i(D-1)}{D} + 1 \right) \right\rceil \implies l_m &\leq \lceil \log_D m \rceil \\ &\leq \log_D m + 1 \\ &\leq D \log_D m - \underbrace{(D-1)\log_D m}_{= \frac{D-1}{\ln D} \ln m \geq \frac{2-1}{\ln 2} \ln 2 = 1} + 1 \\ l_m &\leq D \log_D m \end{aligned}$$

Therefore,

$$\begin{aligned} L^*_{NS} &\geq -\log_D l_m + H(X) \\ &\geq -\log_D (D \log_D m) + H(X) \\ &\geq -\log_D D - \log_D \log_D m + H(X) \\ &\geq -\underbrace{\log_D \log_D m}_{\text{small}} - 1 + \underbrace{H(X)}_{\sim O(\log m)} \end{aligned}$$

Asymptotically, lengths of non-singular codes are constrained by entropy. Nonetheless, this is a bound that may be loose. Is it actually achievable? Can we really code below entropy? We show the answer is yes by an example.

Example: $p(1) = \frac{1}{2}, p(2) = \frac{1}{4}, p(3) = p(4) = \frac{1}{8}$



$E[|C(x)|] = 1.25 \ < \ H(X) = 1.75$

5 Kraft Inequality for Uniquely Decodable Codes

Claim 2 *Given a non-singular code C, there exist prefix-free codes with expected length $L_{NS}+O(\sqrt{L_{NS}})$*

What we are trying to show here is that the additional constraint of unique decodability does not worsen the expected code length much.

Proof First generate $C \to C_1$ by zero padding. Next divide C_1 into consecutive blocks of length $\lceil \sqrt{L_{NS}} \rceil$, and write this as $\lceil \sqrt{L_{NS}} \rceil \mid |C_1(x)|$. Insert 0 between each pair of blocks and append 1 at the end of the code word.

$C_1(x)$ $\boxed{w_1}$ $\boxed{w_2}$ $\boxed{w_3}$ $\ldots$ $\boxed{w_k}$

(w_1: $\lceil \sqrt{L_{NS}} \rceil$, w_2: $\lceil \sqrt{L_{NS}} \rceil$)

$C_2(x)$ $\boxed{w_1}$ $\boxed{0}$ $\boxed{w_2}$ $\boxed{0}$ $\boxed{w_3}$ $\ldots$ $\boxed{w_k}$ $\boxed{1}$

$C_2(x)$ is prefix-free thus uniquely decodable.

$$\begin{aligned}
E\left[|C_2(x)|\right] =& E\left[|C_1(x)|\frac{\lceil\sqrt{L_{NS}}\rceil+1}{\lceil\sqrt{L_{NS}}\rceil}\right] \\
=& \frac{\lceil\sqrt{L_{NS}}\rceil+1}{\lceil\sqrt{L_{NS}}\rceil} \underbrace{E\left[|C_1(x)|\right]}_{\leq E[|C(x)|]+\lceil\sqrt{L_{NS}}\rceil = L_{NS}+\lceil\sqrt{L_{NS}}\rceil} \\
=& L_{NS} + O(\left\lceil\sqrt{L_{NS}}\right\rceil)
\end{aligned}$$

■

For this uniquely decodable code $C^{(k)}: \mathscr{X}^{(k)} \to \mathscr{D}^*$ constructed with k concatenations,

$$\begin{aligned}
l(x_1,\ldots,x_k) =& \sum_{i=1}^{k} l(x_i) \\
\sum_{(x_1,\ldots,x_k)\in\mathscr{X}^k} D^{-l(x_1,\ldots,x_k)} =& \left(\sum_{i=1}^{m} D^{-l_i}\right)^k \leq k l_m \\
\sum_{i=1}^{m} D^{-l_i} \leq& (kl_m)^{1/k} = \exp\frac{\log k + \log l_m}{k} \to \qquad \exp(0) = 1 \text{ as } k\to\infty \\
\sum_{i=1}^{m} D^{-l_i} \leq& 1
\end{aligned}$$

This is the *Kraft Inequality for Uniquely Decodable Code.*

Next consider the expected length L^*_{UD}:

$$\begin{aligned}
L^*_{UD} =& \sum_i p_i l_i \\
=& \sum_i -p_i \log_D \frac{D^{-l_i}}{p_i} + H(X) \\
(Jensen's\ ineq) \quad \geq& -\underbrace{\log_D \sum_i D^{-l_i}}_{\leq 1} + H(X) \\
\geq& H(x)
\end{aligned}$$

Here the code is uniquely decodable, and its expected length is larger than the entropy. Since the AEP code achieves the entropy, this is a tight lower bound as $n \to \infty$. What happens when n is finite?

Recall that the Kraft Inequality is a necessary condition for unique decodability. It is indeed also a sufficient condition for the existence of a prefix-free code satisfying the length assignments. Sufficiency can be proved by examining the tree structure for prefix free codes. Assume $l_1 \leq l_2 \ldots \leq l_m$:

1. assign the first free node at depth l_i to i.

2. prune the subtree of the assigned node so that the descendents are not free to be assigned to any symbol.

3. repeat until l_m

With these assignments, it is feasible to get a prefix free code. We can prove this by contradiction as follows.

Proof If the assignment fails, there exists $k < m$ such than the tree become full[1] without any free nodes after assigning l_k. The fact that the tree is full implies that $\sum_{i=1}^{k} D^{-l_i} = 1 < \sum_{i=1}^{m} D^{-l_i}$, which is a contradiction to the assumption that the set of lengths satisfies the Kraft Inequality. ■

Consider the following length assignment,

Definition 3 (Shannon Code) $l_i = \lceil -\log_D p_i \rceil$.

which satifies the Kraft Inequality

$$\sum_i D^{-l_i} \leq \sum_i D^{\log_D p_i} = \sum_i p_i = 1$$

and therefore a prefix-free code exists with this length assignment. Its expected length L_{SH} is bounded as follows,

$$\log_D \frac{1}{p_i} \leq l_i \leq \log_D \frac{1}{p_i} \Longrightarrow H(X) \leq L^*_{UD} \leq L_{SH} < H(X) + 1$$

which therefore gives an upperbound on the expected length of the optimal uniquely decodable code.

What's the best optimal uniquely decodable code?

Huffman Code optimal prefix free code

To generate the Huffman Code,

1. add $r \in \{0, \ldots, D-1\}$ dummy nodes such that the total number of nodes is equal to $1 + k(D-1)$ for some k. $\Pr(dummy) = 0$.

2. group the D least probable symbols into one symbol; use one D-ary symbol to distinguish these D symbols.

3. repeat step 1 until only one node remains.

6 Summary

$$H(X) - \log_D \log_D m - 1 < L^*_{NS} \leq \underbrace{L^*_{UD}}_{\geq H(X)} < H(X) + 1$$

[1] A D-ary tree is full iff all nodes have either 0 or D children.