LECTURE 6

Last time:

- AEP
- Coding with AEP

Lecture outline

- Kraft inequality
- optimal codes
- Huffman codes

Reading: Scts. 5.2-5.8

Quick Review

- AEP: Typical set $P(A_\epsilon^{(n)}) \to 1$.
- All typical sequences are *approximately* equally likely.
- $|A_\epsilon^{(n)}| \doteq 2^{nH}$.

- Coding, performance metric: average codeword length per source symbol.
- Coding with AEP

$$\frac{1}{n}E[l(X_1^n)] \to H(X)$$

Questions

- Can we do better?
- Can we have a symbol-by-symbol code that is equally good?

Concatenation

Definition The *extension* of a code C is the a code for finite strings of $\mathcal{X}$ given by the concatenation of the individual codewords

$$C(x_1, x_2, \ldots, x_n) = C(x_1)C(x_2)\ldots C(x_n)$$

- A code is called **non-singular** if

$$x_i \neq x_j \Rightarrow C(x_i) \neq C(x_j)$$

- A code is called **uniquely decodable** if its extension is non-singular

Example:

x	a	b	c	d
$C(x)$	1	11	10	101

Prefix code

Example The following code is uniquely decodable,

x	a	b	c	d
$C(x)$	10	00	11	110

consider a coded string 11000000000000010.

Definition A code is called a *prefix code* or *instantaneous code* if no codeword is a prefix of any other codeword.

- Self-punctuating.
- Can decode without reference of the future.

- **Relations between different types of codes**

Kraft's Inequality

Theorem For any prefix code over an alphabet of size D, let the codeword length be $l_1, l_2, \ldots,$, we have

$$\sum_{i=1}^{\infty} D^{-l_i} \leq 1$$

Conversely, for any given set of codeword lengths that satisfy the inequality, we can construct a prefix code with these codeword lengths.

Proof

- Construct a D-ary tree.
- Prefix code means each codeword is a leaf, no codeword can be the descendent of any other codeword.
- Assign weight D^{-l_i} to each codeword.

Consider a codeword $y_1 y_2 \ldots y_{l_i}$, where $y_j \in \{0, \ldots, D-1\}$. Let

$$0.y_1 y_2 \ldots y_{l_i} = \sum_{j=1}^{l_i} y_j D^{-j} \in [0,1]$$

.

This codeword corresponds to an interval

$$\left(0.y_1 y_2 \ldots y_{l_i}, 0.y_1 y_2 \ldots y_{l_i} + \frac{1}{D^{l_i}}\right)$$

Prefix code implies the intervals are disjoint.

- **Converse**: For a given set of lengths $l_1, \ldots, l_m$, construct a D-ary tree, label the first available node of length l_1 for codeword 1, . . .

Kraft's Inequality for Uniquely Decodable Codes

Assume a uniquely decodable code on $|\mathcal{X}| = m$ has the largest codeword length l_{max}, consider the concatenated code for a sequence of k symbols:

$$
\begin{aligned}
\sum_{x_1^k} D^{-l(x_1^k)} &= \sum_{x_1, x_2, \ldots, x_k \in \mathcal{X}^k} D^{-l(x_1)} \ldots D^{-l(x_k)} \\
&= \left(\sum_{x \in \mathcal{X}} D^{-l(x)} \right)^k
\end{aligned}
$$

This says as k increases, the sum weight increases exponentially.

On the other hand, the sum of D^{-l} over all the nodes with the same depth l is 1 for any l. This means as k increases, the sum weight at most increase linearly.

Let $N(m)$ be the number of nodes at depth m, we have $N(m) \le D^m$.

$$\begin{aligned}\sum_{x_1^k} D^{-l(x_1^k)} &\le \sum_{i=1}^{kl_{max}} N(m) D^{-m} \\ &= kl_{max}\end{aligned}$$

Now for any k,

$$\sum_{x \in \mathcal{X}} D^{-l(x)} \le (kl_{max})^{1/k}$$

therefore

$$\sum_{x \in \mathcal{X}} D^{-l(x)} \le 1.$$

Conclusion: Uniquely decodable codes does not offer any more choice for the codeword length than prefix codes.

Optimal codes

Optimal code is defined as code with smallest possible $L(C)$ with respect to P_X

Optimization:

minimize $\sum_{x \in \mathcal{X}} P_X(x) l(x)$

subject to $\sum_{x \in \mathcal{X}} D^{-l(x)} \leq 1$

and $l(x)$s are integers

Optimal codes

Let us relax the integer constraint and replace the first constraint by equality to obtain a lower bound. Use Lagrange multipliers, define

$J = \sum_{x \in \mathcal{X}} P_X(x) l(x) + \lambda \sum_{x \in \mathcal{X}} D^{-l(x)}$

and set $\frac{\partial J}{\partial l(i)} = 0$

$$P_X(i) - \lambda \log(D) D^{-l(i)} = 0$$

equivalently $D^{-l(i)} = \frac{P_X(i)}{\lambda \log(D)}$

solve for $\lambda = \frac{1}{\log(D)}$, yielding $l(i) = -\log_D(P_X(i))$

The expected codeword length

$$\begin{aligned} L(C) &= E[l(X)] = E[-\log_D P_X(X)] \\ &= H_D(X) \\ &= \frac{H(X)}{\log_2 D} \end{aligned}$$

Shannon Code

- Ideal codeword length $l_i = -\log_D P_X(i)$, this is optimal when $-\log_D P_X(i)$ is an integer for any i.
- For general distribution, set

$$l_i = \lceil -\log_D P_X(i) \rceil$$

.

- Bounds for the codeword length.

$$-\log P_X(i) \leq l_i \leq -\log P_X(i) + 1, \forall i$$

  - $\{l_i\}$ satisfy Kraft's inequality, corresponding prefix code exists.
  - Average codeword length

$$H(X) \leq E[l(X)] \leq H(X) + 1$$

Shannon Code

Example X takes four possible values with probabilities $(\frac{1}{3}, \frac{1}{3}, \frac{1}{4}, \frac{1}{12})$.

$$\begin{aligned} H(X) &= 1.8554 \\ l_i &= \lceil -\log P_X(i) \rceil = (2, 2, 2, 4) \\ E[l(X)] &= 13/6 = 2.1667 \end{aligned}$$

Comparing to the obvious codeword length assignment $(2, 2, 2, 2)$, lose $\frac{1}{6}$ bit per source symbol.

Improvement: code over multiple i.i.d. source symbols: look at $(X_1, X_2, \ldots, X_n)$ as one super-symbol, apply Shannon code,

$$H(X_1, \ldots, X_n) \leq E(l(X_1^n)) \leq H(X_1, \ldots, X_n) + 1$$

implies

$$H(X) \leq \frac{1}{n} E[l(X_1^n)] \leq H(X) + \frac{1}{n}$$

Unknown Distribution

If assign the codeword length as

$$l_i = \lceil - \log q(i) \rceil,$$

and the real distribution of X is
$P_X(i) = p_i$,

$$H(p) + D(p||q) \leq E_p[l(X)] \leq H(p) + D(p||q) + 1$$

Proof

$$\begin{aligned}
E_p[l(X)] &= \sum_x p(x) \lceil \log \frac{1}{q(x)} \rceil \\
&\leq \sum_x p(x) \left(\log \frac{1}{q(x)} + 1 \right) \\
&= \sum_x p(x) \log \frac{p(x)}{q(x)} \frac{1}{p(x)} + 1 \\
&= D(p||q) + H(p) + 1
\end{aligned}$$

Penalty of $D(p||q)$ bits per source symbol due to the wrong distribution.

Discussion

- Kraft's inequality gives a lower bound of the average codeword length. For any n, any code over i.i.d. sequence X_1^n, $\frac{1}{n}E[l(X_1^n)]$ cannot be smaller than $H(X)$.
- We can achieve this when $n \to \infty$, AEP code, Shannon code.

$$\lim_{n\to\infty} \frac{1}{n}E[l(X_1^n)] = H(X)$$

True or False: for finite n:

- Shannon code is "optimal"?
- A code with codeword length $l_i = -\log P_X(i), \forall i$ is optimal.
- Any prefix code must satisfy

$$l_i \geq -\log P_X(i), \forall i$$

- The optimal code must satisfy

$$l_i \leq \lceil -\log P_X(i) \rceil, \forall i$$

Constructing the Optimal Prefix Code

X has probability masses $p_1 \geq p_2 \ldots \geq p_m$, construct binary code to minimize $\sum_i p_i l_i$.

What should the optimal code look like?

- If $p_i > p_j$, then $l_i \leq l_j$.
- The two longest codewords have the same length.
- The two longest codewords differ only in the last bit.

Construction:
Take the two least likely symbols, merge them to get a size $m - 1$ problem.

D-ary Huffman Code

Definition Complete tree: every leaf is assigned to a codeword. Every intermediate node has D branches stemming from it.

- A complete tree means Kraft's inequality holds with equality.
- Size of a D-ary complete tree: $1+n(D-1)$ for integer n.
- For an arbitrary $\mathcal{X}$, add 0 probability symbols to make it fit in a complete tree.

What can we say about Huffman Code

- Optimal prefix code for any source.
- Always equally good or better than Shannon code.
- $\frac{1}{n}E[l(X_1^n)] \to H(X)$ as $n \to \infty$.