Information Theory with Applications, Math6397
Lecture Notes from September 25, 2014

taken by Carlos Ortiz

Last Time

- Separable Codes
- Prefix Codes
- Kraft's Inequality

2.4.27 Theorem. *Given DMS with values in $\mathbb{A}$ and induced measure $\mathbb{Q}$, then for all $j \in \mathbb{N}$ and separable codes,*

$$\mathbb{E}[l(X_j)] \geq H_K(\mathbb{Q})$$

where $H_K(\mathbb{Q})$ is the K-ary entropy; e.g. we use $\log$ base K instead of base e.

Proof. If we take the natural logarithm in both sides of the Kraft's inequality notice that,

$$0 \leq -\ln(\sum_{x\in\mathbb{A}} K^{-l(x)}) = \sum_{x\in\mathbb{A}} Q(x)\ln(\frac{\sum_{x\in\mathbb{A}} Q(x)}{K^{-l(x)}}) \leq \sum_{x\in\mathbb{A}} Q(x)\ln(\frac{Q(x)}{K^{-l(x)}})$$

$$= \sum_{x\in\mathbb{A}} Q(x)\ln(Q(x)) - \sum_{x\in\mathbb{A}} -Q(x)l(x)\ln(k) = -H(Q) + ln(K)\mathbb{E}[l(x_j)]$$

Hence we get,

$$0 \leq -H(Q) + ln(K)\mathbb{E}[l(x_j)]$$

Now if we divide by $ln(k)$ both sides we get that,

$$\mathbb{E}[l(x_j)] \geq H_K(\mathbb{Q})$$

□

2.4.28 Question. Can we do any better? What about performance limits of prefix codes?

2.4.29 Theorem. *If a sequence $\{l(x)\}_{x\in\mathbb{A}}$ satisfies the Kraft inequality then there exist a prefix code ϕ with lengths*

$$l(x) = |\phi(X)|$$

Proof. We show existence by constructing an appropriate tree. Let $\alpha_k = |\{x \in \mathbb{A} : l(x) = k\}|$ and $l_{\max} = \sup_{x\in\mathbb{A}} l(x)$. We know by assumption that

$$\sum_{i=1}^{l_{\max}} \alpha_i K^{-i} = \sum_{x\in\mathbb{A}} K^{l(x)} \leq 1$$

Now if $l_{\max} < \infty$ we can rearrange the terms in the above finite sum to deduce that,

$$\alpha_1 \leq K$$

and

$$\alpha_j \leq K^j - \sum_{m=1}^{j-1} \alpha_m K^{j-m}$$

for $j = 2, ..., l_{\max}$. Next we recursively built the code tree by pruning a full infinite K-ary tree. Start at the root ("Level 0") and consider the nodes at the first level. Prune the tree below α_1 nodes and turn them into leaves (codewords). Proceed to the next. The full K-ary tree has kK^2 nodes at this level and after pruning we retain $K^2 - \alpha_1 K$. Prune the tree below α_2 of those nodes and turn them into leaves. At the jth level we have,

$$\alpha_j \leq K^j - \sum_{m=1}^{j-1} \alpha_m K^{j-m}$$

nodes, which is by assumption bigger that α_j. We stop at the $j = l_{\max}$ or continue inductively if $l_{\max} = \infty$. □

2.4.30 Question. How can we assign code lengths based on probability of outcomes in $\mathbb{A}$ to generate short average codewords lengths?

2.4.31 Theorem. *(Shannon-Fano) Given a discrete memoryless source $\{X_j\}_{j=1}^{\infty}$ with induced measure Q and K-ary block codes with $l(x) = \lceil -\log_k(Q(X)) \rceil$, then the Kraft inequality holds and*

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

Proof. Notice that,

$$\sum_{x\in A} K^{-l(x)} = \sum_{x\in A} K^{-\lceil -\log_K(Q(X)) \rceil} \leq \sum_{x\in A} K^{\log_K(Q(X))} = \sum_{x\in A} Q(x) = 1$$

thus the Kraft inequality holds. In addition,

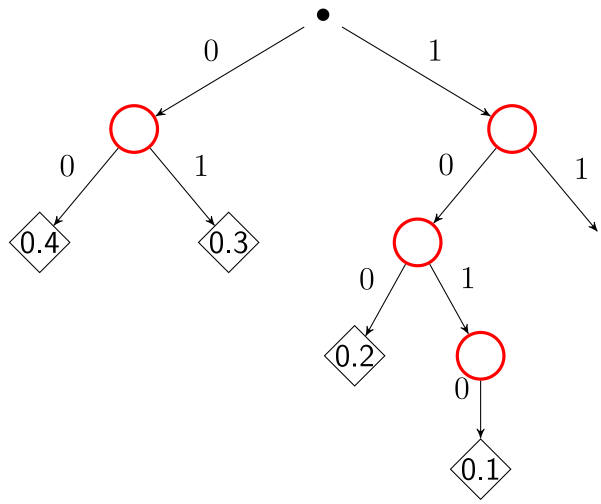
$$-\log_K(Q(x)) \leq l(x) \leq -log_K(Q(x)) + 1$$

and if we average over all $x \in A$ with respect to Q we get,

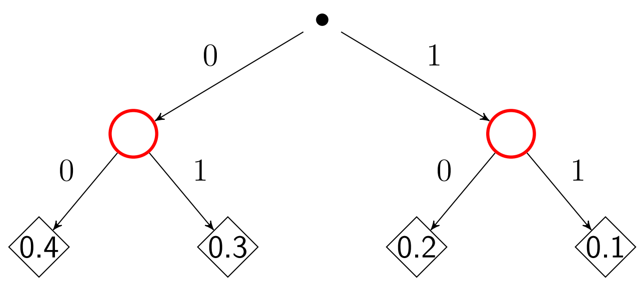
$$H_K(X) \leq E[l(X)] \leq H_K(X) + 1\,.$$

□

2.4.32 Example.
Let $\mathbb{A} = \{1, 2, 3, 4\}$ with $Q(1) = 0.4$, $Q(2) = 0.3$, $Q(3) = 0.2$ and $Q(4) = 0.1$. Now if $K = 2$ we have that $\alpha_1 = \lceil -\log_2(0.4)) \rceil = 2$, $\alpha_2 = \lceil -\log_2(0.3)) \rceil = 2$, $\alpha_3 = \lceil -\log_2(0.2)) \rceil = 3$, and $\alpha_4 = \lceil -\log_2(0.1)) \rceil = 4$. If we take a look at our code tree,



we see that $\mathbb{E}[l(X)] = 2.4$ and this is not optimal since,



has $\mathbb{E}[l(X)] = 2$.

2.4.33 Question. Can we get closer to this lower bound?

2.4.34 Answer. Work with the alphabet $\mathbb{A}^n$ instead of $\mathbb{A}$.

2.4.35 Corollary.

$$H(X_1) \leq \frac{1}{n} E[l(X_1, ..., X_n)] \leq H(X_1) + \frac{1}{n}$$

Proof.

$$nH(X_1) = H(X_1, ..., X_n) \leq E[l(X_1, ..., X_n)] \leq H(X_1, ..., X_n) + 1 = nH(X_1) + 1$$

□

2.4.36 Remark. These bounds have analogous formulations for stationary and ergodic sources.

2.5 Huffman Code

2.5.37 Question. Can we find an optimal average length code for a given discrete memoryless source?

2.5.38 Proposition. *Given discrete memoryless source with induced measure Q such that $H(Q) < \infty$, then the minimum average length binary code has a code tree without unused leaves.*

Proof. Suppose we have a code tree with unused leaves, then we would have one the following two situations,

then we can shorten the tree, just like we did in $2.4.32$, making $\mathbb{E}[l(X)]$ smaller. □

2.5.39 Proposition. *Suppose we have a code tree with unused leaves, there is a optimal binary prefix code such that two given codewords of lowest probability p_1 and p_2 only differ in the last digit.*

Proof. Suppose the smallest probabilites are p_1 and p_2, $p_1 \leq p_2$, associated with symbols a_1 and a_2, respectively. If a_i, a_j are the pair of symbols with the longest codewords, then compare a_1 and a_i, and if $a_1 \neq a_i$, then swap a_1 with a_i. After that, take a_j and compare it to a_2. If $a_2 \neq a_j$ then swap a_2 with a_j. Notice that each step in this procedure will not increase the probability of symbols with longer codewords, so it will not increase the expected codeword length. On the other had, it will lead to the lowest probabilities becoming siblings in the longest branch of the tree. Notice that this is the case even if one or both of a_1 or a_2 are identical with a_i or a_j, because in that case leaves in the longest branch are exchanged if $a_i \neq a_1 = a_j$ or $a_j \neq a_2 = a_i$. □