Information Theory & Coding

Sept 28th 20

Last week(s):

- Source Coding.
  - Codes: injective, u.d., p.f., ...
  - Kraft's Inequality for p.f / u.d codes:

$$KS(c) \le 1$$

$$\left[\sum_{u \in \mathcal{U}}' 2^{-\text{length}(c(u))} \right]$$

- Entropy as a lower bound to average codeword length:

$$\left[E[\text{length}(c(U))] \ge H(U) \right]$$

u.d.

$$= \sum_{u \in \mathcal{U}}' p(u) \log_2 \frac{1}{p(u)}$$

- $\exists$ a p.f. c s.t.

$$\le H(U) + 1$$

Entropy:

Conditional Entropy,

$H(u|v) \leq H(u)$

Mutual Information $I(u;v) = H(u) - H(u|v) \geq 0$

$I(u;v|w)$

$= H(u|w) - H(u|vw) \geq 0.$

Thm: $I(u;v|w) \geq 0.$

$E\left[\log \frac{1}{p(u|w)} - \log \frac{1}{p(u|vw)}\right]$

$= E\left[\log \frac{p(uv|w)}{p(u|w)\, p(v|w)}\right]$

$= \sum_{u,v,w} \underbrace{p(w)\, p(uv|w)}_{p(uvw)} \log_2 \frac{p(uv|w)}{p(u|w)\, p(v|w)}$

$= \sum_{w} p(w) \boxed{\sum_{u,v} p(uv|w) \log \frac{p(uv|w)}{p(u|w)\, p(v|w)}}$

Claim: $\displaystyle\sum_{u,v} p(uv|w) \log \frac{p(uv|w)}{p(u|w)\,p(v|w)} \geq 0.$

Why: $I(u;v) \geq 0 \equiv \displaystyle\sum_{u,v} p(uv) \log \frac{p(uv)}{p(u)p(v)} \geq 0$

$\Rightarrow I(u;v|w) \geq 0.$ //

Back to prefix-free codes & coding for smallest value of $E[\text{length } c(U)]$

Formally what we want to solving is an optimization problem:

Given $\{p(u) : u \in \mathcal{U}\}$ find $\{l(u) : u \in \mathcal{U}\}$ that satisfy Kraft $\sum_u 2^{-l(u)} \leq 1$

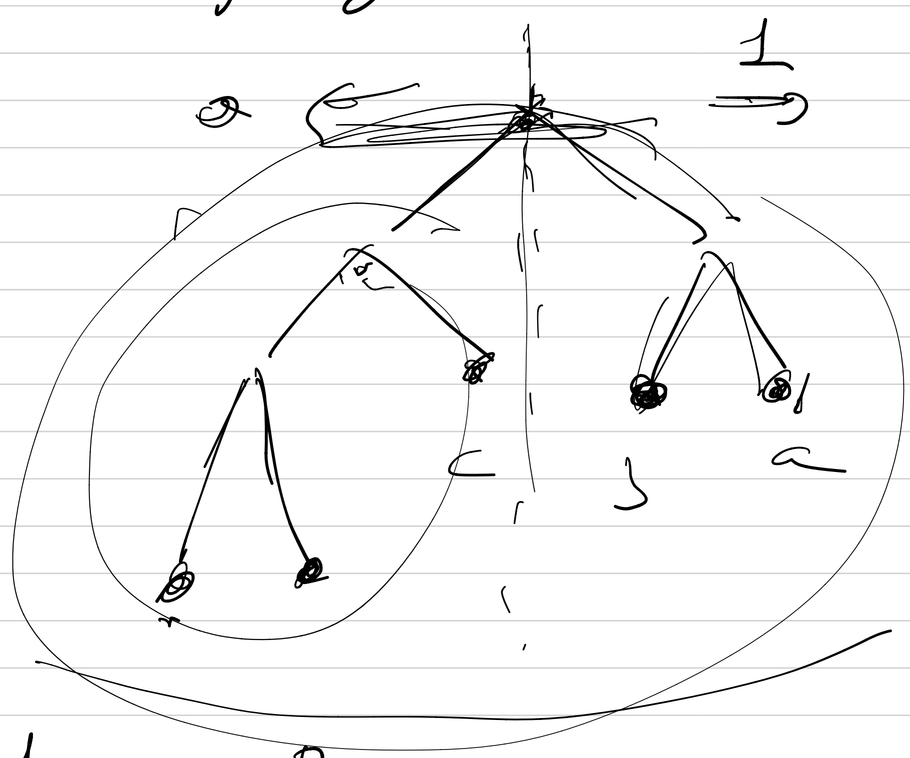
and minimize $\displaystyle\sum_{u \in \mathcal{U}} p(u)\, l(u)$

[Integer programming problem].

Recall that $l(u) = \log \frac{1}{p(u)}$ satisfy Kraft, $E[\ ] = H(u)$ as small as possible

but $l(u) \in \mathbb{R}$ . . .

- Guessing games:

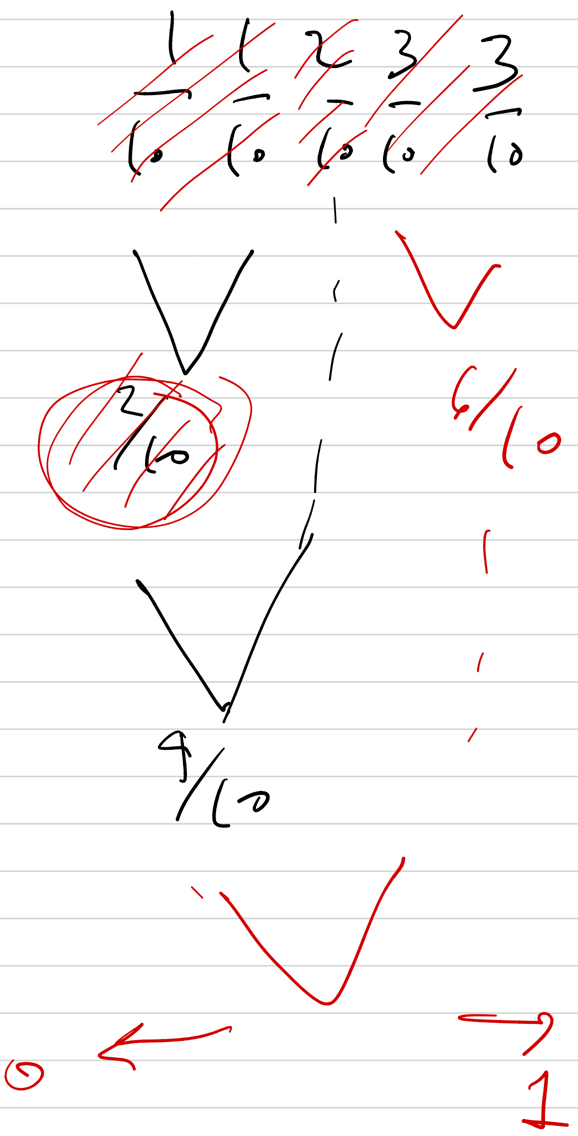


$E(\#\text{ of } q)$

$= E[\text{length}]$

Huffman Procedure:

Example: $\mathcal{U} = \{a, b, c, d, e\}$



$a \to 000$

$b \to 001$

$c \to 01$

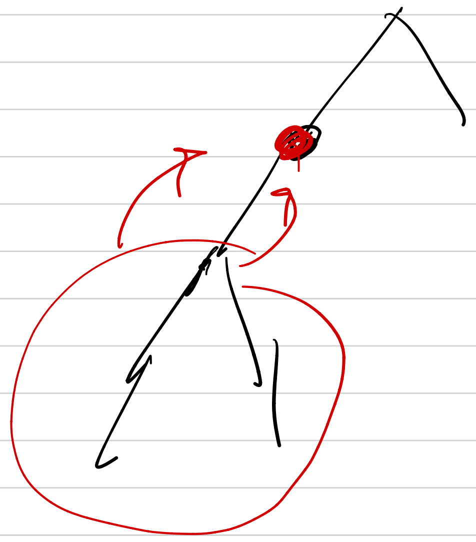
$d \to 10$

$e \to 11$

Optimality of Huffman's Procedure:

- Properties of optimal codes. (p.f.)

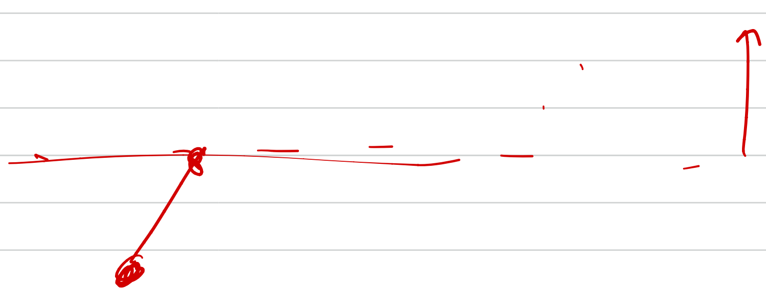
1. The tree representation of the code is indeed binary: each node is either a leaf (codeword) or has exactly two children



Pf. If not, a better code exists.

2. Corollary of 1: the two longest codewords are the same length.

If not:



3. if $p(u) > p(v) \Rightarrow \ell(u) \le \ell(u)$.

Pf if not $p(u) > p(v)$ & $\ell(u) > \ell(v)$, if we swap u & v in the tree then in the new code

$$E[L] = \cdots + p(u)\,l(v) + p(v)\,l(u)$$

old code

$$= \cdots + p(u)\,l(u) + p(v)\,l(v)$$

$$\text{diff} = p(u)[l(v)-l(u)] - p(v)[l(v)-l(u)]$$

$$= \underbrace{(p(u)-p(v))}_{>0}\underbrace{(l(v)-l(u))}_{<0} < 0$$

contradicts the optimality of the original code

④. Corollary of (1,2,3): $\exists$ an optimal code s.t. the two least likely symbols are "siblings".

Pf: by ③ two least likely letters have the largest codewords. by ① they are the same length

swap if the black dots are not siblings. //

With the properties we have shown, if

$p_1 \geq p_2 \geq \dots \geq p_K$ are the

probabilities of a K letter alphabet, then

$l_1 \leq l_2 \leq \dots \leq l_{K-1} = l_K$

Expected length $= \sum p_i l_i =$

$$\underline{p_1 l_1} + \dots + \underline{p_{K-2}}\,\underline{l_{K-2}} + \underline{(p_{K-1}+p_K)}\,\underline{l_{K-1}}$$

$$\left\{\begin{array}{l} p'_1 = p_1 \\ \vdots \\ p'_{K-2} = p_{K-2} \\ p'_{K-1} = p_{K-1} + p_K \end{array}\right. \qquad = \underline{\underline{\sum_{i=1}^{K-1} p'_i l_i}}$$

$$1 \geq \sum_{i=1}^{K} 2^{-l_i} = 2^{-l_1} + \dots + 2^{-l_{K-1}} \cdot 2$$

$$= 2^{-l_1} + \dots + 2^{-l_{K-2}} + 2^{-(l_{K-1}-1)}$$

$l'_1 = l_1, \dots, l'_{K-2} = l_{K-2}, \; l'_{K-1} = l_{K-1} - 1$

$$= \sum_{i=1}^{K-1} 2^{-l'_i}$$

Expected length of the original code

$$= \left(\sum_{i=1}^{k-1} p_i' l_i\right) = \left(\sum_{i=1}^{k-1} p_i' l_i'\right) + p_{k-1}'$$

$p_{k-1} + p_k$

expected length of a new code

for the alphabet $\mathcal{U}' = \{1 \ldots k-1\}$

prob $p_1' \ldots p_{k-1}'$

lengths $l_1' \ldots l_{k-1}'$

So the design problem: find $l_1, \ldots, l_k$ for $p_1 \ldots p_k$ is reduced to find $l_1' \ldots l_{k-1}'$ for $p_1' \ldots p_{k-1}'$.

$\Longrightarrow$ Huffman's procedure.

The Role of Entropy.

Simple Case: we have a source $U_1 U_2 U_3 \ldots$ that produces i.i.d letters.

Ex: $p(0) = p$, $p(1) = 1-p$.

$(U_1 \ldots U_n)$, by the law of large #, will contain $\approx np$ 0's
$\approx n(1-p)$ 1's.

Let $T(p, n, \varepsilon) = \{(u_1 \ldots u_n) : \forall u \in \mathcal{U}$
$\left|\frac{1}{n}\#\{i : u_i = u\} - p(u)\right| \leq \varepsilon p(u)\}$. (*)

$$\Rightarrow np(u)(1-\varepsilon) \leq \#\{i : u_i = u\} \leq np(u)(1+\varepsilon)$$

a sequence $(u_1 \ldots u_n)$ satisfying (*) is said to be ε-typical with respect to p.

Suppose $\mathcal{U} = \{a, b, c\}$

$0.6, 0.3, \underline{0.1}$

$\underline{n = 20 \quad \varepsilon = 0.1}$

for $u_1 \ldots u_{20}$ to be typical it should contain

$$\left.\begin{array}{l} 12 \pm 1.2 \ a\text{'s} \\ 6 \pm 0.6 \ b\text{'s} \\ 2 \pm 0.2 \ c\text{'s} \end{array}\right\} \quad \begin{array}{l} 12 \ a\text{'s} \\ 6 \ b\text{'s} \\ 2 \ c\text{'s} \end{array}$$

Properties of typical sequences:

if $\underline{u_1 \ldots u_n}$ is ε-typical wrt p.

then $\Pr(\underbrace{U_1 \ldots U_n}_{\text{iid} \sim p} = u_1 \ldots u_n)$

$$= \prod_{i=1}^{n} P(U_i = u_i) = \prod_{u \in \mathcal{U}} p(u)^{n(u)}$$

$$n(u) = \#\{i : u_i = u\} = np(u)(1 \pm \varepsilon)$$

$$\prod_{u \in \mathcal{U}} p(u)^{n(u)} = \prod_{u \in \mathcal{U}} 2^{n(u) \log_2 p(u)}$$

$$= 2^{n \sum'_{u \in \mathcal{U}} \frac{n(u)}{n} \log_2 p(u)}$$

$$\Pr(U_1 \ldots U_n = u_1 \ldots u_n) \leq 2^{n \sum_u (1-\varepsilon) p(u) \log p(u)}$$

iid p

$$\geq 2^{n \sum (1+\varepsilon) p(u) \log p(u)}$$

$$2^{-n H(U)(1+\varepsilon)} \leq \Pr(\quad) \leq 2^{-n H(U)(1-\varepsilon)}.$$

Corollary:

$$|T(n, p, \varepsilon)| \leq 2^{n H(U)(1+\varepsilon)}$$

//