Symbol Codes (§5)

Last week: Shannon's source coding theorem: $H(X)$ is "optimal" lossy compr. + "optimal" average lossless compression rate ↯ large block size ↯ complicated

> **Today's goal:** Lossless compression one symbol at a time with $H(X) \leq L \leq H(X)+1$, where L = average length of codeword per symbol.

NOTATION: $S^+ = \bigcup_{N \geq 1} S^N$ = nonempty strings over S

$\ell(w)$ = length of string $w \in S^+$

Symbol code: $C: \mathcal{A} \to \{0,1\}^+$ for alphabet $\mathcal{A}$

* **extended code:**

$$C^+: \mathcal{A}^+ \to \{0,1\}^+, \quad C^+(x_1 \cdots x_N) := C(x_1) \cdots C(x_N)$$

* C is called **uniquely decodable** (UD) if

$$C^+(w) = C^+(w') \implies w = w' \qquad \forall w, w' \in \mathcal{A}^+$$

* C is called a **prefix code** if no codeword $C(x)$ is prefix of any other
* **Any prefix code is UD!**

Examples:

x	$P(x)$	C_3	C_4	C_5	C_6
A	1/2	0	00	0	0
B	1/4	10	01	1	01
C	1/8	110	10	00	011
D	1/8	111	11	11	111
prefix code?		✓	✓	↯	↯
UD?		✓	✓	↯	✓
average length		1.75	2	1.25	1.75

reverse of C_3...

entropy: $H(P) = 1.75$

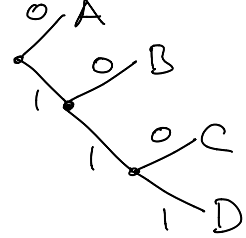
defined as

$$L(C,P) = L(C,X) = \sum_{x \in \mathcal{A}} P(x)\, \ell(C(x)) = E\left[\ell(C(x))\right]$$

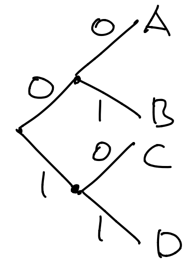
usually want to minimize

Prefix codes = binary trees:

e.g. $C_3 =$



$C_4 =$



* leafs labeled by $x \in \mathcal{A}$
* path to leaf = codeword $C(x)$

What constraint is imposed by UD/prefixness?

> **Kraft–McMillan inequality:** If C is UD then
> $$\sum_{x \in \mathcal{A}} 2^{-\ell(C(x))} \leq 1$$

← optimal codes should saturate this ("complete" code)

Pf: Let $S := \sum_x 2^{-\ell(C(x))}$ and $\ell_{max} = \max_x C(\ell(x))$. Then:

$$\underset{\text{exp. growth}}{S^N} = \sum_{x_1,\dots,x_N} 2^{-\underbrace{\ell(C^+(\overset{N \text{ symbols}}{x_1 \cdots x_N}))}_{\leq N \cdot \ell_{max}}} \leq \sum_{\ell=0}^{N \cdot \ell_{max}} 2^{-\ell} \cdot \underbrace{\#\{\text{codewords of length } \ell\}}_{\leq 2^{\ell} \text{ by UD}}$$

$$\leq \underset{\text{linear growth}}{N \cdot \ell_{max} + 1} \quad \overset{\forall N}{\Longrightarrow} \quad S \leq 1.$$

$\square$

> **Kraft's converse:** If $\sum_x 2^{-\ell(x)} \leq 1$ then $\exists$ prefix code with these lengths.

Thus, prefix codes are as good as any UD code!!!

Pf: Construct as follows:

① Order the lengths:

$\ell(x_1) \leq \ell(x_2) \leq \dots$ where $\mathcal{A} = \{x_1, x_2, \dots\}$

② For $k = 1, 2, \dots$ choose $C(x_k) \in \{0,1\}^{\ell(x_k)}$ s.th. NONE of the $C(x_1), \dots, C(x_{k-1})$ is prefix. This is possible, since

$\#\{\text{bitstrings of length } \ell(x_k) \text{ that have such prefix}\}$

$$\leq \sum_{i=1}^{k-1} 2^{\ell(x_k) - \ell(x_i)} < \sum_{i=1}^{k} 2^{\ell(x_k) - \ell(x_i)} = 2^{\ell(x_k)} \cdot \sum_{i=1}^{k} 2^{-\ell(x_i)}$$

$$\leq 2^{\ell(x_k)} \sum_x 2^{-\ell(x)} \leq 2^{\ell(x_k)}$$

$\square$

How "short" can UD codes be? Need one more tool...

Gibbs inequality: Let P, Q prob. distributions. Then:

$$\sum_x P(x) \log \frac{1}{Q(x)} \geq H(P), \quad \text{"="} \text{ iff } P = Q$$

Pf: $\text{LHS} - \text{RHS} = \sum_x P(x) \log \frac{P(x)}{Q(x)} = -\sum_x P(x) \log \frac{Q(x)}{P(x)}$ & use Jensen. $\square$

$\boxed{\text{Lower bound:}}$ $L(C, P) \geq H(P)$ for every UD code. information content!

(And equality holds iff $\ell(C(x)) = \log \frac{1}{P(x)} \; \forall x$)

Pf: Define

$$Q(x) = \frac{2^{-\ell(C(x))}}{S}, \quad \text{where } S = \sum_x 2^{-\ell(C(x))} \underset{\text{Kraft}}{\leq} 1.$$

Gibbs

$$\Rightarrow H(P) \underset{= \text{ iff } P=Q}{\leq} \sum_x P(x) \log \frac{1}{Q(x)} = L(C, P) + \log S \underset{= \text{ iff } S=1}{\leq} L(C, P) \quad \square$$

We can easily achieve this!

$\boxed{\text{Existence of good codes:}}$ $\exists$ prefix codes with $L(C, X) \leq H(X) + 1$

Pf: Define $\ell(x) = \lceil \log \frac{1}{P(x)} \rceil$ ← ie round equality condition from above!

* $\sum_x 2^{-\ell(x)} \leq \sum_x P(x) = 1 \implies$ prefix code exists by Kraft's converse

* $\sum_x P(x) \ell(x) \leq \sum_x P(x) \left(\log \frac{1}{P(x)} + 1 \right) = H(X) + 1.$ $\square$

NB: This code is in general NOT optimal. E.g.:

x	$P(x)$	$\ell(x)$	$C(x)$
A	1/3	2	00
B	1/3	2	01
C	1/3	2	10

$L(C, X) = 2$, but $H(X) = \log_2(3) = 1.585...$

Optimal prefix (and therefore UD) codes can be achieved as follows:

Huffman's coding algorithm:

Input: probability dist. P on $\mathcal{A}$

Output: binary tree corresponding to prefix code C with minimal $L(C,P)$

algo:
1. Start with forest of $\#\mathcal{A}$ isolated leaves
2. While more than one tree: merge two trees with smallest probabilities

Example:

X	$P(x)$	$C(x)$
A	0.25 — 0.25 — 0.25 — 0.55 — 1	00
B	0.25 — 0.25 — 0.45 — 0.45	10
C	0.2 — 0.2	11
D	0.15 — 0.3 — 0.3	010
E	0.15	011

$H(P) = 2.28..$ $\qquad$ $L(C,P) = 2.3$

Summary:

Source Coding Theorem for Prefix Codes: Let C be the optimal UD/prefix code for $X \sim P$ (e.g., Huffman's). Then: $H(X) \leq L(C,X) \leq H(X)+1$

should be $\ll H_0(X)$

$H(X)+1$ ... ok if $\mathcal{A}$ large, but terrible when compressing bits

Remedy: Look at blocks and use AEP!

Changing symbols + local correlations

QU

very likely