Compression and Symbol Codes (§5)

Want to compress data source modeled by RV X with <u>known</u> distribution P:

by using Symbol codes, which compress one symbol at a time

WANT:

one or more bits

$$X \longrightarrow \boxed{\;C\;} \longrightarrow \{0,1\}^+ \longrightarrow \boxed{\;D\;} \longrightarrow \hat{X} = X$$

Compressor, encoder — decompressor, decoder

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

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

Let's define things properly...

Symbol code: $C : \mathcal{A} \to \{0,1\}^+$ for alphabet $\mathcal{A}$ — $C(x)$ = how we compress x

* average length when encoding $X \sim P$:

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

what we want to minimize

* extended code:

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

how we encode strings

Two important classes of codes: C is called...

* uniquely decodable (UD) if

$$\omega \neq \omega' \implies C^+(\omega) \neq C^+(\omega') \quad \forall \omega, \omega' \in \mathcal{A}^+$$

can unambiguously decode strings!

* prefix (free) code if <u>no</u> codeword $C(x)$ is prefix of any other

entropy: $H(P) = 1.75$

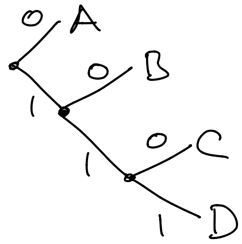
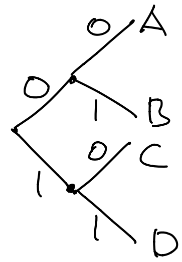
x	$P(x)$	C_3	C_4	C_5	C_6
A	1/2	0	00	0	0
B	1/4	10	01	1	01
C	1/8	110	10	00	011
D	1/8	111	11	11	111
prefix code?		✓	✓	✗	✗
UD?		✓	✓	✗	✓
average length		1.75	2	1.25	1.75

reverse of C_3...

FACT:
Any prefix code is UD!

Prefix codes = binary trees:

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

e.g. $C_3 =$  $C_4 =$ 

What constraints are there on the length of codewords?

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 $\ell_{max} := \max_x \ell(C(x))$. Then:

$$S^N = \sum_{x_1,\dots,x_N} 2^{-\ell(C^+(\overbrace{x_1\cdots x_N}^{N \text{ symbols}}))} \leq \sum_{\ell=1}^{N\cdot\ell_{max}} 2^{-\ell} \cdot \underbrace{\#\left\{\begin{matrix}\text{strings that are}\\ \text{compressed into } \ell \text{ bits}\end{matrix}\right\}}_{\leq 2^{\ell} \text{ by UD}}$$
($\ell(C^+(x_1\cdots x_N)) \leq N\cdot\ell_{max}$)

exp. growth if $S>1$

$$\leq N \cdot \ell_{max} \quad \overset{\forall N}{\Longrightarrow} \quad S \leq 1. \qquad \square$$
linear growth

Kraft's converse: Let $\ell_x \geq 1$ for $x \in \mathcal{A}$ be integers s.th. $\sum_x 2^{-\ell_x} \leq 1$. Then $\exists$ prefix code C with $\ell(C(x)) = \ell_x$ for all $x \in \mathcal{A}$

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

Pf: Construct as follows:
algorithm, but not very efficient

① Order the numbers:
$$\ell_{x_1} \leq \ell_{x_2} \leq \dots \qquad \text{where } \mathcal{A} = \{x_1, x_2, \dots\}$$

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

$\#\{$bitstrings of length ℓ_{x_k} that have one of these as prefix$\}$

$$\leq \sum_{i=1}^{k-1} \underbrace{2^{\ell_{x_k} - \ell_{x_i}}}_{\substack{\#\text{ bitstrings of length } \ell_{x_k}\\ \text{with prefix } C(x_i)}} = 2^{\ell_{x_k}} \cdot \sum_{i=1}^{k-1} 2^{-\ell_{x_i}} < 2^{\ell_{x_k}} \sum_x 2^{-\ell_x} \leq \underbrace{2^{\ell_{x_k}}}_{\substack{\#\text{ bitstrings}\\ \text{of length } \ell_{x_k}}} \qquad \square$$

But what does this mean for the average length? 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!

Equality holds iff $\ell(C(x)) = \log \frac{1}{P(x)}$ $(\forall x)$.

Pf: Define

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

Gibbs

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

$\boxed{\text{Existence of good codes:}}$ $\exists$ prefix codes with $L(C, X) < H(X) + 1$ (assuming X is not deterministic)

Pf: Define $\ell_x = \lceil \log \frac{1}{P(x)} \rceil \geq 1$ ← round up equality condition from above

$* \sum_x 2^{-\ell_x} \leq \sum_x P(x) = 1 \Rightarrow$ by Kraft's converse, there exists a prefix code C with $\ell(C(x)) = \ell_x$

$* L(X, C) = \sum_x P(x) \ell_x < \sum_x P(x) \left(\log \frac{1}{P(x)} + 1 \right) = H(X) + 1.$ $\square$

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

$H(X) = \log_2(3) = 1.585...$

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 we can clearly do better!

0
10
11

$\Rightarrow L = 1.666...$

To find an optimal prefix (and therefore UD) code, can use the following algo:

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) $H(P) = 2.28..$	C(x)
A	0.25	00
B	0.25	10
C	0.2	11
D	0.15	010
E	0.15	011

$L(C,P) = 2.3$

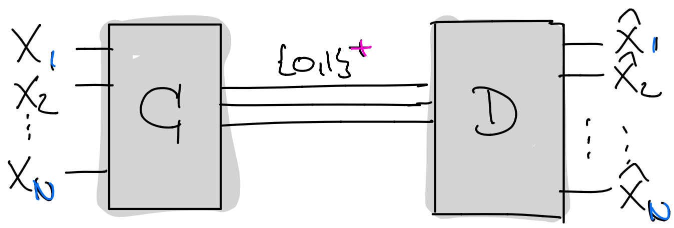
Summary:

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

ok if $H(X)$ large ($\rightarrow \mathcal{A}$ large)
e.g. alphabet of letters

Problem: Completely useless when X is e.g. a bit ϟ

Solution: Compress $\boxed{\text{blocks}}$ of N symbols at a time:



i.e. build code on $\mathcal{A}^N$ for joint distribution of $X_1, \dots, X_N$
$X^N = (X_1, \dots, X_N)$

Result: If $X_1, \dots, X_N \overset{IID}{\sim} P$ then the optimal prefix code satisfies

$$H(P) \leq \frac{L(C, X_1 \cdots X_N)}{N} < H(P) + \frac{1}{N}$$

$\frac{1}{N} \rightarrow 0$ as $N \rightarrow \infty$

$\Rightarrow H(P)$ is optimal asymptotic average rate of compression of IID source

Pf: $H(X_1 \cdots X_N) = N \cdot H(P)$ by the ex. class. □

The IID assumption is not realistic, but a good starting point!

ϟ local correlations ... QU ...
ϟ changing distribution

Two bits of terminology to remember:

* "compression" = "source coding"

* (average) rate of compression = $\dfrac{\text{(average) \# bits used to compress message of length } N}{N}$

NOTATION: R for rate, $\bar{R}$ for average rate