Compression and Symbol Codes (§5)

Consider data source modeled by RV X. Assume we <u>know</u> distribution P_X.
E.g. X could be a letter and we <u>assume</u> $P(x) = P_{english}(x)$

How well can we compress? Today we consider Symbol codes, which compress one symbol (letter, source message, ...) at a time:

WANT:

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

one or more bits

Compressor, encoder

decompressor, decoder

GOAL: Show that lossless compression one symbol at a time can achieve $H(X) \leq L < H(X) + 1$, where L = average length of codeword.

at worst one more bit than entropy

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 $\mathcal{A}$ $\quad$ $C(x)$ = how we compress x

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

 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

<u>Two important classes of codes</u>: C is called...

* uniquely decodable (UD) if

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

 can unambiguously decode strings!

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

FACT:

Any prefix code is UD!

entropy: $H(P) = 1.75$

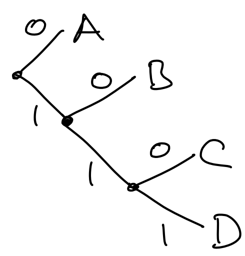
x	$P(x)$	C_3	C_4	C_5	C_6
A	1/2	0	00	0	0
B	1/4	10	01	1	01
C	1/8	110	10	00	011
D	1/8	111	11	11	111
prefix code?		✓	✓	ϟ	ϟ
UD?		✓	✓	ϟ	✓
average length		1.75	2	1.25	1.75

reverse of C_3...

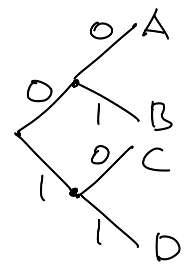
Prefix codes = binary trees:

* 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:

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

$\ell(C^+(x_1 \cdots x_N)) \leq N \cdot \ell_{max}$

exp. growth if $S > 1$

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

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:

$\ell_{x_1} \leq \ell_{x_2} \leq \dots$ 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

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

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

$$\leq 2^{\ell_{x_k}} \quad \#\text{ bitstrings of length } \ell_{x_k} \qquad \square$$

But what does this mean for the average length? Need one more tool...

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

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

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. $\square$

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

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

iff $P=Q$ $\qquad$ $=$ iff $S=$

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

* $\sum_x 2^{-\ell_x} \leq \sum_x P(x) = 1 \Longrightarrow$ 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: The code constructed in the proof is in general NOT optimal. E.g.:

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

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

$L(C_i, X) = 2$

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_i with minimal $L(C_i, P)$

algo: ① Start with "forest" of $\#\mathcal{A}$ isolated leaves

② While more than one tree: merge two trees with smallest probabilities

Example:

X	$P(x)$	$H(P) = 2.28...$	$C_i(x)$
A	0.25	0.55 — 1	00
B	0.25	0.45	10
C	0.2		11
D	0.15	0.3	010
E	0.15		011

$L(C_i, P) = 2.3$

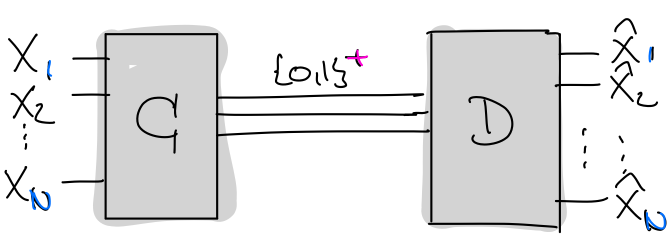
Summary:

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

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

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

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



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

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

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

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

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

Pf: $H(X_1 \cdots X_N) = N \cdot H(P)$ because IID. $\square$

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

↯ local correlations ... QU ...

↯ 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