Information and Compression

File, 100K characters

Each character has 8 possible types a, b, c, d, e, f, g, h
↓
Can be encoded with 3 bits

File: 300K bits

→ you know in addition the frequency of each character

<u>Huffman</u> Coding

<u>Def</u> (Prefix-Freeness)

A variable length encoding is prefix free if the encoding of any character is not a prefix of encoding of any other.

$\underbrace{01}$ a

010 b

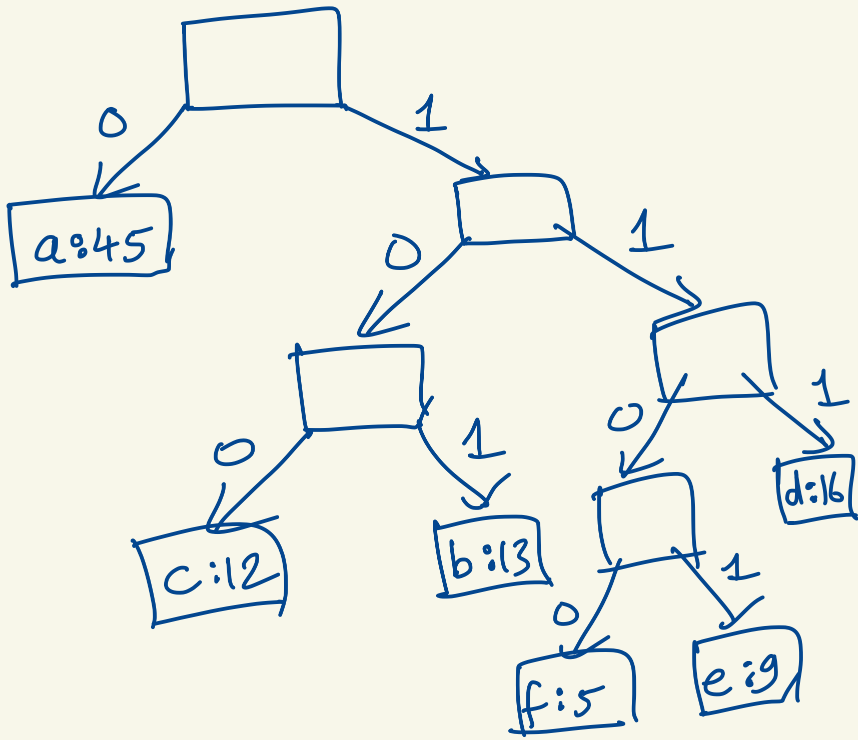
Huffman Coding Algorithm: Greedy Algorithm

A variable length encoding can be described by a tree.

a	b	c	d	e	f	g	h
45K	13K	12K	16K	9K	5K	0K	0K

<u>V.L.E</u>: 0 101 100 111 1101 1100 — —

<u>decode</u>:



1) decoding ≡ tree

2) every leaf is a symbol.

3) if you branch at a node, then both branches lead to proper symbols

---

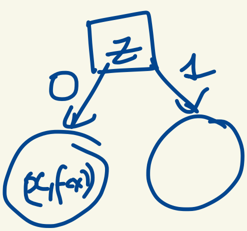
algo: Huffman$(X, f(\cdot))$

$n = |X|$, # of chars.

for all $x \in X$, enqueue$((x, f(x)), Q)$

for $i = 1$ to n,

allocate a tree node z



left-child = delete-min(Q)

right-child = delete-min(Q)

$f(z) = f(\text{left-child}) + f(\text{right-child})$

Make left-child & right-child the children of z

enqueue$((z, f(z)), Q)$.

Runtime: $\Theta(n \cdot \log n)$

Theorem: Alg above produces a tree T s.t. the cost of encoding with T is minimum possible among all prefix free encodings.

Pf: $|X| = 2$ ✓

$|X| > 2$: x, y: lowest freq elements

$$\overline{X} = X - \{x, y\} \cup \{z\}$$

$$|\overline{X}| = |X| - 1$$

Inductively assume that the algo builds the $\overline{T}$ tree for $\overline{X}$.

Goal: by adding x, y as left & right child of z to $\overline{T}$ is also optimal for X.

$$\text{Cost}(T) = \text{cost}(\overline{T}) - f(z) \cdot d(z) + f(x) \cdot d(x) + f(y) \cdot d(y)$$

$$= \text{cost}(\overline{T}) - (f(x) + f(y))(d(x) - 1) + \text{"} + \text{"}$$

$= \text{Cost}(\bar{T}) + f(x) + f(y)$

T': optimal tree.

$$\text{Cost}(T') < \text{Cost}(T) \quad \text{(FSOC)}$$

Assume further that x & y are siblings in T'.

<u>Exercise</u>: Work out the case when x & y are not siblings (HINT: reduce to the above case)

Goal: prove lower bounds on encoding size of File F given frequencies f. $X = \{0,1\}^n$

Thm 1: $C : \{0,1\}^n \to \{0,1\}^*$ lossless encoding (one-one map). Then, there is a file $F \in \{0,1\}^n$ s.t. $|C(F)| \geq n$.

Proof: Suppose not.

$$\#\text{ of possible files} \leq 2^1 + 2^2 + \dots + 2^{n-1}$$
$$= 2^n - 2 < 2^n$$

So contradiction.

Thm 2: $F \in \{0,1\}^n$ be a sequence of n u.a.r. bits. Then,

$$\Pr[\,|C(F)| \leq n-t\,] \leq \frac{1}{2^{t-1}}$$

Proof:

$$= \frac{|\{F : |C(F)| \leq n-t\}|}{2^n}$$

$$= (2^1 + 2^2 + \dots + 2^{n-t})/2^n < \frac{2^{n-t+1}}{2^n}$$

$= \frac{1}{2^{t-1}}.$

<u>Thm 3:</u> C: , prefix free encoding.

F: n-random bits.

Then $\mathbb{E}[|C(F)|] \geq n.$

<u>Thm 4:</u>

$G \subseteq \{0,1\}^n$ be a set of possible files. C: injective function $\{0,1\}^n \to \{0,1\}^*$.

Then:

1) there's a file $F \in G$ s.t. $|C(F)| \geq \log_2 |G| - 1$

2) $\Pr[|C(F)| \leq \log_2 |G| - t] \leq \frac{1}{2^{t-1}}$

3) $\forall$ prefix-free C, F: random

$$\mathbb{E}_F[|C(F)|] \geq \log_2 |C(F)|$$

Given: frequencies of characters.

We will find a set of files G "with typical frequencies"

File contains n characters.

c diff chars possible.

char i has frequency $\frac{f(i)}{n} = p(i)$

Lemma: the number of distinct files of length n and frequencies $f(1), f(2), \dots, f(c)$ is

<u>Proof:</u> If there are 2 chars (i.e. $c=2$), you know this:

$$\binom{n}{f(1)} = \frac{n!}{f(1)!\,(n-f(1))!}$$

Similar formula for $c>2$:

$$|G| = \frac{n!}{f(1)!\,f(2)!\cdots f(c)!}$$

By Thm 4, encoding length

$\sim \log_2 |G|$.

$$= \log_2(n!) - \sum_{i=1}^{c} \log(f(i)!) \,. \quad *$$

Use Stirling's approx.

$$M! \sim (M/e)^M \cdot \sqrt{M} \,.$$

$$\log(M!) \sim M \cdot \log M - \text{"small"}$$

$$* = n \log n - \sum_{i=1}^{c} f(i) \cdot \log f(i)$$

$$= n \cdot \log n - \sum_{i=1}^{c} p(i) \cdot n \cdot \log(p(i) \cdot n)$$

$$= n \cdot \log n - \sum_{i=1}^{c} n \cdot p(i) \cdot \log(p(i))$$

$$- \sum_{i=1}^{c} n\, p(i) \cdot \log(n)$$

Since $\sum_{i=1}^{c} p(i) = 1$, first & last terms cancel

$$= -\sum_{i=1}^{c} n \cdot p(i) \cdot \log(p(i))$$

Now, $-\log p(i) = \log \frac{1}{p(i)}$.

$$= \underbrace{n \sum_{i=1}^{c} p(i) \cdot \log \frac{1}{p(i)}}_{\text{Entropy of freq dist}^n}$$

= minimum possible compressed

size of file with given freq

= "information content" in the file