187 - Day 14

Friday, May 4, 2018 9:30 AM

- Midterm 2 next Friday in class
- Cheat sheet + <u>calculator</u>
- Topic comes later today

Exam 2: Day 7 to 15

Day 14 – Properties of the Entropy (cont.)

Elements of Information Theory

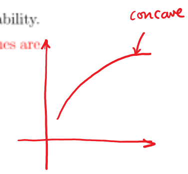
Important inequalities for entropy

1. For a random variable X which takes only k values we always have

$$H(X) \leq \log_2(k)$$

with equality if and only if X takes all its values with equal probability.

<u>Meaning:</u> Maximum uncertainty about X occurs when all its values are equally likely



<u>Fact</u>: If $m_1 + m_2 + \dots + m_k = 1$

and if f is a concave fcn

then

$$\sum_{i=1}^{k} m_i \cdot f(x_i) \leq f\left(\sum_{i=1}^{k} m_i \cdot x_i\right)$$

Here x_i's $\in$ domain of f.

Here, $\log_2(\cdot)$ is a concave fcn.

$$H(X) = \sum_{\substack{\text{all outcomes} \\ a}} \overbrace{\mathbb{P}(X=a)}^{m_i} \cdot \log_2\left(\underbrace{\frac{1}{\mathbb{P}(X=a)}}_{x_i}\right)$$

$$\leq \log_2\left(\sum_a \cancel{\mathbb{P}(X=a)} \cdot \frac{1}{\cancel{\mathbb{P}(X=a)}}\right) = \log_2(k)$$

1

total # of outcomes for X

2. For any two random variables X and Y we always have

$$H(X|Y) \leq H(X)$$

and equality holds if and only if X and Y are independent

Meaning: the info we gain from learning X after we know Y is less than the amount of information we would gain from learning X if we did not know Y

$$H(X|Y) = \sum_b \mathbb{P}(Y=b)\,H(X|Y=b) = \sum_b \mathbb{P}(Y=b) \sum_a \mathbb{P}(X=a|Y=b)\log_2\left(\frac{1}{\mathbb{P}(X=a|Y=b)}\right)$$

$$= \sum_b \mathbb{P}(Y=b) \sum_a \frac{\mathbb{P}(X=a \cap Y=b)}{\mathbb{P}(Y=b)} \cdot \log_2\left(\frac{1}{\mathbb{P}(X=a|Y=b)}\right)$$

$$= \sum_b \sum_a \mathbb{P}(X=a \cap Y=b) \cdot \log_2\left(\frac{1}{\mathbb{P}(X=a|Y=b)}\right)$$

$$= \sum_a \mathbb{P}(X=a) \sum_b \frac{\mathbb{P}(Y=b \cap X=a)}{\mathbb{P}(X=a)} \cdot \log_2\left(\frac{1}{\mathbb{P}(X=a|Y=b)}\right)$$

$$= \sum_a \mathbb{P}(X=a) \cdot \underbrace{\sum_b \mathbb{P}(Y=b|X=a) \cdot \log_2\left(\frac{1}{\mathbb{P}(X=a|Y=b)}\right)}$$

$$\leq \sum_a \mathbb{P}(X=a) \log_2\left(\sum_b \frac{\mathbb{P}(Y=b|X=a)}{\mathbb{P}(X=a|Y=b)}\right)$$

$$= \sum_a \mathbb{P}(X=a) \cdot \log_2\left(\frac{1}{\mathbb{P}(X=a)}\right)$$

$$= H(X)$$

So $H(X|Y) \leq H(X)$.

Here, $\dfrac{\mathbb{P}(Y=b|X=a)}{\mathbb{P}(X=a|Y=b)}$

$$= \frac{\cancel{\mathbb{P}(Y=b \cap X=a)}}{\mathbb{P}(X=a)} \cdot \frac{\mathbb{P}(Y=b)}{\cancel{\mathbb{P}(X=a \cap Y=b)}}$$

$$= \frac{\mathbb{P}(Y=b)}{\mathbb{P}(X=a)}$$

So $\displaystyle\sum_b \frac{\mathbb{P}(Y=b|X=a)}{\mathbb{P}(X=a|Y=b)} = \sum_b \frac{\mathbb{P}(Y=b)}{\mathbb{P}(X=a)}$

$$= \frac{1}{\mathbb{P}(X=a)}$$

2

3. For any two random variables X and Y we always have

$$H(X,Y) \leq H(X) + H(Y)$$

and equality holds if and only if X and Y are independent.

Meaning: the info we gain on learning X and Y simultaneously is less than the info we would gain if we learned them separately

$$H(X,Y) = H(X) + \underset{\substack{\uparrow \\ \text{2nd inequality}}}{H(Y|X) \leq} H(X) + H(Y)$$

Entropy of English

Below is the frequency table for the letters in a sample writing of about 1000 English letters (the Emancipation Proclamation):

letter:	a	b	c	d	e	f	g	h	i	j	k	l	m
frequency:	73	9	30	44	130	28	16	35	74	2	3	35	25

n	o	p	q	r	s	t	u	v	w	x	y	z
78	74	27	3	77	63	93	27	13	16	5	19	1

Here,

$$\text{prob} = \frac{\text{freq.}}{\text{total \# of letters.}}$$

$$\mathbb{P}(\text{letter} = A) = \frac{73}{1000} = 0.073$$

The entropy of English is given by

$$H(X) = \sum_{\alpha=A}^{Z} \mathbb{P}[X=\alpha] \cdot \log_2\left(\frac{1}{\mathbb{P}[X=\alpha]}\right)$$

$$= \mathbb{P}(A).\log_2\left(\frac{1}{\mathbb{P}(A)}\right) + \mathbb{P}(B).\log_2\left(\frac{1}{\mathbb{P}(B)}\right) + \dots + \mathbb{P}(Z).\log_2\left(\frac{1}{\mathbb{P}(Z)}\right)$$

$$= 0.073 \cdot \log_2\left(\frac{1}{0.073}\right) + 0.009 . \log_2\left(\frac{1}{0.009}\right) + \dots + 0.001 . \log_2\left(\frac{1}{0.001}\right)$$

$$\boxed{\text{Entropy of English} \approx 4.1621}$$

↳ Optimally, we need 4.1621 bits to store a letter in English, on average

4

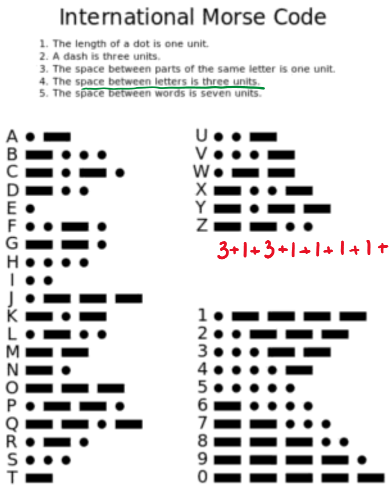
$\bullet\ \bullet\ \bullet\ \bullet$

$=$ H, EEEE, SE, ES, ...

unable to uniquely decrypt w/o the space at the end.

$\bullet\ \text{—}\ \bullet\ \bullet$ end

$1+1\quad 3+1+\quad 1+1+1+\ 3$



A: $1+1+3+3$

E: $1+3$

T: $3+3$

Z: $3+1+3+1+1+1+1+1+3$

Convert

$\bullet := 1$ - bit

$\text{—} := 11$ - bit

space $:= 0$ - bit

end $:= 000$ - bit

$0 = \text{— — —}$

$= 110\,110\,11\,000$

t_α = length of this string

Let T be the time needed to send one letter using Morse code.
Let t_α be the time needed to send the letter α, where $\alpha = A, B, \ldots, Z$.

Then the expectation of T is

$$\mathbb{E}(T) = \sum_{\alpha=A}^{Z} t_\alpha \cdot \mathbb{P}[T = t_\alpha] = \sum_{\alpha=A}^{Z} t_\alpha \cdot \mathbb{P}[\alpha] \approx 7.039$$

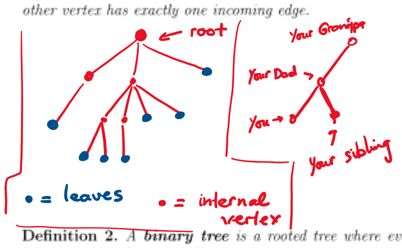
where $\mathbb{P}[\alpha]$ is the probability that a randomly chosen letter from the English alphabet is α.

on averge, we need ≈ 7 bits for 1 letter under Morse code.

5

The Huffman Code

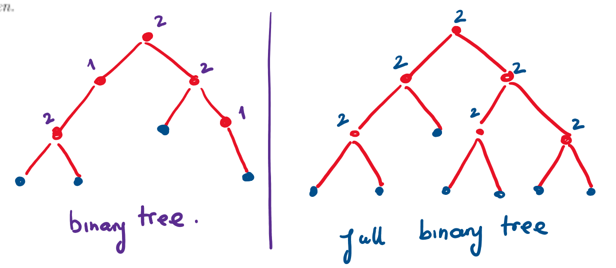
Definition 1. *A* ***rooted tree*** *is a connected directed graph in which one vertex has been designated the root, which has no incoming edges, and every other vertex has exactly one incoming edge.*



Definition 2. *A* ***binary tree*** *is a rooted tree where every internal vertex has* ***no more than*** *2 children.*

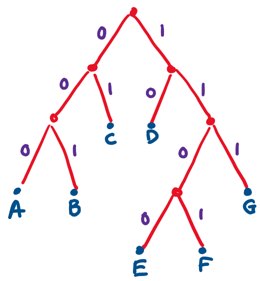
red vertices can only have 1 or 2 children

A ***full*** *binary tree is a rooted tree where every internal vertex has* ***exactly*** *2 children.*



Morse Code is not a comma-free code!

Definition 3. *A binary code is **comma-free** if concatenation of two code words contains a valid code word that overlaps both.*



$\vec{A} = 000$ $E = 1100$

$B = 001$ $F = 1101$

$C = 01$ $G = 111$

$D = 10$

$CA = 01\vdots 000$

To decrypt:

- Start from the root and trace through the branches according to the bits in the code word.
- If we hit a leaf, remove the segment, replace w/ the leaf label.
- Restart from the root.

Question: Given the letter frequencies of a file, which tree will require the least amount of bits? ⇒ The Huffman code.

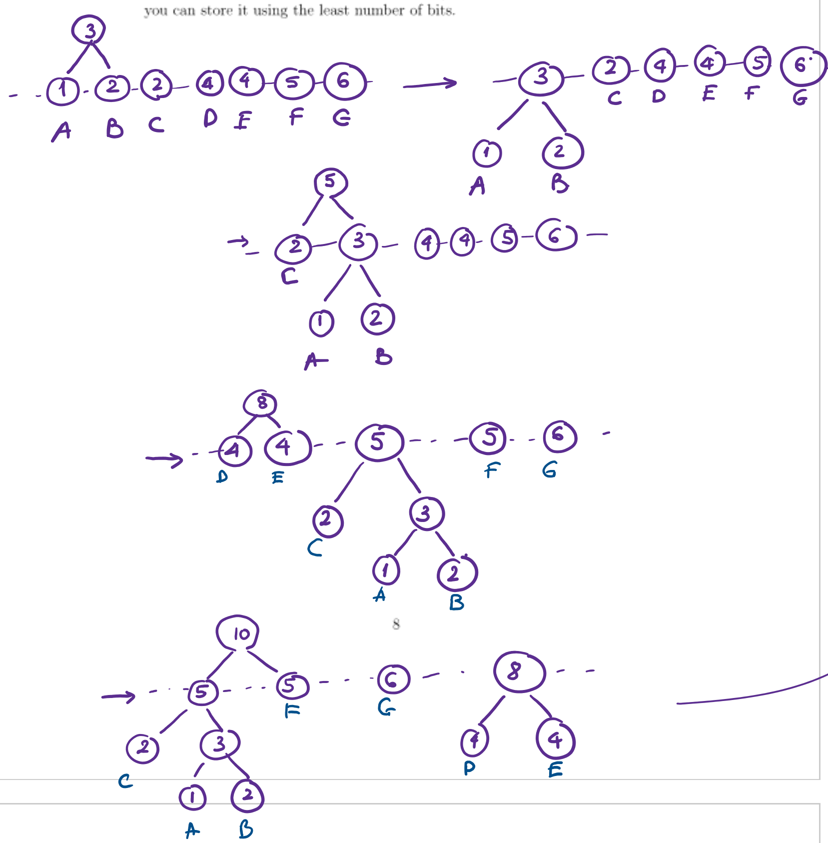
The following algorithm gives the optimal tree:

1. Replace each letter by a node/vertex and label these nodes based on the frequency of each letter. Then sort the nodes by their values in increasing order when reading from left to right.
2. Starting from left to right, group the two smallest numbers together and replace them by their sum.
3. Sort the resulting nodes by their values again. Then repeat these steps until all the nodes are connected.
4. Once we obtain the binary tree, replace the vertex numbers with corresponding letters. Then we label the branches with 0 to the left and 1 to the right.
5. Lastly, we trace along the paths to obtain the code for each letter.

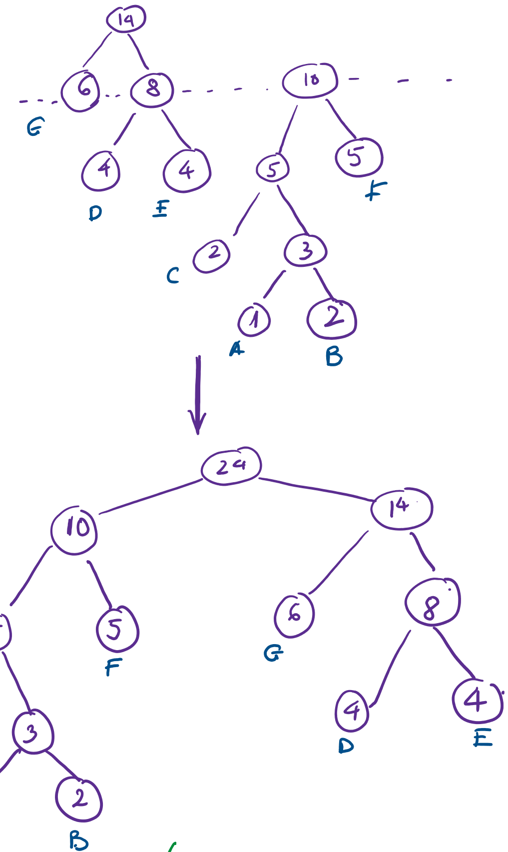
Example. Suppose a certain file contains only the letter with the following frequencies

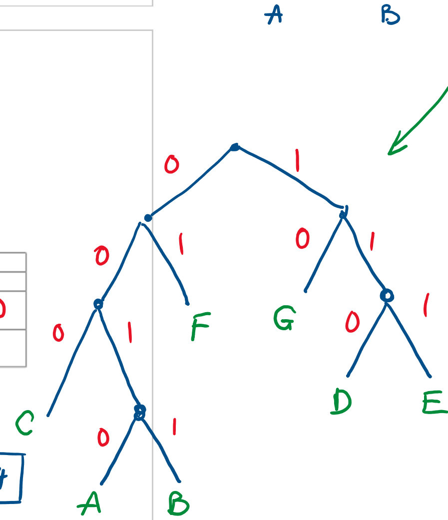
A	B	C	D	E	F	G
1	2	2	4	4	5	6

Construct the comma-free code that enables you to compress the file so that you can store it using the least number of bits.



8





Letter	A	B	C	D	E	F	G
Frequency	1	2	2	4	4	5	6
Code	0010	0011	000	110	111	01	10
Bits	4	4	3	3	3	2	2

File length is (after encrypted)

$$4\cdot N_A + 4\cdot N_B + 3\cdot N_C + \dots + 2\cdot N_G$$
$$= 4\cdot 1 + 4\cdot 2 + 3\cdot 2 + \dots + 2\cdot 6 = \boxed{64}$$

Average number of bits per letter is

$$\frac{\text{after}}{\text{before}} = \frac{64}{24} \approx \boxed{2.66}$$

pretty good :)

Compare to the entropy of the file

$$\sum_\alpha \mathbb{P}(\alpha)\log_2\left(\frac{1}{\mathbb{P}(\alpha)}\right) = \frac{1}{24}\log_2\left(\frac{1}{1/24}\right) + \frac{2}{24}\log_2\left(\frac{1}{2/24}\right) + \frac{2}{24}\log_2\left(\frac{2}{1/24}\right)$$
$$+ \frac{4}{24}\log_2\left(\frac{1}{4/24}\right) + \frac{4}{24}\log_2\left(\frac{1}{4/24}\right)$$
$$+ \frac{5}{24}\log_2\left(\frac{1}{5/24}\right) + \frac{6}{24}\log_2\left(\frac{1}{6/24}\right)$$

$$
\begin{aligned}
&+ \frac{4}{24}\log_2\left(\frac{1}{4/24}\right) + \frac{4}{24}\log_2\left(\frac{1}{4/24}\right) \\
&+ \frac{5}{24}\log_2\left(\frac{1}{5/24}\right) + \frac{6}{24}\log_2\left(\frac{1}{6/24}\right) \\
&\approx 2.62165
\end{aligned}
$$

good

This is the optimal # of bits