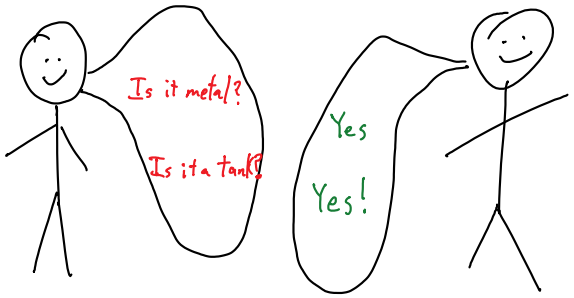
Lecture 19

Monday, March 30, 2015 10:56 AM

20 Questions



Is it animal? No

Is it a plant? No

Is it related to course? Yes

Physical object? Yes

Find in house? No

Is it matal? No

Is it bigger than shoe? Yes

Medical purpose? No

Does it move? No

See it daily? Yes

Electronics? No

Is it in this room? Yes

Large than person? Yes

Building? Yes

Friend Center? Yes

15 Guesses:

If objects are picked from a set $\Omega = \{\text{monkey, ball, apple, tank, ...}\}$

How large can Ω be? $|\Omega| \leq 2^{20} = 1,048,576$

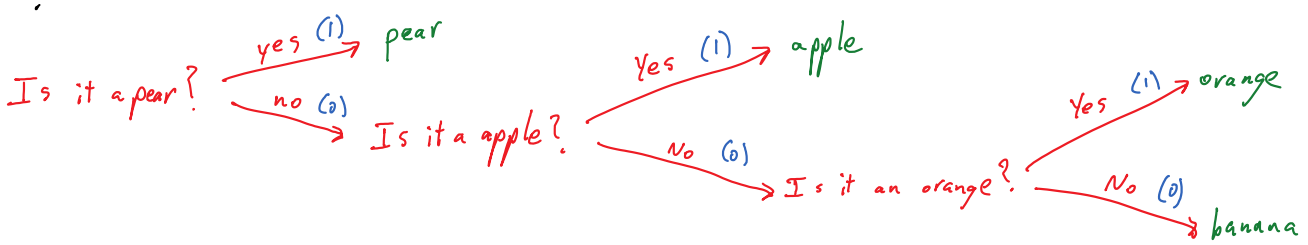
This many combinations of yes-no.
Each question cuts the set in half at best.

Suppose $\Omega = \{\text{apple, orange, pear, banana}\}$

How many question? 2

What if $\text{Prob}(\text{apple}) = \frac{1}{4}$
$\text{Prob}(\text{orange}) = \frac{1}{8}$
$P(\text{pear}) = \frac{1}{2}$
$P(\text{banana}) = \frac{1}{8}$

Minimize the average number of questions needed.



Let X be the answer
and $L(X)$ be the number of questions asked to determine X.

$$\text{Average} = \mathbb{E}L(X) = \sum_{x \in \Omega} p_x(x) L(x) = \frac{1}{4}\cdot 2 + \frac{1}{8}\cdot 3 + \frac{1}{2}\cdot 1 + \frac{1}{8}\cdot 3 = 1.75$$

Ideas for designing optimal question scheme?

* – Find a question that splits prob. in half
- Make that question have fewest number of items.
- Choose set in order
- Sort

Bound: $\mathbb{E}L^*(X) \leq \lceil \log_2 |\Omega| \rceil$

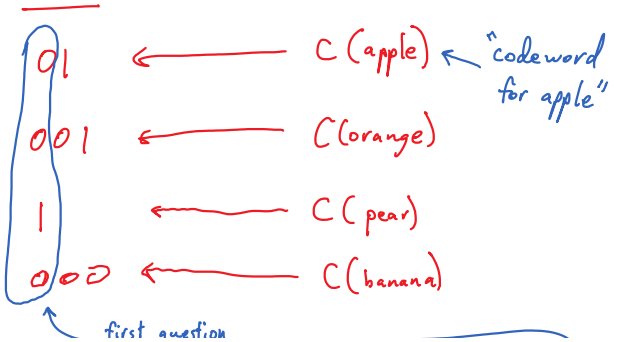
Achieve by using questions to split in half.
i.e. Each object requires the same # of questions.

Question Scheme $\Rightarrow$ Code

Sequence of answers:

Apple :
Orange :
Pear :
Banana :

Code



Banana : 100 ← C(banana)

first question

Notice that the system of questions is implicit in the code!

What are the requirements of the code?

- Unique (Necessary but not sufficient)

Example:

Apple :	0
Orange :	1
Pear :	00
Banana :	01

- Prefix free ("Prefix code"), $C(x)$ is not a prefix of $C(y)$ $\forall y \neq x$

Data Compression (lossless):

Let $X_1, X_2, \ldots$ be a sequence of information that needs to be stored.

Example: - Letters of text
- The output of a quantizer

Let $C(x) \in \{0,1\}^*$ be a code used to store the information

↑ Binary sequences of any length.

$C(X_1)\, C(X_2)\, C(X_3) \ldots$

1.) Non-singular
$C(x) \neq C(y)$ $\forall x \neq y$.
Can decode if there is punctuation.
$C(X_1), C(X_2), \ldots$

2.) Uniquely Decodable.
Can decode without punction.

3.) Prefix Codes.
(defined above)
Instantaneous.

Prefix ⇒ Uniquely Decodable ⇒ Non-singular

↑ We will focus on this.

↑ This is what we need.

No loss of efficiency.

Bounds on Prefix codes:

Kraft inequality: For prefix code: $\sum_{x \in \Omega} 2^{-L(x)} \leq 1$

Notice that any prefix code can be embedded in a binary tree.
Leaves are the objects $x \in \Omega$.

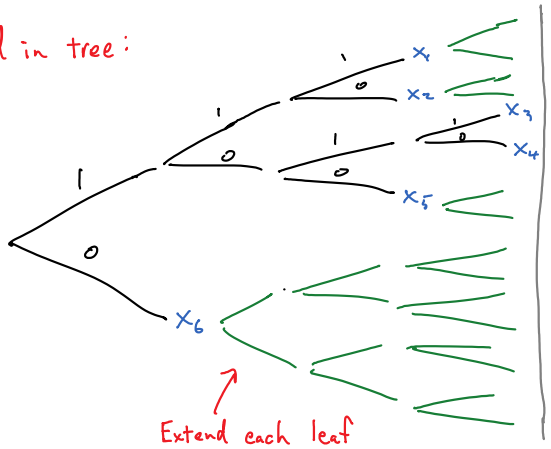
Example:

$C(x_1) = 111$
$C(x_2) = 110$
$C(x_3) = 1011$
$C(x_4) = 1010$
$C(x_5) = 100$
$C(x_6) = 0$

} Prefix code

$L_{max} = 4$

Embed in tree:



Extend each leaf with phantom edges.

$2^{L_{max}}$ phantom leaves.

Item	Phantom leaves	
x_1	2	$= 2^{L_{max} - L(x_1)}$
x_2	2	
x_3	1	
x_4	1	
x_5	2	
x_6	8	

$16 \leq 2^{L_{max}}$

$$\sum_{x \in X} 2^{L_{max} - L(x_i)} \leq 2^{L_{max}}$$

Summary:

1.) Prefix $\Rightarrow \sum_{x \in \Omega} 2^{-L(x)} \leq 1$

2.) $\sum_{x \in \Omega} 2^{-L(x)} \leq 1 \Rightarrow$ There exists a prefix code with lengths $L(x)$

3.) If Kraft inequality is loose, the prefix code is inefficient for any distribution.

(i.e. there is a codeword that can be shortened without changing the others, while maintaining prefix condition.)