Lecture 14

Saturday, April 05, 2014
4:25 PM

Review:

20 Questions = Prefix code

Prefix code $\Rightarrow$ Kraft : $\sum_{x \in \Omega} 2^{-L(x)} \le 1$ $\leftarrow$ Should be tight

Uniquely decodable (no punctuation needed) $\Rightarrow$ Kraft $\leftarrow$ Therefore, prefix is as efficient as uniquely decodable

Kraft $\Rightarrow$ Exists a prefix code with those lengths.

$\uparrow$ Can be fond by embedding in a binary tree.

Kraft $\Rightarrow$ $E L(X) \ge H(X)$ Entropy

- Achieved if $L(x)$ not required to be integer by setting $L(x) = \log_2 \frac{1}{p(x)}$
- Achieved if $\log_2 \frac{1}{p(x)}$ is an integer for each x.

Shannon code : $L_s(x) = \lceil \log_2 \frac{1}{p(x)} \rceil$, $E L_s(X) < H(X) + 1$

Notice that if rounding occurs, Kraft is loose.
$\Rightarrow$ Shannon code is optimal iff $\log_2 \frac{1}{p(x)}$ are integers.

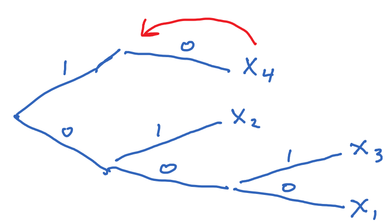
Example: $\Omega = \{x_1, x_2, x_3, x_4\}$

$p(x_1) = \frac{1}{6}$ $\quad L(x_1) = \lceil \log_2 6 \rceil = 3$

$p(x_2) = \frac{1}{3}$ $\quad L(x_2) = 2$

$p(x_3) = \frac{1}{6}$ $\quad L(x_3) = 3$

$p(x_4) = \frac{1}{3}$ $\quad L(x_4) = 2$



Code:

x_1	000
x_2	01
x_3	001
x_4	10

Entropy: Suppose all $x \in \Omega$ are equally likely.

"Uniform distribution" $p(x) = \frac{1}{|\Omega|}$ $\forall x \in \Omega$

$$H(X) = E \log \frac{1}{p(x)} = E \log |\Omega| = \log_2 |\Omega|$$

$$H(X) = E \log \frac{1}{p(x)} = E \log |\Omega| = \log_2 |\Omega|$$

$$= \sum_{x \in \Omega} p(x) \log \frac{1}{p(x)} = \sum_{x \in \Omega} \frac{1}{|\Omega|} \log |\Omega| = \cancel{\frac{1}{|\Omega|}} \log |\Omega| \left(\cancel{\sum_{x \in \Omega} 1} \right)$$

Entropy of a roll of a die:

$$H(X) = \log_2 6 = 2.58 \text{ bits}$$

Entropy of two dice: Let X_1 be the first die.
Let X_2 be the second die.

E.g. $(X_1, X_2) = (1, 5)$

Independence: $P(X_1 = 3 \text{ and } X_2 = 6) = P(X_1 = 3)\ P(X_2 = 6)$

In general, this would be a conditional probability.

$$p(a, b) = \text{Prob}(X_1 = a \text{ and } X_2 = b)$$

$$p(x_1, x_2) = \left(\frac{1}{6}\right)\left(\frac{1}{6}\right) = \frac{1}{36} \quad \forall\, x_1, x_2$$

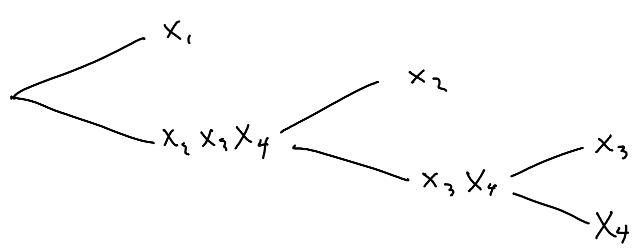
$$H(X_1, X_2) = \log_2 36 = 5.17 \text{ bits} = 2 \cdot (2.58 \text{ bits})$$

If $X_1\ X_2\ X_3 \ldots$ are independent: $H(X_1, X_2, X_3, \ldots) = \sum_i H(X_i)$

in general $H(X_1, X_2, X_3 \ldots) \leq \sum_i H(X_i)$

Shannon-Fano:
- Sort the outcomes by probability
- Make cuts as close to $P = \frac{1}{2}$ as possible

	$P(x)$
x_1	0.45
x_2	0.2
x_3	0.2
x_4	0.15



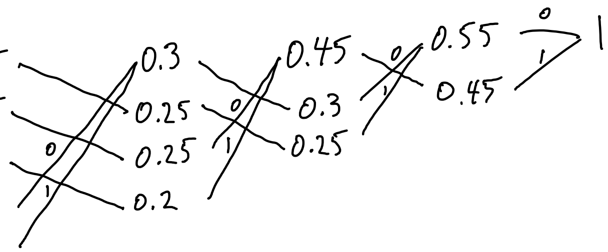
Huffman: 1951

- Design the tree from the leaves.

$\Omega = \{x_1, x_2, x_3, x_4, x_5\}$,

Start by sorting

L	Codewords		$p(x)$
2	01	x_1	0.25
2	10	x_2	0.25
2	11	x_3	0.2
3	000	x_4	0.15
3	001	x_5	0.15



JPEG:

Color is 3-dimensional as perceived by the human eye.

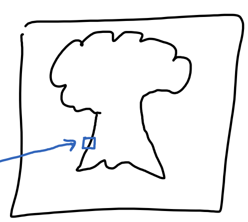
red, green, blue ← basis
red, yellow, blue ← basis

Y, C_B, C_R

Y is intensity
C_B, color } Downsample by 2
C_R, color

Compress each separately.

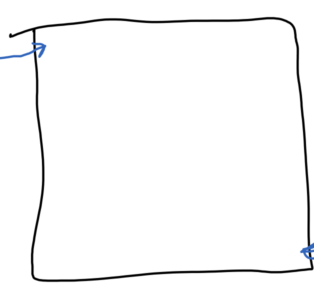
Split into 8×8 pixel blocks:



block

DCT: (think of as DFT)

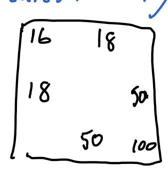
low freq.



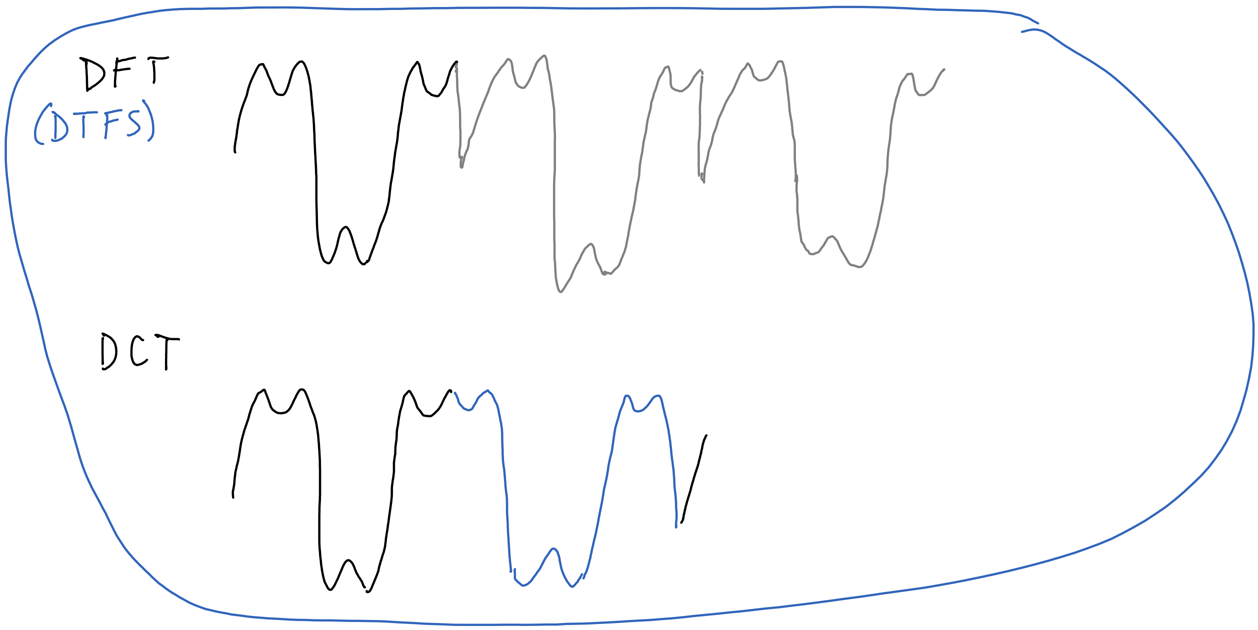
high freq.

Uniform Quantization: Better precision used for low freq.

- Scale each entry (by different values)
- Round to nearest integer.

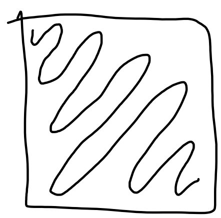


← Matrix of scaling constants.
Fixed part of the protocol.



JPEG Last steps: Lossless

Quantized DCT:



- Run-length code:
  1111100000001111110000
  (5,1), (7,0), (6,1), (4,0)
  5, 7, 6, 4

- Huffman Code

Error-Correction: Add redundancy in a smart way.

Example: Reed-Solomon: In much of today's technology.

We'll investigate this and other error-correction codes in next lecture.