Compression

reduce amount of data to represent same info

data: conveys information

information: ?

redundant data: irrelevant or repeated

b & b' ≡ # bits in 2 rep's for same info

relative data redundancy:

$$R = 1 - \frac{1}{C}$$

where $C = \frac{b}{b'}$ compression ratio

e.g., $b = 100$ & $b' = 10$

then $C = 10 \Rightarrow$ 10 bits in b & 1 bit in b'

90% redundant

b = # bits in image

principal types of redundancy:

Coding code is system of symbols used to represent information
each piece of info is assigned a sequence of code symbols (code word)
symbols in code word ⇒ its length
usually 8-bits in image

Spatial (temporal) spatially correlated video

irrelevant: not used by vision system

Coding redundancy

r_k: random variable in $[0, L-1]$ used to represent intensities in $M \times N$ image

$p(r_k)$: probability of r_k

$$= \frac{n_k}{MN} \quad k = 0, \ldots L-1$$

$n_k \equiv$ count of r_k

G2 | 5

average # of bits / pixel

$$L_{avg} = \sum_{k=0}^{L-1} l(r_k) p_r(r_k)$$

$l(r_k) \equiv$ # bits to represent r_k

total # bits: $MN L_{avg}$

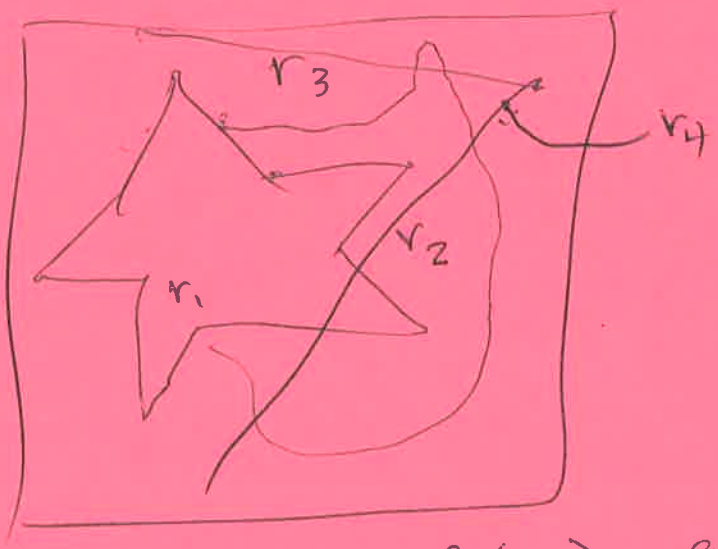
e.g., if $l = m$ bits, then

$$L_{avg} = \sum_{k=0}^{L-1} m\, p_r(r_k)$$

$$= m \sum_{k=0}^{L-1} p_r(r_k)$$

$$= m$$

Consider image in book 6-1



	$P_r(r_k)$	Code1	$l_1(r_k)$	Code2	$l_2(r_k)$
$r_1 = 87$	.25	01010101	8	01	2
$r_2 = 128$	.47	11110000	8	1	1
$r_3 = 186$	.25	10111010	8	000	3
$r_4 = 255$	.03	11111111	8	001	3

Lavg =

$0.25(2) + 0.47(1) + .28(3) = 1.81$ bits

$C = \frac{8}{1.81} = 4.42$

$R = 1 - \frac{1}{4.42} = 0.774$

what about using 2 bits?

$C = \frac{8}{2} = 4$ ≈ 10% worse

8/1

Most images are of some thing, and the intensities are in a narrow range. thus, better to use short codes for these

information theory — modeled as prob. process

E: random event $I(E)$: info contained by E

$$I(E) = \log \frac{1}{P(E)} = -\log P(E)$$

```
> p = [0 : 0.01 : 1];
> b = -log2(p);
> b(56)
> b(76)
> b(end)
```

→ measured in bits

a_j ← source symbols

average info per source output (J events)

entropy:

$$H = -\sum_{j=1}^{J} P(a_j) \log P(a_j)$$

for image

$$\tilde{H} = -\sum_{k=0}^{L-1} p_r(r_k) \log_2 p_r(r_k)$$

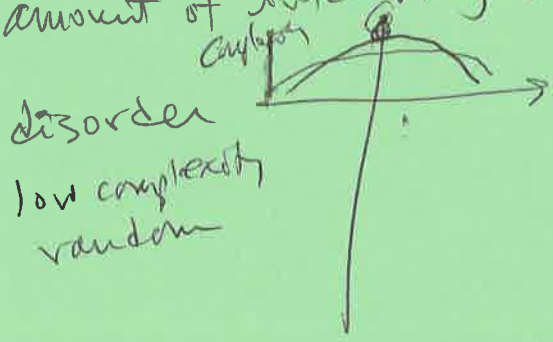
How can this measure be used? To pick compression method

8/11a

Complexity measures

* How difficult to describe or build system
  description length of string
  shortest computer program that generates string

* Amount system is organized or
  amount of interesting structure



high when coexist

order - disorder related to
information + entropy

Shannon: entropy is high if many states
have equal probability
next symbol has most info when
all are equally likely

fidelity criteria

quantify loss : objective vs. subjective

$$f(x,y) \xrightarrow[\text{compress}]{} f_c(x,y) \xrightarrow[\text{decompress}]{} \hat{f}(x,y)$$

error $\quad e(x,y) = \hat{f}(x,y) - f(x,y)$

$$e_{total} = \sum_{x=1}^{M} \sum_{y=1}^{N} \left[\hat{f}(x,y) - f(x,y)\right]$$

root-mean-square error

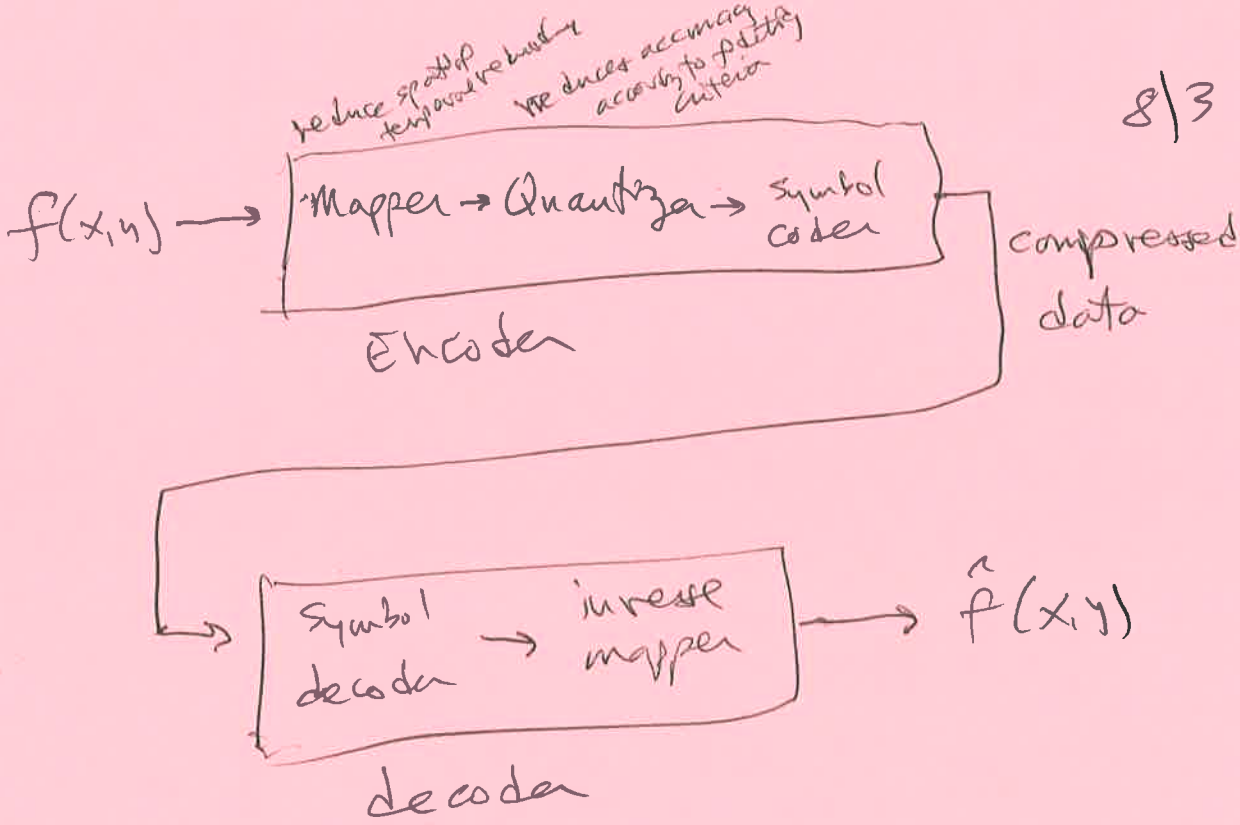
$$e_{rms} = \left[\frac{1}{MN} \sum_{x=1}^{M} \sum_{y=1}^{N} \left[\hat{f}(x,y) - f(x,y)\right]^2\right]^{1/2}$$

mean squared ~~error~~ signal-to-noise ratio

$$SNR_{ms} = \frac{\sum_{x=1}^{M} \sum_{y=1}^{N} \hat{f}(x,y)^2}{\sum_{x=1}^{M} \sum_{y=1}^{N} \left[\hat{f}(x,y) - f(x,y)\right]^2}$$

$$SNR_{rms} = \sqrt{SNR_{ms}}$$

root



codec

if $f = \hat{f}$ error free, lossless, info preserving

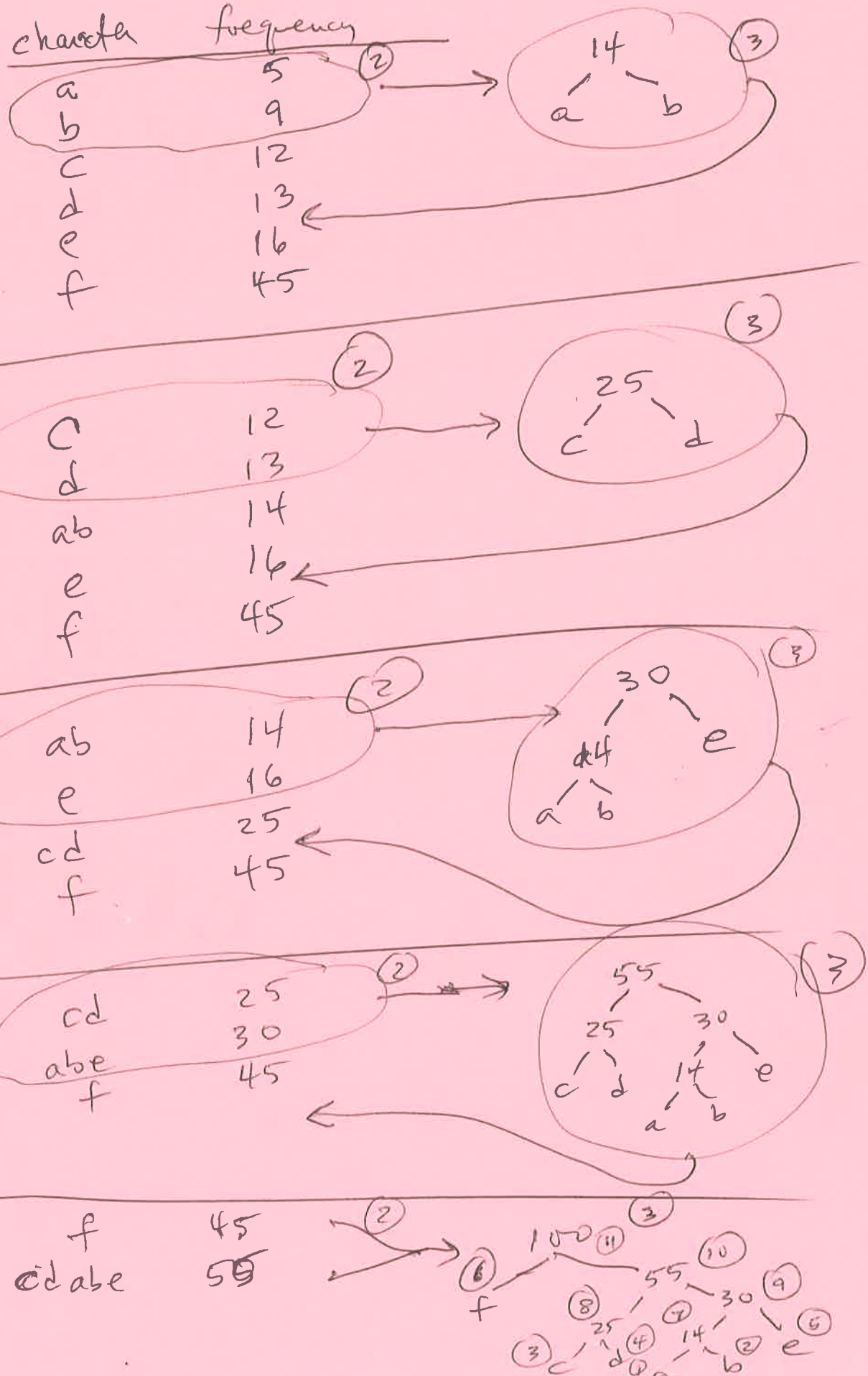
else lossy

image file format : organization standard

image container : multiple types of image data

compression standards : de(compression) procedures

8/5


charcter
frequency
a 5
b 9
c 12
d 13
e 16
f 45
14
a b
c 12
d 13
ab 14
e 16
f 45
25
c d
ab 14
e 16
cd 25
f 45
30
ab e
cd 25
f 45
55
cd 25
abe 30
f 45
f 45
cdabe 55
100
55
30
25
14
e
c d
a b

8/4

Huffman Coding

algorithm (www.geeksforgeeks.org)

Huffman Tree

input: array of unique characters & frequencies

output: Huffman tree

1. create sorted (by frequencies) list of characters (ascending)
2. Extract (remove) two lowest frequency nodes, n_1 & n_2
3. Create new node, n, whose frequency is sum from $n_1 + n_2$ with n_1 as left child & n_2 as right child

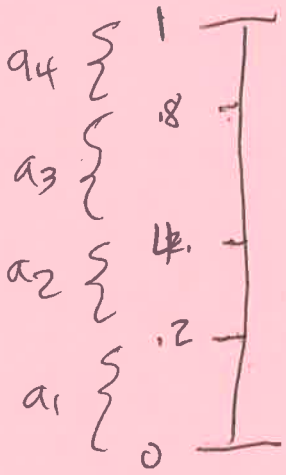
   Insert into sorted list
4. Repeat 2 & 3 until only 1 node

Huffman Codes

Traverse tree to every leaf; for left move assign 0; for right assign 1. Code is sequence generated by path.

Arithmetic coding

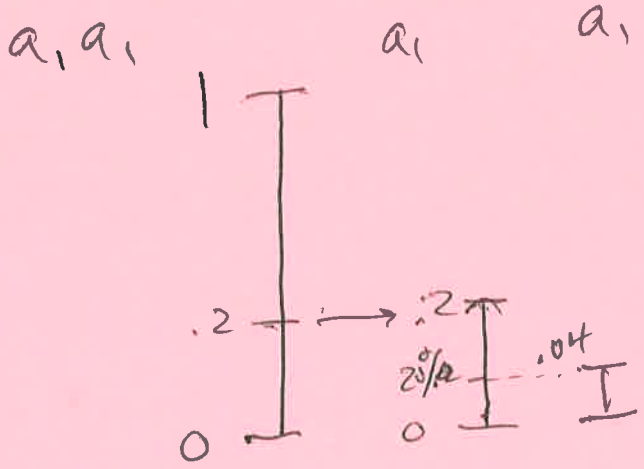
Given symbols, probabilities and a sequence of symbols, convert to an interval in $[0, 1]$.

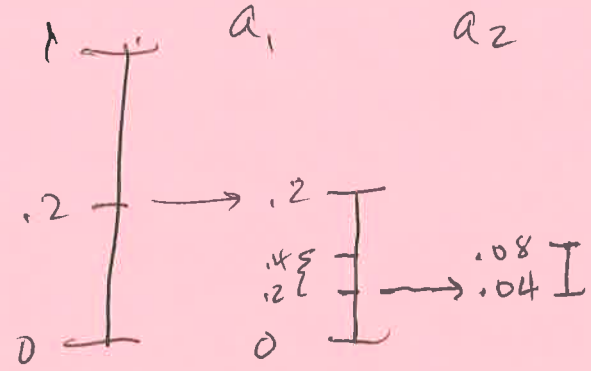


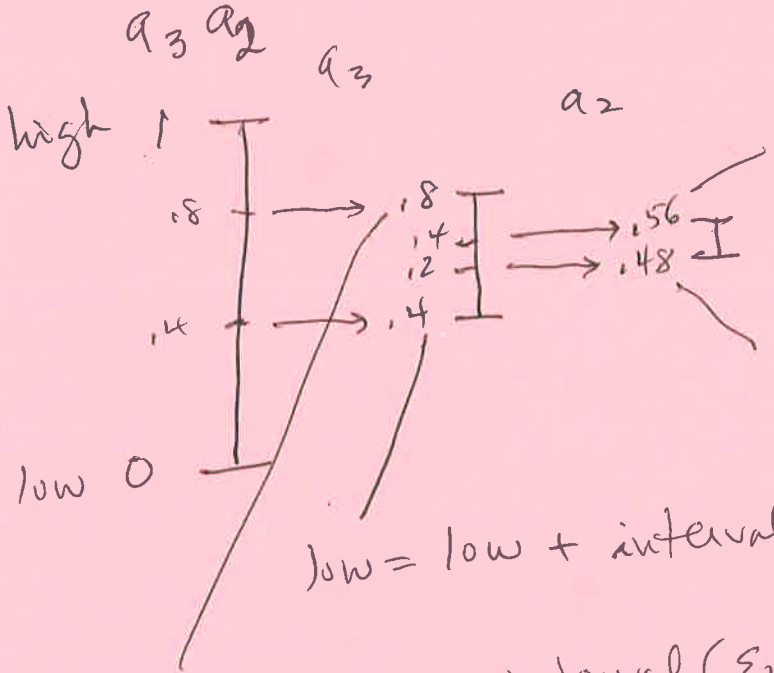
if string has 1 symbol, then know which one by interval

$a_1 \rightarrow [0, .2)$ $a_2 \rightarrow [.2, .4)$ . . .

if string has 2 symbols then find interval as:
1st symbol restricts to its interval,
2nd symbol restricts to its interval on 1st interval







low = low + interval(s,1) * (high - low)

high = low + interval(s,2) * (high - low)

see week7 code