EE274 Lecture 4

Huffman Codes

Oct. 9, 2023

Announcements

- HW1 patches on Ed
  - start early on homeworks!
- SCL tutorial
- SCL on Windows

Recap

- Kraft's inequality
- Entropy $H(X) = \sum_{i=1}^{k} p_i \log_2 1/p_i$ bits
- Joint entropy for i.i.d. variables:

$$H(X^n) = \sum_{i=1}^{n} H(X_i) = nH(X)$$

- KL-divergence : $D(p\|q) \geq 0$ w/ equality iff $p=q$

Recap

Main Result:

1. For every prefix code: $\mathbb{E}\,\ell(X) \geq H(X)$

2. Can achieve $\mathbb{E}\,\ell(X) \approx H(X)$ with prefix codes on blocks.

Achieving $H(X)$

Shannon codes: $H(X) \le \mathbb{E}\,\ell(X) < H(X)+1$

- do better than this - TODAY!

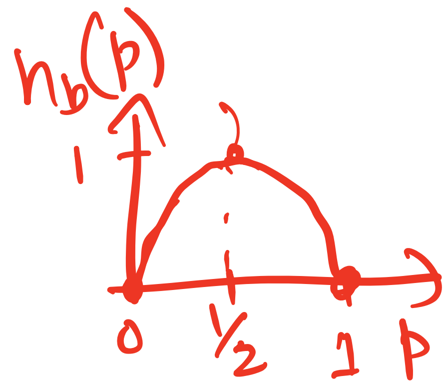
Blocks of n: $H(X) \le \dfrac{\mathbb{E}\,\ell(X^n)}{n} < H(X) + \dfrac{1}{n}$

- practical imblementations next week

Quiz

$\mathrm{Ber}(p) \rightarrow$ $\mathcal{X}=\{0,1\}$

$P(X=0)=1-p \;,\; P(X=1)=p$



1.1 $H(X) = \underbrace{p\log_2 1/p + (1-p)\log_2 \frac{1}{1-p}}_{h_2(p)=h_b(p)}$

1.2 $D(\mathrm{Ber}(p)\|\mathrm{Ber}(q)) = p\log_2 p/q + (1-p)\log_2 \frac{(1-p)}{(1-q)}$

1.3 $D(\mathrm{Ber}(0.3)\|\mathrm{Ber}(0.7)) = 0.4889\ldots$

Quiz

1.4 $\max\limits_{p \in [0,1],\, q \in [0,1)} D(\mathrm{Ber}(p) \| \mathrm{Ber}(q))$

infinity

1.5 Is $D(\mathrm{Ber}(p) \| \mathrm{Ber}(q)) = D(\mathrm{Ber}(q) \| \mathrm{Ber}(p))$?

No

$p = 0.3$

$q = 0.4$

Quiz

Q2: $X \sim Ber(0.001)$

2.1 Shannon code

X	$P(x)$	$\lceil \log_2 1/p(x) \rceil = l(x)$	
0	0.999	1	0
1	0.001	10	1000000000

$$\mathbb{E}[l(x)] = 1.009 = 0.999 \times 1 + 0.001 \times 10$$

Quiz

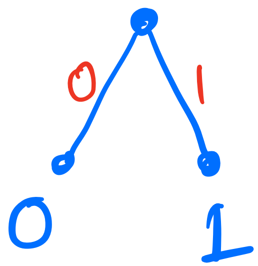
Q2 $\quad X \sim \mathrm{Ber}(0.001)$

2.2 $\quad H(X) \approx 0.011$

2.3 $\quad \mathbb{E}\,\ell(X)/H(X) \approx 88$

Optimal code for Ber(0.001)

Block size 1



$\mathbb{E}[\ell(x)] = 1$ bit/symbol

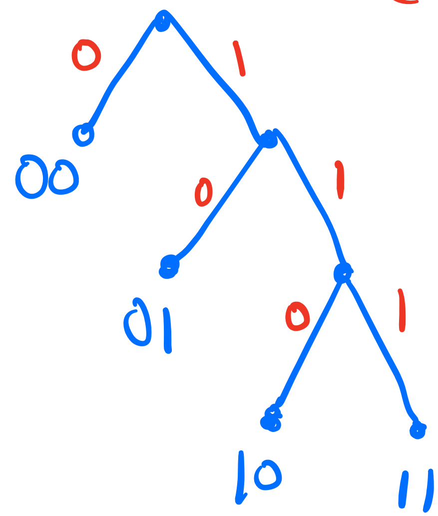
$\gg 0.011 = (H(x))$

Optimal code for Ber(0.001)

Block size 2

$P(x_1, x_2) = P(x_1)\,P(x_2)$ (due to independence)

x^2	$P(x^2)$	$c(x)$
00	0.998001	0
01	0.000999	10
10	0.000999	110
11	0.000001	111



$$\frac{E\left[\ell(x^2)\right]}{2} = \frac{1.002999}{2} \approx 0.501 \text{ bits/symbol}$$

Closer to entropy 0.011!

Outline

- Optimal Prefix Code Conditions
- Huffman Code Construction
- Huffman Coding in Practice

Optimal Prefix codes

$$\min_{l_i} \sum_{i=1}^{k} p_i l_i$$

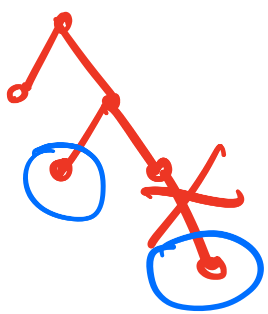
s.t. prefix free cond^n / $\sum_{i=1}^{k} 2^{-l_i} \leq 1$

Conditions for optimality

1. If $p_i > p_j$, $l_i \le l_j$

Otherwise swap the codewords

2. The two longest codewords have the same length.



Why still prefix free?
— If shortening violates prefix property, argue that original code also violates

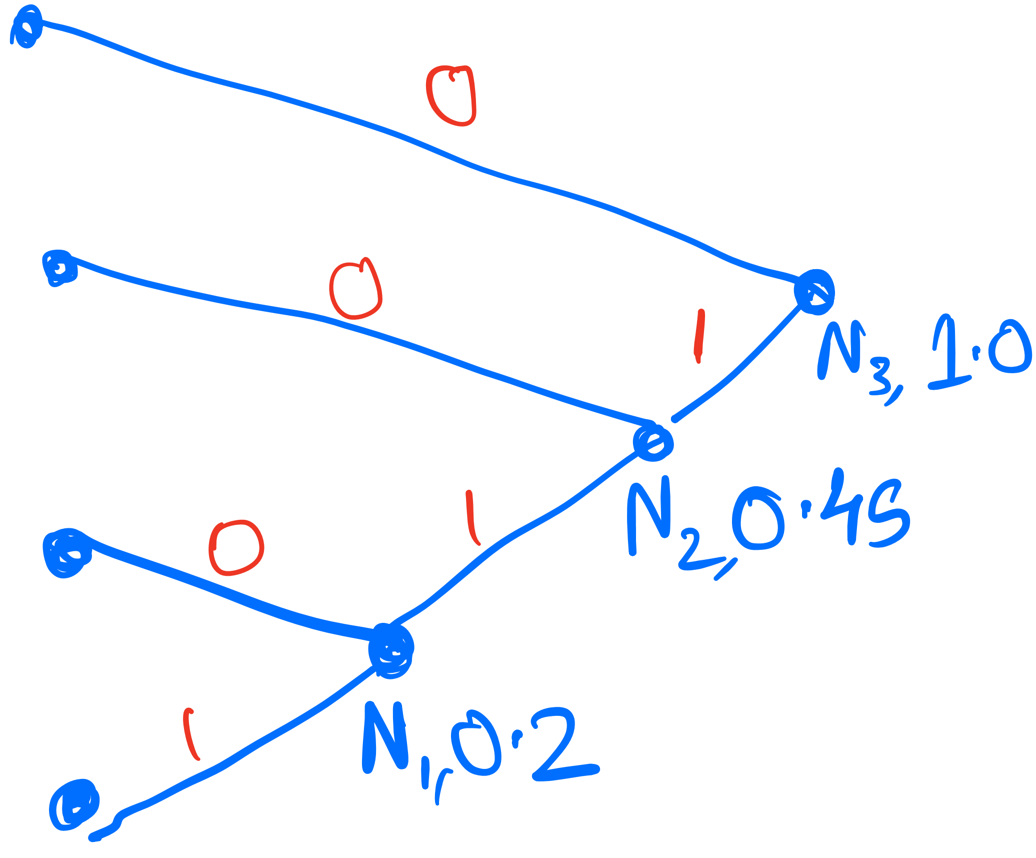
Huffman Code Construction

1. List of nodes = {(symbol, prob.)}
2. While more than one node left:
   i) pick 2 nodes with least prob.*
   ii) merge the 2 nodes:
      - create new node
         - w/ the 2 nodes as children
         - prob. = sum of prob. of children
3. Last remaining node is the root.

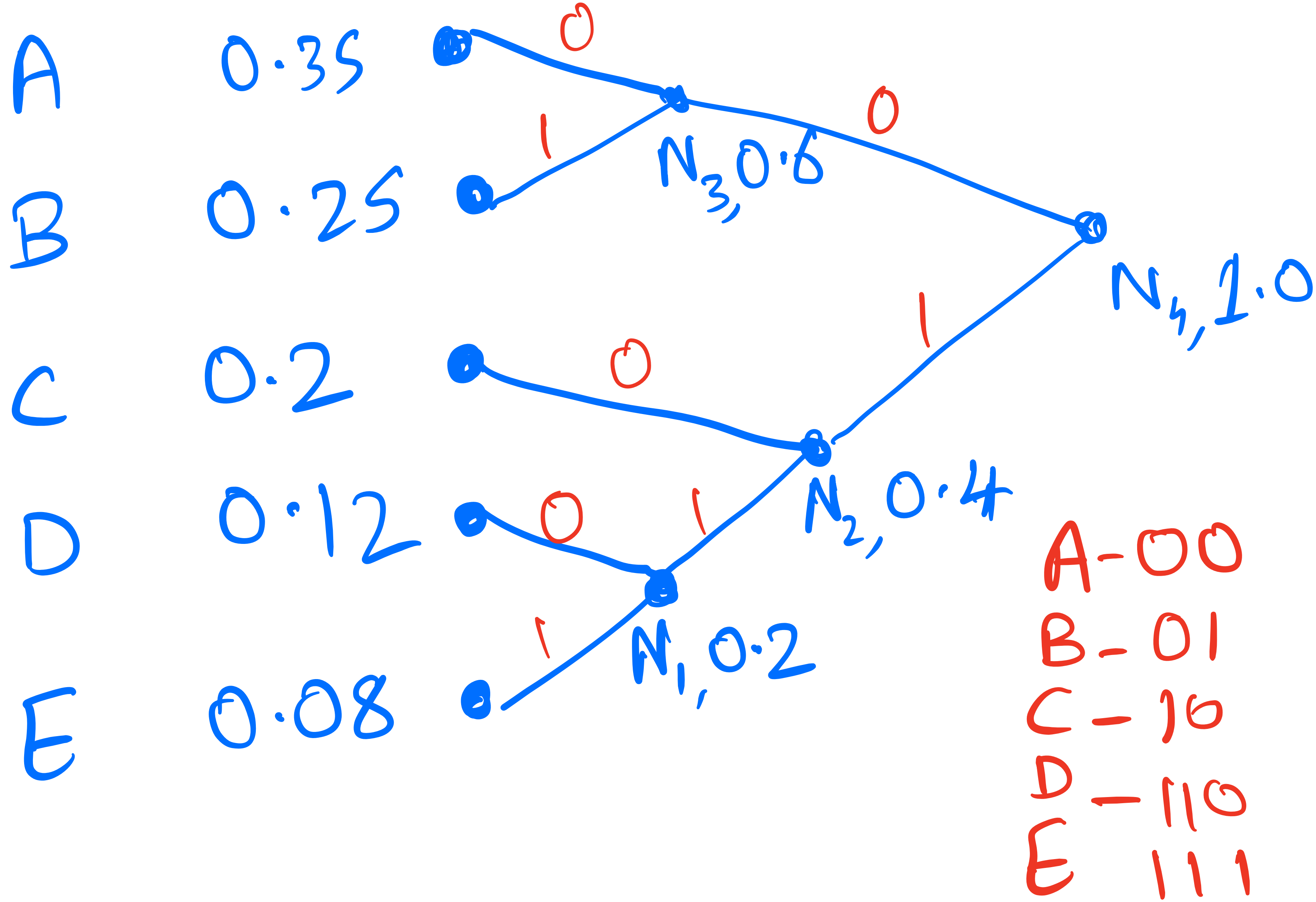
* Break ties arbitrarily

Huffman Example 1

X	P(x)
A	0.55
B	0.25
C	0.13
D	0.07



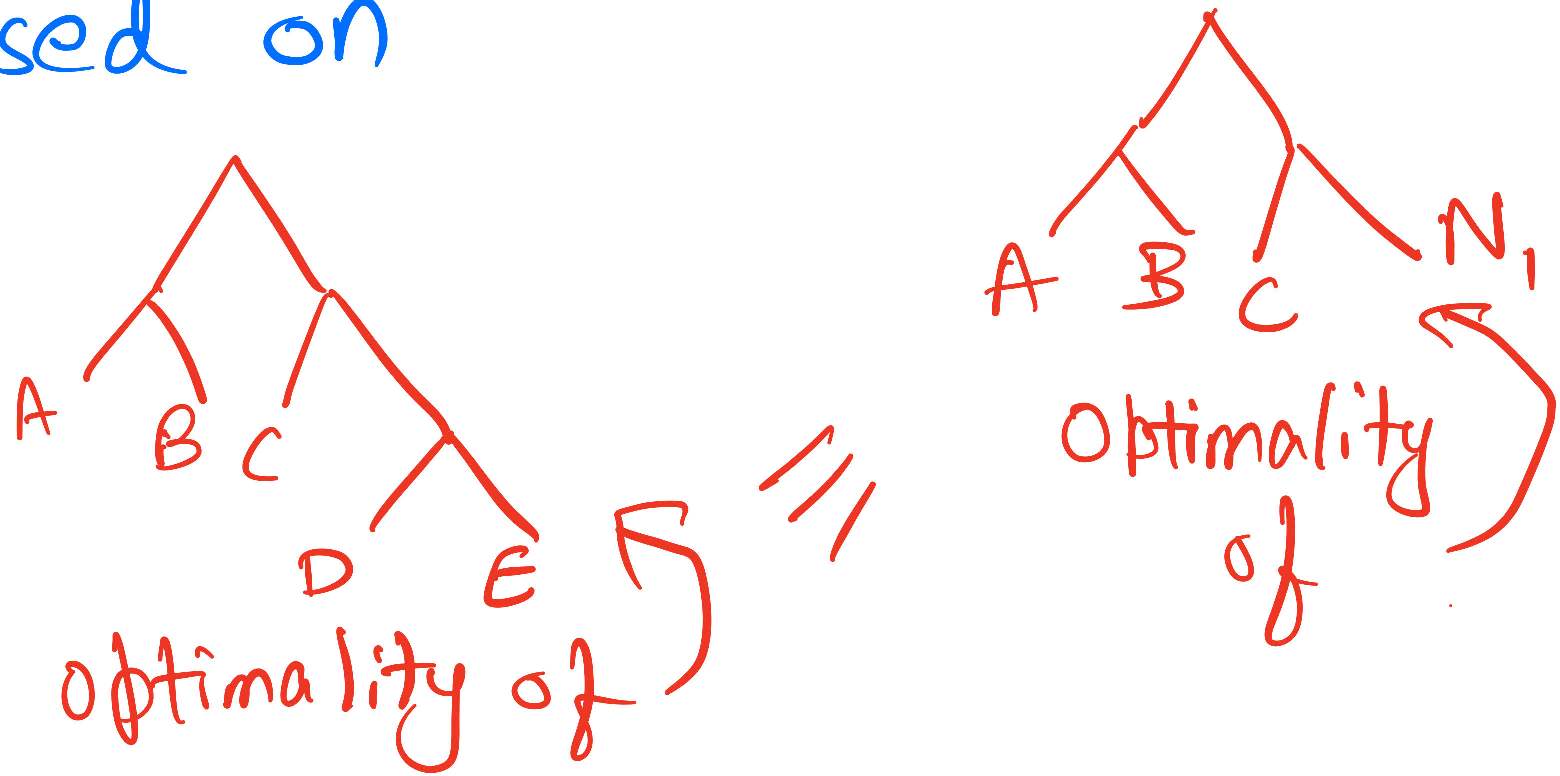
Huffman Example 2



Optimality

See Cover & Thomas Ch. 5

Based on



Huffman codes

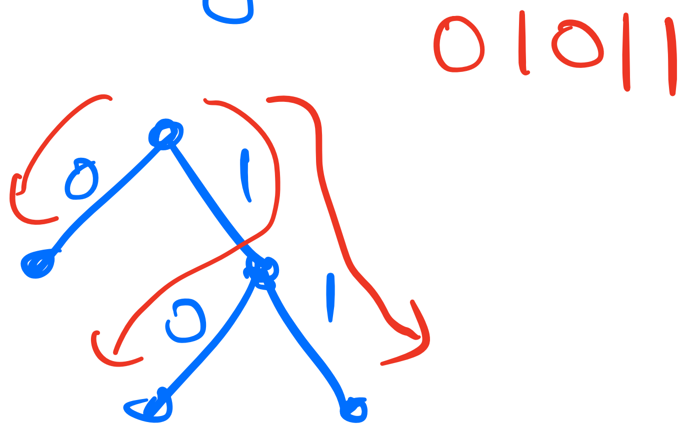
- Greedy algorithm
- Works for general $w_i \geq 0$

$$\sum_{i=1}^{k} w_i l_i \quad \text{s.t. prefix code}$$

- Tie breaking $\Rightarrow$ multiple possible Huffman codes
- $H(X) \leq \mathbb{E}[l_{Huff}(X)] \leq \mathbb{E}[l_{Shan.}(X)] < H(X)+1$

Decoding :→

- Tree based decoding



- Too many branches (if 0, left
  if 1, right)
- Bad for modern computer architectures

Table based decoding

A - 0
B - 10
C - 110
D - 111

decode_state_table

000 - A
001 - A
010 - A
011 - A
100 - B
101 - B
110 - C
111 - D

encode_len

A - 1
B - 2
C - 3
D - 3

```
def decode_symbol_fast(bitarray):
    state = bitarray[:3]
    s = decode_state_table[state]
    num_bits = encode_len[s]
    return s, num_bits
```

Table based decoding

- Size of table = $2^{|max\_depth|}$
- Want to fit in cache
  - → Limit max depth → 16, 24
- Constrained Huffman code
  [best code with max depth constraint]

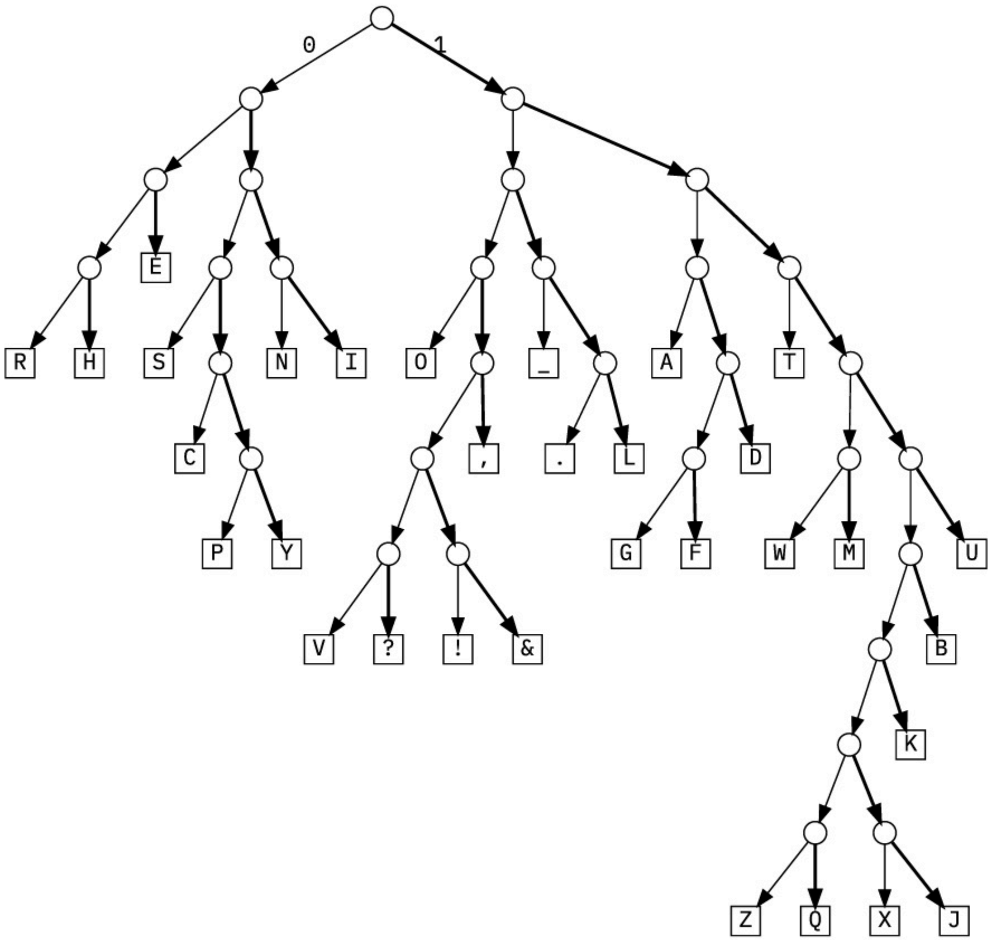
Table based decoding

$$P = \left(\frac{1}{33}, \frac{1}{33}, \frac{2}{33}, \frac{3}{33}, \frac{5}{33}, \frac{8}{33}, \frac{13}{33}\right)$$

Fibonacci

Huffman code: HW!

Max depth $\approx |\mathcal{X}|$


0
1
E
R
H
S
N
I
O
_
A
T
C
,
.
L
D
P
Y
G
F
W
M
U
V
?
!
&
B
K
Z
Q
X
J


Simon Pampena
@mathemaniac
Letter Frequency Keyboard Histogram
4:16 AM - 11 Jan 2015
321 Retweets 288 Likes

Huffman coding SCL demo

https://colab.research.google.com/drive/15eCkqs1FcGMhWaYHjVrABH6eNgXW_Gvj?usp=sharing

Huffman coding in practice

Deflate/gzip:
https://datatracker.ietf.org/doc/html/rfc1951

http/2 header compression:
https://www.rfc-editor.org/rfc/rfc7541#appendix-B

JPEG Huffman coding tables:
https://www.w3.org/Graphics/JPEG/itu-t81.pdf

K.3.1 Typical Huffman tables for the DC coefficient differences

Tables K.3 and K.4 give Huffman tables for the DC coefficient differences which have been developed from the average statistics of a large set of video images with 8-bit precision. Table K.3 is appropriate for luminance components and Table K.4 is appropriate for chrominance components. Although there are no default tables, these tables may prove to be useful for many applications.

Table K.3 – Table for luminance DC coefficient differences

Category	Code length	Code word
0	2	00
1	3	010
2	3	011
3	3	100
4	3	101
5	3	110
6	4	1110
7	5	11110
8	6	111110
9	7	1111110
10	8	11111110
11	9	111111110

What's next?

- Theoretical intuition behind entropy, block coding
- Practical block/stream codes to get closer to entropy

THANK

YOU!