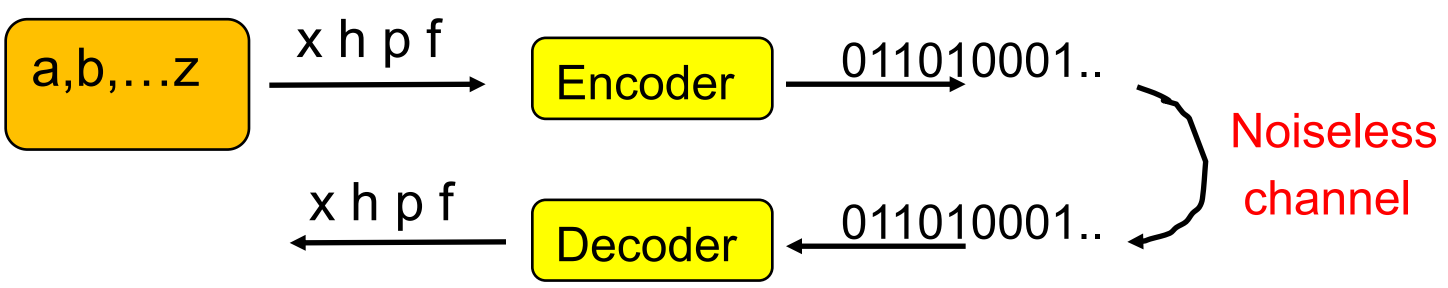
Measure of efficiency for codes

- Expected length of the code

$$L_{\exp} = \sum_{i=1}^{n} p_i l_i$$

- Communication efficiency
  - not computational efficiency



Example

alph	code	prob
a	101	0.125
b	100	0.125
c	11	0.25
d	0	0.5

- Source $\mathcal{X}$= { a, b, c, d}
- p_a=p_b= 0.125, p_c= 0.25, p_d= 0.5
- Code I: d→ 0, c→ 11 a→ 101 b→100
- L_{exp}= 1 x 0.5 + 2 x 0.25 + 2 x 3 x 0.125
- L_{exp}(Code I) = 1.75 binary digits
- Code II: d→ 00 c→ 11 a→ 10 b→01
- L_{exp}(Code II) = 2 binary digits

Noiseless coding theorem

Fundamental Theorem of source coding

- Theorem: $L_{exp} \geq H(X)$
  The expected length of any prefix code for a DMC source in lower bounded by the entropy of the source.

(proof omitted – can be found in CT, or reference below)

- That is, the expected number of binary digits to encode a source will be at least equal to the average information of the source.

http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-450-principles-of-digital-communications-i-fall-2006/lecture-notes/book_2.pdf

Entropy of the source

$H(X) = -\Sigma_i \, p(x_i) \log_2 p(x_i)$

$- \log_2 p(a) = - \log_2 p(b) = 3$ bits

$- \log_2 p(c) = 2$ bits

$- \log_2 p(d) = 1$ bits

$H(S) = \Sigma_i \, p(s_i) \, [- \log_2 p(s_i) \,] =$

$0.5 \times 1 + 0.25 \times 2 + 0.125 \times 3 \times 2 = 1.75$ bits (information)

$H(S) = L_{exp}$ (Code I)

Code I has optimal expected length.

Summary

- Entropy is a measure of
  - Information
  - uncertainty
- Entropy is used for measuring password strength
- Probabilistic modeling of information source
- Encoding source output using binary digits
  - Efficient codes
- Expected length of the best code
- A code with the shortest expected length

Codes with shortest expected length

Theorem

- Let X be a DMS with symbol probabilities $p_1, \ldots, p_m$.
- Let L_{min} be the smallest expected codeword length over all prefix-free codes for X.
- Then

$$H(X) \leq L_{min} < H(X) + 1 \qquad \text{bit/symbol}$$

.

Optimal codes

- A source code with shortest expected length is called optimal.

- Huffman code is a prefix-free optimal source code
  → Lossless compression
  Assumes source distribution is known.

- Universal source coding algorithms do not assume the source distribution is known.
  - ZIP is based on Ziv-Lempel algorithm
    - Optimal
    - Universal

Huffman code

(David Huffman 1952)

- A source with alphabet {1,2,3,4,5} and probabilities {0.25, 0.25,0.2, 0.15, 0.15}

X	$P(X)$
1	0.25
2	0.25
3	0.2
4	0.15
5	0.15

- In 1951, David A. Huffman and his MIT information theory classmates were given the choice of a term paper or a final exam. The professor, Robert M. Fano, assigned a term paper on the problem of finding the most efficient binary code. Huffman, unable to prove any codes were the most efficient, was about to give up and start studying for the final when he hit upon the idea of using a frequency-sorted binary tree and quickly proved this method the most efficient.[3].

wikipedia

(1925-1999)

Huffman code
(David Huffman 1952)

- A source with alphabet {1,2,3,4,5} and probabilities {0.25, 0.25,0.2, 0.15, 0.15}

X	$P(X)$
1	0.25
2	0.25
3	0.2
4	0.15
5	0.15

Huffman code

- A source with alphabet {1,2,3,4,5} and probabilities {0.25, 0.25,0.2, 0.15, 0.15}

X	$P(X)$	
1	0.25	0.3
2	0.25	0.25
3	0.2	0.25
4	0.15	0.2
5	0.15	

Huffman code

- A source with alphabet {1,2,3,4,5} and probabilities {0.25, 0.25,0.2, 0.15, 0.15}

X	$P(X)$		
1	0.25	0.3	0.45
2	0.25	0.25	0.3
3	0.2	0.25	0.25
4	0.15	0.2	
5	0.15		

Huffman code

- A source with alphabet {1,2,3,4,5} and probabilities {0.25, 0.25,0.2, 0.15, 0.15}

X	$P(X)$			
1	0.25	0.3	0.45	0.55
2	0.25	0.25	0.3	0.45
3	0.2	0.25	0.25	
4	0.15	0.2		
5	0.15			

Huffman code

- A source with alphabet {1,2,3,4,5} with probabilities {0.25, 0.25,0.2, 0.15, 0.15}

X	$P(X)$				
1	0.25	0.3	0.45	0.55	0
2	0.25	0.25	0.3	0.45	1
3	0.2	0.25	0.25		
4	0.15	0.2			
5	0.15				

Huffman code

- A source with alphabet {1,2,3,4,5} with probabilities {0.25, 0.25,0.2, 0.15, 0.15}

X	$P(X)$			
1	0.25	0.3	0.45	0.55 0
2	0.25	0.25	0.3 0	0.45 1
3	0.2	0.25	0.25 1	
4	0.15	0.2		
5	0.15			

Huffman code

- A source with alphabet {1,2,3,4,5} with probabilities {0.25, 0.25,0.2, 0.15, 0.15}

X	$P(X)$			
1	0.25	0.3	0.45	0.55 0
2	0.25	0.25	0.3 0	0.45 1
3	0.2	0.25 0	0.25 1	
4	0.15	0.2 1		
5	0.15			

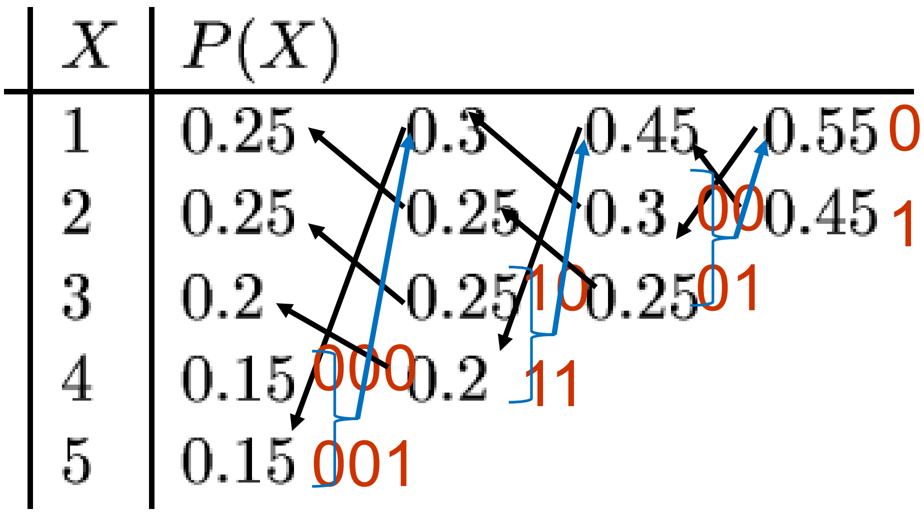
Huffman code

- A source with alphabet {1,2,3,4,5} with probabilities {0.25, 0.25,0.2, 0.15, 0.15}

X	$P(X)$			
1	0.25	0.3	0.45	0.55 0
2	0.25	0.25	0.3 0	0.45 1
3	0.2	0.25 0	0.25 1	
4	0.15 0	0.2 1		
5	0.15 1			

Huffman code

- A source with alphabet {1,2,3,4,5} with probabilities {0.25, 0.25,0.2, 0.15, 0.15}



01
10
11
000
001

- Theorem
- Huffman code is an optimal code.

  $H(X) \leq L_{exp} \leq H(X)+1$

Example

alph	code	prob
a	101	0.125
b	100	0.125
c	11	0.25
d	0	0.5

- Source $\mathcal{X}$= { a, b, c, d}
- p_a=p_b= 0.125, p_c= 0.25, p_d= 0.5

d	0.5	0.5	0.5. 1
c	0.25	0.25 0	0.5 0
a	0.125 0	0.25. 1	
b	0.125. 1		

Reducing the length

- $X=\{0,1\}$ - DMS
- $p_0=1/4$, $p_1=3/4$

- For the optimal code, $L_{exp}=1$

- $H(X)=0.19$ bit

  $0.19 < L_{exp} < 1.19$

Encoding blocks of symbols

- X’={00,01,10,11}
- P00=1/16, p10=p11=3/16, p11=9/16

1	9/16		9/16	1	9/16	1
00	3/16		4/16	01	7/16	0
011	3/16	011	3/16	00		
010	1/16	010				

L_{exp}= (9/16)+(3/16) x 2+(3/16) x3+ (1/16)x 3=1.69 bits/2 sym

→L_{exp}=0.845 bit/symbol

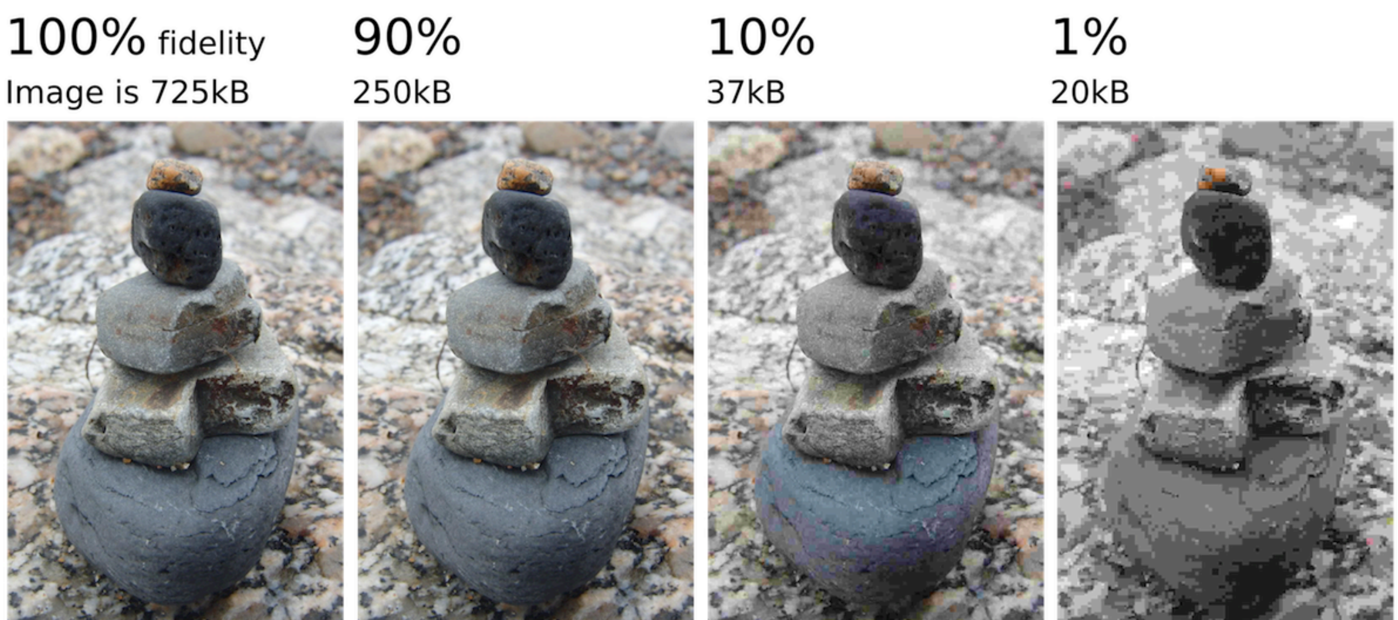
DMC with block coding

- Huffman code when applied to blocks of length n:

$$H(X) \leq L_{exp} \leq H(X)+1/n$$

Lossy and Lossless

- ZIP is an archive file format with lossless data compression.
- JPEG, MPEG, MP3 are lossy compression
  - Trade-off between quality and size of the encoder output



Summary

- Source coding
- Efficiency of source codes
- Optimal codes
- Huffman code