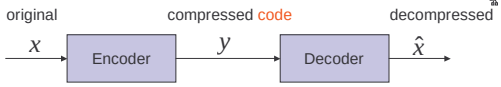
Honey, I Shrunk the Data

Richard Ladner
Computer Science and Engineering
University of Washington

The Plan

- Data compression concepts
- Lossy data compression
- Lossless data compression
- Prefix codes
- Huffman codes

Data Compression Concepts



- Lossless compression $x = \hat{x}$
  - Also called entropy coding, reversible coding.
- Lossy compression $x \neq \hat{x}$
  - Also called irreversible coding.
- Compression ratio = $|x|/|y|$
  - $|x|$ is number of bits in x.

Why Compress

- Conserve storage space
- Reduce time for data transmission
  - Encode, send, decode is faster than send

Braille

- System to read text by feeling raised dots on paper (or on electronic displays). Invented in 1820s by Louis Braille, a French blind man.

a b c z

and the with mother

th ch gh

Braille Example

Clear text:
Call me Ishmael. Some years ago -- never mind how long precisely -- having \\ little or no money in my purse, and nothing particular to interest me on shore, \\ I thought I would sail about a little and see the watery part of the world. (238 characters)

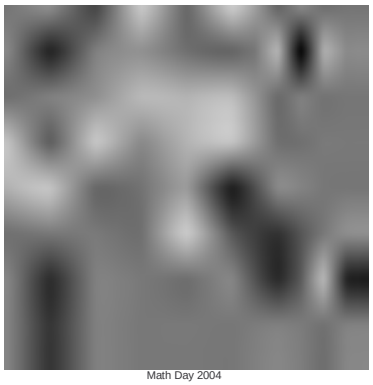
Grade 2 Braille in ASCII.
,call me ,i\%mael4 ,``s ye$>$s ago -- n``e m9d h[l;g precisely -- hav+ \\ ll or no m``oy 9 my purse1 \& no?+ ``picul$>$ 6 9t]e/ me on \%ore1 \\ ,i $?$``$|$ ,i wd sail ab a ll \& see ! wat]y ``p (! _w4 (203 characters)

Compression ratio = 238/203 = 1.17

Lossy Compression

- Data is lost, but not too much.
  - audio
  - video
  - still images, medical images, photographs
- Compression ratios of 10:1 often yield quite high fidelity results.
- Tradoff between compression ratio and fidelity
  - Higher compression means lower fidelity
- Major techniques include
  - JPEG, MPEG, MP3

Ratio
5800 : 1

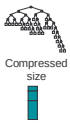


Compressed size

Ratio
2400 :1

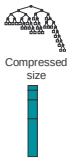
Compressed size

Ratio
1100 :1



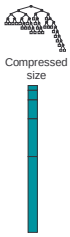
Compressed size

Ratio
533 : 1



Compressed size

Ratio
229 : 1



Compressed size

Ratio
68 : 1

Compressed size

Math Day 2004

Ratio
26 : 1

Compressed size

Math Day 2004

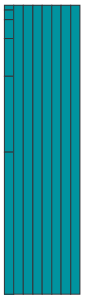
Ratio
13 : 1

Compressed size

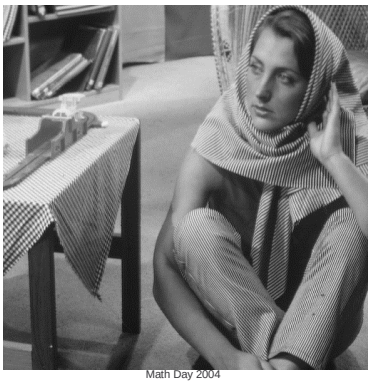
Math Day 2004

Ratio
7 : 1

Compressed size



Math Day 2004



Original

Math Day 2004 17

Why is Compression Possible

- Most data from nature has redundancy
  - There is more data than the actual information contained in the data.
  - Squeezing out the excess data amounts to compression.
  - However, unsqueezing is necessary to be able to figure out what the data means.
- Information theory is needed to understand the limits of compression and give clues on how to compress well.

Math Day 2004 18

Lossless Compression

- Data is not lost - the original is really needed.
  - text compression
  - compression of computer binary files
- Compression ratio typically no better than 4:1 for lossless compression on most kinds of files.
- Statistical Techniques
  - Huffman coding
  - Arithmetic coding
  - Golomb coding
- Dictionary techniques
  - LZW, LZ77
  - Sequitur
  - Burrows-Wheeler Method
- Standards - Morse code, Braille, Unix compress, gzip, zip, bzip, GIF, JBIG, Lossless JPEG

What is Information

- Analog data
  - Also called continuous data
  - Represented by real numbers (or complex numbers)
- Digital data
  - Finite set of symbols $\{a_1, a_2, \ldots, a_m\}$
  - All data represented as sequences (strings) in the symbol set.
  - Example: {a,b,c,d,r} abracadabra
  - Digital data can approximate analog data

Symbols

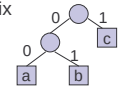
- Roman alphabet plus punctuation
- ASCII - 256 symbols
- Binary - {0,1}
  - 0 and 1 are called bits
  - All digital information can be represented efficiently in binary
  - {a,b,c,d} fixed length representation

symbol	a	b	c	d
binary	00	01	10	11

A Simple Prefix Code

- A prefix code is defined by a binary tree
- Prefix code property
  - no code is a prefix of another

binary tree for prefix code

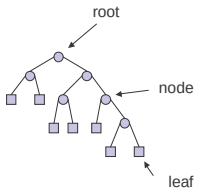


input	output
a	00
b	01
c	1

code

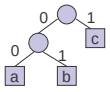
c c a b c c b c c c
1 1 00 01 1 1 01 1 1 1

Binary Tree Terminology



1. Each node, except the root, has a unique parent.
2. Each internal node has exactly two children.

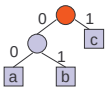
Decoding a Prefix Code




```
repeat
start at root of tree
   repeat
      if bit = 1 then go right
      else go left
   until node is a leaf
   report leaf
until end of the code
```

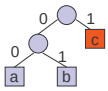
Example 110001

Decoding a Prefix Code



110001

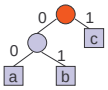
Decoding a Prefix Code



110001

c

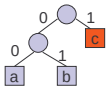
Decoding a Prefix Code



110001

c

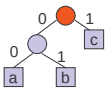
Decoding a Prefix Code



110001

cc

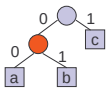
Decoding a Prefix Code



110001

cc

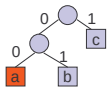
Decoding a Prefix Code



110001

cc

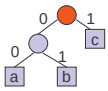
Decoding a Prefix Code



110001

cca

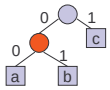
Decoding a Prefix Code



110001

cca

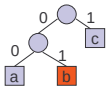
Decoding a Prefix Code



110001

cca

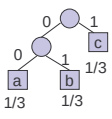
Decoding a Prefix Code



110001

ccab

How Good is the Code



Suppose that all symbols are are equally likely.

Average bit rate = (1/3)2 + (1/3)2 + (1/3)1 = 5/3 = 1.67 bps
Standard code = 2 bps

(bps = bits per symbol)

Statistical Code

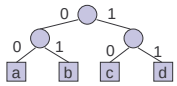
- What if all symbols are not equally likely.
- Example: Letter percentages in English

```
E 12.32 S 6.28 C 2.48 K 0.80
T 9.05  R 5.72 Y 2.11 X 0.15
A 8.17  D 4.31 F 2.09 J 0.10
O 7.81  L 3.97 G 1.82 Q 0.09
I 6.89  U 3.04 P 1.56 Z 0.05
H 6.68  M 2.77 B 1.45
N 6.62  W 2.64 V 1.02
```

Statisitical Coding

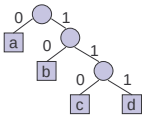
- a : 7/10 b: 1/10 c: 1/10 d: 1/10

Fixed length code



ABR = 2 bps

Variable length code



ABR = (7/10)1 + (1/10)2 + (1/10)3 + (1/10)3
= 15/10 = 1.5 bps

Statisistical Coding Principle

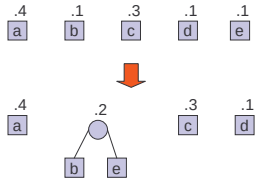
- If you know or can learn the statistics of your data, then even more compression is possible.
- There is an optimal prefix code called the Huffman code.

Huffman Tree Algorithm

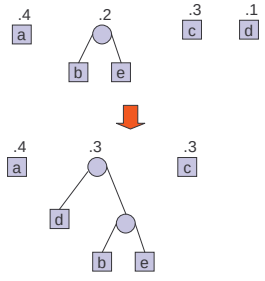
- Initially all symbols are separate with their on probabilities.
- Join two symbols if they have the two lowest probabilities. Add their probabilities.
- Continue this process until there is a single symbol.

Example (1)

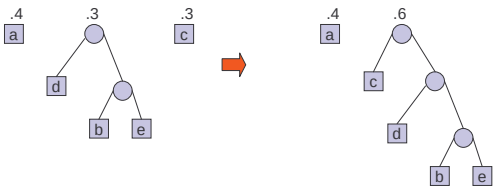
- P(a) =.4, P(b)=.1, P(c)=.3, P(d)=.1, P(e)=.1



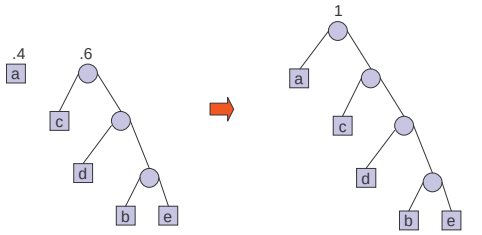
Example (2)



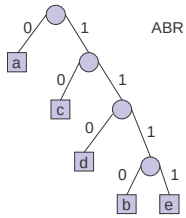
Example (3)



Example (4)



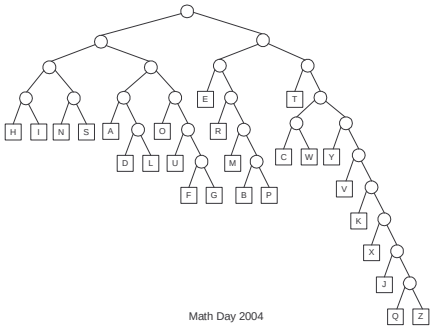
Huffman Code



ABR = .4 x 1 + .1 x 4 + .3 x 2 + .1 x 3 + .1 x 4
= 2.1 bps

a 0
b 1110
c 10
d 110
e 1111

Huffman Code for English



Conclusions

- Huffman coding was invented in 1951 by a graduate student at MIT.
- It is still used today as part of JPEG, MPEG, and other coders.
- The theory of data compression uses probability theory and other parts of mathematics.

Resources

- http://www.cs.washington.edu/homes/ladner
- **Introduction to Information Theory and Data Compression, Second Edition**
  by Greg Harris, Peter D. Johnson, and Darrel R. Hankerson
- **Introduction to Data Compression, Second Edition**
  by Khalid Sayood

000001001011111010111111110

101100100110000001010010011110