CS225 Online Video Series: Huffman Encoding

Chase Geigle

University of Illinois at Urbana-Champaign

October 25, 2012

Motivation

Motivation

- Compression is important:

Motivation

- Compression is important:
  - Improve ability to transfer over a network (bandwidth may be expensive, connections may be slow, etc)

Motivation

- Compression is important:
  - Improve ability to transfer over a network (bandwidth may be expensive, connections may be slow, etc)
  - Reduce storage usage

Motivation

- Compression is important:
  - Improve ability to transfer over a network (bandwidth may be expensive, connections may be slow, etc)
  - Reduce storage usage
- Types of compression:

Motivation

- Compression is important:
  - Improve ability to transfer over a network (bandwidth may be expensive, connections may be slow, etc)
  - Reduce storage usage
- Types of compression:
  - *Lossy* encoding

Motivation

- Compression is important:
  - Improve ability to transfer over a network (bandwidth may be expensive, connections may be slow, etc)
  - Reduce storage usage
- Types of compression:
  - *Lossy* encoding
    - JPEG is an example

Motivation

- Compression is important:
  - Improve ability to transfer over a network (bandwidth may be expensive, connections may be slow, etc)
  - Reduce storage usage
- Types of compression:
  - *Lossy* encoding
    - JPEG is an example
    - Not guaranteed to get the same output bit-for-bit when you decompress!

Motivation

- Compression is important:
  - Improve ability to transfer over a network (bandwidth may be expensive, connections may be slow, etc)
  - Reduce storage usage
- Types of compression:
  - *Lossy* encoding
    - JPEG is an example
    - Not guaranteed to get the same output bit-for-bit when you decompress!
  - *Lossless* encoding

Motivation

- Compression is important:
  - Improve ability to transfer over a network (bandwidth may be expensive, connections may be slow, etc)
  - Reduce storage usage
- Types of compression:
  - *Lossy* encoding
    - JPEG is an example
    - Not guaranteed to get the same output bit-for-bit when you decompress!
  - *Lossless* encoding
    - ZIP and TGZ are examples

Motivation

- Compression is important:
  - Improve ability to transfer over a network (bandwidth may be expensive, connections may be slow, etc)
  - Reduce storage usage
- Types of compression:
  - *Lossy* encoding
    - JPEG is an example
    - Not guaranteed to get the same output bit-for-bit when you decompress!
  - *Lossless* encoding
    - ZIP and TGZ are examples
    - PNG too!

Motivation

- Compression is important:
  - Improve ability to transfer over a network (bandwidth may be expensive, connections may be slow, etc)
  - Reduce storage usage
- Types of compression:
  - *Lossy* encoding
    - JPEG is an example
    - Not guaranteed to get the same output bit-for-bit when you decompress!
  - *Lossless* encoding
    - ZIP and TGZ are examples
    - PNG too!
    - **Guaranteed** to get the *same output* bit-for-bit when you decompress!

Motivation

- Compression is important:
  - Improve ability to transfer over a network (bandwidth may be expensive, connections may be slow, etc)
  - Reduce storage usage
- Types of compression:
  - *Lossy* encoding
    - JPEG is an example
    - Not guaranteed to get the same output bit-for-bit when you decompress!
  - *Lossless* encoding
    - ZIP and TGZ are examples
    - PNG too!
    - **Guaranteed** to get the *same output* bit-for-bit when you decompress!
    - Huffman coding is a form of lossless compression.

Intuition

Intuition

- Say we have some text file.

Intuition

- Say we have some text file.
- There is some kind of redundant (or repeated) information in this file.

Intuition

- Say we have some text file.
- There is some kind of redundant (or repeated) information in this file.
- Moreover, the repetition is uneven (non-uniform) in distribution.

Intuition

- Say we have some text file.
- There is some kind of redundant (or repeated) information in this file.
- Moreover, the repetition is uneven (non-uniform) in distribution.
- **Key Idea:** If we allocate a **small number** of bits to frequent things, and a larger number of bits to infrequent things, we can compress the file.

Algorithm

Algorithm

- Determine how frequently each pattern occurs in the file.

Algorithm

- Determine how frequently each pattern occurs in the file.
- Sort the patterns by frequency.

Algorithm

- Determine how frequently each pattern occurs in the file.
- Sort the patterns by frequency.
- Build a tree from the bottom up, starting with the *least frequent* patterns and ending with the most frequent patterns.

Algorithm

- Determine how frequently each pattern occurs in the file.
- Sort the patterns by frequency.
- Build a tree from the bottom up, starting with the *least frequent* patterns and ending with the most frequent patterns.
- Depth of a node in the tree is directly proportional to the length of its binary code!

An example

feed me more food

An example

feed me more food

- Find frequencies:

f	2
e	4
d	2
␣	3
m	2
r	1
o	3

An example

feed me more food

- Find frequencies:

f	2
e	4
d	2
␣	3
m	2
r	1
o	3

- Sort frequencies:

r	1
f	2
d	2
m	2
␣	3
o	3
e	4

An example

r	1
f	2
d	2
m	2
␣	3
o	3
e	4