Assignment 3 + Compression

(SNLP Tutorial 4)

Vilém Zouhar, Awantee Deshpande, Julius Steuer

18th, 20th May 2021

Assignment 3

- Exercise 1: Entropy Intuition
- Exercise 2: Uncertainty of events
- Exercise 3: KL Divergence
- Bonus: KL Divergence calculation

Compression

- Prefix Codes: No whole code word is a prefix of any other code word
- Uniquely decodable codes: Each word maps to one and only one code word

Compression

- Prefix Codes: No whole code word is a prefix of any other code word
- Uniquely decodable codes: Each word maps to one and only one code word

Prefix codes are a subset of uniquely decodable codes!

Optimal length of code words

$$l_i = -\log_D p(w_i)$$

Kraft's Inequality

$$\sum_{i=1}^{m} D^{-l_i} \leq 1$$

What does the sum < 1 imply?
What does the sum $= 1$ imply?
What does the sum > 1 imply?
What does this tell us about uniquely decodable and prefix codes?

Exercise: Test Kraft's Inequality on Morse Code

(Hint: What is the encoding alphabet?)

ASCII/UTF{8,16,32}/Unicode

Encoding from characters to binary alphabet:

ASCII: 7 bits (byte was standardized to 8 bits later!)

- Q: How many values?
- Q: It has to be aligned to 8 bits nowadays (modern CPU requirement). What do we with the eight bit?

ASCII/UTF{8,16,32}/Unicode

Encoding from characters to binary alphabet:

ASCII: 7 bits (byte was standardized to 8 bits later!)

- Q: How many values?
- Q: It has to be aligned to 8 bits nowadays (modern CPU requirement). What do we with the eight bit?

Windows-1252, Windows-1250

- Full 8 bits, map lower 128 to ASCII
- Individual differences, different encoding for í

ASCII/UTF{8,16,32}/Unicode

Encoding from characters to binary alphabet:

ASCII: 7 bits (byte was standardized to 8 bits later!)

- Q: How many values?
- Q: It has to be aligned to 8 bits nowadays (modern CPU requirement). What do we with the eight bit?

Windows-1252, Windows-1250

- Full 8 bits, map lower 128 to ASCII
- Individual differences, different encoding for í

UTF

- Encoding over Unicode (character alphabet)
- UTF8 - Start with 8 bits, extend to 16 or 32; UTF32 - Always 32 bits
- Compositionality: i with little tail and acute accent `U+0301U+0328U+0069`
- Valid misuse: snowman `U+0301U+0328U+2603`

Encoding

Task

Create encoding (binary) for the following recipe:

`apple apple banana cherries apple dark_chocolate eggplant banana cherries banana ...`

Encoding

Task

Create encoding (binary) for the following recipe:

`apple apple banana cherries apple dark_chocolate eggplant banana cherries banana ...`

Fixed-width encoding

 000  001  010  011  100  101

Length $= 14 \times 3 = 42$

Issues?

- Encoding for  and  ?
- What do 110 and 111 mean?

Encoding - Huffman

4× A 4× B 2× C 2× E 1× D 1× F

Huffman Bonus

- When will the Huffman tree be balanced?

Huffman Bonus

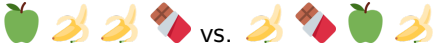
- When will the Huffman tree be balanced?
- How do we store the tree? Does the efficiency of this matter?

Huffman Bonus

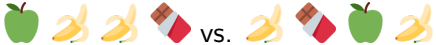
- When will the Huffman tree be balanced?
- How do we store the tree? Does the efficiency of this matter?
- Are there undefined sequences of bits when using Huffman encoding?

Huffman Bonus

- When will the Huffman tree be balanced?
- How do we store the tree? Does the efficiency of this matter?
- Are there undefined sequences of bits when using Huffman encoding?
- Does the result of Huffman encoding depend on the text ordering?

  E.g. 

Huffman Bonus

- When will the Huffman tree be balanced?
- How do we store the tree? Does the efficiency of this matter?
- Are there undefined sequences of bits when using Huffman encoding?
- Does the result of Huffman encoding depend on the text ordering?
  E.g. 

- Can there be two equally good Huffman encodings?

Huffman Bonus

- When will the Huffman tree be balanced?
- How do we store the tree? Does the efficiency of this matter?
- Are there undefined sequences of bits when using Huffman encoding?
- Does the result of Huffman encoding depend on the text ordering?
  E.g.  vs. 
- Can there be two equally good Huffman encodings?
- Can Huffman result in assigning an element code of length 1?

Long Range Dependencies

- Correlation
- Conditional entropy

Assignment 4

- Exercise 1: Encodings (ASCII, UTF, Huffman)

```
a = "Hellp there!"
a[4] = 'o' # substitute_character(string=a, pos=4, newchar='o')
```

- Exercise 2: Conditional Entropy on DNA
- Bonus: Huffman Encoding alphabet

Resources

1. Twitter emojis
2. https://www.ics.uci.edu/~dan/pubs/DC-Sec1.html
3. https://en.wikipedia.org/wiki/Shannon%27s_source_coding_theorem
4. https://en.wikipedia.org/wiki/Huffman_coding
5. http://www.mss.cbi.fau.de/content/uploads/epnat.pdf
6. https://arxiv.org/pdf/adap-org/9507007.pdf
7. https://en.wikipedia.org/wiki/Windows_code_page
8. https://r12a.github.io/app-conversion/