Information Theory

Lecture 6

LZ codes

STEFAN HÖST

Universal Source Coding

Introduction

If the statistics of the source is unknown:

- Use static dictionary.
- Estimate the probability distribution from the source block and use e.g. Huffman. (Two sweep)
- Adaptive algorithm
  - Adaptive Huffman coding. (One sweep or windowed)
  - Sequential algorithms (windowed).

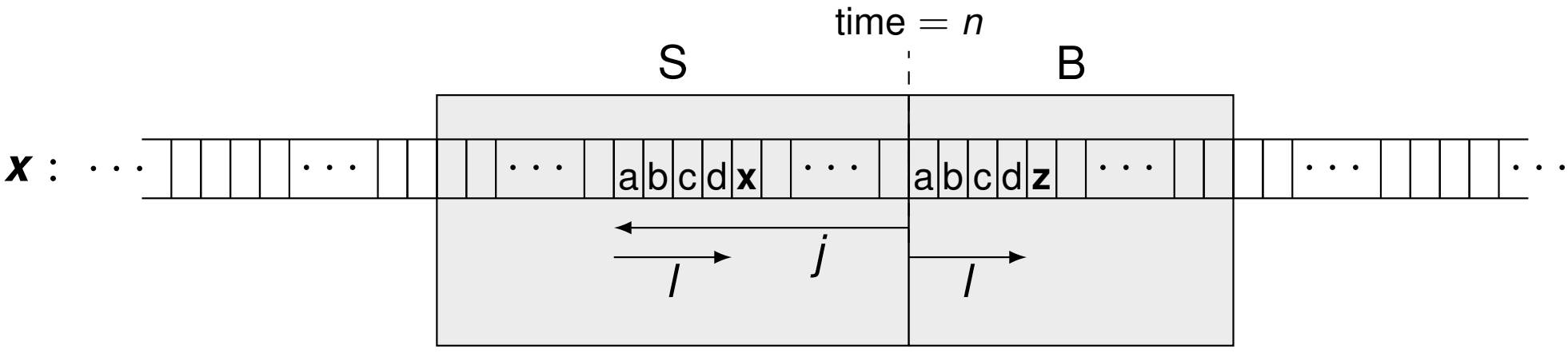
Lempel-Ziv '77

LZ77 (Lempel-Ziv 1977)

A sliding window technique with two buffers:

- Search buffer of length S (recent symbols)
- Look ahead buffer of length B (upcoming symbols)
- The offset j is the number of steps from the first symbol in the look ahead buffer to the same symbol in the search buffer.
- The length of a match l is the number of symbols with match between the buffers, starting at the offset.

Lempel-Ziv '77



Lempel-Ziv '77

LZ77 (Lempel-Ziv 1977)

- Init: Read the first S symbols to the search buffer.
- For the symbol x_n, determine the offset with longest match.
- The codeword is $\boldsymbol{y} = (j, l, r)$, where
  - j is the offset
  - l is the length
  - r is the next symbol in the look ahead buffer

  If there is no match send $(0, 0, x_n)$.

$$\ell(\boldsymbol{x}) = \lceil \log S \rceil + \lceil \log B \rceil + \lceil \log k \rceil$$

Lempel-Ziv '77

Example

Consider a source alphabet $\mathcal{X} = \{a, b, c, d, r\}$
Encode the sequence

$$\boldsymbol{x} = cabracadabrarrarrad\ldots$$

with LZ77 using

- Search buffer of length $S = 7$
- Look ahead buffer of length $B = 6$

Lempel-Ziv '77

Improvements of LZ77

- Encode the triples (codewords) with a second, variable codeword length, code (e.g. Huffman).
- (LZSS) Only reason to include uncoded char is to cope with no match.
  Use binary prefix to distinguish:
  - If x in S: $y = (0, j, l)$
  - In x not in S: $y = (1, x)$

Lempel-Ziv '77

Example

Consider a source alphabet $\mathcal{X} = \{a, b, c, d, r\}$
Encode the sequence

$$\boldsymbol{x} = cabracadabrarrarrad\ldots$$

with LZSS using

- Search buffer of length $S = 7$
- Look ahead buffer of length $B = 6$

Lempel-Ziv '78

LZ77→LZ78

If the repetition period in the source text is longer than the buffer LZ77 will run into problems.

LZ78: Drop the buffer and build a dictionary instead.

Lempel-Ziv '78

LZ78 (Lempel-Ziv 1978)

- Init: Start with an empty dictionary.
- At time n, determine the largest l such that $x_n \dots x_{n+l-1}$ is in the dictionary but $x_n \dots x_{n+l-1} x_{n+l}$ is not. Let j be a pointer to the dictionary position.
- The codeword for $x_n \dots x_{n+l-1} x_{n+l}$ is $\boldsymbol{y} = (j, x_{n+l})$.
  Use $\lceil \log(\text{max index}) \rceil$ bits to encode j
- Add $x_n \dots x_{n+l-1} x_{n+l}$ to the dictionary.
- If there is no match send $(0, x_n)$.

Lempel-Ziv '78

Example

Consider a source alphabet $\mathcal{X} = \{a, b, c, d, r\}$
Encode the sequence

$$\boldsymbol{x} = cabracadabrarrarrad\ldots$$

with LZ78.

Lempel-Ziv '78

LZ78$\rightarrow$LZW

Get rid of the uncoded symbol in the codeword.

Only used together with $(0, r)$ when a single letter is not in the dictionary.

$\Rightarrow$

Initiate the dictionary with the alphabet.

Lempel-Ziv '78

LZW (Lempel-Ziv-Welch)

- Init: Start with a dictionary with all letters.
- At time n, determine the largest l such that $x_n \dots x_{n+l-1}$ is in the dictionary but $x_n \dots x_{n+l-1} x_{n+l}$ is not. Let j be a pointer to the dictionary position.
- The codeword for $x_n \dots x_{n+l-1}$ is $\boldsymbol{y} = (j)$.
- Add $x_n \dots x_{n+l-1} x_{n+l}$ to the dictionary.

Lempel-Ziv '78

Example

Consider a source alphabet $\mathcal{X} = \{a, b, c, d, r\}$
Encode the sequence

$$\boldsymbol{x} = cabracadabrarrarrad\ldots$$

with LZW.

Lempel-Ziv in the real world

Applications of LZ

The LZ algorithms are used in many applications (often followed by Huffman). Among others:

- LZ77
  - deflate (implementation of LZ77 + Huffman)
  - zip, gzip, 7z
  - PNG
- LZW
  - compress (UNIX)
  - GIF, TIFF