Lecture 5

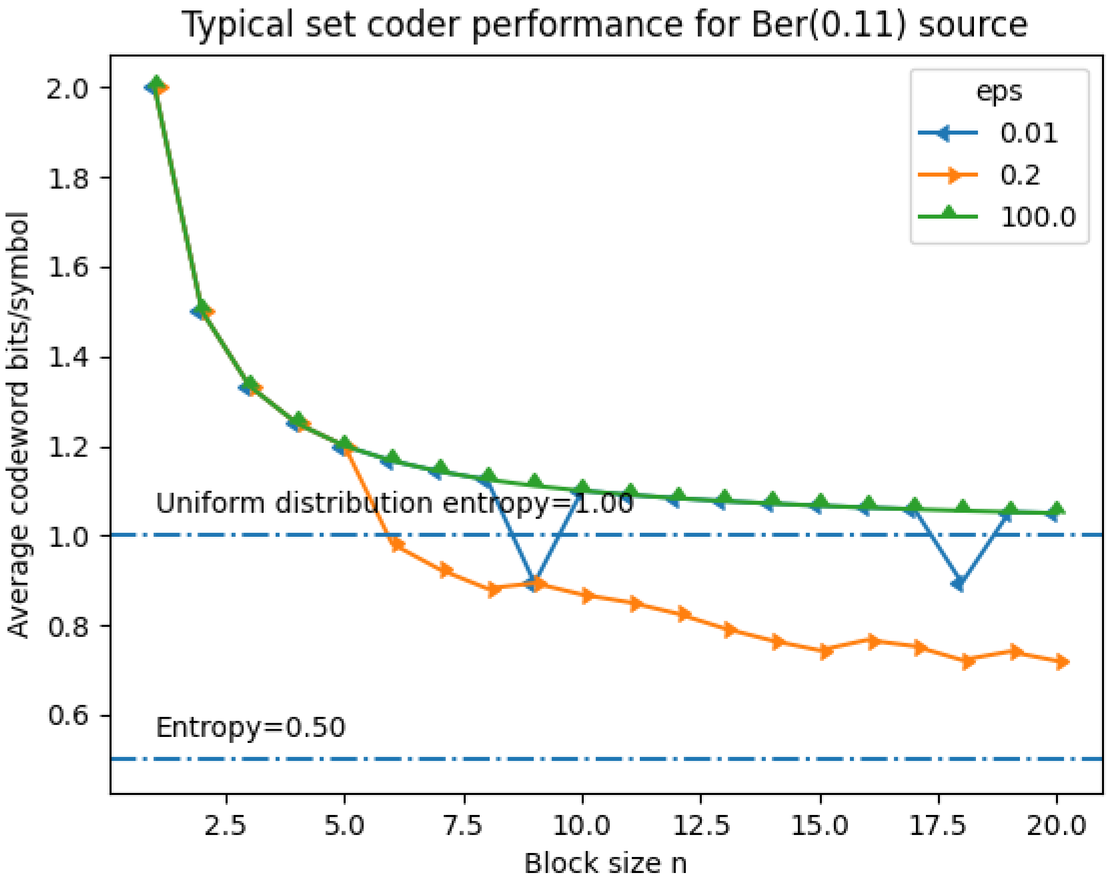
Typical sequences, Typical coder, Arithmetic coding

Typical coder

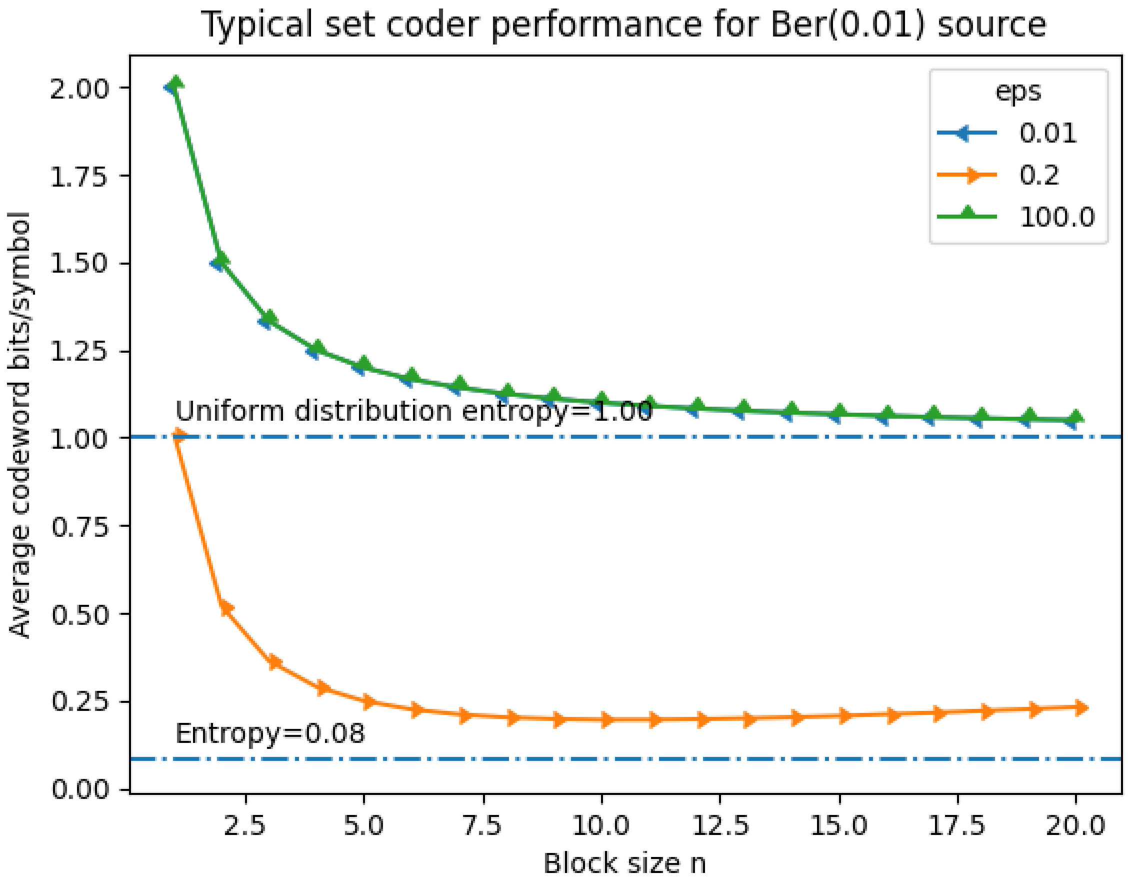
Typical coder is part of SCL!

https://github.com/kedartatwawadi/stanford_compression_library/blob/main/compressors/typical_set_coder.py

Typical coder



Typical coder



Typical coder, Huffman-based block coder

- Typical coder -> Not practical! Codebook size too big as $n \to \infty$
- Huffman-based block coder -> Not practical for large alphabets: decoding tree/table is too big, codebook is too large ...
- Not adaptive, (when probabilities are changing/different per symbol)

Arithmetic coding

1. Operates on the entire input as a single "block". (so if input size is 10,000 -> block size is 10,000)
2. No explicit need to create a "codebook" for each codeword. A codeword is created "on the fly" for the input
3. For an input of size `n`, the overhead of arithmetic coding is `~2/n`
4. Very convenient to adapt to changing probabilities!