Chapter 5

Streaming Source Coding

5.1 Entropy Rate

Suppose that we have a sequence $\mathbf{x}$ with symbols from $\mathcal{A}$. If we treat k consecutive symbols as one symbol from $\mathcal{A}^k$, $\mathbf{x}$ can be regarded as a sequence from alphabet $\mathcal{A}^k$. From this perspective, let $P_{\mathbf{x}}^{(k)}$ be the type of $\mathbf{x}$ with respect to the alphabet $\mathcal{A}^k$. For type $P_{\mathbf{x}}^{(k)}$, we can design a Huffman code that has the average codeword length L_k satisfying

$$\frac{1}{k}H(P_{\mathbf{x}}^{(k)}) \le \frac{L_k}{k} \le \frac{1}{k}H(P_{\mathbf{x}}^{(k)}) + \frac{1}{k}.$$

Regarding $P_{\mathbf{x}}^{(k)}$ as a joint distribution of $(X_1, \ldots, X_k)$ with $X_i \in \mathcal{A}$, we can derive that

$$\frac{1}{k}H(P_{\mathbf{x}}^{(k)}) \le \frac{H(X_1) + \cdots + H(X_k)}{k} \le H((p_1 + \cdots + p_k)/k) = H(P_{\mathbf{x}}).$$

Suppose the sequence $\mathbf{x}$ has length n. Then $H(P_{\mathbf{x}}^{n}) = 0$. Does this mean that we can use almost zero rate to compress any sequence? Of course not. As using Huffman codes, the code itself should be included for decoding. For each sequence $\mathbf{a}$ in $\mathcal{A}^k$, its codeword $f(\mathbf{a})$ should be known for decoding. Using block source coding, we would need as least $k \log_2 |\mathcal{A}|$ bits to represent a sequence $\mathbf{a}$, and $l_{\mathbf{a}}$ bits to represent the codeword. What is the total number of bits for representing the code?

Stochastic Process

- A stochastic process is a sequence of random variables $\mathbf{X} = \{X_i\}$.
- A stochastic process is said to be *stationary* if the joint distribution of any subset of the sequence of random variables is invariant w.r.t. shifts in the time index.

Entropy Rate

- The *entropy rate* of a stochastic process $\mathbf{X}$ is defined by

$$\bar{H}(\mathbf{X}) = \lim_{n\to\infty} \frac{1}{n} H(X_1, X_2, \ldots, X_n),$$

  when the limit exists.
- Let

$$\bar{H}'(\mathbf{X}) = \lim_{n\to\infty} H(X_n | X_1, X_2, \ldots, X_{n-1}).$$

■ **Example 5.1** If $X_1, X_2, \ldots$ are i.i.d. random variables, then

$$\bar{H}(\mathbf{X}) = \lim_{n\to\infty} \frac{1}{n} H(X_1, X_2, \ldots, X_n) = \lim_{n\to\infty} \frac{1}{n} nH(X_1) = H(X_1),$$
$$\bar{H}'(\mathbf{X}) = \lim_{n\to\infty} H(X_n | X_1, X_2, \ldots, X_{n-1}) = \lim_{n\to\infty} H(X_n) = H(X)1.$$

■

■ **Example 5.2** If $X_1, X_2, \ldots$ are independent random variables, then

$$\bar{H}(\mathbf{X}) = \lim_{n\to\infty} \frac{1}{n} H(X_1, X_2, \ldots, X_n) = \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} H(X_i),$$
$$\bar{H}'(\mathbf{X}) = \lim_{n\to\infty} H(X_n|X_1, X_2, \ldots, X_{n-1}) = \lim_{n\to\infty} H(X_n)$$

which may not exist. ■

Exercise 5.1 Find examples of $H(X_i)$ such that $\bar{H}(\mathbf{X})$ does not exist. ■

Theorem 5.1 For a stationary stochastic process, $\bar{H}(\mathbf{X}) = \bar{H}'(\mathbf{X})$.

Proof. By the chain rule for entropy,

$$\frac{1}{n} H(X_1, X_2, \ldots, X_n) = \frac{1}{n} \sum_{i=1}^{n} H(X_i|X_1, \ldots, X_{i-1}).$$

■

Lemma 5.2 For a stationary stochastic process, $H(X_n|X_1, X_2, \ldots, X_{n-1})$ is non-increasing in n and has a limit $\bar{H}'(\mathbf{X})$.

Proof. Conditioning reduces entropy and stationary. ■

Lemma 5.3 — Cesáro Mean. Let a_k and b_k be real numbers. If $a_n \longrightarrow a$ as $n \to \infty$ and $b_n = \frac{1}{n}\sum_{k=1}^{n} a_k$, then $b_n \to a$ as $n \to \infty$.

Proof. For any $\epsilon > 0$, there exists N_ϵ such that $|a_n - a| < \epsilon$ for all $n > N_\epsilon$. ■

See [2, Sec 4.2] and [8, Sec 2.10] for entropy rate.

5.2 Lempel-Ziv Coding

History

- A class of codes pioneered by Lempel and Ziv.
- Widely used in gzip, pkzip, compress, GIF, etc.
- Universally optimal: asymptotic compression rate approaches the entropy rate of the any stationary ergodic source.
- Also called *adaptive dictionary* compression algorithms.

Lempel-Ziv Algorithms

- A *parsing* of a string $x_1 x_2 \cdots$ is a division of the string into phrases (separated by commas).
- Sliding window LZ algorithm:
  - Fix a window size W.
  - The next phrase is the longest string starting from the current postion that is the same to another string starting at a position in W.
- Tree-structured LZ algorithm:
  - Each phrase is the shortest phrase not seen earlier.

Example

- Consider the string

$$ABBABBABBBAABABAA$$

- Sliding window LZ with $W = 4$:

$$A, B, B, ABBABB, BA, A, BA, BA, \ldots$$

  which is represented by $(0, A), (0, B), (1, 1), (3, 6), (4, 2), (1, 1), (3, 2), (2, 2), \ldots$.
- Tree-structured LZ:

$$A, B, BA, BB, AB, BBA, ABA, BAA, \ldots$$

  which is represented by $(0, A)$, $(0, B)$, $(2, A)$, $(2, B)$, $(1, B)$, $(4, A)$, $(5, A)$, $(3, A)$,$\ldots$.

See [2, Chapter 13] for the discussion of universal source coding.

5.3 Optimality of Lempel-Ziv Coding

The compression ratio of a sequence $\mathbf{x} \in \mathcal{A}^*$ compressed by a compressor f is

$$\rho(\mathbf{x}) = \frac{l(f(\mathbf{x}))}{l(\mathbf{x}) \log_2 |\mathcal{A}|}$$

A good compressor should give a small compression ratio for most sequences that we want to compression. Note that though LZ coding is *universal*, it does not mean that it gives a small compression ratio for all sequences.

- Suppose that we use LZ coding to encode all sequences $\mathcal{A}^n$.
- The $|\mathcal{A}|^n$ codewords form a prefix code.
- The average length of all the codewords is at least $n \log_2 |\mathcal{A}|$.

Consider a sequence of n A's. Using the sliding window LZ coding, we codeword is $(0, A)(1, n-1)$. The complexity of representing the codeword is the same as representing $n-1$. If we do a bound on n, say m, we know that $\log M$ bits are sufficient to represent all positive integers no larger than m. As n has no bounds, the representation of integers must be a prefix-free, variable-length code. We know that there exists such a code such that for an integer k, the codeword length is $O(\log k)$. Therefore, the compression ratio is $O(\log n/n)$. In general, LZ coding is good for sequences with certain repetition patterns.

In the original paper of Ziv and Lempel [9], a source is defined as a subset Σ of $\mathcal{A}^*$ satisfying

1. $A \subset \Sigma$;
2. If $\mathbf{x} \in \Sigma$, then $\mathbf{xx} \in Sigma$;
3. If $\mathbf{x} \in \Sigma$, then all subsequences of $\mathbf{x}$ are in Σ.

For such a source, they show that the sliding window LZ coding is no worth than any other algorithms.

Another approach of information theory is to assign certain probability model on $\mathcal{A}^*$. In [7], Wyner and Ziv show that the sliding window LZ coding achieves the entropy rate for all finite-alphabet stationary, ergodic sources.