

Chapter 26

Entropy II

By Sarel Har-Peled, December 30, 2015^①

The memory of my father is wrapped up in white paper, like sandwiches taken for a day at work. Just as a magician takes towers and rabbits out of his hat, he drew love from his small body, and the rivers of his hands overflowed with good deeds.

-- Yehuda Amichai, My Father..

26.1. Compression

In this section, we will consider the problem of how to compress a binary string. We will map each binary string, into a new string (which is hopefully shorter). In general, by using a simple counting argument, one can show that no such mapping can achieve real compression (when the inputs are adversarial). However, the hope is that there is an underlying distribution on the inputs, such that some strings are considerably more common than others.

Definition 26.1.1. A compression function Compress takes as input a sequence of n coin flips, given as an element of $\{H, T\}^n$, and outputs a sequence of bits such that each input sequence of n flips yields a distinct output sequence.

The following is easy to verify.

Lemma 26.1.2. *If a sequence S_1 is more likely than S_2 then the compression function that minimizes the expected number of bits in the output assigns a bit sequence to S_2 which is at least as long as S_1 .*

Note, that this is very weak. Usually, we would like the function to output a prefix code, like the Huffman code.

Theorem 26.1.3. *Consider a coin that comes up heads with probability $p > 1/2$. For any constant $\delta > 0$, when n is sufficiently large, the following holds.*

^①This work is licensed under the Creative Commons Attribution-Noncommercial 3.0 License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc/3.0/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

- (i) There exists a compression function **Compress** such that the expected number of bits output by **Compress** on an input sequence of n independent coin flips (each flip gets heads with probability p) is at most $(1 + \delta)n\mathbb{H}(p)$; and
- (ii) The expected number of bits output by any compression function on an input sequence of n independent coin flips is at least $(1 - \delta)n\mathbb{H}(p)$.

Proof: Let $\varepsilon > 0$ be a constant such that $p - \varepsilon > 1/2$. The first bit output by the compression procedure is '1' if the output string is just a copy of the input (using $n + 1$ bits overall in the output), and '0' if it is compressed. We compress only if the number of ones in the input sequence, denoted by X is larger than $(p - \varepsilon)n$. By the Chernoff inequality, we know that $\Pr[X < (p - \varepsilon)n] \leq \exp(-n\varepsilon^2/2p)$.

If there are more than $(p - \varepsilon)n$ ones in the input, and since $p - \varepsilon > 1/2$, we have that

$$\sum_{j=\lceil n(p-\varepsilon) \rceil}^n \binom{n}{j} \leq \sum_{j=\lceil n(p-\varepsilon) \rceil}^n \binom{n}{\lceil n(p-\varepsilon) \rceil} \leq \frac{n}{2} 2^{n\mathbb{H}(p-\varepsilon)},$$

by **Corollary 26.2.1**. As such, we can assign each such input sequence a number in the range $0 \dots \frac{n}{2} 2^{n\mathbb{H}(p-\varepsilon)}$, and this requires (with the flag bit) $1 + \lceil \lg n + n\mathbb{H}(p - \varepsilon) \rceil$ random bits.

Thus, the expected number of bits output is bounded by

$$(n + 1) \exp(-n\varepsilon^2/2p) + (1 + \lceil \lg n + n\mathbb{H}(p - \varepsilon) \rceil) \leq (1 + \delta)n\mathbb{H}(p),$$

by carefully setting ε and n being sufficiently large. Establishing the upper bound.

As for the lower bound, observe that at least one of the sequences having exactly $\tau = \lfloor (p + \varepsilon)n \rfloor$ heads, must be compressed into a sequence having

$$\lg \binom{n}{\lfloor (p + \varepsilon)n \rfloor} - 1 \geq \lg \frac{2^{n\mathbb{H}(p+\varepsilon)}}{n+1} - 1 = n\mathbb{H}(p - \varepsilon) - \lg(n+1) - 1 = \mu,$$

by **Corollary 26.2.1**. Now, any input string with less than τ heads has lower probability to be generated. Indeed, for a specific strings with $\alpha < \tau$ ones the probability to generate them is $p^\alpha(1 - p)^{n-\alpha}$ and $p^\tau(1 - p)^{n-\tau}$, respectively. Now, observe that

$$p^\alpha(1 - p)^{n-\alpha} = p^\tau(1 - p)^{n-\tau} \cdot \frac{(1 - p)^{\tau-\alpha}}{p^{\tau-\alpha}} = p^\tau(1 - p)^{n-\tau} \left(\frac{1 - p}{p} \right)^{\tau-\alpha} < p^\tau(1 - p)^{n-\tau},$$

as $1 - p < 1/2 < p$ implies that $(1 - p)/p < 1$.

As such, **Lemma 26.1.2** implies that all the input strings with less than τ ones, must be compressed into strings of length at least μ , by an optimal compressor. Now, the Chernoff inequality implies that $\Pr[X \leq \tau] \geq 1 - \exp(-n\varepsilon^2/12p)$. Implying that an optimal compressor outputs on average at least $(1 - \exp(-n\varepsilon^2/12p))\mu$. Again, by carefully choosing ε and n sufficiently large, we have that the average output length of an optimal compressor is at least $(1 - \delta)n\mathbb{H}(p)$. ■

26.2. From previous lecture

Corollary 26.2.1. We have:

(i) $q \in [0, 1/2] \Rightarrow \binom{n}{\lfloor nq \rfloor} \leq 2^{n\mathbb{H}(q)}$. (ii) $q \in [1/2, 1] \Rightarrow \binom{n}{\lceil nq \rceil} \leq 2^{n\mathbb{H}(q)}$.

(iii) $q \in [1/2, 1] \Rightarrow \frac{2^{n\mathbb{H}(q)}}{n+1} \leq \binom{n}{\lfloor nq \rfloor}$. (iv) $q \in [0, 1/2] \Rightarrow \frac{2^{n\mathbb{H}(q)}}{n+1} \leq \binom{n}{\lceil nq \rceil}$.

26.3. Bibliographical Notes

The presentation here follows [MU05, Sec. 9.1-Sec 9.3].

Bibliography

[MU05] M. Mitzenmacher and U. Upfal. *Probability and Computing – randomized algorithms and probabilistic analysis*. Cambridge, 2005.