«Calcul formel avancé et applications». Brief lecture notes.

19.09.2024. Lecture 2.

1. Discussion fo the homework. We proved that for two jointly distributed random variables (X, Y) we have

$$H(X, Y) \leq H(X) + H(Y)$$

(with equality for independent X and Y). The proof uses concavity of logarithm and Jensen's inequality. Applying this inequality twice, we conclude that for a triple of jointly distributed random variables (X, Y, Z) we have

$$H(X, Y, Z) \leq H(X) + H(Y) + H(Z).$$

We used these properties of Shannon's entropy to solve an exercise from the homework:

Homework 1.3. We are given $n = 14$ coins, and again one fake coin can be heavier or lighter than the genuine ones. How many weighings does it take to find a genuine coin? Prove that there is no strategy can do it in *three* operations.

2. Prefix-free codes and Huffman's code. We say that a binary *code* $\mathcal{C}$ is a set of binary words (binary strings), i.e., $\mathcal{C} \subset \{0, 1\}^*$. Elements of $\mathcal{C}$ are called codewords. A binary code $\mathcal{C}$ is called *prefix-free* if for every two codewords $v, w \in \mathcal{C}$, the words v is not a prefix of w. A prefix-free binary code can represented as a rooted binary tree, whose branches (paths from the root to the leaves) correspond to the codewords. In the class we discussed that a prefix-free binary code with n codewords is an equivalent description of a *strategy* for the game "guess a number between 1 and n" with yes-or-no answers.

Prefix free codes have the useful property of unique decoding: if a binary string x was obtained as a concatenation of several words from a prefix-free code $\mathcal{C}$,

$$X = c_{i_1} c_{i_2} \ldots c_{i_\ell},$$

then this representation (i.e., the sequence of codewords $c_{i_1}, c_{i_2}, \ldots, c_{i_\ell}$) can be reconstructed uniquely.

For a distribution of probabilities on n messages $(p_1, \ldots, p_n)$ and a prefix-free binary code $(c_1, \ldots, c_n)$ associated with this distribution, the *average length ofthe codewords* is

$$(*) \qquad \sum_{i=1}^{n} p_i |c_i|,$$

where $|c_i|$ denotes the length (number of binary digits) in c_i. For a fixed distribution of probabilities, we may ask how to find a prefix-free code that minimises the average length of the codewords. We will call the minimal average length of the codewords associated with a given distribution of probabilities by the *cost* of this distribution.

In the class we discussed the construction of Huffman, which allows to find for a given distribution of probabilities $(p_1, \ldots, p_n)$ the prefix-free code that provides the minimum of (*) and, respectively, to compute the *cost* of this distribution. The correctness of Huffman's construction is based on the following lemmas.

Lemma 1. *For every distribution of probabilities $(p_1, \ldots, p_n)$ such that $p_1 \geq p_2 \geq \ldots \geq p_n$, in every prefix-free binary code $(c_1, \ldots, c_n)$ that provides the minimum of (*), we have*

$$|c_1| \leq |c_2| \leq \ldots \leq |c_n|.$$

In other words, in an optimal code, the smaller probabilities are associated with the longer codeword.

Lemma 2. *For every distribution of probabilities $(p_1, \ldots, p_n)$ such that $p_1 \geq p_2 \geq \ldots \geq p_n$, in every prefix-free binary code $(c_1, \ldots, c_n)$ that provides the minimum of (*), we have $|c_{n-1}| = |c_n|$. In other words, an optimal code cannot have a unique longest codeword.*

Lemma 3. *For every distribution of probabilities* $(p_1, \ldots, p_n)$ *such that* $p_1 \ge p_2 \ge \ldots \ge p_n$*, in every prefix-free binary code* $(c_1, \ldots, c_n)$ *that provides the minimum of (*), there exists* $i < n$ *such that the codewords* c_i *and* c_n *are of the same length and, moreover, the words* c_i *and* c_n *differ in only the very last bit.*

Lemma 4. *Let*

$$\mathcal{P}_1 = (p_1, p_2, \ldots, p_{n-1}, p_n, p_{n+1})$$

be a probability distribution where $p_1 \ge p_2 \ge \ldots \ge p_{n-1} \ge p_n \ge p_{n+1}$*, and*

$$\mathcal{P}_2 = (p_1, p_2, \ldots, p_{n-1}, q)$$

be another probability distribution where $q = p_n + p_{n+1}$*. The the difference between the* costs *of these distributions is equal to* q*.*

Recursive construction of Huffman's code

`input:` probability distribution $(p_1, \ldots, p_n)$

- `if` $n = 2$ `then return the code` $(0, 1)$
- `otherwise, sort the list of probabilities in descending order`

  /* so in what follows we may assume that $p_1 \ge p_1 \ge \ldots \ge p_n$ */
- `let` $q := p_{n-1} + p_n$
- $(d_1, \ldots, d_{n-1}) \leftarrow$ `result of the recursive call of the algorithm on the distribution` $(p_1, \ldots, p_{n-2}, q)$)
- `return the result` $(c_1, \ldots, c_n)$ `where`

  $c_i := d_i$ `for` $i = 1 \ldots (n-2)$,

  $c_{n-1} = d_{n-1}0$ (d_{n-1} `concatenated with` *`zero`*),

  $c_n = d_{n-1}1$ (d_{n-1} `concatenated with` *`one`*)

In the class we proved that Huffman's algorithm returns for every distribution an optimal prefix-free code, i.e., a code that provides the minimal possible value for the average length of the codeword (*),

Theorem 1. *For every distribution* $(p_1, \ldots, p_n)$ *Huffman's code (the coded constructed by the algoruthm explained above) provides the minimum to the average length of the codeword (*).*

The proof of this theorem is based on Lemma 1-4.

Exercise 1. Find Huffman's codes for the following probability distributions:

(a) $(0.25, 0.35, 0.4)$

(b) $(0.32, 0.3, 0.23, 0.12, 0.03)$

(c) $(\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}, \frac{1}{32})$

Exercise 2. Show that for some distributions $(p_1, \ldots, p_n)$ the expression

$$\sum_{i=1}^{n} p_i \left\lceil \log \frac{1}{p_i} \right\rceil$$

is strictly greater than the optimal average number of questions in a strategy of guessing a number with yes-or-no questions.

Exercise 3. We have n stones and the scale that can compare the weights of any pair of stones. How many operations de we need to sort all the stones in in descending order ?

(a) $n = 3$,

(b) $n = 4$,

(c) $n = 5$.

For each of these n, suggest a sorting strategy and prove its optimality.

3. Uniquely decodable codes and Kraft's inequality. We say that binary code $\mathcal{C} = \{c_1, \ldots, c_n\}$ is *uniquely decodable* if for every $x \in \{0,1\}^*$ there exists at most one way to represent x as a concatenation of codewords from $\mathcal{C}$,

$$x = c_{i_1} c_{i_2} \ldots c_{i_\ell}.$$

We have seen prefix-free codes are uniquely decodable. Some uniquely decodable code are *not* prefix-free. However, for every uniquely decodable code there exists a prefix-free code with the same lengths of codewords (so the minimum of the average length of the codewords can be always achieved in the class of prefix-free codes):

Theorem 2. *For every uniquely decodable binary code* $(c_1, \ldots, c_n)$ *there exists a prefix-free binary code* $(d_1 \ldots, d_n)$ *such that* $|d_i| = |c_i|$ *for all* $i = 1, \ldots, n$.

This theorem follows from two lemmas.

Lemma 5 (Kraft's inequality)**.** *For every uniquely decodable binary code* $(c_1, \ldots, c_n)$

$$\sum_{i=1}^{n} 2^{-|c_i|} \leq 1.$$

(We proved this lemma in the class.)

Lemma 6. *For every set of natural numbers* $\ell_1, \ldots \ell_n$ *such that*

$$\sum_{i=1}^{n} 2^{-\ell_i} \leq 1$$

there exists a prefix-free binary code $(d_1 \ldots, d_n)$ *such that* $|d_i| = \ell_i$ *for all* $i = 1, \ldots, n$.

(We proved this lemma in the class a week ago in terms of "strategies" for games with yes-and-no questions.)

References

[1] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory.* Wiley Inc., New York.