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

25.09.2025. Lecture 3.

1. Efficient encoding for the Hamming code. For an error-correcting code $\mathcal{C} = \{c_1, \dots, c_{2^n}\}$ with $c_i \subset \{0,1\}^k$, one can choose an encoding procedure as a mapping

$$\mathrm{Enc}\colon \{0,1\}^n \to \{0,1\}^k$$

that assigns to each binary string of length k the corresponding codeword. Even if we fix the set of codewords $\mathcal{C}$ (for a binary linear code, it is a linear subspace of dimension k in $(\mathbb{Z}/2\mathbb{Z})^k$), we can establish a bijection between $\{0,1\}^n$ and $\mathcal{C}$ in many different ways. Moreover, even a *linear* mapping between $(\mathbb{Z}/2\mathbb{Z})^n$ and $\mathcal{C}$ can be defined in many different ways. However, not all encoding procedures are equally useful in practice. In what follows we discuss a simple and natural encoding for the Hamming codes.

We consider a Hamming code with the length of the codewords $n = 2^m - 1$. Let us recall that the checksum matrix consists of all non-zero binary columns of size m, e.g., for $m = 3$ we have

$$H = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ * & * & & * & & & \end{pmatrix}$$

Observe that m of these columns contain exactly one bit 1 (and $m-1$ bits 0). Let us call these columns *special* (in our example the special are the columns *one*, *two*, and *four*). If we are looking for a solution of the linear system defined by the checksum matrix,

$$\begin{cases} & & & y_4 & +\,y_5 & +\,y_6 & +\,y_7 & = & 0 \\ & y_2 & +\,y_3 & & & +\,y_6 & +\,y_7 & = & 0 \\ y_1 & & +\,y_3 & & +\,y_5 & & +\,y_7 & = & 0 \\ * & * & \uparrow & * & \uparrow & \uparrow & \uparrow & & \\ & & x_1 & & x_2 & x_3 & x_4 & & \end{cases}$$

and we fix all the values x_i for *non-special* columns, then we can easily compute the corresponding values of the *special* x_i. In our example we can encode a string of four bits (x_1, x_2, x_3, x_4) into a codeword $(y_1, y_2, y_3, y_4, y_5, y_6, y_7)$ as follows:

$$\begin{aligned} y_3 &:= x_1 \\ y_5 &:= x_2 \\ y_6 &:= x_3 \\ y_7 &:= x_4 \\ y_1 &:= x_1 + x_2 + x_4 \mod 2 \\ y_2 &:= x_1 + x_3 + x_4 \mod 2 \\ y_4 &:= x_2 + x_3 + x_4 \mod 2 \end{aligned}$$

Such a string of 7 bits will satisfy the system of linear equations determined by the matrix H and, therefore, this string is a codeword of the Hamming code.

Thus, we have a simple encoding algorithm, which computes for a bit string $(x_1 \dots x_k)$ the corresponding codeword $(y_1 \dots y_n)$. Observe that in our encoding procedure every bit x_i of the initial message is embedded directly in the codeword: in the example above, the bits $x_1, \dots, x_4$ appear in the corresponding codeword $(y_1 \dots y_7)$ at the positions $3, 5, 6, 7$ respectively. Such a code is called *systematic*. For a systematic code, if there is no error in the codeword, the procedure of decoding is trivial (in the example discussed above, it is enough to "erase" in the codeword the bits y_1, y_2, and y_4, and the remaining four bits will give the original message $(x_1x_2x_3x_4)$).

2. Asymptotic bound for the size of a ball in the Hamming space. We have defined in the space $\{0,1\}^k$ the Hamming distance $\mathrm{Dist}_H(x,y)$ as the number of positions where the k-bit words x and y differ from each other. In this space, a *sphere* and a *ball* of radius r (centered at x) are defined as

$$S_r(x) = \{y \in \{0,1\}^k \ : \ \mathrm{Dist}_H(x,y) = r\}$$

and

$$B_r(x) = \{y \in \{0,1\}^k \ : \ \mathrm{Dist}_H(x,y) \le r\}$$

respectively.

It is easy to see that for $r \le k$ the number of points in a ball is equal to

$$|B_r(x)| = \binom{k}{0} + \binom{k}{1} + \ldots + \binom{k}{r}.$$

As we will see later, in coding theory it is more important to know this value in the logarithmic scale, i.e., $\log |B_r(x)|$. So let us estimate the asymptotic behavior of $\log |B_r(x)|$ for $r = \alpha k$ (we will need this value for a constant $\alpha < 1/4$). Since $\binom{k}{m} = \frac{k!}{m!(k-m)!}$, we need to compute the sum

$$1 + k + \frac{k!}{2!(k-2)!} + \frac{k!}{3!(k-3)!} + \ldots + \frac{k!}{r!(k-r)!}.$$

Observe that $\frac{k!}{r!(k-r)!}$ is the biggest term in this sum. Therefore,

$$\frac{k!}{r!(k-r)!} \le 1 + k + \frac{k!}{2!(k-2)!} + \frac{k!}{3!(k-3)!} + \ldots + \frac{k!}{r!(k-r)!} \le (r+1) \cdot \frac{k!}{r!(k-r)!}$$

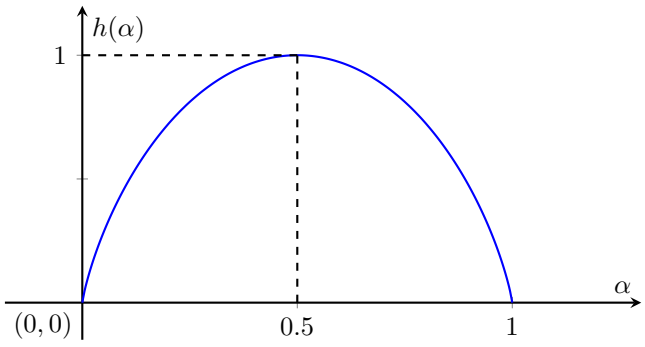
Thus, our aim is to estimate the value $\log\left(\frac{k!}{r!(k-r)!}\right)$. To this end, we use Stirling's approximation of the factorial: $n! = \sqrt{2\pi n}\left(\frac{n}{e}\right)^n \cdot (1 + o(1))$ as $n \to \infty$. In the class we verified that for $r = \alpha k$ we have

$$\log\left(\frac{k!}{r!(k-r)!}\right) = \left(\alpha \log \frac{1}{\alpha} + (1-\alpha)\log\frac{1}{1-\alpha}\right) k + O(\log k).$$

It follows that for $r = \alpha k$

$$2^{h(\alpha)k - O(\log k)} \le |B_r(x)| \le 2^{h(\alpha)k + O(\log k)},$$

where $h(\alpha) = \alpha \log \frac{1}{\alpha} + (1-\alpha)\log\frac{1}{1-\alpha}$.



3. Asymptotic versions of the Hamming and Gilbert bounds. Now we know the number of points in a ball of a given radius in the Hamming space. We can revisit the upper and lower bounds for the sizes of binary codes correcting e errors and estimate the asymptotic meaning of Hamming's and Gilbert's inequalities.

Proposition 1 (Hamming's bound, a necessary condition for the existence of a code)**.** *Let $N = 2^n$ and $e = \alpha k$ (for $\alpha < 1/2$). If $\{c^1, \dots, c^N\} \subset \{0,1\}^k$ is a code correcting e errors, then*

$$N \le \frac{2^k}{1 + \binom{k}{1} + \binom{k}{2} + \dots + \binom{k}{e}},$$

and, therefore,

$$n \le (1 - h(\alpha))k + O(\log k).$$

Therefore, the "capacity" n/k of a binary code that allows to correct a fraction α of errors cannot be greater than

$$1 - h(\alpha) + o(1)$$

as $k \to \infty$, see the red line in the figure below.

Proposition 2 (Gilbert's bound, a sufficient condition for the existence of a code)**.** *Let $N = 2^n$ and $e = \alpha k$ (for $\alpha < 1/4$). If*

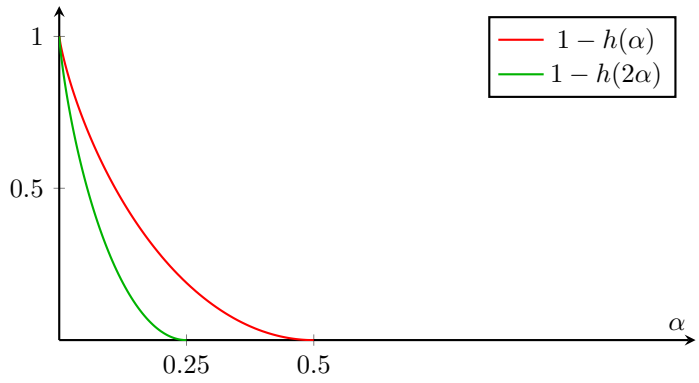
$$N \le \frac{2^k}{1 + \binom{k}{1} + \binom{k}{2} + \dots + \binom{k}{2e}},$$

then there exists a binary code $\{c^1, \dots, c^N\} \subset \{0,1\}^k$ correcting e errors. In other words, the inequality

$$n < (1 - h(\alpha))k - O(\log k)$$

is a sufficient condition for the existence of a binary code correcting e errors.

Therefore, if a point (α, β) lies below the curve $1 - h(\alpha)$ (green line in the graph below), then for large enough k there exists a code $\{c^1, \dots, c^{2^n}\} \subset \{0,1\}^k$ with $\frac{n}{k} \ge \beta$ and correcting $e \ge \alpha k$ errors.



4. Shannon's entropy. For a random variable X with k possible values $c_1, \ldots, c_k$ such that for $i = 1 \ldots k$ $\mathrm{Prob}[X = c_i] = p_i$, we define its Shannon's entropy as

$$H(X) := \sum_{i=1}^{k} p_i \log \frac{1}{p_i}$$

(with the usual convention $0 \cdot \log \frac{1}{0} = 0$).

In the class we proved:

Proposition 1. Let us fix an alphabet with k letters $\{c_1, \ldots, c_k\}$ and consider all words in this alphabet that consist of N letters and contain $p_1 N$ copies of letter c_1, ..., p_k copies of letter c_k. In other words, the numbers p_i are the frequencies of letter c_i for $i = 1, \ldots, k$ (we assume that each number $p_i N$ is an integer). Then the number of all such words is

$$2^{\left(\sum_{i=1}^{k} p_i \log \frac{1}{p_i}\right) N \pm O(\log N)}.$$

In other words, Shannon's entropy is the asymptotic rate of possible compression of a text with given frequencies of the letters.

Proposition 2. *For every random variable X distributed on a set of k values*

$$H(X) \geq 0$$

Moreover, $H(X) = 0$ if and only if the distribution is concentrated at one point (one probability p_i is equal to 1, and the other p_j for $j \neq i$ are equal to 0).

Proposition 3. *For every random variable X distributed on a set of k values*

$$H(X) \leq \log k.$$

Moreover, $H(X) = \log k$ if and only if the distribution is uniform ($p_1 = \ldots = p_k = \frac{1}{k}$).

Proposition 4. Let (X, Y) be a pair of jointly distributed random variables, with joint distribution

$$p_{ij} = \mathrm{Prob}[X = a_i \text{ and } Y = b_j]$$

for $i = 1, \ldots, n$ and $j = 1, \ldots, m$. Then $H(X, Y) \leq H(X) + H(Y)$.

Homework 1. Show that $H(X, Y) \leq H(X) + H(Y)$ if and only if X and Y are independent, i.e., for all i, j

$$\mathrm{Prob}[X = a_i \text{ and } Y = b_j] = \mathrm{Prob}[X = a_i] \cdot \mathrm{Prob}[Y = b_j].$$

5. Entropy in combinatorial problems. In class, we revisited the exercises of finding a single fake coin among n identical coins with a balance scale, and solved some of them using Shannon's entropy.