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

18.09.2025. Lecture 2.

1. Binary error-correcting codes. We define a set of binary strings $\mathcal{C} = \{y^1, \ldots, y^N\} \subset \{0,1\}^k$ to be a *binary code correcting e errors* if for every $w \in \{0,1\}^k$ there exists at most one y^j in the code such that

$$\mathrm{Dist}_H(w, y^j) \leq e.$$

The words (bit strings) in a code are called *codewords*.

Observation 1. A set $\mathcal{C} \subset \{0,1\}^k$ is a binary code correcting e errors if and only if

$$\min_{\substack{y^i,\, y^j \in \mathcal{C} \\ y^i \neq y^j}} \mathrm{Dist}_H(y^i, y^j) \geq 2e+1.$$

In class we proved the following two statements.

Proposition 1 (Hamming's bound, a necessary condition for the existence of a code)**.** *If $\{y^1, \ldots, y^N\} \subset \{0,1\}^k$ is a code correcting e errors, then*

$$N \leq \frac{2^k}{1 + \binom{k}{1} + \binom{k}{2} + \ldots + \binom{k}{e}}.$$

In particular, for codes correcting one error we have $N \leq \frac{2^k}{1+k}$.

Proposition 2 (Gilbert's bound, a sufficient condition for the existence of a code)**.** *If*

$$N \leq \frac{2^k}{1 + \binom{k}{1} + \binom{k}{2} + \ldots + \binom{k}{2e}},$$

then there exists a binary code $\{y^1, \ldots, y^N\} \subset \{0,1\}^k$ correcting e errors.

Homework 1. A mathematician has chosen in $\{0,1\}^{10}$ a subset of k binary words $x_1, x_2, \ldots, x_k$ so that the Hamming distance between any two of these words is $\mathrm{dist}_H(x_i, x_j) \geq 6$ for all $i \neq j$. Prove that $k < 20$.

2. Linear algebra for arithmetic modulo 2. We can understand the set $\{0,1\}^k$ as a *vector space* $(\mathbb{Z}/2\mathbb{Z})^k$, i.e., the k-dimensional space over the field of two elements. In this space, the sum of two vectors is defined as the coordinate-wise XOR. The vector space $(\mathbb{Z}/2\mathbb{Z})^k$ contains a special element

$$\mathbf{0} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

(zero vector) such that for all $\mathbf{x} \in (\mathbb{Z}/2\mathbb{Z})^k$ we have $\mathbf{x} + \mathbf{0} = \mathbf{x}$. In the field $\mathbb{Z}/2\mathbb{Z}$ there are only two elements (zero and one), so scalar multiplication is trivial: multiplication by 1 does not change the vector, multiplication by 0 gives the zero vector.

A family of vectors $\mathbf{x}_1, \ldots, \mathbf{x}_m$ in $(\mathbb{Z}/2\mathbb{Z})^k$ is called *linearly independent* if for all indices $1 \leq i_1 < \ldots < i_s \leq m$ the sum

$$\mathbf{x}_{i_1} + \ldots + \mathbf{x}_{i_s} \neq \mathbf{0}.$$

A linear *subspace* in $(\mathbb{Z}/2\mathbb{Z})^k$ is a set $V \subset (\mathbb{Z}/2\mathbb{Z})^k$ such that for all vectors $\mathbf{x}, \mathbf{y} \in (\mathbb{Z}/2\mathbb{Z})^k$ the sum $\mathbf{x} + \mathbf{y}$ also belongs to V. In class we proved that in every subspace $V \subset (\mathbb{Z}/2\mathbb{Z})^k$ there is a subset $B = \{\mathbf{z}_1, \ldots, \mathbf{z}_m\} \subset V$ called a *basis* such that

- the elements of B are linearly independent,
- every element $\mathbf{x} \in V$ can be represented as the sum of several elements of B,

$$\mathbf{x} = \mathbf{z}_{i_1} + \ldots + \mathbf{z}_{i_s}.$$

We observed that if a basis of V consists of m vectors, then the cardinality of V is 2^m. This implies, in particular, that all bases of V must contain the same number of elements m. This number is called the *dimension* of the linear subspace V.

An m-dimensional linear subspace $V \subset (\mathbb{Z}/2\mathbb{Z})^k$ can be defined in two different ways:

- by m vectors that provide a basis in V,
- by $(k-m)$ linear equations such that V is the set of all solutions of this system of linear equations.

Equivalently, these two ways can be explained as follows. A linear subspace $V \subset (\mathbb{Z}/2\mathbb{Z})^k$ can be specified

- by a *generator matrix* G of size $k \times m$ such that the vectors of V are exactly those that can be represented as

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{pmatrix} = G \cdot \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{pmatrix}$$

  for $w_1, \ldots, w_m \in \{0, 1\}$ (the m columns of G form a basis of V);
- by a *parity-check matrix* H with k columns and $m-k$ rows, such that the codewords are all vectors satisfying

$$H \cdot \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

3. The Hamming codes. A set $\mathcal{C} \subset \{0,1\}^k$ is called a binary *linear code* correcting e errors if $\mathcal{C}$ is a binary code correcting e errors (as defined above) and at the same time this set is a linear subspace of $(\mathbb{Z}/2\mathbb{Z})^k$. Observe that the zero vector (the vector whose coordinates are all equal to zero) is always a codeword in a linear code, and the total number of codewords in a binary linear code is always a power of 2.

Proposition 3. *A binary linear code corrects e errors if and only if every non-zero codeword contains at least $2e+1$ ones.*

Proof. First of all, we observe that a code corrects e errors if and only if the distance between every two codewords is at least $2e+1$. Thus, we need to characterize the linear codes where the distance between every two codewords is greater than or equal to $2e+1$.

If v and w are codewords in a linear code, then the bitwise XOR of v and w is also a codeword u. Therefore, the Hamming distance between v and w is equal to the Hamming weight of some codeword u. Thus, we can guarantee that the distance between every two codewords is at least $2e+1$ if the Hamming weight of every non-zero codeword z is at least $2e+1$.

The *only if* part follows from the observation that the Hamming weight of a codeword z is equal to the Hamming distance between z and the zero codeword $\underbrace{000\ldots0}_{n}$. □

In class we proved the proposition:

Proposition 4. *A parity-check matrix H with k columns and m rows defines a linear code that corrects e errors if and only if every $2e$ columns in H are linearly independent.*

Corollary 1. *A parity-check matrix H with k columns and m rows defines a linear code that corrects one error if and only if (i) H contains no column of all zeros, and (ii) all columns in H are pairwise different.*

A matrix H with m rows such that *there is no column of all zeros and all columns in H are pairwise different* can contain at most $k = 2^m - 1$ columns (all non-zero binary vectors of length m, without repetition). Here is an example of such a matrix for $m = 3$ and $k = 2^m - 1 = 7$:

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

Such a checksum matrix represents a system of linear equations (as usual, in the arithmetic modulo 2) with k variables :

$$H \cdot \begin{pmatrix} x_1 \\ \vdots \\ x_k \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$$

In particular, for the 3×7-matrix H from the example above we have the system of equations

$$\begin{cases} x_4 + x_5 + x_6 + x_7 &= 0, \\ x_2 + x_3 + x_6 + x_7 &= 0, \\ x_1 + x_3 + x_5 + x_7 &= 0. \end{cases}$$

This system contains m (independent) linear equations for k variables. Therefore, the set of all solutions $\mathbf{x} = (x_1, \ldots, x_k)$ of this system is a linear subspace in $(\mathbb{Z}/2\mathbb{Z})^k$ of dimension

$$k - m = 2^m - m - 1.$$

Therefore, we obtain a linear code with

$$N = 2^{k-m} = 2^{2^m - m - 1}$$

codewords. This code is called the *Hamming code*.

Important fact: The Hamming codes are optimal among all codes correcting one error: we cannot obtain any code correcting one error with the same length of codewords k and a greater number of codewords. Indeed, the Hamming balls of radius 1 contain $k + 1$ points; since the balls of radius 1 constructed around all codewords must be disjoint, we have

$$(k + 1) \cdot [\text{number of codewords}] \leq 2^k.$$

Thus, the number of codewords is at most

$$\frac{2^k}{k+1} = \frac{2^k}{2^m} = 2^{k-m} = 2^{2^m - m - 1}$$

(see the proof of the *Hamming bound* above), which is exactly the number of codewords in the code that we constructed.

Homework 2. The parity check matrix of the Hamming code with codewords of length 15 is

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

(a) How many errors does this code correct?

(b) Does there exist a binary word x of length 15 that is not a codeword for this code and that cannot be obtained from any codeword by inverting one bit?

(c) Is the bit sequence $x = (001010011000011)$ a codeword of this code? If not, which bit should be inverted to obtain a codeword?

(d) Is the bit sequence $x = (111111100000000)$ a codeword of this code? If not, which bit should be inverted to obtain a codeword?

4. Correcting errors in the Hamming code. If H is the checksum matrix of the Hamming code, then for every vector

$$\mathbf{w} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

(with 1 at the i-th position and 0s at all other positions), the product $H \cdot \mathbf{x}$ is exactly the i-th column of H. (In our construction, the i-th column of H represents the binary expansion of the number i.) We will use this fact to describe a simple algorithm for decoding the Hamming code that involves revealing an eventual error in the codeword.

Let $\mathbf{x} = (x_1 \dots x_k)$ be a codeword of the Hamming code, i.e., a string of bits such that

$$H \cdot \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_k \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

(where H is the checksum matrix of the Hamming code, with m rows and $k = 2^m - 1$ columns). Let us flip one bit in $\mathbf{x}$ and denote the result $\mathbf{x}'$. The Hamming code corrects one error, so we can reconstruct $\mathbf{x}$ uniquely given the "corrupt" vector $\mathbf{x}'$. We will show how to do it quite efficiently.

Observe that $\mathbf{x}' = \mathbf{x} + \mathbf{w}$, where

$$\mathbf{w} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

is the vector with 1 at the only position where $\mathbf{x}$ and $\mathbf{x}'$ differ and with 0s at all other positions. If we multiply $\mathbf{x}'$ by the check-sum matrix H, we will get a string of m bits, which is called the *syndrome* of $\mathbf{x}'$. It is equal to

$$H \cdot \mathbf{x}' = H \cdot (\mathbf{x} + \mathbf{w}) = H \cdot \mathbf{x} + H \cdot \mathbf{w} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} + H \cdot \mathbf{w} = H \cdot \mathbf{w}.$$

So the syndrome of $\mathbf{x}'$ is equal to $H \cdot \mathbf{w}$. The feature of the Hamming code is that the syndrome is the binary expansion of the index i of the corrupt bit. For example, if $m = 3$ and

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

the string $\mathbf{x} = (1010101)$ is a codeword. Let us flip the 4-th bit, $\mathbf{x}' = (1011101)$. When we compute the syndrome of $\mathbf{x}'$ (the product of the checksum matrix H and the column-vector $\mathbf{x}'$), we get

$$\begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0\cdot 1 + 0 \cdot 0 + 0 \cdot 1 + 1 \cdot 1 + 1 \cdot 1 + 1 \cdot 0 + 1 \cdot 1 \\ 0 \cdot 1 + 1 \cdot 0 + 1 \cdot 1 + 0 \cdot 1 + 0 \cdot 1 + 1 \cdot 0 + 1 \cdot 1 \\ 1 \cdot 1 + 0 \cdot 0 + 1 \cdot 1 + 0 \cdot 1 + 1 \cdot 1 + 0 \cdot 0 + 1 \cdot 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

(the calculations done modulo 2). The resulting syndrome (100) is the binary expansion of the number $4 = 1 \cdot 2^2 + 0 \cdot 2^1 + 0 = 100_2$, which is the index of the position of the corrupt bit.

5. The game *guess a number* with one false response. In the class we discussed how to use the Hamming codes to play the game *guess a number*, where the first player chooses a natural number x between 1 and N, and the second player asks *yes/no* questions. The first player is allowed to give at most one false answer. The challenge is to suggest a strategy with the minimum number of questions so that the second player can reveal the number x, despite one possible false answer of the first player.

In what follows we discuss lower and upper bounds on the size of the optimal strategy for a specific example of N. Let us take, for instance, $N = 2048$. Assume that there is a strategy where the second player can determine $x \in \{1, \ldots, N\}$ after asking k questions. We fix this strategy of the second player. We know that for every x, the first player may choose a question between 1 and k to give a false answer, or decide to give all true answers. Thus, for every possible x there are $k + 1$ possible sequences of answers. Since each sequence of answers must correspond to only one number x, we have

$$N \leq 2^k/(k+1).$$

As $N = 2048$, we see that k must be *at least* 15 (for $k = 1, \ldots, 14$ we would have $2^k/(k+1) < 2048$, which gives a contradiction with the inequality above).

Let us show now that 15 questions is enough to guess an integer x between 1 and $n = 2048$ with one false answer. We will construct such a strategy using the Hamming code with parameters $m = 4$ and $k = 2^4 - 1 = 15$. Indeed, in this code we have $2^{k-m} = 2048$ codewords of length 15, and every two codewords differ in at least 3 positions. We can associate these codewords with the numbers $\{1, \ldots, N\}$. The second player should ask questions

- Does the 1st bit of the xth codeword equal to 1 ?
- Does the 2nd bit of the xth codeword equal to 1 ?
- Does the 3rd bit of the xth codeword equal to 1 ?

  $\vdots$

- Does the 15-th bits of your codeword equal to 1 ?

The answers to this question give us the bits of the x-th codeword, possibly with one error (if one answer was false). But even if one of the answers is false, the Hamming code allows to reconstruct the entire codeword. This is enough to reveal the number x chosen by the first player.

In the class we also discussed an optimal strategy in this game for $N = 100$.