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

11.09.2025. Lecture 1.

1. The game "guess a number" : one player chooses an integer x between 1 and $n = 100$, another player must determine this number by asking questions with answers *yes* or *no*. How many questions should be asked to determine the number x with certainty ? There is a simple strategy that allows to find the chosen number in $\lceil \log_2 n \rceil$ questions (bisection). Moreover, there is a non-adaptive strategy with the same number of questions (the second player asks bits of the binary expansion of the chosen number ; in this strategy all questions can be formulated in advance, before the first response).

These strategies are optimal : no strategy allows to reveal the chosen number in less than $\lceil \log_2 n \rceil$ questions (in the worst case). Indeed, every guessing strategy can be represented as a binary rooted tree (with questions in the internal nodes and the guessed numbers in the leaves). Since such a tree must have at least n leaves (one leave for each possible answer), the depth of the tree must be at least $\lceil \log_2 n \rceil$.

2. Weighing problems (finding a counterfeit coin). In the class we discussed several variations of the classic problem of finding a counterfeit coin. In all these problems we assume that there are several identical-looking coins, one of which differs from the others in weight (while all other coins have the same weight). We have at our disposal a scale without supplementary weights. With this scale we can compare any two groups of coins and find out whether they differ in weight or not (and if they differ, which group is lighter and which one is heavier).

Example 1. We are given $n = 9$ coins and one of them is fake. It is known that the fake coin is lighter than the genuine ones. How many weighings does it take to find the fake coin ?

It is not hard to see that this task requires 2 weighings : there exists a strategy that finds the fake coin in *two* operations, and there is no strategy which does the same in only *one* operation.

Example 2. We are given again $n = 9$ coins and one if them is fake. It is known that the fake coin is lighter or heavier than the genuine ones, but it is not known which is the case. How many weighings does it take to find the fake coin ?

In this setting we need 3 weighings : there exists a strategy that finds the fake coin in *three* operations, and there is no strategy which does the same in only *two* operations. The intuitive explanation is that in any strategy we learn some unnecessary information (in most cases, we also learn whether the fake coin is lighter or heavier than the genuine ones). In the class we discussed a complete proof of this statement.

Example 3. We are given $n = 12$ coins and one of them is fake. The fake coin can be heavier or lighter than the genuine ones. How many weighings does it take to find a fake coin ? In the class we found out that the fake one can be found in *four* weighings, and that *two* operations is not enough.

Homework 1. What is the number of weighings required to find among 12 coins the fake one (which can be heavier or lighter than the genuine ones) ? *Hint :* improve the lower bound 3 or the upper bound 4 proven in the class.

Homework 2. What is the number of weighings required to find among 14 coins the fake one (which can be heavier or lighter than the genuine ones) ?

3. The game "guess a number" revisited : we assume again that one player chooses an integer number x between 1 and $n = 100$, and another player should find this number by asking questions with answers *yes* or *no*. However, this time the first player is allowed to lie once (i.e., may give at most one false answer). How many questions should be asked to determine the number x in this setting ?

In the class we discussed an adaptive strategy that requires $2\lceil \log_2 n \rceil + 1$ questions (repeat each question ; if the answers are different, ask the same question for the third time). We also observed that in every strategy the second player will learn a lot of extra information — whether the first player has lied or not, and in which answer. This observation allows us to prove that every strategy should contain *at least* 11 questions (this is the minimum number k such that $2^k \geq n \cdot (k+1) = 100(k+1)$).

In this lecture we did not determine the optimal number of questions in this guessing game. We only have the lower bound $k \geq 11$ and the upper bound $k \leq 2\lceil \log_2 100 \rceil + 1 = 15$.

4. Hamming distance.

Definition 1. The *Hamming distance* between binary words $\bar{x} = x_1 \dots x_k, \bar{y} = y_1 \dots y_k$ (both of the same length k) is defined as the number of indices i such that $x_i \neq y_i$ (the number of positions between 1 and k where the words $\bar{x}$ and $\bar{y}$ differ from each other). We denote the Hamming distance between $\bar{x}$ and $\bar{y}$ as $\mathrm{Dist}_H(\bar{x}, \bar{y})$.

Observe that the Hamming distance satisfies the basic properties of distance familiar from Euclidean space :

— $\mathrm{Dist}_H(x, x) = 0$,

— $\mathrm{Dist}_H(x, y) = \mathrm{Dist}_H(y, x)$,

— $\mathrm{Dist}_H(x, y) \leq \mathrm{Dist}_H(x, z) + \mathrm{Dist}_H(y, z)$.

In the space $\{0,1\}^k$ with the Hamming distance we can re-use the standard notion of a sphere and a ball. A *ball* of radius r with the center at x is the set

$$B_r(x) = \{y \ : \ \mathrm{Dist}_H(x, y) \leq r\}.$$

A *sphere* of radius r with the center at x is the set

$$S_r(x) = \{y \ : \ \mathrm{Dist}_H(x, y) = r\}.$$

Let us mention that the Hamming distance between $x_1 \dots x_k$ and the word that consists of n zeros $\underbrace{00\dots0}_{k}$ is equal to the number of *ones* in the words $x_1 \dots x_k$. This number is called the *Hamming weight* of $x_1 \dots x_k$.

5. Binary error correcting codes. We say that a set of binary strings $\{y^1, \dots, y^N\}$ (where each y^i belongs to $\{0,1\}^k$) is 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 strings (words) in a code are called *codewords* of the code. The definition can be reformulated as follows : a set of strings is a code correcting e errors, if the balls of radius e (in the Hamming metrics) centered at the codewords are pairwise disjoint. This definition has a clear combinatorial meaning : if at most e bits of a codeword are altered by noise, the original codeword can still be reconstructed. Observe that a set of words is a code correcting e errors, if and only if for every two codewords y^i and y^j we have $\mathrm{Dist}_H(y^i, y^j) \geq 2e+1$.

Proposition 1. *If $\{y^1, \dots, y^N\} \subset \{0,1\}^k$ is a code correcting 1 error, then $2^k \geq N(k+1)$.*

Démonstration. In the space $\{0,1\}^k$, every ball of radius 1 contains $1 + k$ points (the point which is the center of the ball and the k points at the distance exactly 1 from the center). The balls of radius 1 centered at the codewords must be disjoint, and they all belong to the set $\{0,1\}^k$ of cardinality 2^k. □

Thus, if we need to correct one error, then we cannot have more than $2^k/(k+1)$ codewords of length k. For example, we cannot have more than $2^7/(7+1) = 16$ binary codewords of length 7 in a code correcting one error.