Artificial Intelligence and Machine Learning 2023/2024

Week 2 Tutorial and Additional Exercises

Dr Ruchit Agrawal

School of Computer Science

January 29, 2024

Complexity of Programs

1 Mathematical Preliminaries

1.1 Laws of probability

The probability of an event A is written $\mathbb{P}(A)$. It is an element of $\{x \in \mathbb{R} \mid 0 \leqslant x \leqslant 1\}$.

The basic laws of probability are as follows:

- An impossible event has probability 0.
- A certain event has probability 1.
- For an event A, we have $\mathbb{P}(\text{not } A) = 1 - \mathbb{P}(A)$. For example, suppose the probability that it's raining is $\frac{1}{3}$. Then the probability that it's not raining is $\frac{2}{3}$.
- For events A and B that are mutually exclusive, we have $\mathbb{P}(A \text{ or } B) = \mathbb{P}(A) + \mathbb{P}(B)$. For example, suppose the probability that it's raining is $\frac{1}{3}$ and the probability that it's sunny is $\frac{1}{5}$. If these are mutually exclusive events, then the probability that it's either raining or sunny is $\frac{8}{15}$.
- For events A and B that are independent, we have $\mathbb{P}(A \text{ and } B) = \mathbb{P}(A) \times \mathbb{P}(B)$. For example, suppose the probability that it's raining is $\frac{1}{3}$ and the probability that John is happy is $\frac{1}{5}$. If these are independent events, then the probability that it's raining and John is happy is $\frac{1}{15}$.

1.2 Important summations

Here are some summations that come up again and again, so make sure you know them.

$$
\begin{aligned}
0 + 1 + 2 + \cdots + (n-1) &= \frac{1}{2}n(n-1) \\
1 + 2 + 3 + \cdots + n &= \frac{1}{2}n(n+1) \\
1 + b + b^2 + \cdots + b^{n-1} &= \frac{b^n - 1}{b - 1} \qquad (b \neq 1) \\
1 + b + b^2 + \cdots + b^n &= \frac{b^{n+1} - 1}{b - 1} \qquad (b \neq 1)
\end{aligned}
$$

1.3 Upper and lower bounds

- It will take me at least an afternoon to clear my office. *Lower bound.*
- Clearing the office will take me a week at most. *Upper bound.*
- Building the new railway will cost no more than 70 billion pounds. *Upper bound.*
- For the café to be viable, we need at least 30 customers a day, maybe more. *Lower bound.*

Note that an upper bound gives a guarantee.

2 Running time of a program

2.1 Best, average and worst cases

Consider a sample problem, e.g. sorting (arranging items in order)

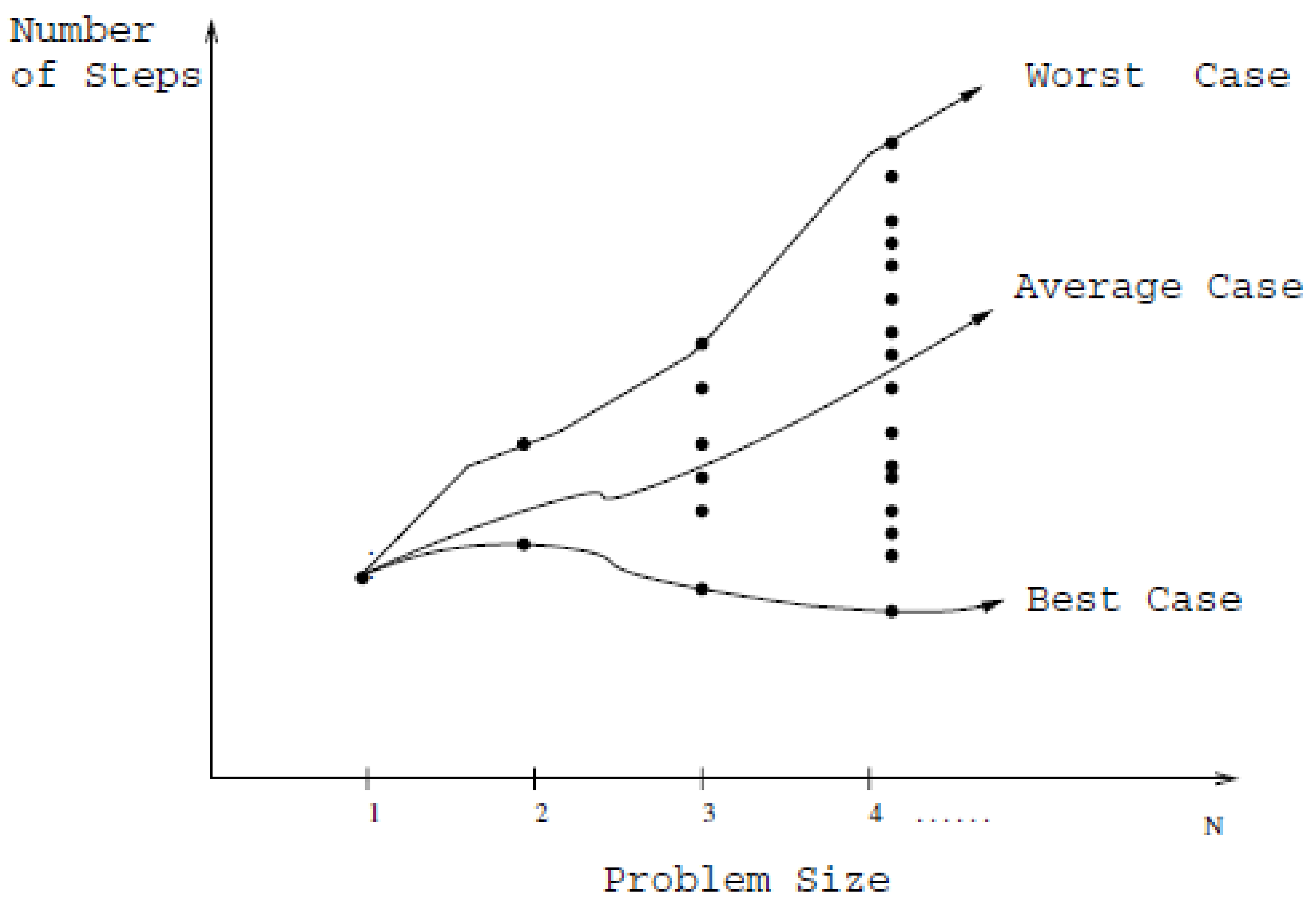

Figure 2.1: Best, worst, and average-case complexity

The above plot illustrates three functions:

- **Worst-case complexity**: It gives us an *upper bound* on the cost. It is determined by the *most difficult* input and provides a guarantee for all inputs.
- **Best-case complexity**: It gives us a *lower bound* on the cost. It is determined by the *easiest input* and provides a goal for all inputs.
- **Average-case complexity**: It gives us the *expected cost* for a random input. It requires a model for *random input* and provides a way to predict performance.

We will mainly focus input size and average/worst case complexity analysis.

Lets consider the following program that operates on an array of characters that are all `a` or `b` or `c`.

```
void f (char[] p) {
  elapse(1 second);
  for (nat i = 0; i<p.length(), i++) {
    if (p[i]=='a') {
      elapse(1 second);
    } else {
      elapse(2 seconds);
    }
    elapse (1 second);
  }
}
```

For an array of length 0, the running time is 1 second.
For an array of length 1, assuming `a`, `b`, `c` are equally likely:

Array contents	Probability	Time	Probability $\times$ time
`a`	$\frac{1}{3}$	$3s$	$1s$
`b`	$\frac{1}{3}$	$4s$	$1\frac{1}{3}s$
`c`	$\frac{1}{3}$	$4s$	$1\frac{1}{3}s$
Worst case: `b`		$4s$	
Average case			$3\frac{2}{3}s$

For an array of length 2, assuming `a`, `b`, `c` are equally likely and the characters are independent:

Array contents	Probability	Time	Probability $\times$ time
`aa`	$\frac{1}{9}$	$5s$	$\frac{5}{9}s$
`ab`	$\frac{1}{9}$	$6s$	$\frac{2}{3}s$
`ac`	$\frac{1}{9}$	$6s$	$\frac{2}{3}s$
`ba`	$\frac{1}{9}$	$6s$	$\frac{2}{3}s$
`bb`	$\frac{1}{9}$	$7s$	$\frac{7}{9}s$
`bc`	$\frac{1}{9}$	$7s$	$\frac{7}{9}s$
`ca`	$\frac{1}{9}$	$6s$	$\frac{2}{3}s$
`cb`	$\frac{1}{9}$	$7s$	$\frac{7}{9}s$
`cc`	$\frac{1}{9}$	$7s$	$\frac{7}{9}s$
Worst case: `bb`		$7s$	
Average case			$6\frac{1}{3}s$

Now consider an array of length n.

1. When does the worst case arise? (Just give one example.) What is its running time?
2. Assuming `a`, `b`, `c` are equally likely and the characters are independent, what is the average case running time?

Formal optimisation problem

- Recall the canonical form of an optimisation problem:

$$\begin{array}{rlr} \text{maximise/minimise} & f(\mathbf{x}) & \\ \text{subject to} & g_i(\mathbf{x}) \leq 0, & i = 1, \ldots, m \\ & h_j(\mathbf{x}) = 0, & j = 1, \ldots, n \end{array}$$

- $\mathbf{x}$ is the vector of *design variables*.
- f is the *objective function*, e.g. the cost or quality of a solution.
- $g_1, \ldots, g_m$ are the *inequality constraints* and $h_1, \ldots, h_n$ are the *equality constraints*.
- In a *multi-objective* optimisation problem, there are more than one objective functions, e.g. $f_1, f_2, \ldots, f_k$.

Formal optimisation problem (continued)

Some more definitions:

- Each value of $\mathbf{x}$ is a *solution* to the optimisation problem.
- The *search space* consists of all possible solutions.
- A solution that satisfies the constraints is called *feasible*. A solution that does not satisfy the constraints is called *infeasible*.

Exercise 1

Consider the following problem:

- A company makes square boxes and triangular boxes. Square boxes take 2 minutes to make and sell for a profit of 4. Triangular boxes take 3 minutes to make and sell for a profit of 5. No two boxes can be created simultaneously. A client wants at least 25 boxes including at least 5 of each type in one hour. What is the best combination of square and triangular boxes to make so that the company makes the most profit from this client?
- Formalize this problem as a canonical optimisation problem, but consider it is ok for the objective to be a function to be maximised instead of minimised. Identify the design variables, the objective function and the constraints.

Exercise 1: Solution

- Let $x_1 \in \mathbb{N}$ be the number of square boxes and $x_2 \in \mathbb{N}$ be the number of triangular boxes. These are the design variables. Write $\mathbf{x} = (x_1, x_2)$. The formal optimisation problem is the following:

$$\begin{aligned} \text{maximise} \quad & 4x_1 + 5x_2 \\ \text{subject to} \quad & 2x_1 + 3x_2 \leq 60 \\ & x_1 \geq 5 \\ & x_2 \geq 5 \\ & x_1 + x_2 \geq 25 \end{aligned}$$

- Let us find the objective function and the constraints so that they follow the canonical formulation.

Exercise 1: Solution (continued)

- Objective function:

$$f(\mathbf{x}) = 4x_1 + 5x_2$$

- Constraints:

$$\begin{aligned} g_1(\mathbf{x}) &= 2x_1 + 3x_2 - 60 \\ g_2(\mathbf{x}) &= 5 - x_1 \\ g_3(\mathbf{x}) &= 5 - x_2 \\ g_4(\mathbf{x}) &= 25 - x_1 - x_2 \end{aligned}$$

  There are no equality constraints, so no $h_1, h_2, \ldots$ functions.

- With these definitions, the canonical problem can be written

$$\begin{array}{ll} \text{maximise} & f(\mathbf{x}) \\ \text{subject to} & g_i(\mathbf{x}) \leq 0, \qquad i = 1, 2, 3, 4 \end{array}$$

Exercise 2

Consider the following problem:

- A woman makes pins and earrings. Each pin takes 1 hour to make and sells for a profit of 8. Each earring takes 2 hours to make and sells for a profit of 20. She wants to make exactly as many pins as earrings. She has 40 hours and wants to have made at least 20 items, including at least 4 of each item. How many each of pins and earrings should the woman make to maximise her profit?
- Formalize this problem as a canonical optimisation problem, but consider it is ok for the objective to be a function to be maximised instead of minimised. Identify the design variables, the objective function and the constraints.

Exercise 2: Solution

- Let $x_1 \in \mathbb{N}$ be the number of pins and $x_2 \in \mathbb{N}$ be the number of earrings. These are the design variables. Write $\mathbf{x} = (x_1, x_2)$. The formal optimisation problem is the following:

$$\begin{array}{ll} \text{maximise} & 8x_1 + 20x_2 \\ \text{subject to} & x_1 + 2x_2 \leq 40 \\ & x_1 + x_2 \geq 20 \\ & x_1 \geq 4 \\ & x_2 \geq 4 \\ & x_1 = x_2 \end{array}$$

- Let us find the objective function and the constraints so that they follow the canonical formulation.

Exercise 2: Solution (continued)

- Objective function:

$$f(\mathbf{x}) = 8x_1 + 20x_2$$

- Constraints:

$$\begin{aligned} g_1(\mathbf{x}) &= x_1 + 2x_2 - 40 \\ g_2(\mathbf{x}) &= 20 - x_1 - x_2 \\ g_3(\mathbf{x}) &= 4 - x_1 \\ g_4(\mathbf{x}) &= 4 - x_2 \\ h_1(\mathbf{x}) &= x_1 - x_2 \end{aligned}$$

- With these definitions, the canonical problem can be written

$$\begin{array}{lll} \text{maximise} & f(\mathbf{x}) & \\ \text{subject to} & g_i(\mathbf{x}) \leq 0, & i = 1, 2, 3, 4 \\ & h_j(\mathbf{x}) = 0, & j = 1 \end{array}$$

Formulation

Design variable	Objective function	Constraints
$x_1 \leftarrow$ # square boxes $x_2 \leftarrow$ # triangular boxes. $x = (x_1, x_2)$	Profit (maximize) $4x_1 + 5x_2$	Time $2x_1 + 3x_2 \le 60$ Min. quantity $x_1 \geq 5$ — sq. boxes $x_2 \geq 5$ — trig " $x_1 + x_2 \geq 25$

Design variable.

formalize

→ canonical formulation