Mathematical Foundations

Data Science Project: An Inductive Learning Approach

Prof. Dr. Filipe A. N. Verri

About these slides

These slides are companion material for the book

Data Science Project: An Inductive Learning Approach

Prof. Dr. Filipe A. N. Verri

`https://leanpub.com/dsp`

All intellectual content comes from the book and is not AI-generated. Slides were produced with the assistance of Claude Code.

Maar ik maak steeds wat ik nog niet kan om het te leeren kunnen.

— Vincent van Gogh, The Complete Letters, Volume Three

Overview

Contents

- Algorithms and data structures
- Set theory
- Linear algebra
- Probability

Objectives

- Consolidate notations and definitions used throughout the book
- Remind the reader of the main computational, mathematical, and statistical concepts

Algorithms and data structures

Computational complexity

- Total amount of resources (time, space) as a function of input size
- We focus on **asymptotic** complexity — behavior as input grows

Big-O notation: f is $O(g)$ if $\exists\, c > 0$ such that $f(n) \leq c\, g(n)$ for all $n \geq n_0$.

Common complexity classes:

$$O(1) < O(\log n) < O(n) < O(n \log n) < O(n^2) < O(2^n) < O(n!)$$

Big-O properties

- Worst-case analysis: upper bound on resources for any input of size n
- An $O(n)$ algorithm is not always faster than an $O(n^2)$ algorithm — only for large enough n
- Sequential composition:

$$O(f) + O(g) = O(\max(f, g))$$

Divide and conquer

1. Divide the problem into smaller subproblems
2. Solve each subproblem recursively
3. Combine the solutions

Examples: merge sort, quick sort, binary search.

Binary search

Given a sorted array $\vec{a} = [a_1, a_2, \dots, a_n]$ and key x:

1. Set $l \leftarrow 1$, $r \leftarrow n$
2. While $l \leq r$:
   - $m \leftarrow \lfloor (l + r)/2 \rfloor$
   - If $x = a_m$: return *true*
   - If $x < a_m$: $r \leftarrow m - 1$; else $l \leftarrow m + 1$
3. Return *false*

Search space halved at each step: $\frac{n}{2^{i-1}} = 1 \implies i = 1 + \log n$

Time complexity: $O(\log n)$.

Greedy algorithms

- Solved with incremental, locally optimal steps
- Overall solution is **not** guaranteed to be optimal
- Examples: Dijkstra's algorithm, Prim's algorithm

Knapsack problem:

$$\text{maximize} \sum_{i=1}^{n} v_i x_i \quad \text{subject to} \sum_{i=1}^{n} w_i x_i \leq W$$

Greedy heuristic: sort by value, add if it fits.
Time complexity: $O(n \log n)$ (dominated by sorting).

Brute force and backtracking

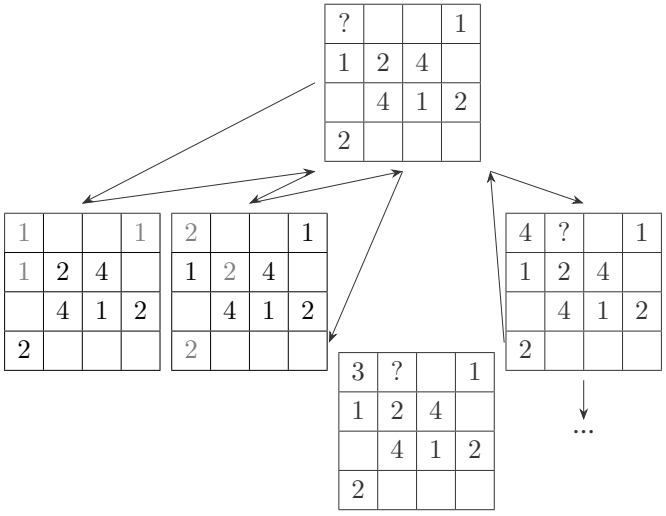
Brute force:

- Try all possible solutions
- Guaranteed optimal but usually exponential: $O(2^n)$ for knapsack

Backtracking:

- Solve incrementally; undo when a constraint is violated
- Special case of brute force, often exponential
- Example: Sudoku — n^m possible fillings for m empty cells

Backtracking — Sudoku example



Constraint violations shown in gray cause backtracking.

Data structures

- **Array**: homogeneous collection accessed by index; $\vec{a} = [a_1, a_2, \ldots, a_n]$
- **Stack**: LIFO; push and pop from top only
- **Queue**: FIFO; enqueue at back, dequeue from front
- **Tree**: nodes with children, no cycles; root, leaves
- **Graph**: nodes with edges; directed or undirected

Binary tree

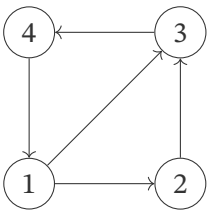
Recursive definition:

$$T = \begin{cases} \emptyset & \text{if empty} \\ (v, T_l, T_r) & \text{value } v \text{ with left } T_l \text{ and right } T_r \end{cases}$$

Parentheses notation: $(1, (2, \emptyset, \emptyset), (3, \emptyset, \emptyset))$
is a tree with root 1, left child 2, right child 3.

Graphs

A graph $G = (V, E)$ where V = vertices, $E \subseteq V \times V$ = edges.



Adjacency matrix:

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

Weighted graphs: $w : E \to \mathbb{R}$ assigns a weight to each edge.

Set theory

Sets

A set is an unordered collection of unique elements: $\{1, 2, 3\}$.

Special sets:

- **Universe set** Ω: all elements in a given context
- **Empty set** $\emptyset$: no elements

Set operations

- **Union**: $A \cup B$ — elements in A or B
- **Intersection**: $A \cap B$ — elements in both A and B
- **Difference**: $A \setminus B$ — elements in A but not B
- **Complement**: $A^c = \Omega \setminus A$

Set relations

- **Inclusion**: $A \subseteq B$ — all elements of A are in B
- **Proper inclusion**: $A \subset B$ — $A \subseteq B$ and $A \neq B$
- **Equality**: $A = B$ iff $A \subseteq B$ and $B \subseteq A$
- **Disjointness**: $A \cap B = \emptyset$

Properties of set operations and relations

Given sets A, B, C:

- *Commutativity:* $A \cup B = B \cup A, \quad A \cap B = B \cap A$
- *Associativity:* $(A \cup B) \cup C = A \cup (B \cup C)$
- *Distributivity:* $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$

De Morgan's laws:

$$(A \cup B)^c = A^c \cap B^c \qquad (A \cap B)^c = A^c \cup B^c$$

Difference in terms of complement: $A \setminus B = A \cap B^c$

Inclusion properties

- *Reflexivity:* $A \subseteq A$
- *Transitivity:* $A \subseteq B$ and $B \subseteq C$ implies $A \subseteq C$

Relation to Boolean algebra

Operation	Set	Boolean
Union / OR	$A \cup B$	$A \vee B$
Intersection / AND	$A \cap B$	$A \wedge B$
Complement / NOT	A^c	$\neg A$

De Morgan's laws also hold: $\neg(A \vee B) = \neg A \wedge \neg B$

Boolean algebra is the foundation of digital electronics and programming control flow.

Linear algebra

Basic objects

- **Vector**: ordered collection of numbers; $\vec{v} = [v_i]_{i=1,\ldots,n}$
- **Matrix**: rectangular array; $A = (a_{ij})_{i=1,\ldots,n;\ j=1,\ldots,m}$
- **Tensor**: generalization to k indices (rank k)
  - Scalar: rank 0; Vector: rank 1; Matrix: rank 2

Addition and scalar multiplication

Addition:

- Vectors: $(\vec{v} + \vec{w})_i = v_i + w_i$
- Matrices: $(A + B)_{ij} = a_{ij} + b_{ij}$

Scalar multiplication:

- $(\alpha\vec{v})_i = \alpha v_i$
- $(\alpha A)_{ij} = \alpha a_{ij}$

Dot product and matrix multiplication

Dot product (inner product):

$$\vec{v} \cdot \vec{w} = \sum_{i=1}^{n} v_i w_i$$

Matrix multiplication: $C = AB$, where

$$c_{ij} = \sum_{k=1}^{n} a_{ik}\, b_{kj}$$

Columns of A must equal rows of B.
Vectors are column matrices unless stated otherwise.

Transpose, determinant, inverse

Transpose: $(A^T)_{ij} = a_{ji}$

Determinant: $\det(A)$ — measure of signed volume.

$$\begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc$$

$\det(A) \neq 0$ iff A is invertible. $\det(AB) = \det(A)\det(B)$.

Inverse: $AA^{-1} = A^{-1}A = I_n$

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} = \frac{1}{ad - bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$$

Inverse — general formula

$$A^{-1} = \frac{1}{\det(A)} \operatorname{adj}(A)$$

where $\operatorname{adj}(A)$ is the transpose of the cofactor matrix.

Cofactor of entry a_{ij}: determinant of A with row i and column j removed, times $(-1)^{i+j}$.

Systems of linear equations

$$A\vec{x} = \vec{b}$$

- A: matrix of constants
- $\vec{x}$: vector of unknowns
- $\vec{b}$: vector of constants

Unique solution iff A is invertible: $\vec{x} = A^{-1}\vec{b}$.

Eigenvalues and eigenvectors

An eigenvalue λ of A satisfies:

$$A\vec{v} = \lambda\vec{v}$$

for some non-zero eigenvector $\vec{v}$.

Eigenvalues are roots of the **characteristic polynomial**:

$$\det(A - \lambda I_n) = 0$$

Probability

Kolmogorov axioms

1. $\mathrm{P}(A) \geq 0$ for any event A
2. $\mathrm{P}(\Omega) = 1$ where Ω is the sample space
3. If $A \cap B = \emptyset$: $\mathrm{P}(A \cup B) = \mathrm{P}(A) + \mathrm{P}(B)$

Sum rule (non-disjoint):

$$\mathrm{P}(A \cup B) = \mathrm{P}(A) + \mathrm{P}(B) - \mathrm{P}(A \cap B)$$

Joint, conditional, independence

Joint probability: $\mathrm{P}(A, B) = \mathrm{P}(A \cap B)$

Law of total probability: if $B_1, \ldots, B_n$ partition Ω:

$$\mathrm{P}(A) = \sum_{i=1}^{n} \mathrm{P}(A, B_i)$$

Conditional probability: $\mathrm{P}(A \mid B)$

Independence: $\mathrm{P}(A \mid B) = \mathrm{P}(A)$, equivalently $\mathrm{P}(A, B) = \mathrm{P}(A) \cdot \mathrm{P}(B)$

Bayes' rule

$$\mathrm{P}(A \mid B) = \frac{\mathrm{P}(B \mid A) \cdot \mathrm{P}(A)}{\mathrm{P}(B)}$$

One of the most important formulas in probability theory.
Foundation of Bayesian statistics and machine learning.

Random variables

A random variable $X : \Omega \to E$ maps outcomes to values.

$$\mathrm{P}(X \in A) = \mathrm{P}(\{\omega \in \Omega : X(\omega) \in A\})$$

- $E = \mathbb{R}$: continuous random variable
- $E = \mathbb{Z}$: discrete random variable
- $X \sim P$: X follows distribution P

Probability mass function (discrete):

$$p_X(x) = \mathrm{P}(X = x)$$

Probability density function (continuous):

$$\mathrm{P}(a \leq X \leq b) = \int_a^b f_X(x)\,dx$$

Cumulative distribution function:

$$F_X(x) = \mathrm{P}(X \leq x)$$

Expectation

$$\mathrm{E}[X] = \sum_x x \cdot p_X(x) \qquad \text{or} \qquad \mathrm{E}[X] = \int_{-\infty}^{\infty} x \cdot f_X(x)\, dx$$

Properties (linearity):

$$\begin{aligned} \mathrm{E}[cX] &= c\ \mathrm{E}[X] \\ \mathrm{E}[X + c] &= \mathrm{E}[X] + c \\ \mathrm{E}[X + Y] &= \mathrm{E}[X] + \mathrm{E}[Y] \end{aligned}$$

General: $\mathrm{E}[g(X)] = \sum_x g(x) \cdot p_X(x)$ or $\int g(x) \cdot f_X(x)\, dx$.

Variance

$$\operatorname{Var}(X) = \operatorname{E}\big[(X - \operatorname{E}[X])^2\big]$$

Equivalent form:

$$\operatorname{Var}(X) = \operatorname{E}[X^2] - \operatorname{E}[X]^2$$

Proof:

$$\begin{aligned}
\operatorname{Var}(X) &= \operatorname{E}\big[X^2 - 2X\operatorname{E}[X] + \operatorname{E}[X]^2\big] \\
&= \operatorname{E}[X^2] - 2\operatorname{E}[X]\operatorname{E}[X] + \operatorname{E}[X]^2 \\
&= \operatorname{E}[X^2] - \operatorname{E}[X]^2
\end{aligned}$$

Sample statistics

Sample mean: $\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$

Law of large numbers: $\lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} X_i = \mathrm{E}[X]$ (for i.i.d. $X_i \sim X$)

Sample variance: $S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2$

Denominator $n-1$ corrects for bias (Bessel's correction).

Higher moments

k-th moment: $\mathrm{E}[X^k]$

Sample skewness (3rd moment):

$$\text{Skewness} = \frac{\frac{1}{n}\sum_{i=1}^{n}(X_i - \bar{X})^3}{S^3}$$

Zero for symmetric; positive = right-skewed; negative = left-skewed.

Sample kurtosis (4th moment):

$$\text{Kurtosis} = \frac{\frac{1}{n}\sum_{i=1}^{n}(X_i - \bar{X})^4}{S^4} - 3$$

Positive = heavier tails than normal; negative = lighter tails.

Bernoulli distribution

$X \sim \text{Bern}(p)$, two outcomes (success/failure).

- $\text{E}[X] = p$
- $\text{Var}(X) = p(1-p)$

Poisson distribution

$X \sim \text{Poisson}(\lambda)$, number of events in a fixed interval.

PMF:

$$p_X(x) = \frac{e^{-\lambda}\,\lambda^x}{x!}$$

- $\text{E}[X] = \lambda$
- $\text{Var}(X) = \lambda$

Normal distribution

$X \sim \mathcal{N}(\mu, \sigma^2)$, bell-shaped density.

PDF:

$$f_X(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

- $\mathrm{E}[X] = \mu, \quad \mathrm{Var}(X) = \sigma^2$
- Standard normal: $\mathcal{N}(0, 1)$

Central limit theorem

Given $X_1, \dots, X_n$ i.i.d. with mean μ and finite variance σ^2:

$$\sqrt{n}\,(\bar{X} - \mu) \;\sim\; \mathcal{N}(0, \sigma^2) \quad \text{as } n \to \infty$$

The sample mean is approximately normal for large n, with mean μ and variance σ^2/n.

One of the most important results in probability and statistics.

T distribution

$X \sim \mathcal{T}(\nu)$, bell-shaped, heavier tails than normal.

- $\nu > 0$: degrees of freedom
- Location-scale generalization: $\mu + \sigma X \sim \mathcal{T}(\mu, \sigma^2, \nu)$
- Converges to normal: $\lim_{\nu \to \infty} \mathcal{T}(\nu) = \mathcal{N}(0, 1)$

Gamma distribution

$X \sim \text{Gamma}(\alpha, \beta)$, right-skewed density.

PDF:

$$f_X(x) = \frac{\beta^\alpha \, x^{\alpha-1} \, e^{-\beta x}}{\Gamma(\alpha)}$$

where $\Gamma(\alpha) = \displaystyle\int_0^\infty t^{\alpha-1} \, e^{-t} \, dt$.

Commonly used as a **conjugate prior** in Bayesian analysis.

Permutations and combinations

Factorial: $n! = n \cdot (n-1) \cdots 2 \cdot 1, \quad 0! = 1$

Permutation: number of arrangements of n elements $= n!$

Combination: number of ways to choose k from n:

$$\binom{n}{k} = \frac{n!}{k!\,(n-k)!}$$

Questions?