Learning from Data

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.

To understand God's thoughts we must study statistics, for these are the measures of His purpose.

— Florence Nightingale, her diary

Overview

Contents

- Introduction
- The learning problem
- Optimal solutions
- ERM inductive principle
- SRM inductive principle
- Linear problems

Objectives

- Define the learning problem and the common predictive tasks
- Understand the main principles that guide the learning process

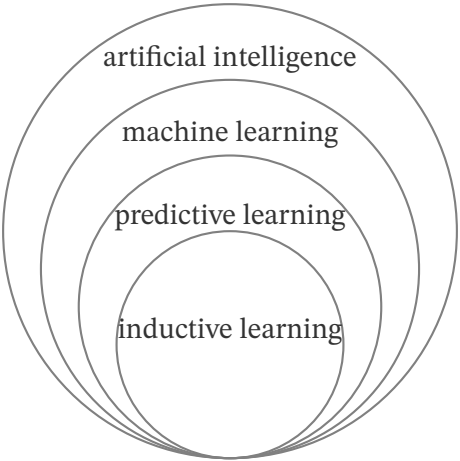
Introduction

From AI to inductive learning

- **Artificial intelligence** — algorithms that exhibit intelligent behavior
- **Machine learning** — algorithms that learn from experience/data
- **Predictive learning** — making predictions about outcomes
- **Inductive learning** — deriving general rules from specific observations

The inferred general rule can make predictions about *any* new instance.

Learning field hierarchy



Statistical Learning Theory

- General framework for predictive learning[1]
- Formalizes the learning problem
- Provides bounds on generalization ability
- Guides the design of learning machines

[1] V. N. Vapnik (1999). ***The nature of statistical learning theory***. 2nd ed. Springer-Verlag New York, Inc. ISBN: 978-1-4419-3160-3.

The learning problem

Setup

Training set:

$$\{(\vec{x}_i, y_i) : i = 1, \dots, n\}$$

- $\vec{x}_i \in \mathcal{X}$ — feature vector
- $y_i \in \mathcal{Y}$ — target variable
- Samples are i.i.d. drawn from $\mathrm{P}(x, y) = \mathrm{P}(y \mid x)\ \mathrm{P}(x)$
- Both $\mathrm{P}(x)$ and $\mathrm{P}(y \mid x)$ are **fixed but unknown**

Learning machine and risk

A **learning machine** generates models $f_\theta : \mathcal{X} \to \mathcal{Y}$ for $\theta \in \Theta$.

Given a **loss** $\mathcal{L}(y, f_\theta(x))$, the **risk** is

$$R(\theta) = \int \mathcal{L}(y, f_\theta(x)) \, d\mathrm{P}(x, y)$$

Goal: find f_θ that minimizes $R(\theta)$ using only the training set.

Binary data classification

- Output $y \in \{0, 1\}$ (negative and positive class)
- Loss: indicator function

$$\mathcal{L}(y, f_\theta(x)) = \begin{cases} 0 & \text{if } y = f_\theta(x) \\ 1 & \text{if } y \neq f_\theta(x) \end{cases}$$

- Risk $=$ probability of classification error
- f_θ is called a **classifier**

Regression estimation

- Output $y \in \mathbb{R}$
- Loss: squared error

$$\mathcal{L}(y, f_\theta(x)) = \left(y - f_\theta(x)\right)^2$$

- f_θ is called a **regressor**

A few remarks

Supervised vs. semisupervised:

- Supervised: target known for all training samples
- Semisupervised: only a small subset labeled

Generative vs. discriminative:

- Generative: models joint $\mathrm{P}(x, y)$ (more complex, more information)
- Discriminative: models $\mathrm{P}(y \mid x)$ directly (prefer if only predicting)

More remarks

Multiclass classification:

- y takes more than two values
- One-versus-all: l binary classifiers
- One-versus-one: $l(l-1)/2$ binary classifiers

Number of inputs and outputs:

- Input/output can be scalar, vector, matrix, or tensor
- The learning machine must handle the structure of the data

Optimal solutions

Why study optimal solutions?

- Optimal solutions assume $\mathrm{P}(y \mid x)$ is known
- Unrealistic but useful to understand how good a solution can be
- They depend only on $\mathrm{P}(y \mid x)$ (discriminative)
- Establish the **irreducible error** for each task

Bayes classifier

Optimal solution for binary classification:

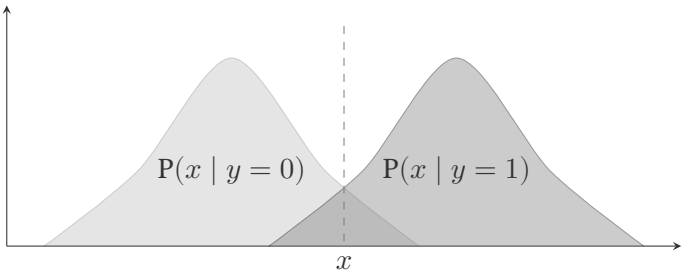
$$f_{\text{Bayes}}(x) = \arg\max_{y \in \mathcal{Y}} \mathrm{P}(y \mid x)$$

Bayes error rate (irreducible error):

$$R_{\text{Bayes}} = \int \min\{b(x),\ 1 - b(x)\}\, d\mathrm{P}(x)$$

where $b(x) = \mathrm{P}(y = 1 \mid x)$.

Bayes classifier — illustration



The dashed line is the Bayes decision boundary.
The darker intersection area causes the irreducible error.

Regression function

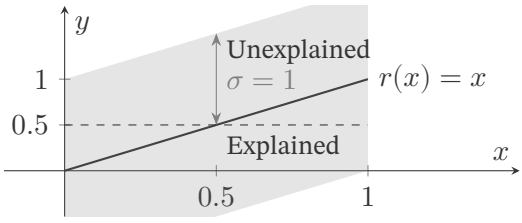
Optimal solution for regression estimation:

$$r(x) = \int y \, d\mathrm{P}(y \mid x) = \mathrm{E}[y \mid x]$$

Irreducible error (law of total variance):

$$R(r) = \mathrm{E}\big[\,\mathrm{Var}(y \mid x)\big] \quad \text{(unexplained variance)}$$

Explained vs. unexplained variance



$$\mathrm{P}(y \mid x) = \mathcal{N}(x, 1), \quad \mathrm{P}(x) = \mathcal{U}(0, 1)$$

ERM inductive principle

Empirical Risk Minimization

Since $\mathrm{P}(z)$ is unknown, replace the risk by the **empirical risk**:

$$R_n(\theta) = \frac{1}{n} \sum_{i=1}^{n} L(z_i, \theta)$$

- $z_i = (x_i, y_i)$ — training samples
- Traditional methods (least squares, maximum likelihood) are realizations of ERM
- Question: does $R_n(\theta) \to R(\theta)$?

Consistency of the learning process

ERM is consistent if the empirical risk converges **uniformly**:

$$\lim_{n\to\infty} \mathrm{P}\left(\sup_{\theta\in\Theta} \left|R_n(\theta) - R(\theta)\right| > \epsilon\right) = 0$$

Fast rate of convergence if, for $n > n_0$:

$$\mathrm{P}\big(R(\theta_n) - R(\theta) > \epsilon\big) < \exp\big(-c\, n\, \epsilon^2\big)$$

VC entropy

For bounded loss functions $|L(z, \theta)| < M$:

- $N(z_1, \dots, z_n; \Theta, \epsilon)$ — size of minimal ϵ-net
- **VC entropy:**

$$H(n; \Theta, \epsilon) = \mathrm{E}[\ln N(z_1, \dots, z_n; \Theta, \epsilon)]$$

Necessary and sufficient condition for uniform convergence:

$$\lim_{n \to \infty} \frac{H(n; \Theta, \epsilon)}{n} = 0$$

Growth function and VC dimension

Growth function (distribution-independent):

$$G(n;\Theta) = \ln \sup_{z_1,\dots,z_n} N(z_1,\dots,z_n;\Theta)$$

Vapnik & Chervonenkis (1968): $G(n;\Theta)$ is either

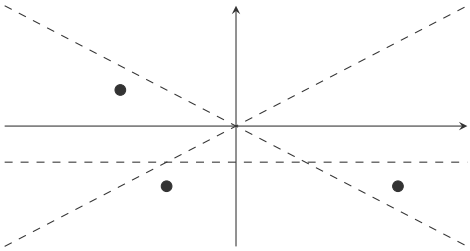
- $n \ln 2$ (VC dimension is **infinite**), or
- bounded by $h(\ln \frac{n}{h} + 1)$, where h is the **VC dimension**

Finite VC dimension $\Rightarrow$ consistency + fast convergence.

Shattering

- VC dimension $=$ max number of points that can be **shattered**
- h points are shattered if they can be separated into two classes in all 2^h possible ways
- VC dimension measures **complexity of the hypothesis space**, not the number of parameters

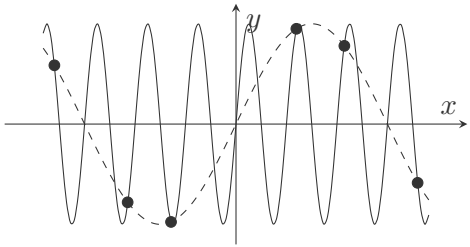
VC dimension — lines in the plane



VC dimension of lines in the plane $= 3$.

A line can shatter 3 points in all $2^3 = 8$ ways, but not 4 points.

Infinite VC dimension — sine functions



$f(z;\theta) = \mathbb{1}_{\sin \theta x > 0}$ has infinite VC dimension with only **one** parameter θ.

SRM inductive principle

Generalization bound

$$R(\theta_n) \leq R_n(\theta_n) + \frac{B\mathcal{E}}{2}\left(1 + \sqrt{1 + \frac{4R_n(\theta_n)}{B\mathcal{E}}}\right)$$

with

$$\mathcal{E} = 4\frac{h\left(\ln\frac{2n}{h} + 1\right) - \ln\frac{\eta}{4}}{n}$$

- B — upper bound of the loss, h — VC dimension
- Small $n/h \Rightarrow$ small empirical risk does **not** guarantee small true risk

Overfitting and underfitting

Problem	Empirical risk	Confidence interval
Underfitting	High	Low
Overfitting	Low	High

- **Underfitting**: model too simple (low VC dim.)
- **Overfitting**: model too complex (high VC dim.)
- Must balance both terms to generalize well

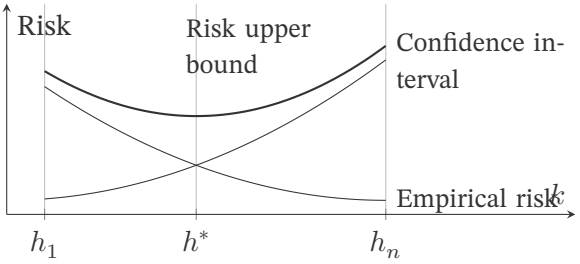
Structural Risk Minimization

- Build an **admissible structure**:

$$S_1 \subset S_2 \subset \cdots \subset S_n \subset \cdots \quad \text{with} \quad h_1 \leq h_2 \leq \cdots$$

- Choose the subset S_k that minimizes the **guaranteed risk** (empirical risk + confidence interval)

SRM trade-off



The smallest guaranteed risk is found at h^* in the admissible structure.

Bias-variance decomposition

For regression with noise ϵ ($\mathrm{E}[\epsilon] = 0$, $\mathrm{Var}(\epsilon) = \sigma^2$):

$$\mathrm{E}_D\left[\left(y - \hat{f}(x; D)\right)^2\right] = \underbrace{\sigma^2}_{\text{irreducible}} + \underbrace{\left(f(x) - \mathrm{E}[\hat{f}]\right)^2}_{\text{bias}^2} + \underbrace{\mathrm{Var}_D(\hat{f}(x; D))}_{\text{variance}}$$

- **Bias**: failure to capture relevant relationships
- **Variance**: modeling the noise in training data
- This is the general case of the SRM trade-off

Regularization

Modify the loss to penalize model complexity:

$$R_n(\theta) + \lambda\, \Omega(\theta)$$

- $\Omega(\theta)$ — complexity penalty (e.g., $\|\theta\|^2$)
- λ — hyperparameter controlling the trade-off
- Acts as a proxy for the confidence interval in SRM
- Implicit regularization: early stopping, dropout, ensembles, pruning

Linear problems

Linear classification

- Realize SRM concepts in practice
- Two learning machines:
  1. **Perceptron** — fix complexity, minimize empirical risk
  2. **Maximal margin classifier** — fix empirical risk (zero), minimize confidence interval

AND and XOR datasets

x_1	x_2	$y = x_1 \wedge x_2$
0	0	0
0	1	0
1	0	0
1	1	1

x_1	x_2	$y = x_1 \oplus x_2$
0	0	0
0	1	1
1	0	1
1	1	0

AND is linearly separable. XOR is not.

Perceptron

Linear classifier with model:

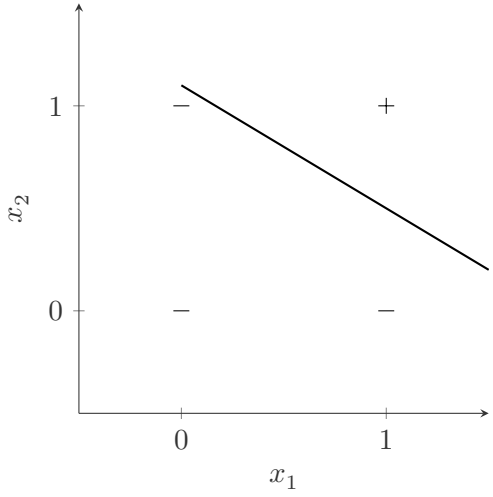
$$f(x_1, x_2; \vec{w}) = u(w_0 + w_1 x_1 + w_2 x_2)$$

where u is the Heaviside step function:

$$u(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases}$$

The equation $\vec{w} \cdot \vec{x} = 0$ defines a **hyperplane** ($\vec{x} = [1, x_1, x_2]$).

Perceptron — AND dataset

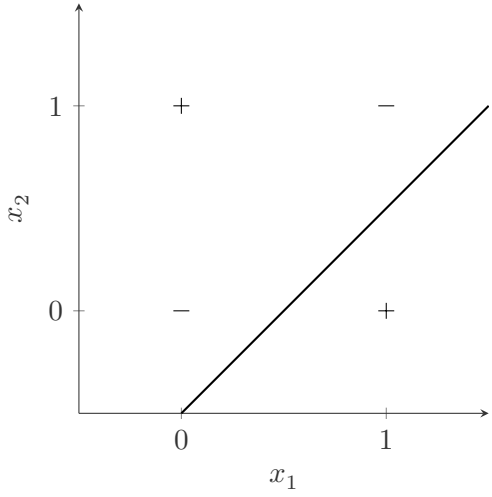


$\vec{w} = [-1.1,\ 0.6,\ 1]$ — correctly classifies all samples.

Perceptron AND — truth table

x_1	x_2	y	$-1.1 + 0.6\,x_1 + x_2$	$\hat{y}$
0	0	0	-1.1	0
0	1	0	-0.1	0
1	0	0	-0.5	0
1	1	1	0.5	1

Perceptron — XOR dataset



$\vec{w} = [-0.5,\ 1,\ -1]$ — no perceptron can solve XOR.

Perceptron XOR — truth table

x_1	x_2	y	$-0.5 + x_1 - x_2$	$\widehat{y}$
0	0	0	-0.5	0
0	1	1	-1.5	0
1	0	1	0.5	1
1	1	0	-0.5	0

The perceptron fails to classify $(0, 1)$ correctly.

Perceptron training

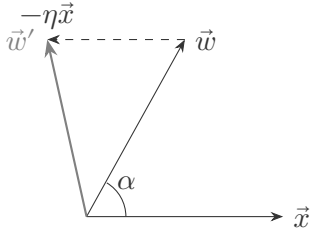
For a misclassified sample, update the weights:

$$\vec{w}' = \vec{w} + \eta\, e\, \vec{x}$$

where $e = y - u(\vec{w} \cdot \vec{x})$ and $\eta > 0$ (step size).

- Converges if η is small enough and data is linearly separable
- Makes no effort to reduce the confidence interval
- Simplest artificial neural network

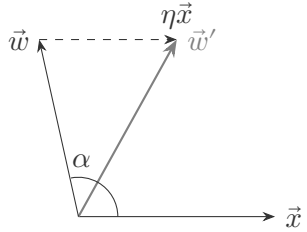
Weight update — positive output error



$\vec{w} \cdot \vec{x} > 0 \implies \alpha < 90°$.

Subtract $\eta\vec{x}$ to increase the angle.

Weight update — negative output error



$\vec{w} \cdot \vec{x} < 0 \implies \alpha > 90°$.

Add $\eta\vec{x}$ to decrease the angle.

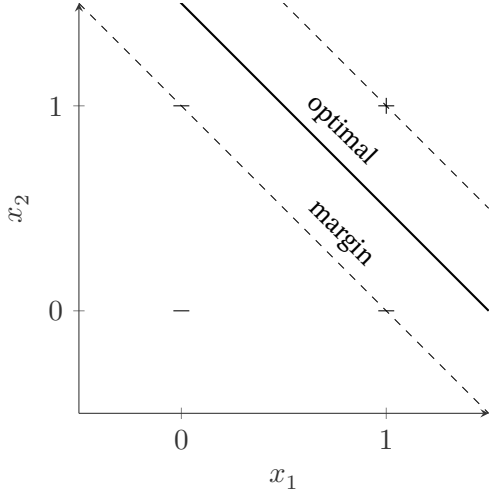
Maximal margin classifier

- Fix empirical risk to **zero** (assume linearly separable)
- Minimize the **confidence interval** by maximizing the margin
- Δ-margin hyperplane: $(\vec{w} \cdot \vec{x}) - b = 0$, $\|\vec{w}\| = 1$
- VC dimension:

$$h \leq \min\left(\left\lfloor \frac{R^2}{\Delta^2} \right\rfloor, d\right) + 1$$

- Larger margin $\Delta \Rightarrow$ lower VC dimension $\Rightarrow$ better generalization

Maximal margin — AND dataset



Support vectors: $(1, 0)$, $(0, 1)$, and $(1, 1)$.

Maximal margin classifier — properties

The classifier is built from **support vectors** only:

$$f(x) = \text{sign}\left(\sum_{i=1}^{n} y_i \, a_i \, (\vec{x}_i \cdot x) - b\right)$$

where $a_i > 0$ for support vectors, $a_i = 0$ otherwise.

- Number of parameters depends on data $\Rightarrow$ **nonparametric**
- Soft margin: allows $R_{\text{emp}} > 0$ for non-separable data
- Kernel trick: handle nonlinear problems

Takeaways

- Optimal solutions establish how good a solution can possibly be
- Reducing error is not enough to guarantee a good solution
- Controlling model complexity is crucial for generalization
- The perceptron minimizes empirical risk (fixed complexity)
- The maximal margin classifier minimizes complexity (fixed risk)

Questions?