Theory of Machine Learning

Prof. Damien Garreau

Julius-Maximilians-Universität Würzburg

Winter term 2024–2025

1. Course organization

Organization of the course

- **Wuestudy Course ID:** 08134700
- **Name on Wuecampus:** Theory of Machine Learning
- **Who?**
  - **Lectures:** myself
  - **Exercises:** M. Taimeskhanov
- **Format** = slides (available on Moodle after each lecture)
- **Exercises** = mostly pen and paper, regular coding (in Python)
- **Schedule:**
  1. lectures on Fridays, 4-5:30pm
  2. exercise sessions on Fridays, 2-3:30pm *(starting next week)*
- **Room:** SE 2, CAIDAS building

Evaluation

- do not forget to register to the exam
- **Evaluation:**
  - written exam at the end of the semester
  - content = definitions, similar derivations to the exercises, more ambitious problem
  - exercises sessions $\rightarrow$ bonus points
- **How does the bonus work?**
  - attend the sessions
  - send your work to Magamed at the end of the session
  - global grade $\rightarrow$ up to 10% bonus
- **Examples:** (based on 10 sessions)
  - exam = 76%, I attended all exercise sessions and made a good effort for each: I get full bonus and my final grade is $76 + 10 = 86\%$
  - exam = 96%, I attended all exercise sessions and made a good effort for each: I get full bonus and my final grade is $96 + 10 = 100\%$
  - exam = 76%, I skipped two sessions and during one session I was not paying attention and handed out something subpar: bonus = 7.5%, final grade = 83.5%

Goals and pre-requisites

- **Pre-requisites:**
  - linear algebra (matrix, eigenvectors, diagonalization)
  - analysis (derivative, gradient, global maximum)
  - probability theory (random variable, density, expectation)
  - I am glad to interrupt the lecture if some maths notion is not clear
- **Goals of the lecture:**
  - know about the **basic vocabulary**
  - look into the **details of the fundamental machine learning algorithms** (linear regression, gradient descent, etc.)
  - prove **key easy theoretical results** (*e.d.*, convergence rate for least squares)
  - **check experimentally** that these results hold

Outline I

Outline II

Useful resources

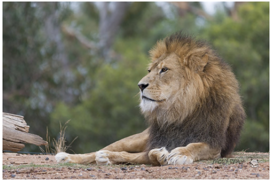
- **Main references:**
  - *for general learning theory:* Francis Bach, *Learning Theory from First Principles*, 2023
  - *for methodology:* Hastie, Tibshirani, Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer Series in Statistics, 2001 (second edition: 2009)
  - *for kernel methods specifically:* Bernhard Schölkopf, Alexander Smola, *Learning with kernels*, MIT Press, 2002
- **Wikipedia:** as good as ever.
- **Wolfram alpha:** if you have computations to make and you do not know want to use a proper language: `https://www.wolframalpha.com/`
- **Remedials:**
  - *linear algebra:* Gilbert Strang, *Introduction to Linear Algebra*, Cambridge Press, 2009
  - *probability theory:* William Feller, *An introduction to probability theory and its applications*, Wiley, 1950

2. Introduction

2.1. First concepts

Fundamental example

- **Fundamental example:** image classification
- input = image x
- **Goal:** given any input, we want to predict which object / animal is in the image
- output = label y



$\longmapsto$ "lion"

- **Successful philosophy:** instead of defining the function f ourselves, we are going to *learn it* from data

Supervised learning

Definition: we call *predictor* (or *model*) any mapping between inputs and outputs.

- supervised learning $\rightarrow$ we will find a good predictor using *annotated* examples
- **Remark (i):** why is it difficult?
  - output may not be a deterministic function of input
  - link between the two may be incredibly complex
  - only a few observations available, potentially not where we want them
  - high dimensionality
  - ...
- **Remark (ii):** large part of machine learning: *unsupervised learning* (no annotations)
- **Examples:** clustering, dimension reduction, etc.
- out of the scope of this lecture

Input space

Definition: we call *input space* (or *domain*, or *domain set*) the set of all possible inputs of our machine learning model. We will denote it by $\mathcal{X}$.

- **Example (i):** tabular data = spreadsheet data; x has well-defined *features* such as age, `income`, `has_a_car`
- **Example (ii):** text data = ordered sequence of tokens; generally have to be pre-processed to be understood by our computer
- **Example (iii):** images = $H \times W \times C$ arrays of numbers

$$\in [\![0, 255]\!]^{299 \times 299 \times 3}$$

Input space as vector space

- **Remark:** elements $x \in \mathcal{X}$ are usually described as *vectors*
- **Reminder:** vectors are 1D arrays of number, here are two vectors with three *coordinates*:

$$u = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix}, \qquad v = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix}.$$

- they can be
  - *added:* $(u + v)_i = u_i + v_i$
  - *multiplied by a number:* $(\lambda u)_i = \lambda u_i$
- vectors belong to a **vector space**, its dimension is the number of coordinates
- $\dim = d \Rightarrow$ canonical identification with $\mathbb{R}^d$
- **Intuition:** d copies of $\mathbb{R}$ with a special structure
- **Remark:** d typically high in modern machine learning
- **Example:** ImageNet images $\rightarrow 299 \times 299 \times 3 = 268,203$

Classification and regression

- we will consider two fundamental tasks: **classification** and **regression**
  - in classification, we want to associate to each $x \in \mathcal{X}$ a given *class*
  - in regression, we want to associate to each $x \in \mathcal{X}$ a given *value*
- **Example (i):** for each image on my hard drive, I want to predict what appears in it

```
{0: 'tench, Tinca tinca',
 1: 'goldfish, Carassius auratus',
 2: 'great white shark, white shark, man-eater, man-eating shark, Carcharodon carcharias',
 3: 'tiger shark, Galeocerdo cuvieri',
 4: 'hammerhead, hammerhead shark',
 5: 'electric ray, crampfish, numbfish, torpedo',
 6: 'stingray',
 7: 'cock',
 8: 'hen',
 9: 'ostrich, Struthio camelus',
```

- **Example (ii):** for each customer in my database, I want to predict how many euros he will spend next year

Labels / responses

Definition: we call *target space* (or *output space*) the set of all possible outputs of our machine learning model. We will denote it by $\mathcal{Y}$.

- **Example (i):** in image classification, $\mathcal{Y}$ is the set of all names of object and animals of the dataset
- we identify it with $\{1, 2, \ldots, 1000\} = [1000]$
- **Remark (i):** no notion of order (3 is not better than 2)
- **Remark (ii):** we will often restrict ourselves to $\mathcal{Y} = \{0, 1\}$ or $\{-1, +1\}$ for simplicity
- **Example (ii):** in regression, $\mathcal{Y} = \mathbb{R}$ (or $\mathbb{R}^k$ if we want to predict several targets simultaneously)

Training data

Definition: we call *training data* (or *training set*) a *finite* sequence of elements of $\mathcal{X} \times \mathcal{Y}$, denoted as

$$S = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\} \, .$$

Here, n is the size of the training set.

- **Example (i):** S is a collection of 10^6 images, each associated to the correct label
- **Example (ii):** S is a spreadsheet with the customer data from the last 25 years
- **Remark:** in real-life, there are many complications:
  - labels may be *corrupt*
  - some data (= feature value for some observations) may be *missing*
- we do not consider these complications in this lecture

Machine learning algorithm

- we can now be a bit more precise:

Definition: we call *machine learning algorithm* a mapping A transforming a training set $S \in (\mathcal{X} \times \mathcal{Y})^n$ into a predictor $f : \mathcal{X} \to \mathcal{Y}$. Thus $f = A(S)$.

- of course, we want to devise a "good" algorithm
- **Question:** what does good even mean?
- **Definition that machine learning uses:** performance on new, unseen data
- there are two difficulties here: we need to define
  1. performance
  2. new, unseen data

Loss functions

Definition: we call loss function any mapping $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$.

- **Intuitively:** $\ell(y, y')$ measures the cost of predicting y' whereas the true target is y
- generally, we require that:
  - ℓ is symmetric;
  - ℓ has non-negative (≥ 0) values
  - $\ell(y, y) = 0$.
- **Example (i):** classification $\to$ $0 - 1$ loss

$$\ell(y, y') = \mathbb{1}_{y \neq y'} \, .$$

- here, $\mathbb{1}_E = 1$ if E is true, 0 otherwise
- **Remark:** does not matter how many classes

Loss functions

- **Example (ii):** regression $\to \mathcal{Y} \subseteq \mathbb{R} \to$ square loss

$$\ell(y, y') = (y - y')^2 \,.$$

- other possibility: absolute loss

$$\ell(y, y') = |y - y'| \,.$$

- **Other examples:** structured prediction,[1] functional regression, etc.
- **Remark (i):** in addition to the properties already lister, regression loss tend to tend to ∞ when the prediction errs far away from the ground truth
- **Remark (ii):** loss function also tend to be convex, but there are exceptions

[1] Osokin, Bach, Lacoste-Julien, *On structured prediction theory with calibrated convex surrogate losses*, NeurIPS, 2017

Expected risk: informal definition

- we model new, unseen data by a random variable $(X, Y) \in \mathcal{X} \times \mathcal{Y}$ with distribution p
- **Intuition:** new annotated data coming from the same distribution as the training data
- **Informal definition:** expected risk is the expected loss on new data
- **Reminder:** expectation $=$ average value of a random variable
- in the discrete case, $X \in \{x_1, \ldots, x_p\}$,

$$\mathbb{E}[X] = \sum_{i=1}^{p} x_i \cdot \mathbb{P}(X = x_i) \, .$$

- **Intuition:** sum of outcome values weighted by how often they occur

Expected risk

- let us give a formal definition:

Definition: for a given data distribution p and loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, we define the *expected risk* (or *test error*) of a predictor $f : \mathcal{X} \to \mathcal{Y}$ as

$$\mathcal{R}(f) := \mathbb{E}\left[\ell(Y, f(X))\right] .$$

- **Remark (i):** depends on both the loss function and the data distribution p
- **Remark (ii):** hidden assumption: data distribution is equal to p...
- unfortunately, we do not know the data distribution...
- expected risk is the key quantity: ideally, we want to find f such that it is minimal

Special cases

- general definition, often specified in two key examples:
- **Binary classification:** $\mathcal{Y} = \{0, 1\}$ and $\ell(y, y') = \mathbb{1}_{y \neq y'}$, risk can be rewritten as

$$\begin{aligned}\mathcal{R}(f) = \mathbb{E}\left[\mathbb{1}_{Y \neq f(X)}\right] &= 0 \cdot \mathbb{P}\left(Y = f(X)\right) + 1 \cdot \mathbb{P}\left(f(X) \neq Y\right) \\ &= \mathbb{P}\left(f(X) \neq Y\right) .\end{aligned}$$

- **Remark:** probability of disagreement $= 1-$ accuracy
- **Regression:** $\mathcal{Y} = \mathbb{R}$ and $\ell(y, y') = (y - y')^2$

$$\mathcal{R}(f) = \mathbb{E}\left[(Y - f(X))^2\right]$$

- also known as **mean squared error** $(=$ MSE$)$
- in any case, *lower is better*

Expected risk

- **Example (i):** in the classification setting, consider the following predictor:

$$\forall x \in \mathcal{X}, \qquad f(x) = 1\,.$$

- let us assume balanced data, that is, $\mathbb{P}\left(Y = 0\right) = \mathbb{P}\left(Y = 1\right) = 1/2$
- then the expected risk of f is

$$\mathcal{R}(f) = \mathbb{P}\left(f(X) \neq Y\right) = \mathbb{P}\left(Y \neq 1\right) = \mathbb{P}\left(Y = 0\right) = 1/2\,.$$

- **Example (ii):** regression setting, assume that $Y = X + \varepsilon$, with $\varepsilon \sim \mathcal{N}\left(0, \sigma^2\right)$
- consider $f(x) = x$ (perfect predictor!)

$$\mathcal{R}(f) = \mathbb{E}\left[(Y - f(X))^2\right] = \mathbb{E}\left[(X + \varepsilon - X)^2\right] = \mathbb{E}\left[\varepsilon^2\right] = \sigma^2 > 0\,.$$

- **Reminder:** $\mathrm{Var}\left(\varepsilon\right) = \mathbb{E}\left[(\varepsilon - \mathbb{E}\left[\varepsilon\right])^2\right]$

Bayes risk

- **Question:** what is the *best* prediction function for our criterion (expected risk)?
- **Intuitively:** we want to find f that **minimizes** expected risk

Definition: we define the *Bayes risk* as the minimal possible risk over all possible predictors, for a given loss function and data distribution. Formally,

$$\mathcal{R}^\star := \inf_f \mathcal{R}(f) = \inf_f \mathbb{E}\left[\ell(Y, f(X))\right] .$$

- **Reminder:** $\inf_{x \in E} r(x)$ is the minimal value of $r(x)$ on the set E
- **Remark (i):** this is not necessarily $= 0$
- **Remark (ii):** $\mathcal{R}^\star$ is our true yardstick

Bayes predictors

- in some cases, one can actually give predictors achieving $\mathcal{R}^\star$

Definition: we call *Bayes predictor* any predictor with minimal risk and denote it by $f^\star$. Formally,

$$\mathcal{R}(f^\star) = \mathcal{R}^\star \left(= \inf_f \mathcal{R}(f) = \inf_f \mathbb{E}\left[\ell(Y, f(X))\right] \right) .$$

- **Question:** how do we do that?
- first step $=$ using the **tower property**: let g be a predictor,

$$\mathcal{R}(g) = \mathbb{E}_{x\sim p}[\mathbb{E}\left[\ell(Y, g(x)) \mid X = x\right]]$$

Reminder: conditional probability

Proposition: given two events A and B such that $\mathbb{P}(B) \neq 0$, we define the *conditional probability* of A "given" B by

$$\mathbb{P}(A \,|\, B) := \frac{\mathbb{P}(A \text{ and } B)}{\mathbb{P}(B)}\,.$$

- **Example:** let us consider two Bernoulli with parameter $1/2$, A_1 and A_2
- we can compute

$$\mathbb{P}(A_1 + A_2 = 1 \,|\, A_1 = 0) = \frac{\mathbb{P}(A_1 + A_2 = 1 \text{ and } A_1 = 0)}{\mathbb{P}(A_1 = 0)} = \frac{\mathbb{P}(A_1 = 0 \text{ and } A_2 = 1)}{\mathbb{P}(A_1 = 0)}$$

$$= \frac{1/4}{1/2} = \frac{1}{2}\,.$$

Reminder: conditional expectation

Proposition: let X and Y be discrete rndom variables. The *conditional expectation* of X given Y is given by

$$\mathbb{E}\left[X \mid Y = y\right] = \sum_x x \cdot \mathbb{P}\left(X = x \,|\, Y = y\right) .$$

- **Remark:** undefined if $\mathbb{P}\left(Y = y\right) = 0$ (but still possible for continuous random variables)
- **Example:**

$$\begin{aligned}
\mathbb{E}\left[A_1 + A_2 \mid A_1 = 0\right] &= 0 \cdot \mathbb{P}\left(A_1 + A_2 = 0 \,|\, A_1 = 0\right) + 1 \cdot \mathbb{P}\left(A_1 + A_2 = 1 \,|\, A_1 = 0\right) \\
&\quad + 2 \cdot \mathbb{P}\left(A_1 + A_2 = 2 \,|\, A_1 = 0\right) \\
&= 0 \cdot \frac{1}{2} + 1 \cdot \frac{1}{2} + 2 \cdot 0 = \frac{1}{2} .
\end{aligned}$$

Reminder: tower property

Proposition: Let X and Y be two random variables. Then $\mathbb{E}_Y[\mathbb{E}[X \mid Y]] = \mathbb{E}[X]$.

- **Proof (in the discrete case):** using the previous slide:

$$\begin{aligned}
\mathbb{E}_Y[\mathbb{E}[X \mid Y]] &= \sum_y \left(\sum_x x \cdot \mathbb{P}(X = x \,|\, Y = y) \right) \mathbb{P}(Y = y) \\
&= \sum_x x \cdot \sum_y \mathbb{P}(X = x \,|\, Y = y) \, \mathbb{P}(Y = y) \\
&= \sum_x x \cdot \sum_y \mathbb{P}(X = x, Y = y) \\
&= \sum_x x \cdot \mathbb{P}(X = x) \\
\mathbb{E}_Y[\mathbb{E}[X \mid Y]] &= \mathbb{E}[X] \quad \square
\end{aligned}$$

Back to Bayes predictors

- according to the tower property:

$$\mathcal{R}(g) = \mathbb{E}_{x\sim p}[\mathbb{E}\left[\ell(Y, g(x)) \mid X = x\right]]$$

- **Remark:** $\mathbb{E}\left[\ell(Y, g(x)) \mid X = x\right]$ is also sometimes called the *conditional risk*
- we can *define* $f^\star$ such that, for all $x \in \mathcal{X}$, it minimizes

$$C(g, x) := \mathbb{E}\left[\ell(Y, g(x)) \mid X = x\right] .$$

- by positivity of the integral, this gives us the best possible risk

Bayes predictors

- summarizing everything:

Proposition: The expected risk is minimized at a *Bayes predictor* $f^\star : \mathcal{X} \to \mathcal{Y}$ satisfying for all $x \in \mathcal{X}$

$$f^\star(x) \in \arg\min_{z \in \mathcal{Y}} \mathbb{E}\left[\ell(Y, z) \mid X = x\right] .$$

All Bayes predictor have the same risk, equal to the Bayes risk. It can be computed as

$$\mathcal{R}^\star = \mathbb{E}_{x \sim p}\left[\inf_{z \in \mathcal{Y}} \mathbb{E}\left[\ell(Y, z) \mid X = x\right]\right] .$$

- **Remark:** $f^\star$ seems complicated to compute... and it is
- we can still get some interesting statements

Examples

- **Binary classification:** for the $0-1$ loss, Bayes predictor can be written

$$f^\star(x) \in \operatorname*{arg\,min}_{z\in\{0,1\}} \mathbb{P}\left(Y \neq z \,|\, X = x\right) = \operatorname*{arg\,max}_{z\in\{0,1\}} \mathbb{P}\left(Y = z \,|\, X = x\right) .$$

- set $\eta(x) = \mathbb{P}\left(Y = 1 \,|\, X = x\right)$, then $f^\star(x) = \mathbb{1}_{\eta(x)>1/2}$
- Bayes risk is equal to

$$\mathcal{R}^\star = \mathbb{E}\left[\min(\eta(x), 1-\eta(x))\right] .$$

- **Regression:** for the square loss, Bayes predictor is such that

$$f^\star(x) \in \operatorname*{arg\,min}_{z\in\mathbb{R}} \mathbb{E}\left[(Y-z)^2 \mid X = x\right] = \mathbb{E}\left[Y \mid X = x\right]$$