Introduction to Supervised Learning

Clément Lalanne

Objectives of this lecture

- Model a supervised learning problem mathematically
- Define the metrics used to evaluate performance
- Present the classical families of algorithms that solve it
- Introduce **why assumptions on the data are unavoidable**
- Survey the other classical paradigms of statistical learning

I. Data and predictions

Observations

We are given n pairs, the **training data**:

$$(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}, \quad i = 1, \dots, n$$

Inputs $x_i \in \mathcal{X}$

- images
- sounds
- videos
- texts
- vectors in $\mathbb{R}^d$
- ...

Outputs $y_i \in \mathcal{Y}$

- binary classification: $\{0, 1\}$ or $\{-1, 1\}$
- multiclass: $\{1, \dots, K\}$, or one-hot vectors $\{(1, 0, \dots, 0), \dots, (0, \dots, 0, 1)\}$
- regression: $\mathbb{R}^d$
- ...

The objective

Given a **new** input x, possibly never seen during training, predict the output y that best corresponds to it.

$$y \approx f(x)$$

where f is **learned from the training data**.

Four difficulties

y may be a random function of x. Two identical inputs can carry different labels. There is no deterministic f to recover.

f may be complex. Nothing guarantees the relationship is simple, linear, or smooth.

We observe only finitely many training points. We must find a compromise between *interpolating* the training data and *extrapolating* to new data. This is the **underfitting / overfitting** trade-off.

The dimension d may be large. Intuition from $\mathbb{R}^2$ is a poor guide.

II. Mathematical formalisation

The probabilistic model

The modelling is done through probability theory. We posit a probability distribution $\mathbb{P}$ on $\mathcal{X} \times \mathcal{Y}$:

$$\mathbb{P} \text{ on } \underbrace{\mathcal{X}}_{\text{inputs}} \times \underbrace{\mathcal{Y}}_{\text{outputs}}$$

Independence assumption. The observations, and any future test point, are drawn independently from the same distribution:

$$(x, y), (x_1, y_1), \ldots, (x_n, y_n) \overset{\text{i.i.d.}}{\sim} \mathbb{P}$$

We write $S := ((x_1, y_1), \ldots, (x_n, y_n)) \sim \mathbb{P}^{\otimes n}$ for the training sample seen as a single random object. We will need this whenever a quantity depends on the whole sample.

What we would ideally estimate

The complete answer to the prediction problem is the **conditional distribution**

$$\mathbb{P}_{y|x=x'}, \quad x' \in \mathcal{X}$$

the law of the consequences given the causes.

This is often more than we need, and harder than we can manage. The problem is therefore simplified: find $f : \mathcal{X} \to \mathcal{Y}$ such that

$$\mathbb{P}_{y|x=x'} \approx \delta_{f(x')} \quad \text{or, informally,} \quad y \approx f(x)$$

Keep both levels in mind: some methods model $\mathbb{P}_{y|x}$ explicitly, others go straight for f.

III. Evaluating performance

1) Loss function

Given $f : \mathcal{X} \to \mathcal{Y}$, how do we evaluate it? We first need a notion of pointwise error.

$$\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$$

$\ell(y, \hat{y})$ is the error induced by predicting $\hat{y}$ when the truth was y.

Binary and multiclass classification

$$\ell(y, \hat{y}) = \mathbf{1}\{y \neq \hat{y}\} \quad \text{(0–1 loss)}$$

Regression

$$\ell(y, \hat{y}) = (y - \hat{y})^2 \quad \text{or} \quad \ell(y, \hat{y}) = |y - \hat{y}|$$

2) Risk

With a loss in hand, we can measure the error of a predictor f **over the whole population**.

Average risk / test error / generalisation error

$$R(f) := \mathbb{E}_{(x,y)\sim\mathbb{P}}[\ell(y, f(x))] = \int_{\mathcal{X}\times\mathcal{Y}} \ell(y, f(x)) d\mathbb{P}(x, y)$$

Binary classification. With the 0–1 loss,

$$R(f) = 0 \times \mathbb{P}(f(x) = y) + 1 \times \mathbb{P}(f(x) \neq y) = \mathbb{P}(f(x) \neq y)$$

the probability, under $\mathbb{P}$, of misclassifying a fresh pair. Multiclass: identical.

Regression. $R(f) = \mathbb{E}_{(x,y)\sim\mathbb{P}}[(y - f(x))^2]$ or $R(f) = \mathbb{E}_{(x,y)\sim\mathbb{P}}[|y - f(x)|]$.

The obstruction

We would like to choose f minimizing $R(f)$. But $R(\cdot)$ is an expectation under $\mathbb{P}$, the **true distribution of the data, which is unknown**.

We will therefore need an *estimator* of the risk. The sample gives us one immediately.

Empirical risk / training error

$$\widehat{R}_S(f) := \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f(x_i))$$

An average of n i.i.d. random variables, exactly the object the classical limit theorems are about. We keep the subscript S only when the dependence on the sample matters, and write $\widehat{R}(f)$ otherwise.

Why the approximation $R(f) \approx \widehat{R}_S(f)$ is legitimate

Under integrability assumptions, and for a **fixed** f (chosen before seeing S):

Law of large numbers

$$\widehat{R}_S(f) \xrightarrow[n\to\infty]{\text{a.s.}} R(f)$$

Central limit theorem

$$\sqrt{n}(\widehat{R}_S(f) - R(f)) \xrightarrow[n\to\infty]{d} \mathcal{N}(0, \mathrm{Var}_{(x,y)\sim\mathbb{P}}(\ell(y, f(x))))$$

The first says the substitution is consistent. The second gives its **rate**: the error is of order $n^{-1/2}$, and, as we will see, it gives us error bars.

But optimizing breaks this simple concentration

Both theorems apply to a predictor f chosen **independently of the sample** S.

The predictor we will actually build, $\hat{f}_S$, is chosen *by minimizing* $\widehat{R}_S$ over that very sample. The variables $\ell(y_i, \hat{f}_S(x_i))$ are no longer independent of $\hat{f}_S$, and the estimator loses its guarantees. In fact it becomes **optimistically biased**.

Let $f^\dagger$ be a minimizer of R over the same class, fixed independently of S.

$$\begin{aligned}
\mathbb{E}_{S\sim\mathbb{P}^{\otimes n}}[\widehat{R}_S(\hat{f}_S)] &\le \mathbb{E}_{S\sim\mathbb{P}^{\otimes n}}[\widehat{R}_S(f^\dagger)] && \text{definition of } \hat{f}_S \\
&= R(f^\dagger) && f^\dagger \text{ independent of } S \\
&= \inf_f R(f) \\
&\le \mathbb{E}_{S\sim\mathbb{P}^{\otimes n}}[R(\hat{f}_S)] && \text{the infimum bounds every } R(\hat{f}_S)
\end{aligned}$$

The training error **systematically underestimates** the true risk of the model that produced it. It is not a matter of bad luck or small n.

Consequence: hold data out

Split the sample **before doing anything else**, into three parts with three distinct roles.

S_{train}, 70%

Used to fit $\hat{\theta}$ by empirical risk minimization. The predictor depends on these points, so any error measured on them is biased.

S_{val}, 15%

Used to choose everything the fitting procedure does not choose by itself: the model class, λ, k, when to stop. May be reused freely.

S_{test}, 15%

Used **once**, at the very end, to report $\widehat{R}_{S_{\text{test}}}(\hat{f})$. Never used to make a decision.

Why this works. Conditionally on S_{train} and S_{val}, the predictor $\hat{f}$ is fixed and independent of S_{test}. The law of large numbers and the central limit theorem apply again, so

$$\mathbb{E}_{S_{\text{test}} \sim \mathbb{P}^{\otimes m}}[\widehat{R}_{S_{\text{test}}}(\hat{f})] = R(\hat{f})$$

The bias of the previous slide has disappeared, but only for a predictor that was never exposed to S_{test}.

The test error is itself a random variable

Apply the CLT from three slides ago to a test sample S_{test} of size m. For the 0–1 loss, conditionally on $\hat{f}$,

$$\ell(y_j, \hat{f}(x_j)) \sim \mathcal{B}(R(\hat{f})) \quad \text{i.i.d.},$$
$$\sqrt{m}(\widehat{R}_{S_{\text{test}}}(\hat{f}) - R(\hat{f})) \xrightarrow{d} \mathcal{N}(0, R(\hat{f})(1 - R(\hat{f})))$$

giving, writing $\widehat{R} := \widehat{R}_{S_{\text{test}}}(\hat{f})$, the asymptotic 95% confidence interval

$$\widehat{R} \pm 1.96 \sqrt{\frac{\widehat{R}(1 - \widehat{R})}{m}}$$

Numerically. With $m = 1000$ test points and an error rate $\widehat{R} = 0.10$, the half-width is $1.96\sqrt{0.1 \times 0.9/1000} \approx 0.019$. Your test error is $10\% \pm 1.9\%$.

3) Bayes predictor and Bayes risk

Proposition–Definition. The risk $R(\cdot)$ is minimized by the **Bayes predictor** $f^\star : \mathcal{X} \to \mathcal{Y}$, which satisfies, for every $x' \in \mathcal{X}$,

$$f^\star(x') \in \arg\min_{\hat{y} \in \mathcal{Y}} \underbrace{\mathbb{E}_{y|x=x'}[\ell(y, \hat{y})]}_{=:r(\hat{y}|x')}$$

The **Bayes risk** is the risk of the Bayes predictor:

$$R^\star := R(f^\star) = \mathbb{E}_{x \sim \mathbb{P}_x}[\inf_{\hat{y} \in \mathcal{Y}} r(\hat{y} \mid x)]$$

Note that the minimization is *pointwise in* x': the optimal predictor is built by solving one independent problem at each input. The outer expectation over $x \sim \mathbb{P}_x$ then averages the resulting conditional risks.

Proof

$$
\begin{aligned}
R(f) - R^\star &= R(f) - R(f^\star) \\
&= \int_{\mathcal{X}} [r(f(x') \mid x') - \min_{\hat{y} \in \mathcal{Y}} r(\hat{y} \mid x')] d\mathbb{P}_x(x') \\
&\geq 0
\end{aligned}
$$

The first equality uses the tower property, $R(f) = \mathbb{E}_{x \sim \mathbb{P}_x}[r(f(x) \mid x)]$. The bracket is non-negative for every x' by definition of the minimum, and the integral of a non-negative function against $\mathbb{P}_x$ is non-negative. ■

> **Excess risk.** For any predictor f we call $R(f) - R^\star \geq 0$ the excess risk. It is the quantity every theoretical result in this course will try to bound.

Example: classification

With the 0–1 loss $\ell(y, \hat{y}) = \mathbf{1}\{y \neq \hat{y}\}$:

$$f^\star(x') \in \arg\min_{\hat{y}} \mathbb{E}_{y \mid x = x'}[\mathbf{1}\{y \neq \hat{y}\}] = \arg\min_{\hat{y}} \mathbb{P}(y \neq \hat{y} \mid x = x')$$

$$= \arg\max_{\hat{y}} \mathbb{P}(y = \hat{y} \mid x = x')$$

The Bayes classifier predicts **the most probable output** given the input.

Its risk is $R^\star = \mathbb{E}_{x \sim \mathbb{P}_x}[1 - \max_k \mathbb{P}(y = k \mid x)]$, which vanishes only when the label is a deterministic function of the input.

Example: regression

With the squared loss $\ell(y, \hat{y}) = (y - \hat{y})^2$:

$$f^\star(x') \in \arg\min_{\hat{y}} \mathbb{E}_{y|x=x'}[(y - \hat{y})^2]$$

$$= \arg\min_{\hat{y}} \{\mathbb{E}_{y|x=x'}[(y - \mathbb{E}_{y|x=x'}[y])^2] + (\hat{y} - \mathbb{E}_{y|x=x'}[y])^2\}$$

$$\text{the first term does not depend on } \hat{y}$$

$$f^\star(x') = \mathbb{E}_{y|x=x'}[y],$$

$$R^\star = \mathbb{E}_{x \sim \mathbb{P}_x}[\mathrm{Var}_{y|x}(y)].$$

Regression is the estimation of a conditional expectation, and the Bayes risk is the average conditional variance, the **irreducible noise**.

Example: regression with the ℓ_1 loss

With the absolute loss $\ell(y, \hat{y}) = |y - \hat{y}|$:

$$f^\star(x') \in \arg\min_{\hat{y}} \mathbb{E}_{y|x=x'}[|y - \hat{y}|]$$

A useful identity.

$$\begin{aligned}
\mathbb{E}_{y|x=x'}[|y - \hat{y}|] &= \mathbb{E}_{y|x=x'}\left[\int_{-\infty}^{\infty} |\mathbf{1}\{y > s\} - \mathbf{1}\{\hat{y} > s\}| ds\right] \\
&= \int_{-\infty}^{\infty} \mathbb{E}_{y|x=x'}[|\mathbf{1}\{y > s\} - \mathbf{1}\{\hat{y} > s\}|] ds \quad \text{(Fubini)} \\
&= \int_{-\infty}^{\hat{y}} \mathbb{E}_{y|x=x'}[\mathbf{1}\{y \leq s\}] ds + \int_{\hat{y}}^{\infty} \mathbb{E}_{y|x=x'}[\mathbf{1}\{y > s\}] ds \quad (\mathbf{1}\{\hat{y} > s\} = \mathbf{1}\{s < \hat{y}\}) \\
&= \int_{-\infty}^{\hat{y}} \mathbb{P}(y \leq s \mid x = x') ds + \int_{\hat{y}}^{\infty} \mathbb{P}(y > s \mid x = x') ds
\end{aligned}$$

Example: regression with the ℓ_1 loss

Recall, $g(\hat{y}) = \int_{-\infty}^{\hat{y}} \mathbb{P}(y \leq s \mid x = x')ds + \int_{\hat{y}}^{\infty} \mathbb{P}(y > s \mid x = x')ds$

$$a \mapsto |y - a| \text{convex} \forall y \quad \implies \quad g \text{convex, hence subdifferentiable everywhere.}$$

$$\hat{y} \in \arg\min_a g(a) \quad \iff \quad 0 \in \partial g(\hat{y})$$

From the two-integral expression above: $s \mapsto \mathbb{P}(y \leq s \mid x = x')$ is right-continuous and $s \mapsto \mathbb{P}(y < s \mid x = x')$ is left-continuous (as a CDF and its left limit), which is enough to get the one-sided derivatives of g at $\hat{y}$, hence its subdifferential:

$$\partial g(\hat{y}) = [\mathbb{P}(y < \hat{y} \mid x = x') - \mathbb{P}(y \geq \hat{y} \mid x = x'), \mathbb{P}(y \leq \hat{y} \mid x = x') - \mathbb{P}(y > \hat{y} \mid x = x')]$$

$$0 \in \partial g(\hat{y}) \iff \mathbb{P}(y \leq \hat{y} \mid x = x') \geq \tfrac{1}{2} \text{and} \mathbb{P}(y \geq \hat{y} \mid x = x') \geq \tfrac{1}{2} \iff f^\star(x') = \text{median}(\mathbb{P}_{y|x=x'})$$

Same recipe as before, a different loss produces a different Bayes predictor: the ℓ_2 loss recovers the conditional **mean**, the ℓ_1 loss recovers the conditional **median**. The median ignores how far outliers are, only how many points lie on each side, so it is far more **robust** to heavy-tailed noise or mislabelled points.

Two remarks before moving on

$R^\star > 0$ in general. No algorithm, no volume of data removes $\mathbb{E}_{x \sim \mathbb{P}_x}[\mathrm{Var}_{y|x}(y)]$. When your test error stops improving, the honest hypothesis is that you have reached the noise floor, not that you need a bigger model.

Knowing $f^\star$ does not solve the problem. Its expression involves $\mathbb{P}_{y|x}$, which is exactly what we do not know. The Bayes predictor is a *target*, not a method. The next part is about the different ways of aiming at it.

Three strategies:

1. restrict to a parametric family and minimize the empirical risk
2. estimate the distribution, then plug it into the formula for $f^\star$
3. exploit a regularity assumption to estimate $f^\star(x')$ locally

IV. Classical learning paradigms

1) Empirical risk minimization

Setting. Restrict the search to a parametric family of predictors

$$f_\theta : \mathcal{X} \to \mathcal{Y}, \quad \theta \in \Theta$$

Idea. Replace the unknown objective by its estimator, $R(f_\theta) \approx \widehat{R}_S(f_\theta)$.

Estimator.

$$\hat{\theta} \in \arg\min_{\theta \in \Theta} \widehat{R}_S(f_\theta) = \arg\min_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f_\theta(x_i))$$

Two approximations are now stacked: Θ instead of all functions, and $\widehat{R}_S$ instead of R. The next slides quantify each of them.

Example: linear regression

Take $f_\theta(x) = \theta^\top \varphi(x)$, where φ is a **feature map**, together with the squared loss:

$$\widehat{R}_S(f_\theta) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \theta^\top \varphi(x_i))^2$$

Equivalently, from the generative side: assume the model

$$y = \theta^\top x + \epsilon, \quad \epsilon \text{ noise, independent of } x$$

Squared loss plus linear model gives back exactly

$$\hat{\theta} \in \arg\min_{\theta \in \Theta} \frac{1}{n}\sum_{i=1}^{n}(y_i - \theta^\top x_i)^2$$

The feature map is what makes this non-trivial: $\varphi(x) = (1, x, x^2, \ldots, x^k)$ turns linear regression into polynomial regression without changing a single line of the analysis.

Decomposition of the excess risk

For an estimator $\hat{\theta}$:

$$R(f_{\hat{\theta}}) - R^\star = \underbrace{\{R(f_{\hat{\theta}}) - \inf_{\theta \in \Theta} R(f_\theta)\}}_{\textbf{estimation error}} + \underbrace{\{\inf_{\theta \in \Theta} R(f_\theta) - R^\star\}}_{\textbf{approximation error}}$$

Estimation error. How much we lose by having only n samples instead of $\mathbb{P}$ itself. Random, since it depends on S. Decreases with n, **increases** with the richness of Θ.

Approximation error. How much we lose by restricting to Θ. Deterministic, independent of n, **decreases** with the richness of Θ.

The two terms pull in opposite directions. Everything that follows is about that tension.

Controlling the estimation error

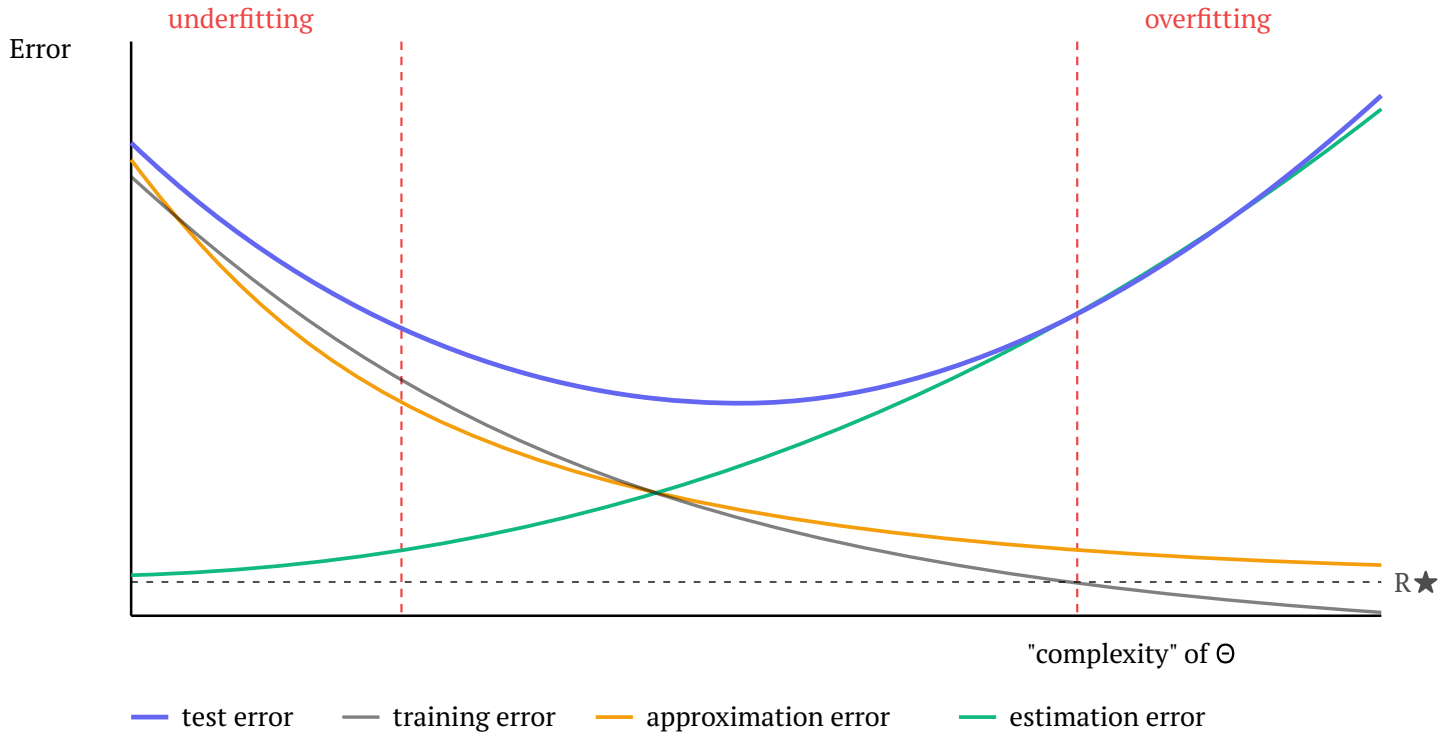
Let $\theta' \in \arg\min_{\theta \in \Theta} R(f_\theta)$. Then

$$R(f_{\hat{\theta}}) - R(f_{\theta'}) = \underline{(R(f_{\hat{\theta}}) - \widehat{R}_S(f_{\hat{\theta}}))} + (\widehat{R}_S(f_{\hat{\theta}}) - \widehat{R}_S(f_{\theta'})) + \underline{(\widehat{R}_S(f_{\theta'}) - R(f_{\theta'}))}$$

$$\leq \underbrace{2 \sup_{\theta \in \Theta} |\widehat{R}_S(f_\theta) - R(f_\theta)|}_{\text{uniform deviation}} + \underbrace{\sup_{\theta \in \Theta} (\widehat{R}_S(f_{\hat{\theta}}) - \widehat{R}_S(f_\theta))}_{\text{optimisation error}}$$

The first term is a **supremum of centred random variables**: for each fixed θ, $\mathbb{E}_{S \sim \mathbb{P}^{\otimes n}}[\widehat{R}_S(f_\theta) - R(f_\theta)] = 0$, but the supremum over Θ does not have mean zero. Bounding it is the object of statistical learning theory (later chapters: uniform bounds, Rademacher complexity). The second is zero if the minimization is solved exactly, and measures how far the optimisation algorithm stopped from the true minimizer otherwise.

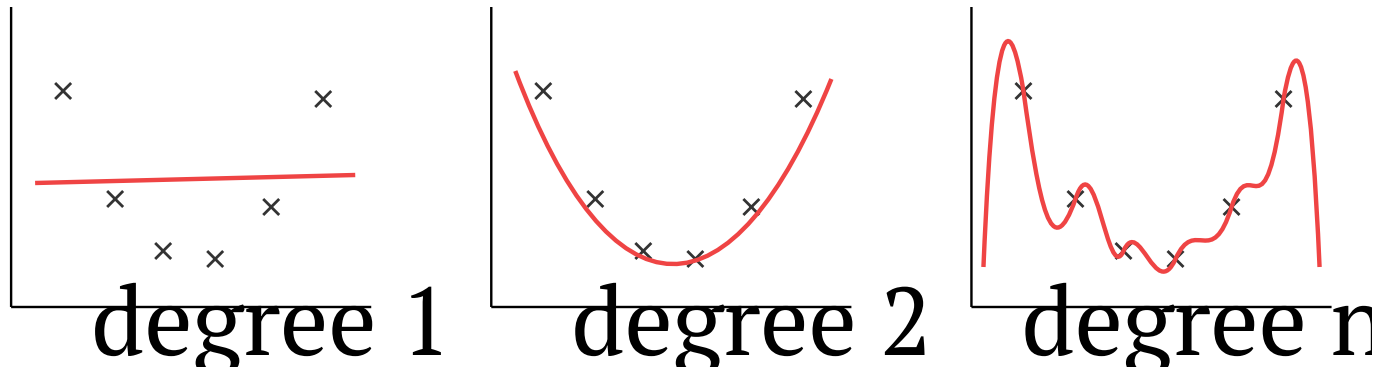
Underfitting vs overfitting



The test error is the sum of $R^\star$, plus the approximation error, plus the estimation error.

Example: polynomial regression

Fitting polynomials of increasing degree; at degree $n - 1$ we recover the Lagrange interpolating polynomial, which passes through every point exactly.



The third fit has **zero training error** and is useless. Empirical risk minimization, taken literally over a rich enough class, is a bad idea.

Regularisation

The remedy: penalise complexity inside the objective itself.

$$\hat{\theta} \in \arg\min_{\theta \in \Theta} \{ \widehat{R}_S(f_\theta) + \underbrace{\Omega(\theta)}_{\text{complexity penalty}} \}$$

Instead of restricting Θ by hand, we keep a rich class and make the expensive parts of it costly. The penalty controls the estimation error without sacrificing all of the approximation gain.

Classical choices: $\Omega(\theta) = \lambda \|\theta\|_2^2$ (ridge), $\lambda \|\theta\|_1$ (lasso).

λ is a hyperparameter: it cannot be chosen by minimizing the training error, which is decreasing in λ^{-1} by construction. It is chosen on the **validation set**.

How λ is actually tuned

Write, for a fixed $\lambda \geq 0$, the parameter fitted **on S_{train} alone**

$$\hat{\theta}_\lambda(S_{\text{train}}) \in \arg\min_{\theta \in \Theta} \{\widehat{R}_{S_{\text{train}}}(f_\theta) + \lambda \|\theta\|_2^2\}, \qquad \hat{f}_\lambda := f_{\hat{\theta}_\lambda(S_{\text{train}})}$$

1. Fix a finite grid $\Lambda = \{\lambda_1, \dots, \lambda_M\}$, usually geometric, e.g. $\lambda_j = 10^{a + (b-a)j/M}$.
2. For each $\lambda \in \Lambda$: fit $\hat{f}_\lambda$ on S_{train}, then record the validation error $\widehat{R}_{S_{\text{val}}}(\hat{f}_\lambda)$.
3. Select $\hat{\lambda} \in \arg\min_{\lambda \in \Lambda} \widehat{R}_{S_{\text{val}}}(\hat{f}_\lambda)$.
4. Refit on $S_{\text{train}} \cup S_{\text{val}}$ with $\lambda = \hat{\lambda}$ to get the final $\hat{f}$.
5. Report $\widehat{R}_{S_{\text{test}}}(\hat{f})$ **once**.

Why it is legitimate. For each fixed λ, $\hat{f}_\lambda$ depends only on S_{train}, so conditionally on S_{train} the LLN and CLT of Part III give $\mathbb{E}[\widehat{R}_{S_{\text{val}}}(\hat{f}_\lambda)] = R(\hat{f}_\lambda)$. Taking the $\min$ over the grid reintroduces a mild optimistic bias, growing like $\sqrt{\log M / |S_{\text{val}}|}$: keep M moderate, and never read the final number off S_{val}.

When data is scarce, replace the single split by K-fold cross-validation on S_{train}.

2) Probabilistic methods, maximum likelihood

Idea. Make an assumption on the class of probability measures relevant to the problem, and choose the one that maximises the chance of observing the training set.

Let $\{p_\theta : \theta \in \Theta\}$ be a parametric family of densities on $\mathcal{X} \times \mathcal{Y}$. Since $S \sim \mathbb{P}^{\otimes n}$, the joint density of the sample factorises:

$$\hat{\theta} \in \arg\max_{\theta \in \Theta} \underbrace{p_\theta(x_1, y_1) \times \cdots \times p_\theta(x_n, y_n)}_{=: \mathcal{L}(\theta; S)}$$

the **likelihood** of θ given S.

Equivalently, taking logarithms, which turns the product into a sum and the maximisation into something computable:

$$\hat{\theta} \in \arg\max_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} \ln p_\theta(x_i, y_i) \quad \textbf{(log-likelihood)}$$

Example: logistic regression

Sigmoid function:

$$\sigma(z) := \frac{1}{1 + e^{-z}}$$

Logistic model, with labels coded in $\{-1, 1\}$:

$$p_\theta(y = 1 \mid x) = \frac{1}{1 + e^{-\theta^\top x}} = \sigma(\theta^\top x)$$

Remark. The two cases collapse into one formula:

$$p_\theta(y = -1 \mid x) = 1 - \sigma(\theta^\top x) = \frac{e^{-\theta^\top x}}{1 + e^{-\theta^\top x}} = \frac{1}{1 + e^{\theta^\top x}} = \sigma(-\theta^\top x)$$

hence, for both values of y,

$$p_\theta(y \mid x) = \sigma(y\theta^\top x)$$

Logistic regression: the resulting objective

Maximum likelihood on that model:

$$\begin{aligned}\hat{\theta} &\in \arg\max_{\theta} \sum_{i=1}^{n} \ln\left(\frac{1}{1+e^{-y_i\theta^\top x_i}}\right) \\ &= \arg\min_{\theta} \frac{1}{n}\sum_{i=1}^{n} \ln(1+e^{-y_i\theta^\top x_i})\end{aligned}$$

Look at what we obtained: an **empirical risk minimization** problem $\min_\theta \widehat{R}_S(f_\theta)$, with the loss

$$\ell(y, \theta^\top x) = \ln(1+e^{-y\theta^\top x})$$

Maximum likelihood and ERM are not two competing paradigms. The first produces instances of the second.

The bridge between the paradigms

Linear regression. Model $y = \theta^\top x + \epsilon$ with $\epsilon \sim \mathcal{N}(0, \sigma^2)$. Then

$$-\ln p_\theta(y \mid x) = \frac{(y - \theta^\top x)^2}{2\sigma^2} + \text{const}$$

Maximum likelihood $\Longleftrightarrow$ least squares.

Logistic regression. Bernoulli model $\Longleftrightarrow$ ERM with the logistic loss.

General principle. Choosing a parametric model $p_\theta(y \mid x)$ and setting $\ell(y, \hat{y}) := -\ln p_\theta(y \mid x)$ turns maximum likelihood into empirical risk minimization. Most losses used in practice arise this way rather than being invented.

Why not minimize the 0–1 loss directly?

In classification, the quantity we actually care about is $\ell_{01}(y, \hat{y}) = \mathbf{1}\{y \neq \hat{y}\}$. Yet every method above optimised something else.

Two obstructions:

- ℓ_{01} is piecewise constant: its gradient is zero almost everywhere, so no descent method can move
- minimizing it over a linear class is NP-hard in general

The logistic loss $\ln(1 + e^{-y\theta^\top x})$ is a **convex surrogate**: differentiable, convex, and an upper bound on ℓ_{01} up to a constant.

Keep the two apart: the loss you **optimise** and the risk you **report** are almost never the same function.

3) Local averaging methods

Idea. Approximate the Bayes predictor directly, by assuming a **regularity in** x: points that are close should have similar conditional distributions $\mathbb{P}_{y|x}$.

Classification

$$f^\star(x') = \arg\max_{\hat{y}} \mathbb{P}(y = \hat{y} \mid x = x')$$

k-nearest neighbours: predict the majority class among the k nearest neighbours of x' in the training set.

Regression

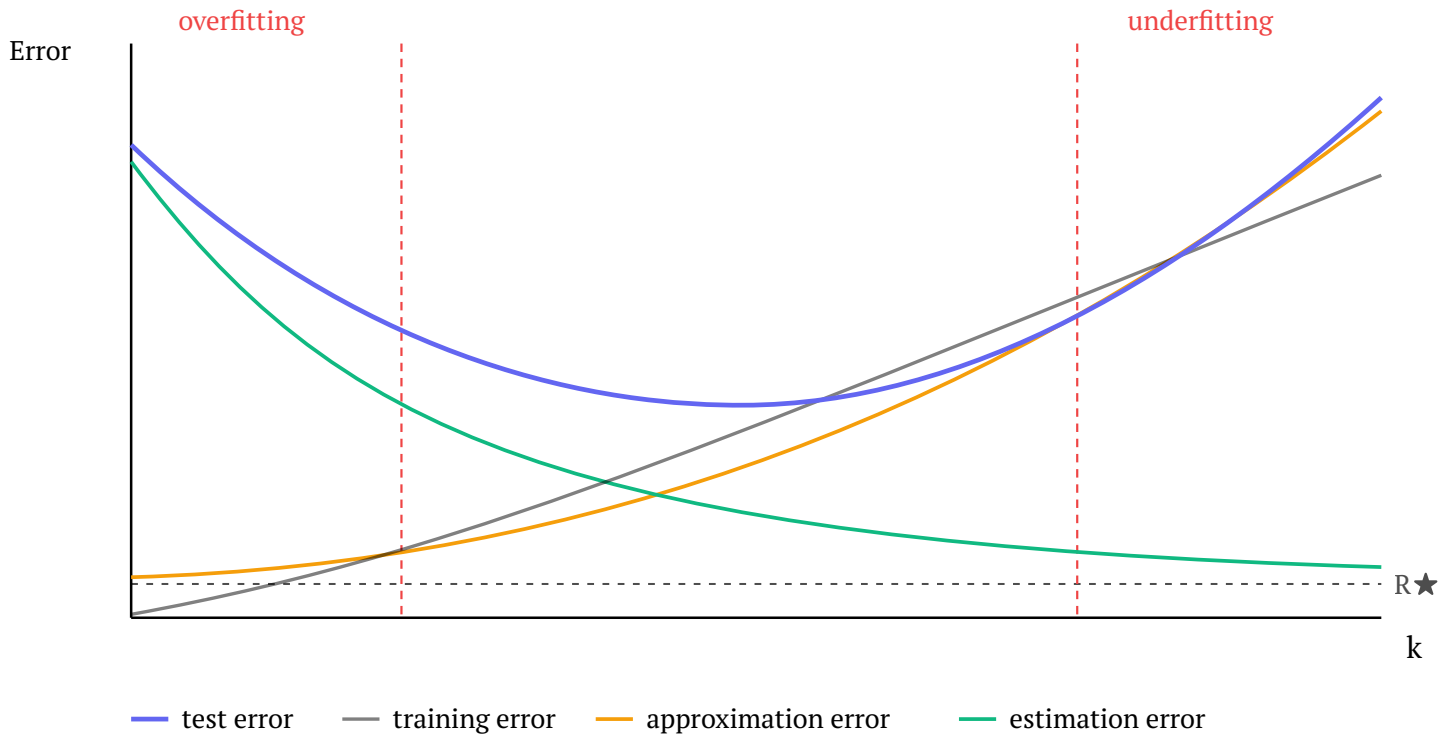
$$f^\star(x') = \mathbb{E}_{y|x=x'}[y]$$

k-nearest neighbours: predict the average of the y of the k nearest neighbours of x' in the training set.

In both cases, an expectation under $\mathbb{P}_{y|x=x'}$ is replaced by an empirical average over a neighbourhood of x'. No parameter is fitted: the training set *is* the model.

Underfitting vs overfitting, for k-NN

Same phenomenon, but the complexity axis is reversed: small k means a rich, flexible predictor.



4) Generative methods

Idea. Rather than going straight for f, estimate the full joint law, then plug it into the expression of the Bayes predictor.

Step 1. Estimate the densities $p(x \mid y)$ and $p(y)$.

Step 2. Deduce the conditional by Bayes' formula:

$$p(y \mid x) \propto p(x \mid y)p(y)$$

and predict $\arg\max_{\hat{y}} p(\hat{y} \mid x)$, the most probable class.

The name comes from step 1: a model of $p(x \mid y)$ lets you *generate* synthetic inputs for a given class, which a model of $p(y \mid x)$ alone does not.

Example: the Gaussian case in 1D

Take a simplified model where both laws are known, so nothing has to be estimated:

$$y \sim \mathcal{B}(\tfrac{1}{2}), \quad x \mid y \sim \mathcal{N}(y, 1), \quad \text{i.e.} \quad p(x \mid y) = \frac{1}{\sqrt{2\pi}} \exp(-(x-y)^2/2)$$

By Bayes' formula, for a given $x' \in \mathbb{R}$,

$$\mathbb{P}(y = 0 \mid x = x') \propto \tfrac{1}{2} \exp(-x'^2/2), \quad \mathbb{P}(y = 1 \mid x = x') \propto \tfrac{1}{2} \exp(-(x'-1)^2/2)$$

Hence

$$\frac{\mathbb{P}(y = 0 \mid x = x')}{\mathbb{P}(y = 1 \mid x = x')} \geq 1 \iff (x'-1)^2 \geq x'^2 \iff -2x' + 1 \geq 0 \iff x' \leq \tfrac{1}{2}$$

The Bayes classifier is the threshold rule at $x' = 1/2$, the midpoint of the two means, as symmetry demands.

V. The No Free Lunch theorem

No free lunch

In short. There is no magic algorithm that performs well on every distribution. Each algorithm has advantages and drawbacks, and **assumptions on the data are necessary** to obtain interesting results.

Theorem. Consider binary classification with the 0–1 loss, with $\mathcal{X}$ infinite. Let $\mathcal{P}$ be the set of all probability distributions on $\mathcal{X} \times \{0, 1\}$, and write $R_{\mathbb{P}}$, $R^\star_{\mathbb{P}}$ for the risk and Bayes risk under $\mathbb{P}$. Then for every n and every algorithm $\mathcal{A}$,

$$\sup_{\mathbb{P} \in \mathcal{P}} \{ \mathbb{E}_{S \sim \mathbb{P}^{\otimes n}} [R_{\mathbb{P}}(\mathcal{A}(S))] - R^\star_{\mathbb{P}} \} \geq \frac{1}{2}$$

See Devroye, Györfi and Lugosi, *A Probabilistic Theory of Pattern Recognition*, for a precise statement and proof.

Reading the theorem

The supremum is over **all** distributions, including adversarial ones. For any algorithm you fix, there exists a $\mathbb{P}$, depending on the algorithm, on which it does no better than random guessing, and this even though $R^\star_{\mathbb{P}}$ can be taken equal to zero.

It does **not** say that learning is impossible. It says that a guarantee valid uniformly over $\mathcal{P}$ is impossible. Note the order of the quantifiers: the bad $\mathbb{P}$ is chosen *after* the algorithm.

Consequence for the rest of the course. Every theorem we prove will hold under assumptions restricting $\mathcal{P}$: smoothness of $f^\star$, a margin condition, a bounded complexity of Θ, a sparse θ. Those assumptions are not technical conveniences. Without some of them, nothing can be proved at all.

VI. Other areas of machine learning

Beyond supervised learning

Unsupervised learning. We observe only the x, not pairs (x, y). The goal is to learn the structure of $\mathbb{P}_x$. Examples: principal component analysis, clustering, ...

Online learning. The data arrive as a stream, and the predictor must be updated as they come.

Reinforcement learning. The algorithm can act on its environment and receives rewards in return. Example: video games.

Data generation. Learn the structure of the data well enough to produce new samples from it. Examples: large language models, diffusion models.

What to remember

- The data are i.i.d. from an unknown $\mathbb{P}$ on $\mathcal{X} \times \mathcal{Y}$
- $R(f) = \mathbb{E}_{(x,y)\sim\mathbb{P}}[\ell(y, f(x))]$ is unobservable; $\widehat{R}_S(f)$ estimates it
- LLN and CLT justify this **for f fixed independently of S**
- $\hat{f}_S$ depends on the sample, so $\widehat{R}_S(\hat{f}_S)$ is optimistically biased, hence held-out data
- The Bayes predictor $f^\star$ is the target; $R^\star > 0$ in general
- Excess risk = estimation + approximation error
- Four paradigms, one shared requirement: assumptions
- No free lunch: without assumptions, nothing can be guaranteed