Introduction to Machine learning

Sylvain Le Corff

Prerequisites: basics of probability and statistics.

Objectives:

- Learn and apply the fundamental concepts of statistical learning;
- Understand the basic theory underlying data science;
- Implement (Python) the most classical learning algorithms;
- Be able to read current research books and papers.

This course:

- 4h of lecture/TD/TP per week;
- Evaluation: 1/4 Moodle quiz + 3/4 exam.

References

→ Slides adapted from Claire Boyer's lectures.

→ Probabilistic Machine Learning: An Introduction by Kevin P. Murphy

(Complete book & figures)

→M1 Lecture notes - statistical learning introduction and basic topics on regression

(Lecture notes)

Material

→ Mini tests after each session on Moodle for training.

→ Notebooks to create all illustrations of the lectures (Experiments)

→ Materials on Moodle and (Github).

1. Some applications
2. Learning settings
3. Mathematical framework
4. Some criterion for regression and supervised classification
   - Regression
   - Binary classification
   - Scoring function
5. Empirical risk
6. Case of a finite class
7. Overfitting
8. Cross-validation

Summary

Computer vision

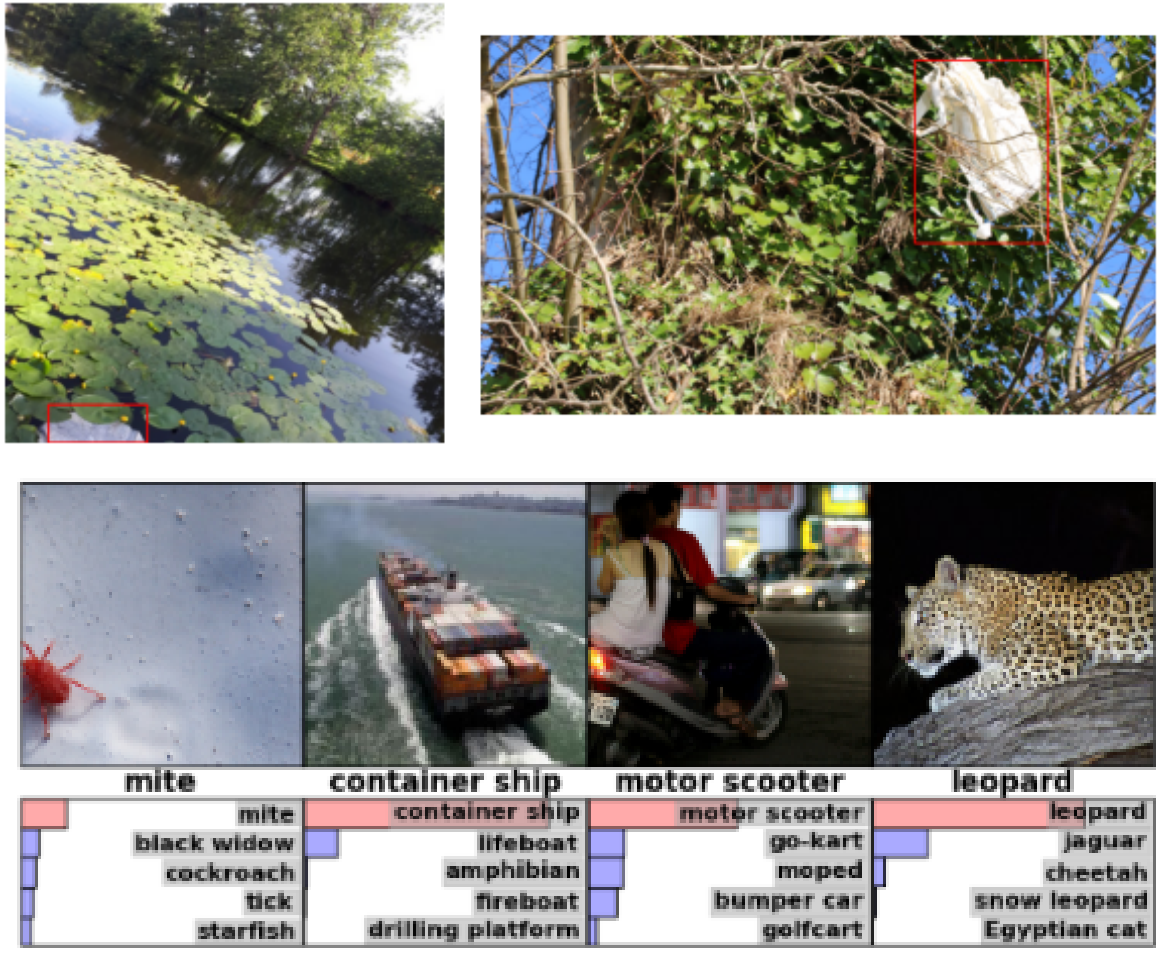


Figure: Detection of plastic trash in videos, image classification. (ImageNet, Krizhevsky et al., 2012), (Chagneux et al. 2022).

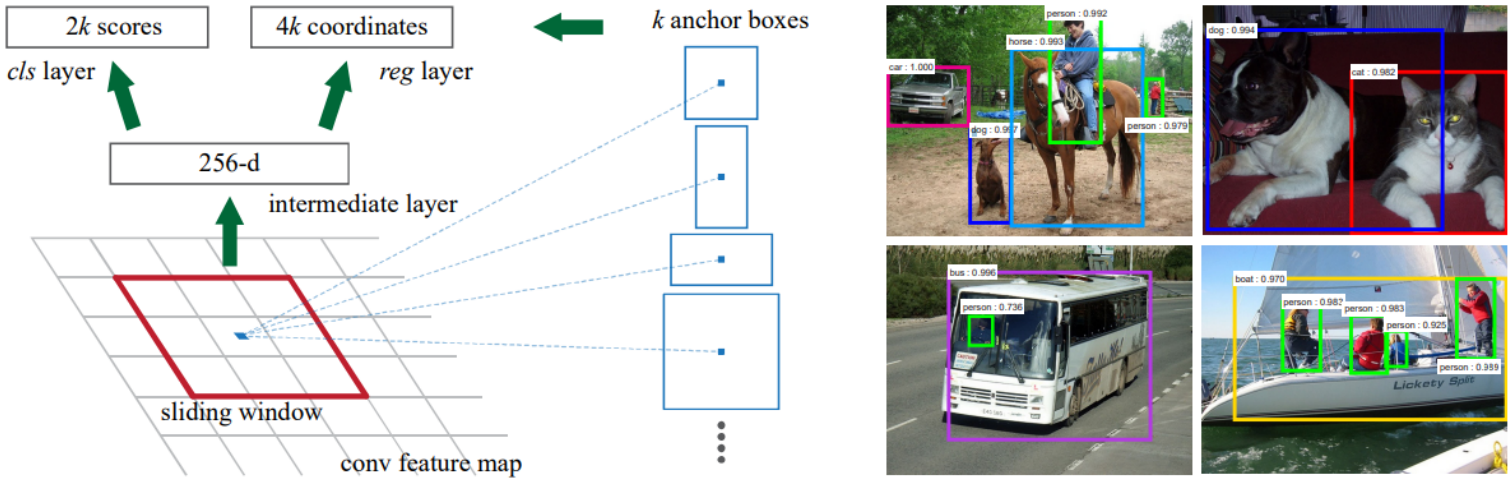
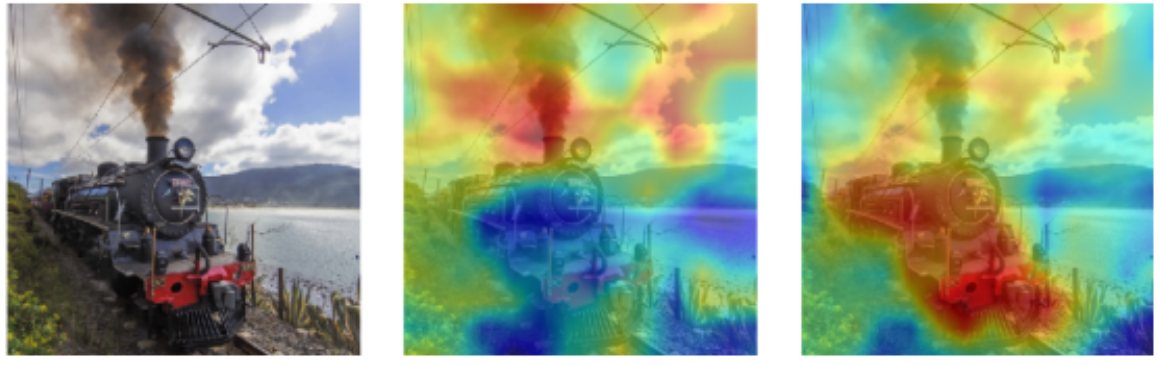


Figure: Automatic detections using Region Proposal Network (bounding box design). (Faster-RCNN, Ren et al. 2015). Code is available at `https://github.com/ShaoqingRen/faster_rcnn`.

Vision + Natural Language Processing

(a) Question: Where is the woman ? - Answer: on the elephant



(b) Question: Where is the smoke coming from ? - Answer: train

Figure: Visual Question Answering (VQA) tasks. (Mutan, Ben-Younes et al. 2017).

Medical applications

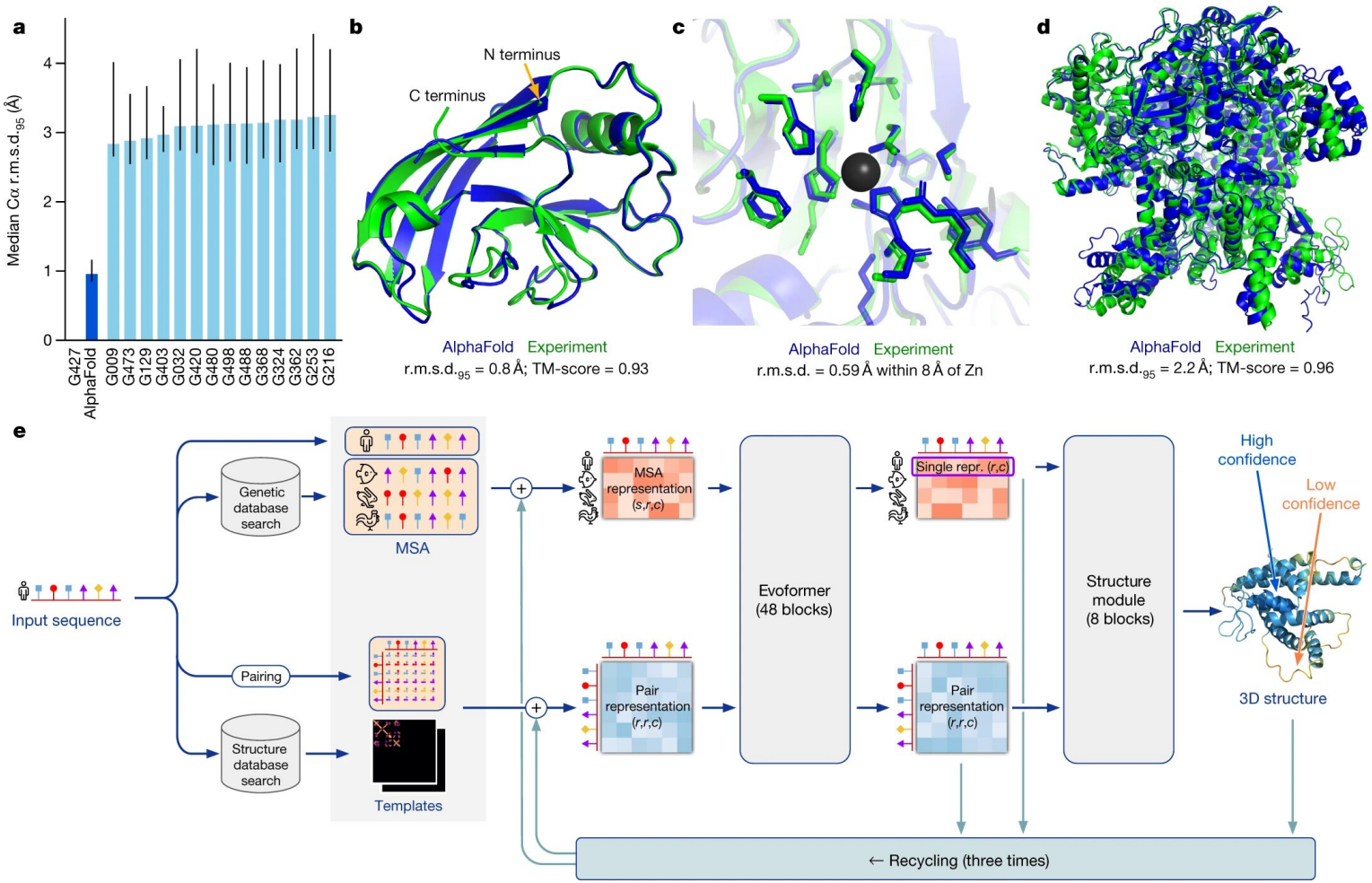


Figure: Predicting the three-dimensional structure that a protein will adopt based solely on its amino acid sequence—the structure prediction component of the "protein folding problem". Performance of AlphaFold on the CASP14 dataset. (AlphaFold, Jumper et al. 2022).

Challenges - reduce GHG emissions from cities

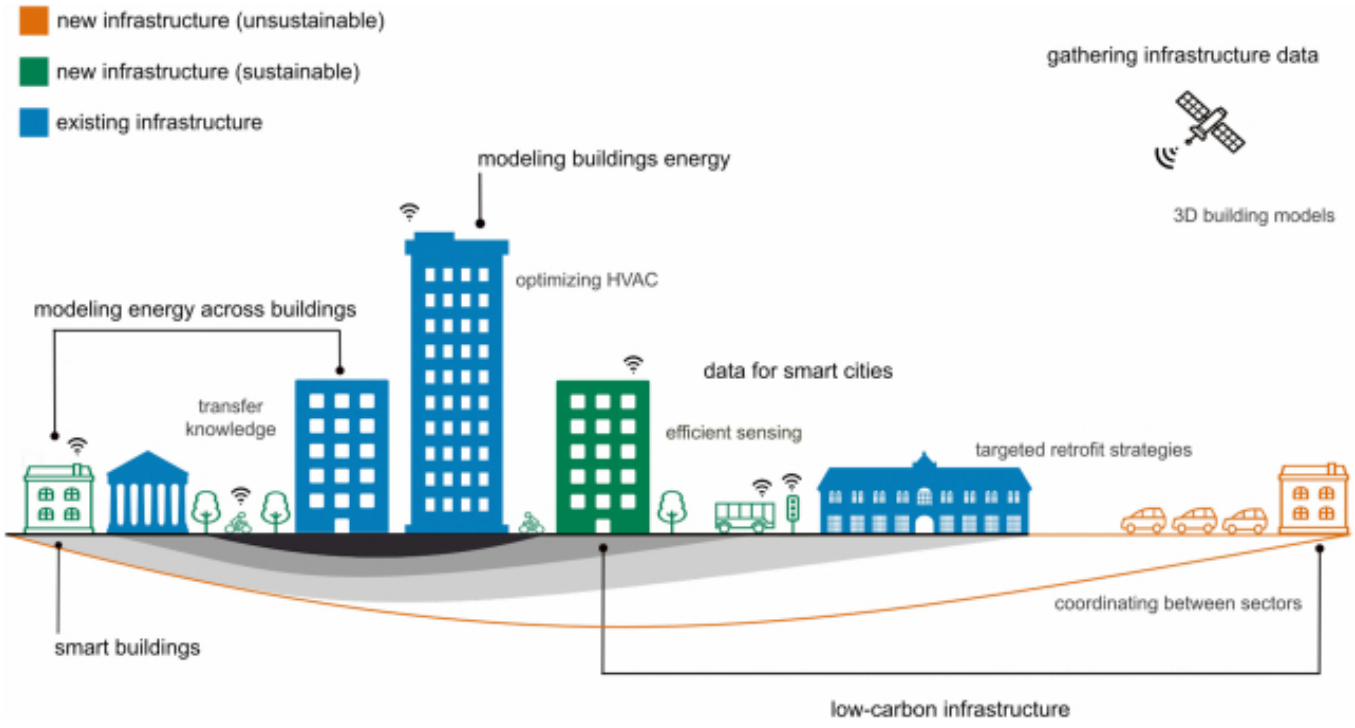


Figure: Selected opportunities to reduce GHG emissions from buildings and cities using ML. (Tackling Climate Change with Machine Learning, Rolnick et al. 2022).

Challenges - reduce GHG emissions from land use

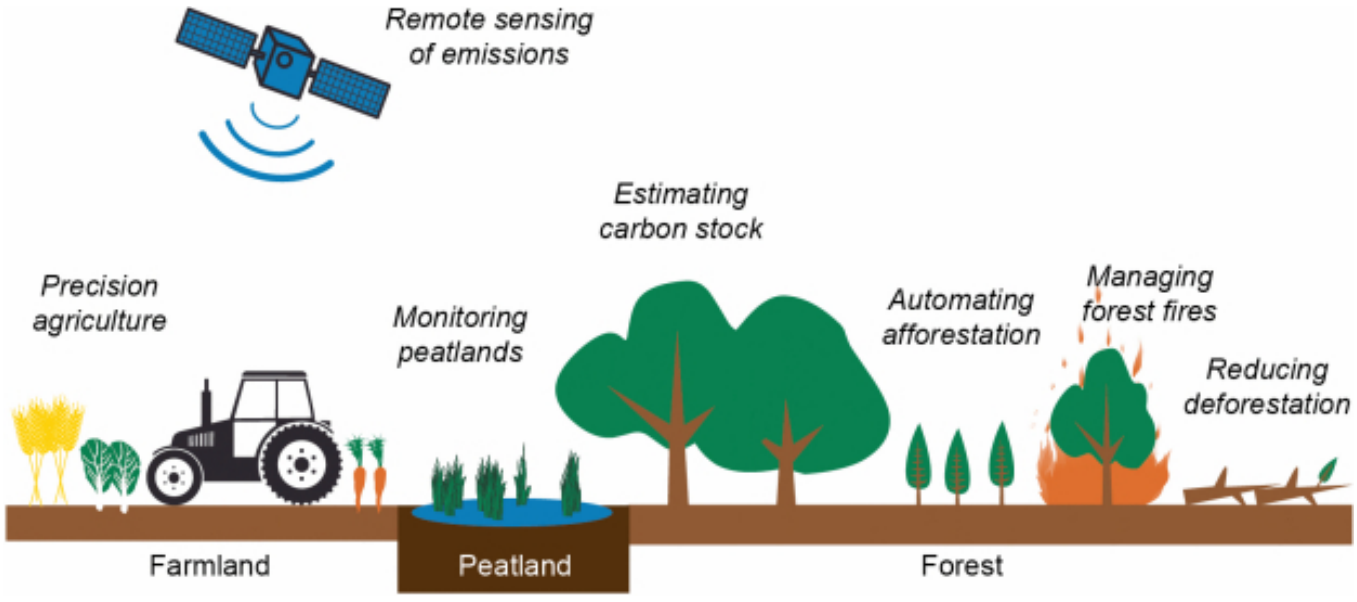


Figure: Selected opportunities to reduce GHG emissions from land use using ML. (Tackling Climate Change with Machine Learning, Rolnick et al. 2022).

Many other applications

- Efficient simulation in physics-based problem (Fluid simulation, Tompson et al., 2017), (Metamodels for building energy models, Cohen et al., 2021).
- Natural language processing (Dynamic memory networks, Kumar A. et al., 2016), (Dall-E, OpenAI).
- Medical diagnosis (Causal inference for medical diagnosis, Richens, J.G. et al., 2020).
- Generativel models for speech processing (WaveNet, van den Oord A. et al., 2016).
- Many more...

Summary

Typical machine learning problems

- **Classification**: Assign a category to each input data.
- **Regression**: Predict a vector associated with each input data.
- **Ranking**: Order input data according to some criterion.
- **Clustering**: Partition input data into homogeneous regions.
- **Dimensionality reduction or manifold learning**: Transform an initial representation of input data into a lower-dimensional representation while preserving some properties.

Focus on classification and regression

Classification: $(X, Y) \in \mathbb{R}^d \times \{1, \dots, M\}$

$\rightharpoonup$ Learn whether an individual from $\mathbb{R}^d$ belongs to some class.

$\rightharpoonup$ Focus with known number M of classes, $\mathcal{Y} = \{1, \dots, M\}$.

$\rightharpoonup$ Objective: define a function $f : \mathbb{R}^d \to \{1, \dots, M\}$, such that $f(X)$ is the best prediction of Y in a given context.

Regression: $(X, Y) \in \mathbb{R}^d \times \mathbb{R}^m$

$\rightharpoonup$ The observation associated with X is assumed to be given by

$$Y = f_*(X) + \varepsilon\,,$$

where ε is a centered noise independent of X.

$\rightharpoonup$ Objective: define the best estimator of f_* in a given context.

Learning settings

The two main settings are supervised and unsupervised learning.

- **Supervised learning**:
  - Training data: a set of labeled examples
  - Prediction for all unseen points.

  $\rightsquigarrow$ classification, regression, and ranking problems
- **Unsupervised learning**:
  - Training data: a set of unlabeled examples
  - Prediction for all unseen points.

  $\rightsquigarrow$ clustering and dimensionality reduction
- **Semi-supervised learning**:
  - Training data: a set of both labeled and unlabeled examples
  - Prediction for all unseen points.
- and also **Online learning**, etc...

Summary

Supervised Learning Framework

- Input measurement $\mathbf{X} \in \mathcal{X}$ (often $\mathcal{X} \subset \mathbb{R}^d$).
- Output measurement $Y \in \mathcal{Y}$.
- The joint distribution of $(\mathbf{X}, Y)$ is unknown.
- $Y \in \{1, \ldots, M\}$ (classification) or $Y \in \mathbb{R}^m$ (regression).
- A predictor is a measurable function in $f : \mathcal{X} \to \mathcal{Y}$.

Training data

$\mathcal{D}_n = \{(\mathbf{X}_i, Y_i)\}_{1 \leqslant i \leqslant n}$ i.i.d. with the same distribution as $(\mathbf{X}, Y)$.

Goal

- Construct a good predictor $\widehat{f}_n$ from the training data.
- Predict the output for input data not in the training dataset.

Mathematical framework

- We use a cost function to evaluate the quality of a prediction: $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^+$ is such that

$$\begin{aligned} \ell(y, y') &= 0 \quad \text{if} \quad y = y' \\ &> 0 \quad \text{if} \quad y \neq y'. \end{aligned}$$

- Interpretation: $\ell(y, y')$ measures an error between the prediction y' and the observation y.

Classical choices

$\rightharpoonup$ Prediction loss: $\ell(Y, f(\mathbf{X})) = \mathbf{1}_{Y \neq f(\mathbf{X})}$.
$\rightharpoonup$ Quadratic loss: $\ell(Y, \mathbf{X}) = \|Y - f(\mathbf{X})\|_2^2$.

We assume for the whole lecture that

- data $d_n = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ are realizations of a n-sample $\mathcal{D}_n := \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$, meaning that the (X_i, Y_i), $1 \leqslant i \leqslant n$, are i.i.d. copies of (X, Y) taking value in $\mathcal{X} \times \mathcal{Y}$.
- For a given cost function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^+$, the performance of a predictor $f : \mathcal{X} \to \mathcal{Y}$ is measured as the average loss.

Risk of a predictor

The risk of a predictor $f : \mathcal{X} \to \mathcal{Y}$ is defined by

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

Classical choices

$\rightharpoonup$ Prediction loss: $\mathbb{E}[\ell(Y, f(\mathbf{X}))] = \mathbb{P}(Y \neq f(\mathbf{X}))$.
$\rightharpoonup$ Quadratic loss: $\mathbb{E}[\ell(Y, f(\mathbf{X}))] = \mathbb{E}[\|Y - f(\mathbf{X})\|_2^2]$.

THE problem in Machine Learning

The problem is then to find a minimizer of the risk for a fixed cost function ℓ

$$f^\star \in \operatorname{argmin}_{f:\mathcal{X}\to\mathcal{Y}} \{\mathcal{R}(f) = \mathbb{E}[\ell(Y, f(X))]\}.$$

Such a function $f^\star$ (if it exists) is called the optimal predictor for the cost function ℓ and we define $\mathcal{R}^\star := \mathcal{R}(f^\star)$.

- $f^\star$ generally depends on the unknown probability distribution of (X, Y), then $f^\star$ is unknown in practice.
- Using $\mathcal{D}_n$, our work consists in finding a good estimate f_n of $f^\star$, i.e. finding $f_n : \mathcal{X} \to \mathcal{Y}$ such that $\mathcal{R}(f_n) \simeq \mathcal{R}(f^\star)$.
- As f_n depends on $\mathcal{D}_n$, $\mathcal{R}(\widehat{f_n})$ is a random variable!

Choice of a cost function ℓ?

- The proposed mathematical framework implies that a predictor is performant with respect to a criterion (represented by the cost function ℓ).
- It means that a predictor f could be good for a cost function ℓ_1 ($\mathcal{R}_1(f)$ small) but not for another cost function ℓ_2 ($\mathcal{R}_2(f)$ large).

- Crucial to choose a relevant cost function for the problem we are faced.
- Can reflect a prior that you know on your problem

Summary

Regression

In regression ($\mathcal{Y} = \mathbb{R}$), the quadratic cost is often used, defined as follows:

$$\ell : \mathbb{R} \times \mathbb{R} \to \mathbb{R}^+$$
$$(y, y') \mapsto (y - y')^2.$$

Define the quadratic risk for a machine or regression function $m : \mathcal{X} \to \mathbb{R}$:

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

As seen in your previous statistics lectures, the optimal regression function $m^\star$ for the quadratic risk is

$$m^\star(X) := \mathbb{E}\left[Y|X\right].$$

Indeed, for all m,

$$\mathcal{R}(m^\star) = \mathbb{E}\left[(Y - m^\star(X))^2\right] \leqslant \mathbb{E}\left[(Y - m(X))^2\right] =: \mathcal{R}(m).$$

Binary classification

- The output can only take 2 values ($Y \in \{0, 1\}$).
- Note that the distribution of (X,Y) is entirely characterized by the marginal distribution of X and the conditional distribution of Y given X. More precisely, for all $A \in \mathscr{B}(\mathbb{R}^d)$, write $\mu_X(A) = \mathbb{P}(X \in A)$, and

$$r(X) = \mathbb{E}[Y|X] = \mathbb{P}(Y = 1|X).$$

The error probability or the risk for a classification rule

For a classifier $g : \mathbb{R}^d \to \{0, 1\}$,

$$\mathcal{R}(g) = \mathbb{E}\left[\mathbb{1}_{g(X)\neq Y}\right] = \mathbb{P}(g(X) \neq Y).$$

Does an optimum exist?

The Bayes classifier $g^\star$ is defined as:

$$g^\star(X) = \begin{cases} 1 & \text{if} \quad \mathbb{P}(Y=1|X) > \mathbb{P}(Y=0|X), \\ 0 & \text{otherwise.} \end{cases}$$

Equivalently,

$$g^\star(X) = \begin{cases} 1 & \text{if} \quad r(X) > 1/2, \\ 0 & \text{otherwise,} \end{cases}$$

Lemma

For any classification rule $g : \mathbb{R}^d \to \{0,1\}$, *one has*

$$\mathcal{R}(g^\star) \leqslant \mathcal{R}(g).$$

The Bayes risk

$$\mathcal{R}^\star := \mathcal{R}(g^\star) = \inf_{g:\mathbb{R}^d \to \{0,1\}} \mathbb{P}(g(X) \neq Y).$$

See blackboard

- $g^\star$ depends on the distribution of (X, Y).
- The explicit solution requires to know $\mathbb{E}[Y|\mathbf{X}]$.
- If not, we cannot know $g^\star$ and $\mathcal{R}^\star$ and we will use a n-sample, i.e. n i.i.d. copies of (X, Y) to retrieve information on those two quantities.

Scoring function

- Still in the setting of binary classification,
- instead of a classification rule $g : \mathbb{R}^d \to \{0, 1\}$, we want to find a function $S : \mathcal{X} \to \mathbb{R}$.

Definition

- Perfect score: S is perfect if there exists $s^\star$ such that

$$\mathbb{P}(Y = 1 | S(X) \geqslant s^\star) = 1 \quad \text{and} \quad \mathbb{P}(Y = 0 | S(X) < s^\star) = 1.$$

- Random score: S is random if $S(X)$ and Y are independent.

Scoring function I

Link between a score and a classification rule For a given score S and a threshold s, we obtain a classification rule:

$$g_s(x) = \begin{cases} 1 & \text{if} \quad S(x) \geqslant s, \\ 0 & \text{otherwise.} \end{cases}$$

	$g_s(X) = 0$	$g_s(X) = 1$
$Y = 0$	✓	✗
$Y = 1$	✗	✓

Therefore, for any threshold, we define two types of errors:

$$\alpha(s) := \mathbb{P}(g_s(X) = 1 | Y = 0) = \mathbb{P}(S(X) \geqslant s | Y = 0),$$
$$\beta(s) := \mathbb{P}(g_s(X) = 0 | Y = 1) = \mathbb{P}(S(X) < s | Y = 1).$$

One can also define the following related quantities

- the specificity: $sp(s) = \mathbb{P}(S(X) < s | Y = 0) = 1 - \alpha(s)$
- the sensibility: $se(s) = \mathbb{P}(S(X) \geqslant s | Y = 1) = 1 - \beta(s)$

The ROC curve I

We can measure the performance of a score by visualizing errors $\alpha(s)$ and $\beta(s)$ on a same graph for all threshold s.

Definition

The ROC curve of a score S is the parametrized curve $(x(s), y(s))$ defined by

$$\begin{cases} x(s) & = \alpha(s) = 1 - sp(s) \\ & = \mathbb{P}(g_s(X) = 1 | Y = 0) = \mathbb{P}(S(X) \geqslant s | Y = 0) \\ y(s) & = 1 - \beta(s) = se(s) \\ & = \mathbb{P}(g_s(X) = 1 | Y = 1) = \mathbb{P}(S(X) \geqslant s | Y = 1). \end{cases}$$

ROC stands for "receiver operating characteristic".

The ROC curve - example of a linear classifier

For a linear classifier, the score writes $S : x \mapsto \omega^\top x + b$ where $\omega \in \mathbb{R}^d$ and $b \in \mathbb{R}$ (see next lecture on discriminant analysis and logistic regression).

For each choice of the threshold s^*, $x(s^*) = \mathbb{P}(S(X) \geqslant s^* | Y = 0)$ and $y(s^*) = \mathbb{P}(S(X) \geqslant s^* | Y = 1)$ are unknown and replaced by the empirical false positive rate (FPR) and true positive rate (TPR).

$$\mathrm{FPR} = \frac{\sum_{i=1}^n \mathbb{1}_{S(X_i) \geqslant s^*; Y_i = 0}}{\sum_{i=1}^n \mathbb{1}_{Y_i = 0}},$$

$$\mathrm{TPR} = \frac{\sum_{i=1}^n \mathbb{1}_{S(X_i) \geqslant s^*; Y_i = 1}}{\sum_{i=1}^n \mathbb{1}_{Y_i = 1}}.$$

The ROC curve - example of a linear classifier

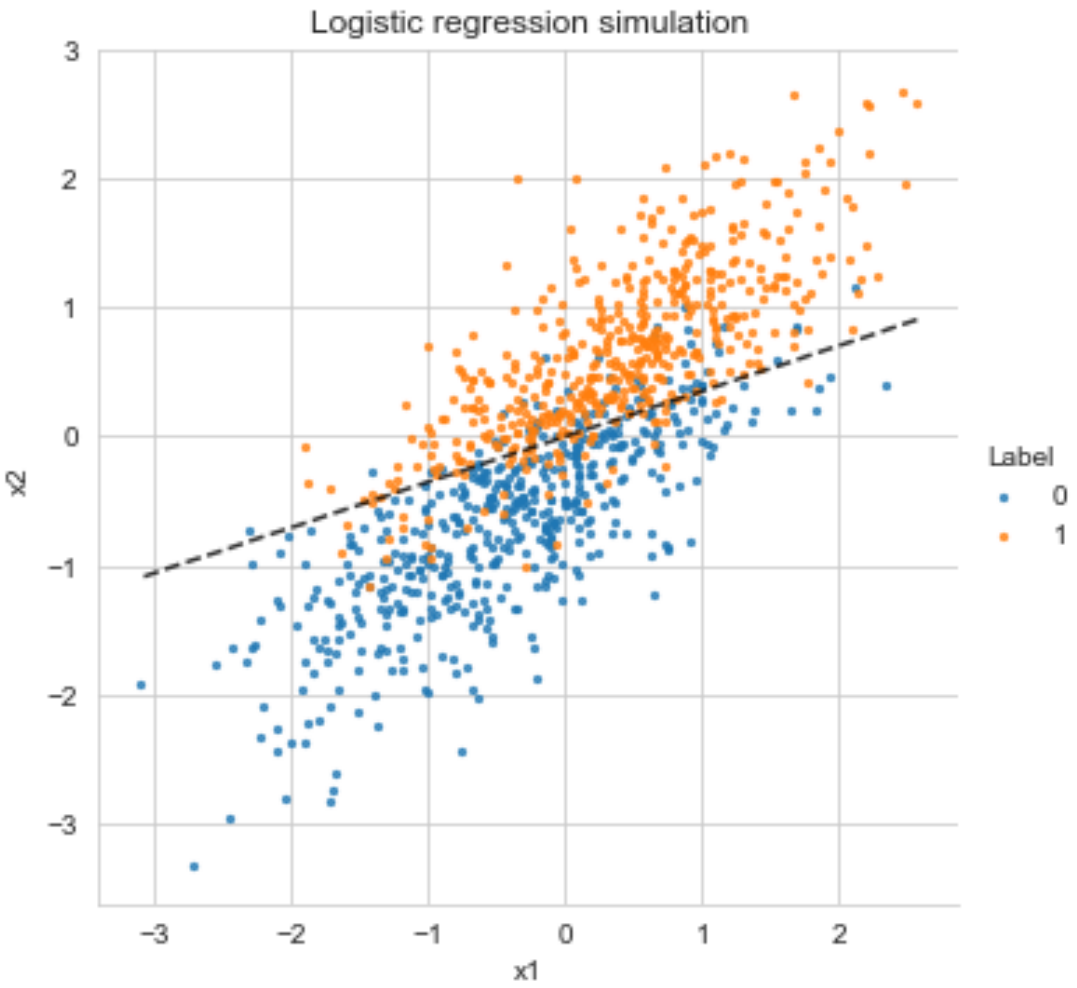


Figure: 1000 data points are randomly generated ($d = 2$) with Gaussian distribution. A Logistic regression model is used to label each data randomly.

The ROC curve - example of a linear classifier

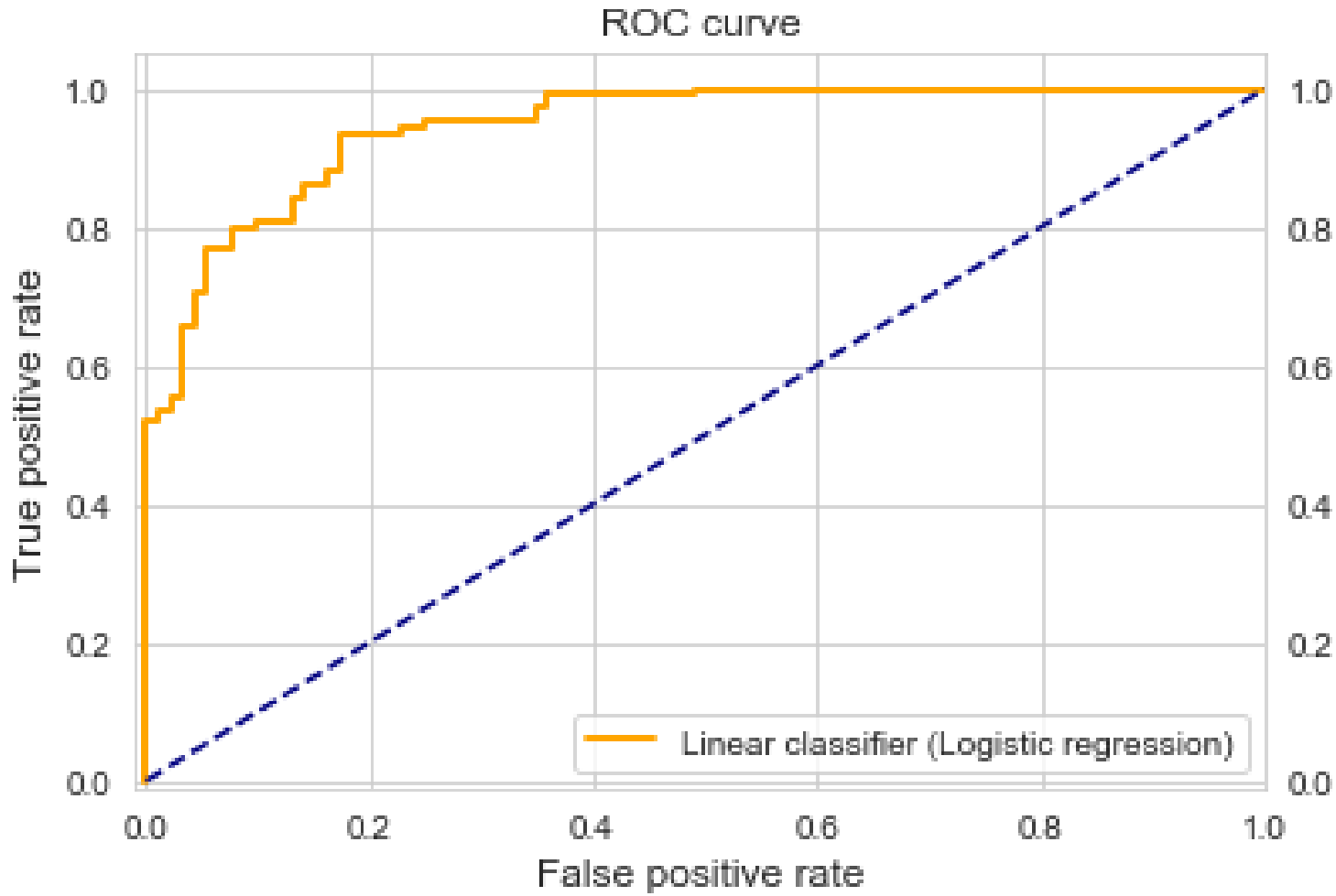


Figure: A ROC curve for a logistic regression classifier 200 additional data points not in the training set. Simulations are inspired by (Roc curve with scikit-learn) and can be found here https://sylvainlc.github.io/.

Area Under ROC I

The Area Under ROC (AUC) is often used to measure performance of a score S. Note that

- Perfect score: $AUC(S) = 1$.
- Random score: $AUC(S) = 1/2$.

Proposition

Let (X_1, Y_1) and (X_2, Y_2) be two i.i.d. copies of (X, Y). Then,

$$AUC(S) = \mathbb{P}\left(S(X_1) \geqslant S(X_2) | (Y_1, Y_2) = (1, 0)\right).$$

AUC(S) measures the probability that S correctly orders two observations with different labels.
AUC(S) can be seen as a cost function for a score S;

Area Under ROC II

Question: does there exist an optimal score $S^\star$ for this cost function?

Theorem (S Clémençon, G Lugosi, N Vayatis, 2008)

Let $S^\star(X) = \mathbb{P}(Y = 1|X)$, then for any score S we have

$$AUC(S^\star) \geqslant AUC(S).$$

- The distribution of (X,Y) is unknown, so we do not have access to $S^\star(x)$.
- We should find a good empirical estimate S_n of $S^\star(X) = \mathbb{P}(Y = 1|X)$.

Summary

	Cost function $\ell(y, f(x))$	Risk $\mathbb{E}[\ell(y, f(x))]$	Optimum $f^\star$
Regression	$(y - f(x))^2$	$\mathbb{E}\left[(Y - f(X))^2\right]$	$\mathbb{E}[Y\|X]$
Binary classif.	$\mathbb{1}_{y \neq f(x)}$	$\mathbb{P}(Y \neq f(X))$	Bayes rule
Scoring		$AUC(S)$	$\mathbb{P}(Y = 1\|X)$.

Summary

To the empirical risk

- Consider a n-sample $\mathcal{D}_n = (X_1, Y_1), \ldots, (X_n, Y_n)$, where (X_i, Y_i), $1 \leqslant i \leqslant n$, are i.i.d. copies of (X, Y).
- Let $\mathcal{C}$ be a class of potential classifiers.

Since the distribution of (X, Y) is generally unknown, the minimization of the risk is impossible in practice.

Goal

Therefore, given a cost function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$, we search a predictor $f_n(x) = f_n(x, D_n)$ "close" to the optimal $f^\star$ defined by

$$f^\star \in \operatorname{argmin} \mathcal{R}(f)\,,$$

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

The problem is then to find $f_n^\star \in \mathcal{C}$ such that $\mathcal{R}(f_n^\star) \simeq \inf_{f \in \mathcal{C}} \mathcal{R}(f)$.

Empirical risk minimization (ERM)

Empirical risk

Since the risk is an expectation, a first natural choice is to choose the one that minimizes its empirical version:

$$\widehat{\mathcal{R}}_n(f) = \frac{1}{n}\sum_{i=1}^{n} \ell(Y_i, f(X_i)).$$

A simple reformulation reads as follows

$$\widehat{\mathcal{R}}_n(f_n) - \mathcal{R}^\star = \underbrace{\left[\widehat{\mathcal{R}}_n(f_n) - \inf_{f\in\mathcal{C}} \mathcal{R}(f)\right]}_{\text{estimation error}} + \underbrace{\left[\inf_{f\in\mathcal{C}} \mathcal{R}(f) - \mathcal{R}^\star\right]}_{\text{approximation error}}$$

- The estimation error is random. It reflects the discrepancy in terms of $\mathbb{P}$, between the chosen classifier and the "local champion" in $\mathcal{C}$.
- The approximation error is deterministic and measures the closeness in terms of $\mathbb{P}$ between the class $\mathcal{C}$ and the optimal choice $f^\star$.

Some comments on Empirical risk minimization

$\mathcal{C}$ should be wide enough for the approximation error to be small.

$\mathcal{C}$ should not be too wide for the control of the estimation error.

Consider for instance that $\mathcal{C}$ is the set of all measurable functions from $\mathbb{R}^d$ to $\{0,1\}$. The approximation error is zero, but the estimation error can be large: think of the choice

$$f_n(x) = \begin{cases} Y_i, & \text{if } \; x = X_i \,, 1 \leqslant i \leqslant n \,, \\ 0 & \text{otherwise,} \end{cases}$$

for which the empirical risk is zero!
$\rightsquigarrow$ no generalization ability.
$\rightsquigarrow$ overfitting (to be continued).

Empirical risk minimization

Lemma

Define $f_n^\star \in \operatorname{argmin}_{f \in \mathcal{C}} \widehat{\mathcal{R}}_n(f)$. *Then,*

1.
$$\mathcal{R}(f_n^\star) - \inf_{f \in \mathcal{C}} \mathcal{R}(f) \leqslant 2 \sup_{f \in \mathcal{C}} \left| \widehat{\mathcal{R}}_n(f) - \mathcal{R}(f) \right|$$

and,

2.
$$\left| \widehat{\mathcal{R}}_n(f_n^\star) - \mathcal{R}(f_n^\star) \right| \leqslant \sup_{f \in \mathcal{C}} |\widehat{\mathcal{R}}_n(f) - \mathcal{R}(f)|.$$

Proof on blackboard

The objective is now to control $\sup_{f \in \mathcal{C}} |\widehat{\mathcal{R}}_n(f) - \mathcal{R}(f)|$.

Summary

The case of a finite class of binary classifiers

- $(X, Y) \in \mathbb{R}^d \times \{0, 1\}$.
- A natural choice for the cost function in this case is

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

- The risk can be then written for this loss function:

$$\mathcal{R}(f) = \mathbb{P}(Y \neq f(X)).$$

- The optimal choice for the classifier is the Bayes rule

$$g^\star(X) = \begin{cases} 1 & \text{if} \quad \mathbb{P}(Y = 1|X) > \mathbb{P}(Y = 0|X), \\ 0 & \text{otherwise.} \end{cases}$$

- The empirical risk is then

$$\widehat{\mathcal{R}}_n(f) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{f(X_i) \neq Y_i}.$$

ERM and a finite class

Objective: Provide a theoretical control of $\sup_{f \in \mathcal{C}} |\widehat{\mathcal{R}}_n(f) - \mathcal{R}(f)|$.

We need a uniform deviation of $\widehat{\mathcal{R}}_n(f)$ from its expectation $\mathcal{R}(f)$.

Given $f \in \mathcal{C}$,

$$n\widehat{\mathcal{R}}_n(f) = \sum_{i=1}^{n} \mathbb{1}_{f(X_i) \neq Y_i} ,$$

so that $n\widehat{\mathcal{R}}_n(f) \overset{\mathcal{L}}{\sim} \mathcal{B}(n, \mathcal{R}(f))$.

Therefore, we need a uniform control for binomial randm variables.

Theorem (Hoeffding's inequality)

Let $X_1, \ldots, X_n$ be independent real-valued random variables. Assume that each X_i takes its values in $[a_i, b_i]$ ($ai < bi$) with probability one. Then, for all $t > 0$,

$$\mathbb{P}\left(\sum_{i=1}^{n} X_i - \mathbb{E}\left[\sum_{i=1}^{n} X_i\right] \geqslant t\right) \leqslant \mathrm{e}^{-2t^2 / \sum_{i=1}^{n} (b_i - a_i)^2},$$

and

$$\mathbb{P}\left(\sum_{i=1}^{n} X_i - \mathbb{E}\left[\sum_{i=1}^{n} X_i\right] \leqslant -t\right) \leqslant \mathrm{e}^{-2t^2 / \sum_{i=1}^{n} (b_i - a_i)^2}.$$

In particular,

$$\mathbb{P}\left(\left|\sum_{i=1}^{n} X_i - \mathbb{E}\left[\sum_{i=1}^{n} X_i\right]\right| \geqslant t\right) \leqslant 2\mathrm{e}^{-2t^2 / \sum_{i=1}^{n} (b_i - a_i)^2}.$$

Proof of Hoeffding's inequality by using Chernoff's bounding method and the following lemma.

Lemma

Let X be a real-valued random variable with $\mathbb{E}[X] = 0$ and $X \in [a, b]$ $(a < b)$ with probability one. Then, for all $s > 0$,

$$\mathbb{E}\left[\mathrm{e}^{sX}\right] \leqslant \mathrm{e}^{s^2(b^2-a^2)/8}.$$

Proof on blackboard

3. Getting back to the deviation of binomial r.v.

Corollary

Let $X \sim \mathcal{B}(n,p)$, $n \geqslant 1, p \in [0,1]$. Then for all $t \geqslant 0$,

$$\mathbb{P}\left(|X - np| \geqslant t\right) \leqslant 2\mathrm{e}^{-2t^2/n}.$$

A union bound ($|\mathcal{C}| < \infty$) leads to the following result.

Theorem

Assume that $|\mathcal{C}|$ is finite, with $|\mathcal{C}| \leqslant N$. Then, for all $t > 0$,

$$\mathbb{P}\left(\sup_{f \in \mathcal{C}} \left|\widehat{\mathcal{R}}_n(f) - \mathcal{R}(f)\right| \geqslant t\right) \leqslant 2N\mathrm{e}^{-2nt^2}.$$

In the case where $|\mathcal{C}|$ is finite, with $|\mathcal{C}| \leqslant N$,

$$\mathbb{P}\left(\sup_{f\in\mathcal{C}}\left|\widehat{\mathcal{R}}_n(f)-\mathcal{R}(f)\right| \geqslant t\right) \leqslant 2Ne^{-2nt^2}.$$

- The bound is universal.
- Borel-Cantelli: $\sup_{f\in\mathcal{C}}|\widehat{\mathcal{R}}_n(f)-\mathcal{R}(f)| \to 0$ almost surely.
- Consequence: $\mathcal{R}(f_n^\star) - \inf_{f\in\mathcal{C}}\mathcal{R}(f) \to 0$ almost surely.
  $\Rightarrow$The estimation error tends to 0 almost surely
  meaning that learning is asymptotically optimal.
- Bound on $\mathbb{E}[\mathcal{R}(f_n^\star)] - \inf_{f\in\mathcal{C}}\mathcal{R}(f)$?

4. From probability to expectation

Lemma ($\mathbb{P}$ to $\mathbb{E}$)

Let Z be a random variable taking values in $\mathbb{R}_+$. Assume that there exists a constant $C \geqslant 1$ such that, for all $t > 0$, $\mathbb{P}(Z \geqslant t) \leqslant C\mathrm{e}^{-2nt^2}$. Then,

$$\mathbb{E}[Z] \leqslant \sqrt{\frac{\log(C\mathrm{e})}{2n}}.$$

Combined with the previous result, this yields

$$\mathbb{E}\left[\sup_{f\in\mathcal{C}}\left|\widehat{\mathcal{R}}_n(f) - \mathcal{R}(f)\right|\right] \leqslant \sqrt{\frac{log(2\mathrm{e}N)}{2n}},$$

and

$$\mathbb{E}[\mathcal{R}(f_n^\star)] - \inf_{f\in\mathcal{C}} \mathcal{R}(f) \leqslant 2\sqrt{\frac{log(2\mathrm{e}N)}{2n}}.$$

ERM with a finite class

In the case where $|\mathcal{C}|$ is finite, with $|\mathcal{C}| \leqslant N$,

$$\mathbb{E}\left[\mathcal{R}(f_n^\star)\right] - \inf_{f \in \mathcal{C}} \mathcal{R}(f) \leqslant 2\sqrt{\frac{log(2eN)}{2n}}.$$

Take-home message

For a finite class $\mathcal{C}$, such that $|\mathcal{C}| \leqslant N$

$$\text{Expectation of Estimation error} = O\left(\sqrt{\frac{log(N)}{n}}\right).$$

The next objective is to handle more complex classes of functions, that would be the purpose of next sessions

Additional results at [A few notes]

Summary

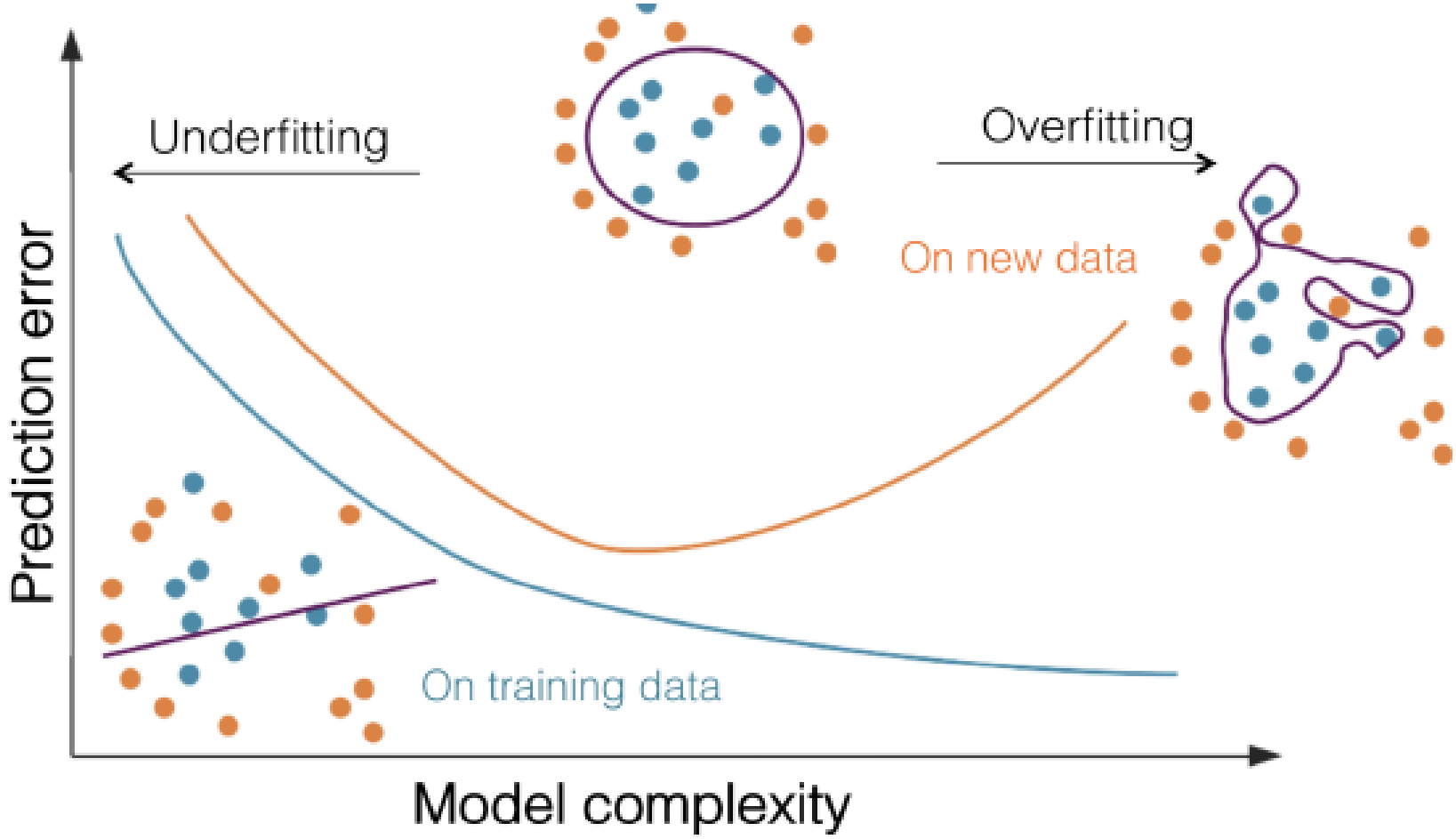
Model complexity

Most of statistical learning algorithms depends on parameters λ.
Some examples are:

- number of input variables in linear and logistic models,
- penalty parameters for lasso and ridge regressions,
- depth for tree algorithms,
- number of nearest neighbors,
- number of iterations for boosting algorithms,
- The choice of theses parameters reveals crucial for the performance of the machine...

Parameter λ often measures model complexity.

With a **fixed** sample size, varying the model complexity.



Bias/Variance

- Bias: difference between the expected value of the estimator (model) and the true value being estimated!

$$\text{Bias}(\hat{f}(x)) = \mathbb{E}\left[\hat{f}(x) - y\right]$$

  - A simpler model has a higher bias (naturally a simple model will do some errors)!
  - High bias can cause underfitting!
- Variance: deviation from the expected value of the estimates!

$$\text{Var}(\hat{f}(x)) = \mathbb{E}\left[\left(\hat{f}(x) - \mathbb{E}(\hat{f}(x))\right)^2\right]$$

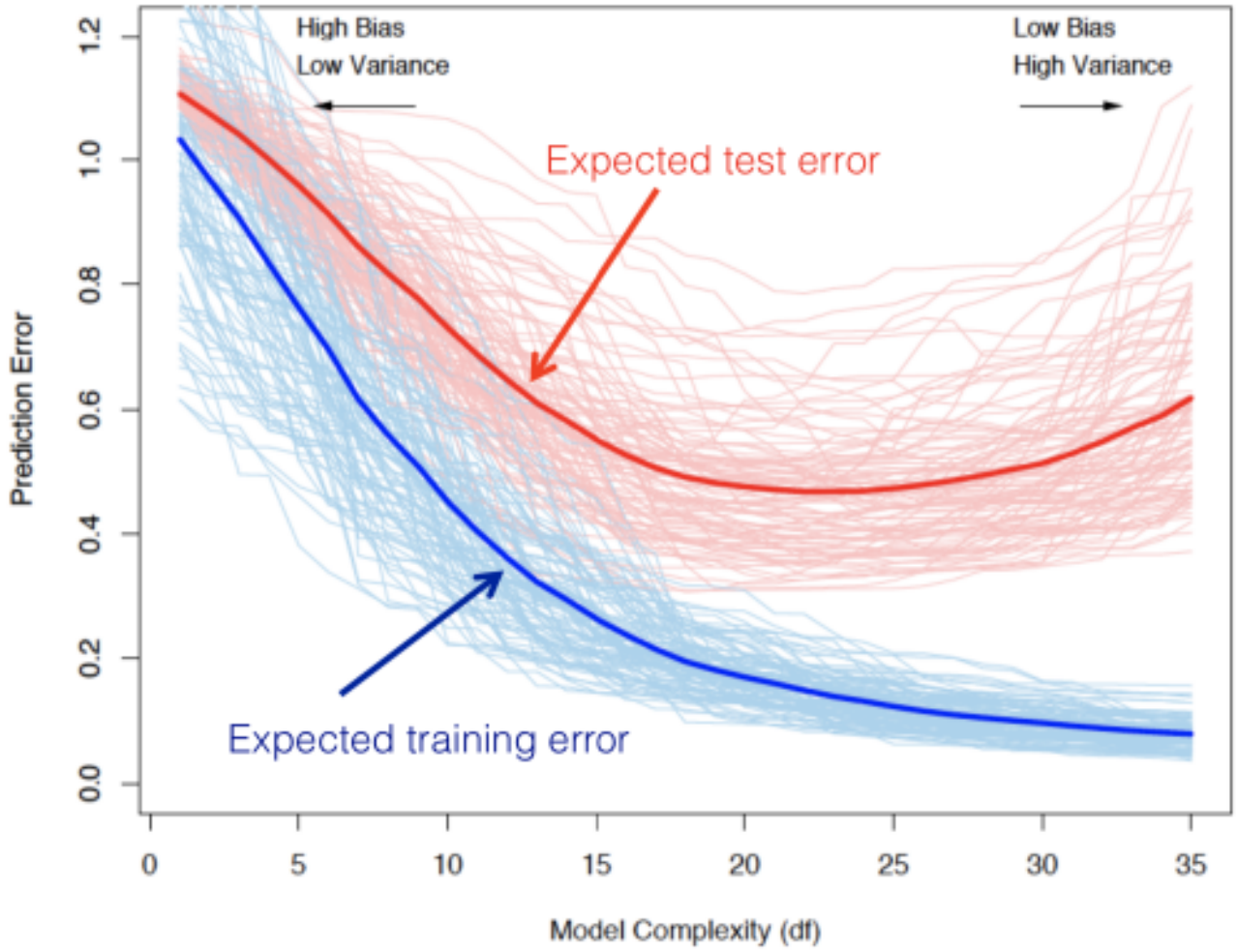
  - A more complex model has a higher variance!
  - High variance can cause overfitting!

Ideally we want to optimize both of them.

Bias-Variance tradeoff

- When do we have high bias?
  - We have high bias when the model (function) cannot model the true data distribution well!
  - This doesn't depend on the training data size!
  - Underfitting!
- When do we have high variance?
  - We have high variance when there is a small amount of training data and a very complex model!
  - Overfitting!
  - Variance decreases with larger training data, and increases with more complicated classifiers!

Bias/Variance tradeoff



Take-home message

- High bias $\Longrightarrow$ high training and test errors!
- High variance $\Longrightarrow$ low training error, high test errors!

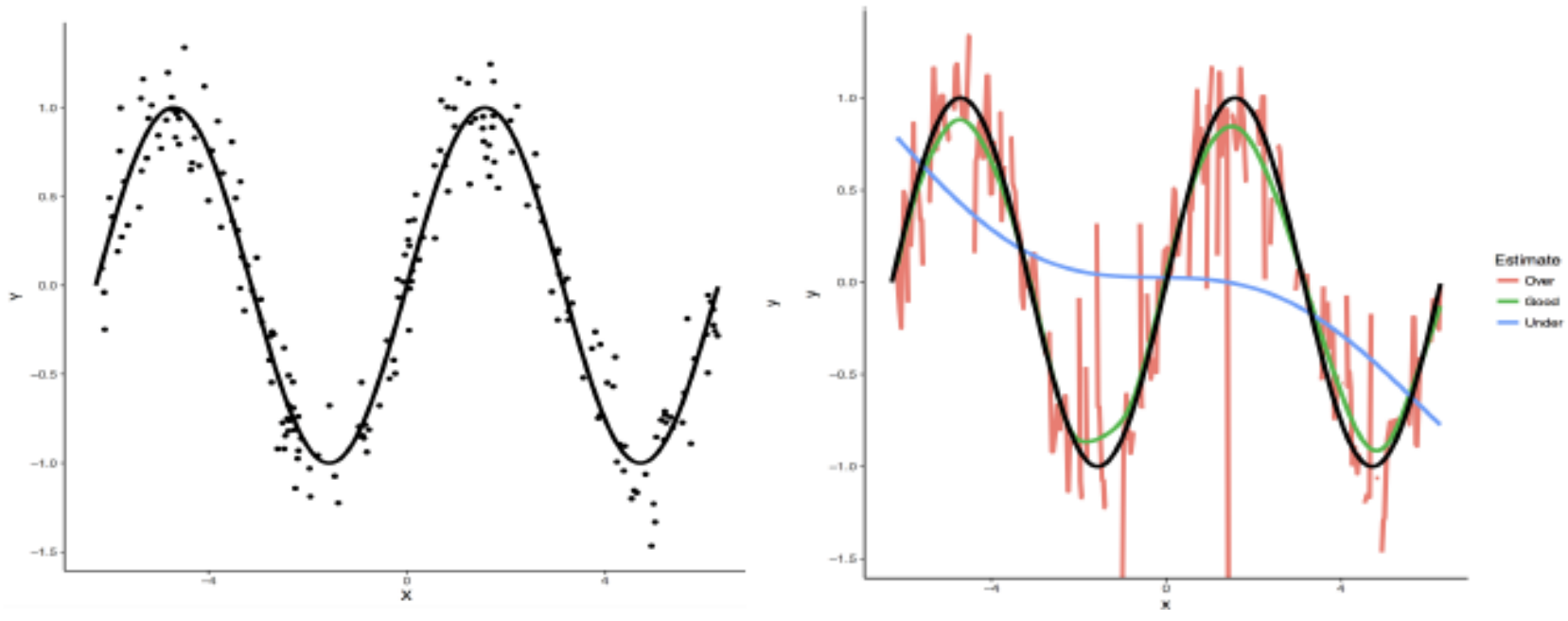


Figure: Illustration of overfitting in regression. Here, λ controls the regressor smoothness.

Summary

Generalization

Goal of supervised learning

- A trained classifier has to be generalizable: it must be able to work on other data than the training dataset
- Generalizable means also "works without overfitting".

- The empirical error on the training set is a poor estimate of the generalization error (expected error on new data)!
  $\rightsquigarrow$ If the model is overfitting, the generalization error can be arbitrarily large!
- We would like to estimate the generalization error on new data, which we do not have!

Validation sets

- Choose the model that performs best on a validation set separate from the training set!

- Because we have not used the validation data at any point during training, the validation set can be considered "new data" and the error on the validation set is an estimation of the generalization error!

Validation hold out

The simplest approach consists in splitting the data $\mathcal{D}_n$ into:

1. a learning or training set $\mathcal{D}_{n,train}$ used to learn the machine f_n,
2. a validation or test set $\mathcal{D}_{n,test}$ used to estimate the risk of f_n.

Algorithm 1 Validation hold out

Inputs: $\mathcal{D}_n$ data, $(\mathcal{T}, \mathcal{V})$ a partition of $\{1, \ldots, n\}$.

1. Learn the machine with
   $\mathcal{D}_{n,train} = \{(X_i, Y_i), i \in \mathcal{T}\} \Longrightarrow f_{n,train}$;
2. Compute

$$\widehat{\mathcal{R}}_n(f_{n,train}) = \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} \ell\left(Y_i, f_{n,train}(X_i)\right).$$

Remark

$n_{train} = |\mathcal{T}|$ and n_{test} should be large enough.

Model selection

- What if we want to choose among k models?!
  - Train each model on the train set!
  - Compute the prediction error of each model on the validation set!
  - Pick the model with the smallest prediction error on the validation set!
- What is the generalization error?!
  - We don't know!!
  - Validation data was used to select the model!
  - We have "cheated" and looked at the validation data: it is not a good proxy for new, unseen data any more!

Model selection

- Hence we need to set aside part of the data, the test set, that remains untouched during the entire procedure and on which we'll estimate the generalization error!

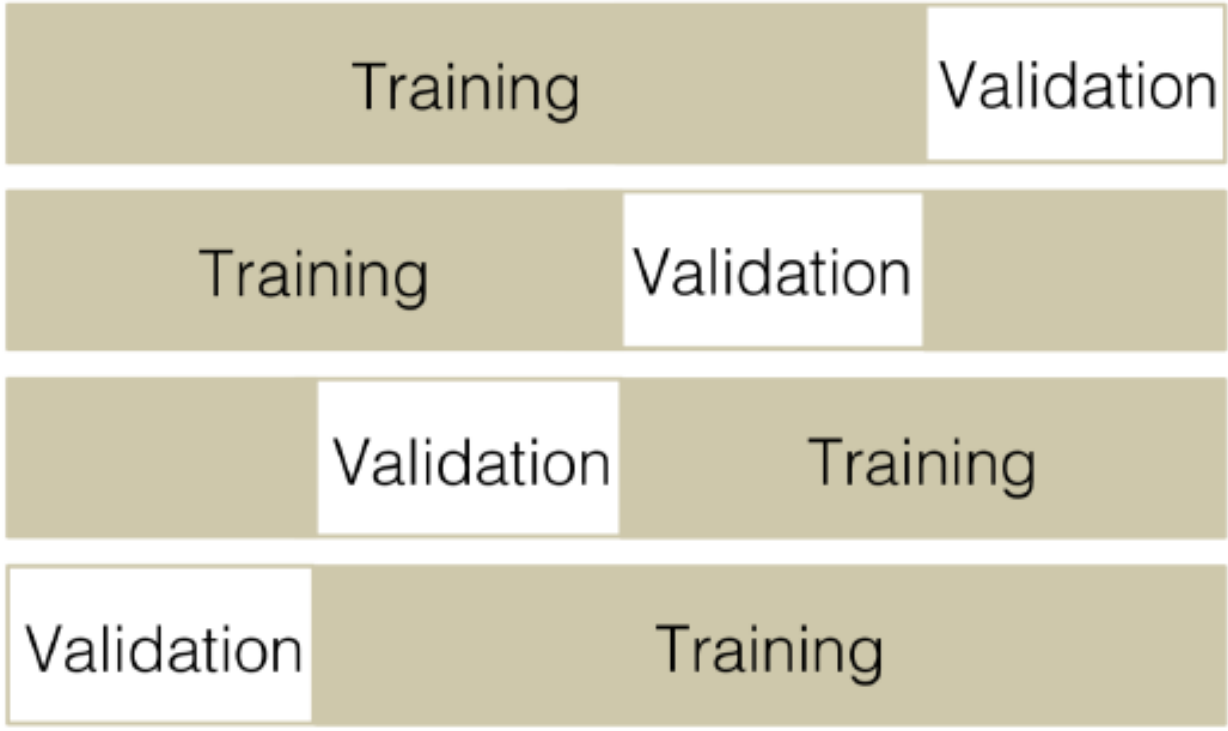
- Model selection: pick the best model!
- Model assessment: estimate its prediction error on new data!

Validation sets

- How much data should go in each of the training, validation and test sets?!
- How do we know that we have enough data to evaluate the prediction and generalization errors?!
- Empirical evaluation with sample re-use!
  - Cross-validation
  - Bootstrap (random sampling with replacement)

Cross-validation

- Cut the training set in K separate folds!
- For each fold, train on the $(K-1)$ remaining folds!



K-fold cross validation

Algorithm 2 K-fold CV

Inputs: $\mathcal{D}_n$ data, K an integer;

1. Define a random partition $\{\mathcal{I}_1, \ldots, \mathcal{I}_K\}$ of $\{1, \ldots, n\}$;
2. For a fixed λ, for $k = 1, \ldots, K$
   - $\mathcal{I}_{train} = \{1, \ldots, n\} \setminus \mathcal{I}_k$ and $\mathcal{I}_{test} = \mathcal{I}_k$;
   - Learn the machine with $\mathcal{D}_{n,train} = \{(X_i, Y_i), i \in \mathcal{T}\} \Longrightarrow f_{n,k}^{(\lambda)}$;
   - Compute the test error
     $\mathrm{Err}_{test}(f_{n,k}^{(\lambda)}) = \frac{1}{n-|\mathcal{I}_k|} \sum_{i \in \mathcal{I}_k} \ell\left(Y_i, f_{n,k}^{(\lambda)}(X_i)\right)$.
3. Choose

$$\hat{\lambda}^{(CV)} \in \mathrm{argmin}_{\lambda \in \Lambda} \frac{1}{K} \sum_{k=1}^{K} \mathrm{Err}_{test}(f_{n,k}^{(\lambda)}).$$

K-fold CV

- K has to be chosen by the user. Usually $K = 5$ or 10.
- Advantage of this method over repeated random sub-sampling (bootstrap) is that all observations are used for both training and validation, and each observation is used for validation exactly once.

Leave-one-out CV

- When $K = n$, we obtain the leave-one-out (LOO) cross validation, since at each iteration exactly one instance is left out of the training sample.
- In general, the leave-one-out error is very costly to compute, since it requires training n times on samples of size $n - 1$, but for some algorithms it admits a very efficient computation.
- Exercise: LOO-CV in least squares regression