Machine learning

Claire Boyer

September 9th

Hi!

- Prerequisites: basics of probability and statistics.
- Objectives:
  - Learn and apply the fundamental concepts of statistical learning;
  - Understand the basic theory underlying data science and IA;
  - Be able to read current research books and papers.

Contact information

- 15-25 office 220
- `claire.boyer@sorbonne-universite.fr`

- I will put some material on Moodle

Modality

- 4h of lecture/TD/TP per week
- Evaluation by
  1/4 QCM en cours + 1/4 project + 1/2 exam
- You need to install the following on your laptop

- Easy way to do it (not tensorflow): install Anaconda.

To get a freshener in Python

- You can visit Maxime Sangnier's webpage (LPSM)
- If you have some installation problem, do NOT contact him !

- Instead google your problem.

Thank you to...

... the following people helps me directly or indirectly to make those lectures

Statisticial learning

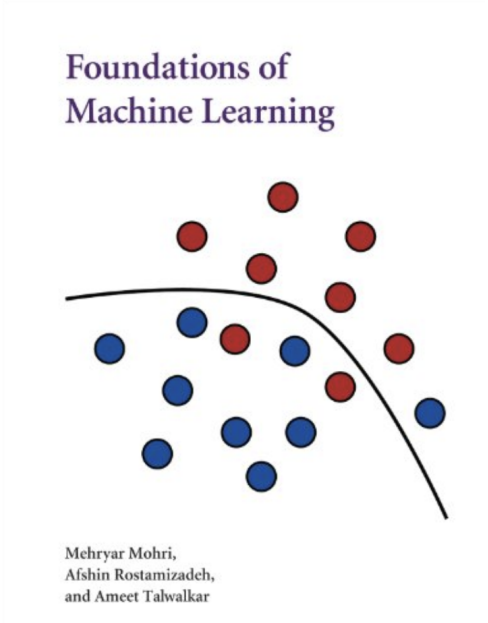
- Gérard Biau,
- Erwan Scornet,
- Laurent Rouvière,
- Pierre Gaillard,

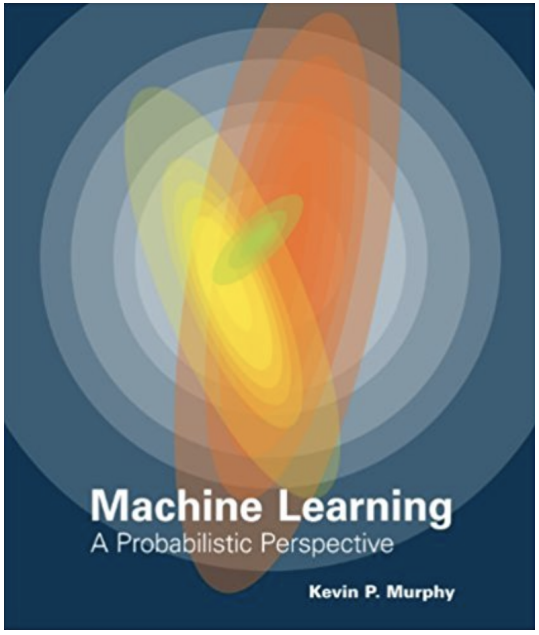
Machine learning

- Maxime Sangnier,
- Stéphane Gaïffas, ...

A few references

- Foundations of Machine Learning. M. Mohri, A. Rostamizadeh and A. Talwalkar, MIT Press
- Machine Learning, K.M. Murphy, MIT Press

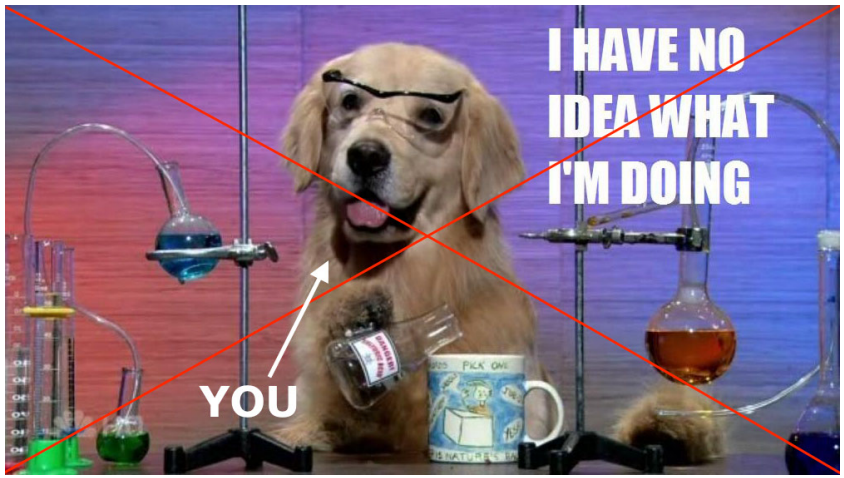




and other references for specific chapters to come.

This lecture

- Bridge between theory and practice
  - No time to go deep in theory
  - Not enough teachers to go deep in practice
- ML theory is challenging
- But necessary not to do bull**it in practice



Today

- Slides
- Blackboard time
- Your first notebook

For next time

Need for people to present stuff not covered by this course

Today

1. Some applications
2. Learning scenarii
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

Introduction

Learning algorithms have been successfully deployed in

- Text or document classification, e.g., spam detection;
- Natural language processing, e.g., morphological analysis, part-of-speech tagging, named-entity recognition;
- Speech recognition, speech synthesis, speaker verification;
- Optical character recognition (OCR);
- Computational biology applications, e.g., protein function or structured prediction;
- Computer vision tasks, e.g., image recognition, face detection;
- Fraud detection (credit card, telephone) and network intrusion;
- Games, e.g., chess, backgammon;
- Unassisted vehicle control (robots, navigation);
- Medical diagnosis;
- Recommendation systems, search engines, information extraction systems.

Motivating examples: Handwritten digit recognition

from MNIST dataset

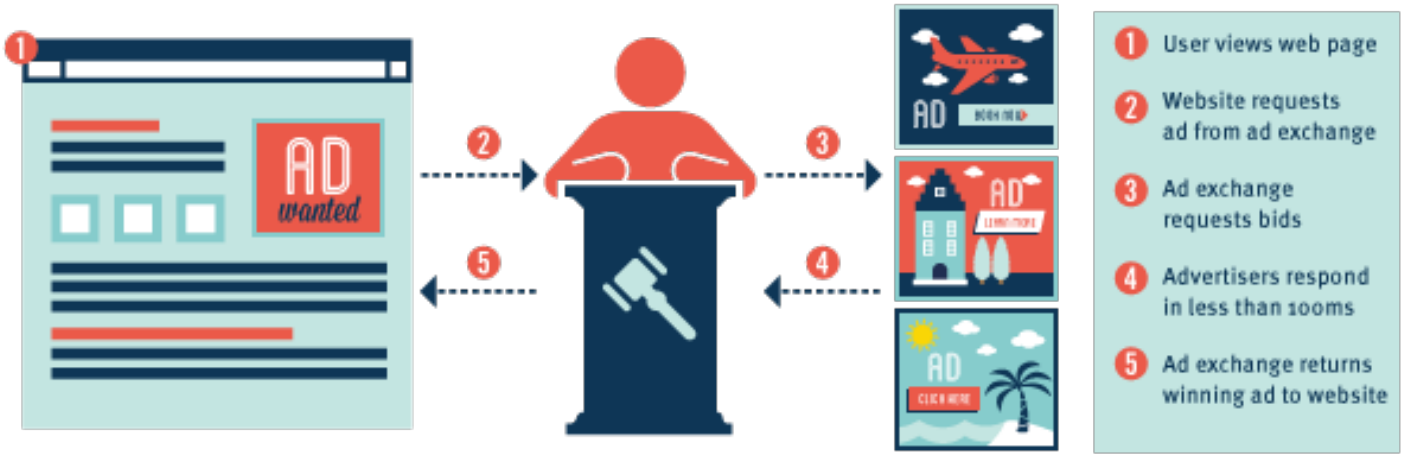
Can you tell which digit is it?

A noble cause: Real-time bidding I

a.k.a. "enchère en temps réel"

- A customer visits a webpage with his browser: a complex process of content selection and delivery begins.
- An advertiser might want to display an ad on the webpage where the user is going. The webpage belongs to a publisher.
- The publisher sells ad space to advertisers who want to reach customers.

In some cases, an auction starts: RTB (Real Time Bidding)



A noble cause: Real-time bidding II

- Advertisers have 10ms (!) to give a price: they need to assess quickly how willing they are to display the ad to this customer
- Machine learning is used here to predict the probability of click on the ad.
- Time constraint: few model parameters to answer quickly
  Feature selection / dimension reduction is crucial here!

  Full process takes < 100ms

- Need for **efficient algorithms**

A noble cause: Real-time bidding III

Some figures:

- 10 million prediction of click probability per second answers within 10ms
- stores 20Terabytes of data daily

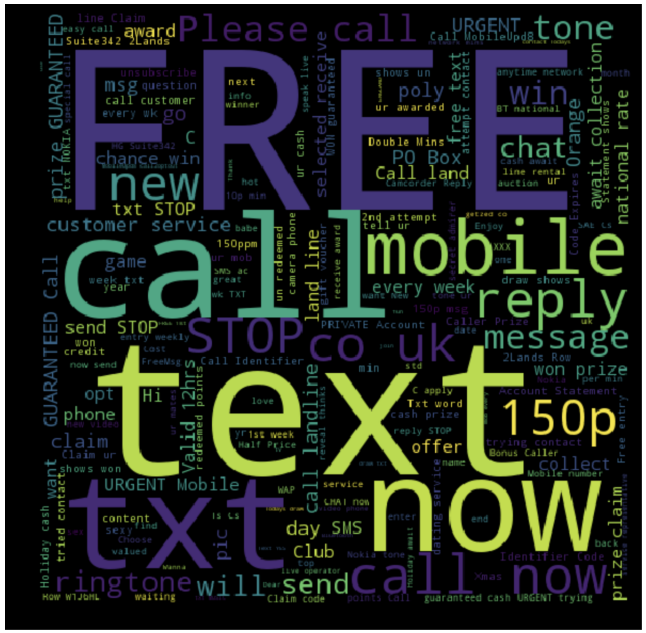
Aim

- Based on past data, you want to find users that will click on some ads
- This problem can be formulated as a binary classification problem

Spam detection

1

- Given a dataset, one can list the most repeated words in the spam messages or punctuation signs.
- Goal: predict if a new message is a spam from its words/punctuation signs.

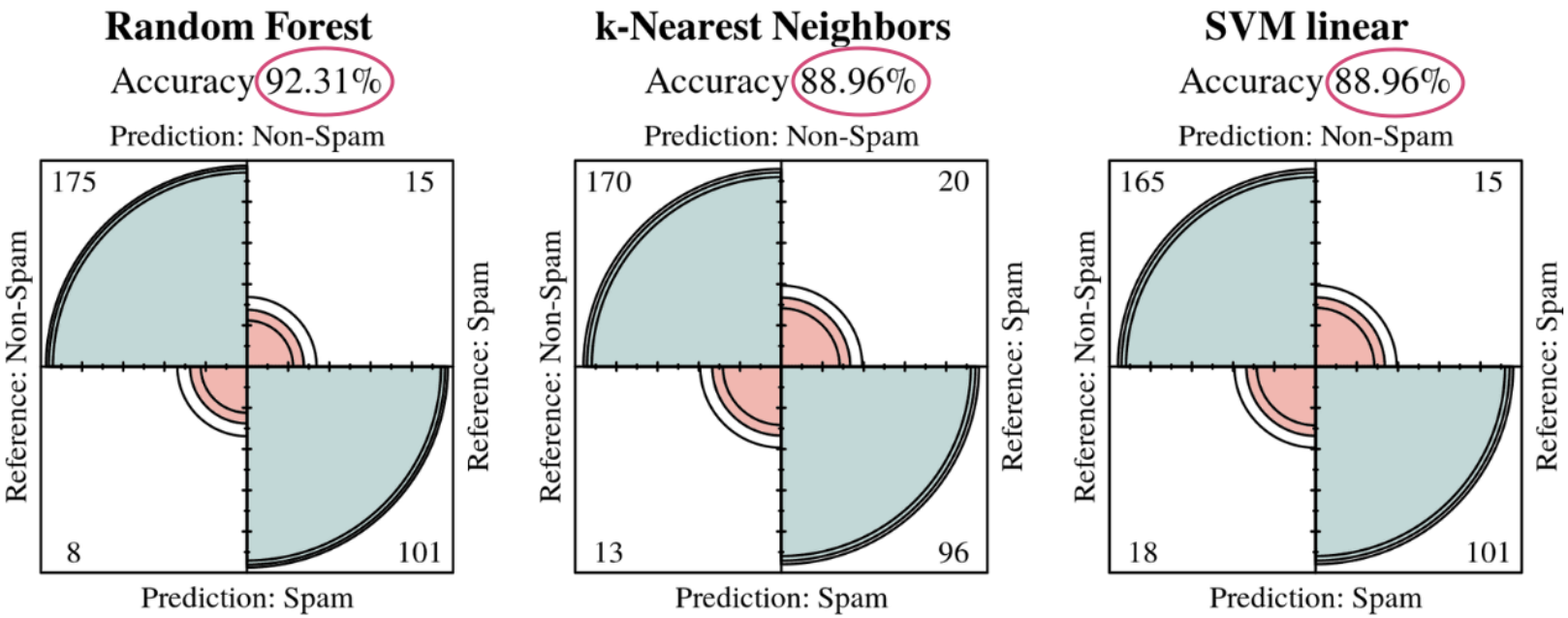


Spam wordcloud

Ham wordcloud

[1] from here

Spam detection

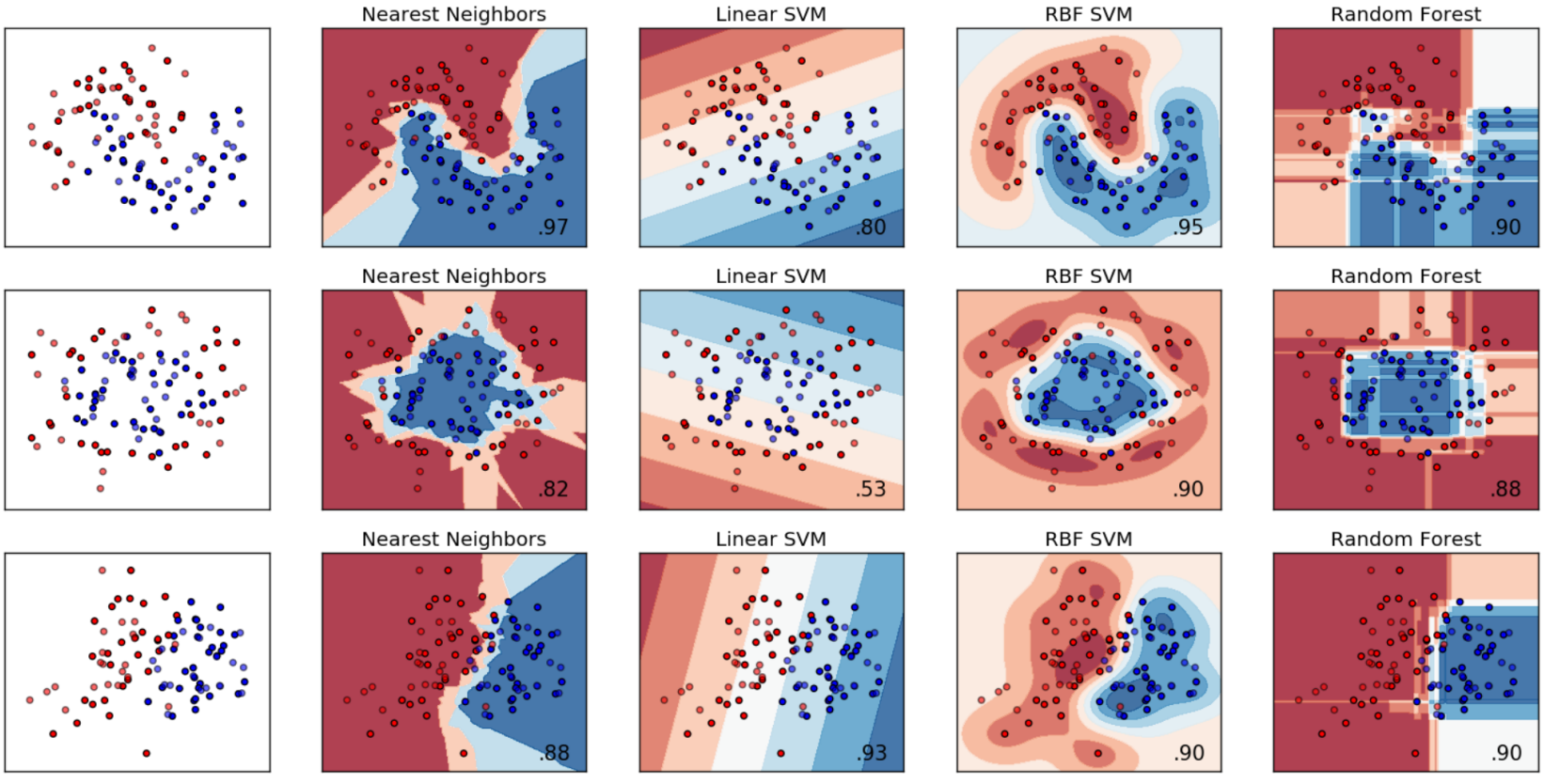


Accuracy: percentage of all emails that are correctly categorised

[2] from here

More generally, different ways to separate points

Examples in supervised learning



Summary

Typical machine learning problems

Stat/machine learning

Understand and learn a behavior from examples.

- **Classification**: Assign a category to each item.
- **Regression**: Predict a real value for each item. e.g. prediction of stock value
- **Ranking**: Order items according to some criterion.
- **Clustering**: Partition items into homogeneous regions.
- **Dimensionality reduction or manifold learning**: Transform an initial representation of items into a lower-dimensional representation of these items while preserving some properties of the initial representation.

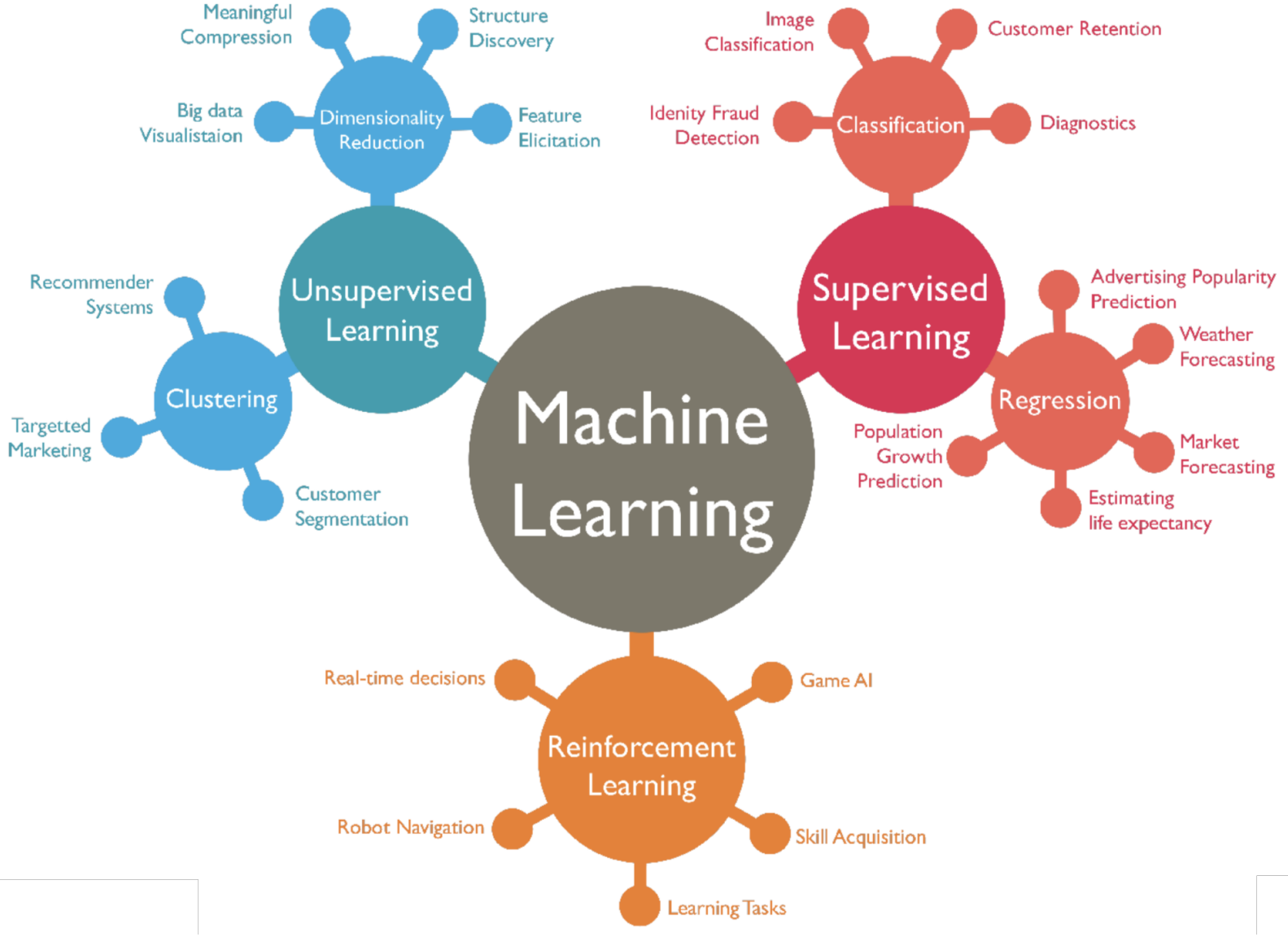
Learning scenarii I

The two main scenarii are supervised and unsupervised learning, which description follows:

- **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...


Machine Learning
Unsupervised Learning
Dimensionality Reduction
Meaningful Compression
Structure Discovery
Big data Visualistaion
Feature Elicitation
Clustering
Recommender Systems
Targetted Marketing
Customer Segmentation
Supervised Learning
Classification
Image Classification
Customer Retention
Idenity Fraud Detection
Diagnostics
Regression
Advertising Popularity Prediction
Weather Forecasting
Market Forecasting
Estimating life expectancy
Population Growth Prediction
Reinforcement Learning
Real-time decisions
Game AI
Robot Navigation
Skill Acquisition
Learning Tasks

MACHINE LEARNING

SUPERVISED LEARNING

(X_i, Y_i)

↙ ↘

Y_i : quantitative — Y_i : categorical "labels"

REGRESSION

Finding h such that $h(x) \simeq \mathbb{E}(Y|X=x)$

CLASSIFICATION

$h(x) > 0$ if $\mathbb{P}(Y=-1|X=x) > \mathbb{P}(Y=1|X=x)$

PARAMETRIC METHODS

- GLM (regression / logistic regression)
- TREES - FORESTS
- MODEL AGGREGATION
  Boosting
- NEURAL NETWORKS

NON-PARAMETRIC METHODS

- kNN
- SVM
- ...

UNSUPERVISED LEARNING

(X_i)

Guessing the communities groups.

CLUSTERING : class prediction up to permutation

↳ if model stat : EM ALGO
↳ if not : SPECTRAL clustering.
HIERARCHICAL

DIMENSION REDUCTION

linear : PCA, Lindenstrauss
non-linear :

SCORING

COMPLETION

Machine learning vs. statistics

Machine Learning

- Weights
- Learning
- Generalization
- Supervised learning
- Unsupervised learning
- Large grant: $1,000,000$

Statistics

- Parameters
- Fitting
- Test set performance
- Regression/classification
- Density estimation, clustering
- Large grant: $50,000$

Summary

Mathematical framework

Given observations, $d_n = \{(x_1, y_1), (x_2, y_2), \dots (x_n, y_n)\}$ we want to explain/predict outputs $y_i \in \mathcal{Y}$ from inputs $x_i \in \mathcal{X}$.

Goal

- Explain/Learn connections between inputs x_i and outputs y_i ;
- Predict the output y for a new input $x \in \mathcal{X}$

To do so, we have to find a machine or function $f : \mathcal{X} \to \mathcal{Y}$ such that

$$f(x_i) \simeq y_i, i = 1, \dots, n.$$

Jargon

- When the output y is continuous, $\rightsquigarrow$ regression problem
- When the output y is categorical, $\rightsquigarrow$ classification problem

Mathematical framework

- Need a criterion to measure the performance of a given machine f.
- We often use a cost function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^+$ 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 the cost (error) between the prevision/estimation y' and the observation y.

Risk

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)'s are i.i.d. copies of (X, Y) taking value in $\mathcal{X} \times \mathcal{Y}$.
- For a given cost function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^+$, we can measure the global (for all possible values of X and Y) performance of a machine $f : \mathcal{X} \to \mathcal{Y}$ by $\ell(Y, f(X))$. But this quantity is random! So quite difficult to minimize.

Risk of a machine

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

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

THE problem in Machine Learning

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

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

Such a function $f^\star$ (if it exists) is called the optimal machine for the cost function ℓ. Define

$$\mathcal{R}^\star := \mathcal{R}(f^\star).$$

In practice...

- The optimal machine $f^\star$ generally depends on the unknown probability distribution of (X, Y), then $f^\star$ is unknown in practice.
- Using $\mathcal{D}_n$, statistician's job consists in finding a good estimate f_n of $f^\star$, i.e. finding f_n such that $\mathcal{R}(f_n) \simeq \mathcal{R}(f^\star)$.

Universal consistency

Definition

We said that the estimate $(f_n)_n$ is universally consistent if for all distribution of (X, Y):

$$\lim_{n \to +\infty} \mathcal{R}(f_n) = \mathcal{R}(f^\star),$$

and strongly consistent if

$$\mathcal{R}(f_n) \to \mathcal{R}(f^\star) \quad a.s.$$

Exercise: Show that

$$\text{Consistency} \Leftrightarrow \mathcal{R}(f_n) \xrightarrow{L^1} \mathcal{R}^\star \Leftrightarrow \mathcal{R}(f_n) \xrightarrow{\mathbb{P}} \mathcal{R}^\star.$$

Hint: the first equivalence is given since $\mathcal{R}(g_n) \geqslant \mathcal{R}^\star$.

Choice of a cost function ℓ?

- The proposed mathematical framework implies that a machine is performant with respect to a criterion (represented by the cost function ℓ).
- It means that a machine 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 machine or regression function $m^\star$ for the quadratic risk is ???

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

Indeed, for all m one has,

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

- The problem is that $m^\star$ is generally unknown, so we have to find an estimate $m_n(x)$ of $m(x)$ such that $m_n(x) \simeq m^\star(x)$.
- Therefore, m_n will be universally consistant if

$$\lim_{n \to +\infty} \mathcal{R}(m_n) = \mathcal{R}(m).$$

Binary classification

- Setting: the output can only take 2 values ($Y \in \{0, 1\}$).
- Note that the distribution of (X,Y) is entirely characterized by (μ_X, r) with μ the marginal distribution of X and r is the regression function of Y on X. More precisely, for all $A \in \mathscr{B}(\mathbb{R}^d)$, $\mu_X(A) = \mathbb{P}(X \in A)$, and

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

  where the last equality comes from $Y \in \{0, 1\}$.
- There is a classification error (or misclassification) as soon as $g(X) \neq Y$

The error probability or the risk for a classification rule

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

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

Does an optimum exist?

Yes, it is called the the Bayes rule $g^\star$

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

the equality favoring 0 by convention. 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

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

Exercise: Prove it.

The Bayes risk

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

Exercise: Show that

1. $\mathcal{R}^\star = 1 - \mathbb{E}\left[\mathbb{1}_{r(X)>1/2} r(X) + \mathbb{1}_{r(X)\leqslant 1/2}(1 - r(X))\right]$,
2. $\mathcal{R}^\star = \mathbb{E}\left[\min(r(X), 1 - r(X))\right] = \frac{1}{2} - \frac{1}{2}\mathbb{E}\left|2r(X) - 1\right|$,
3. $\mathcal{R}^\star = 0 \iff Y = \varphi(X)$ with probability one.

Problem

$g^\star$ depends on the distribution of (X, Y).

- If it is known, the job is done.
- 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 $S : \mathcal{X} \to \mathbb{R}$ such that

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

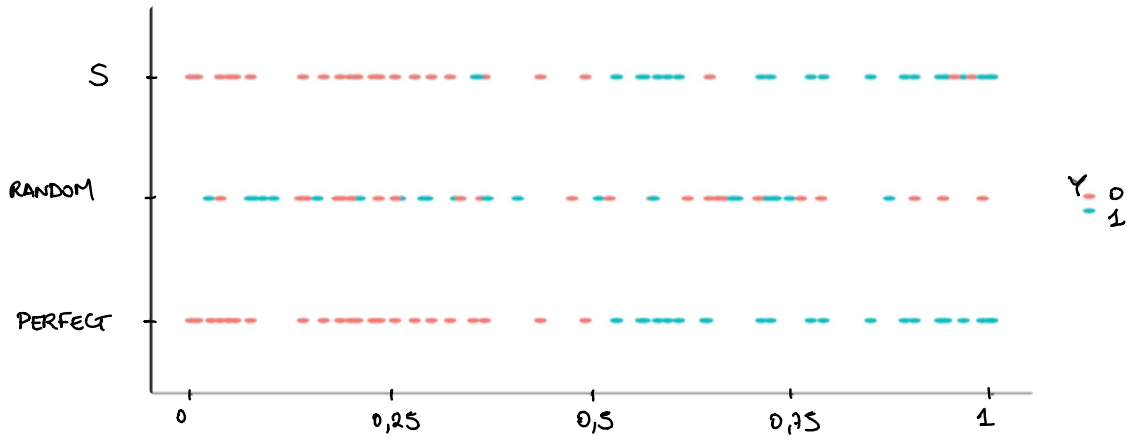


Figure: Illustration of different scores

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

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

Scoring function II

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

Therefore, for any threshold, one can 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

- Idea: 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 II

Remark

- *For any score S:* $x(-\infty) = y(-\infty) = 1$ *and* $x(+\infty) = y(+\infty) = 0$.
- *For a perfect score:* $x(s^\star) = 0$ *and* $y(s^\star) = 1$.
- *For a random score:* $x(s) = y(s) \forall s$.

The ROC curve III

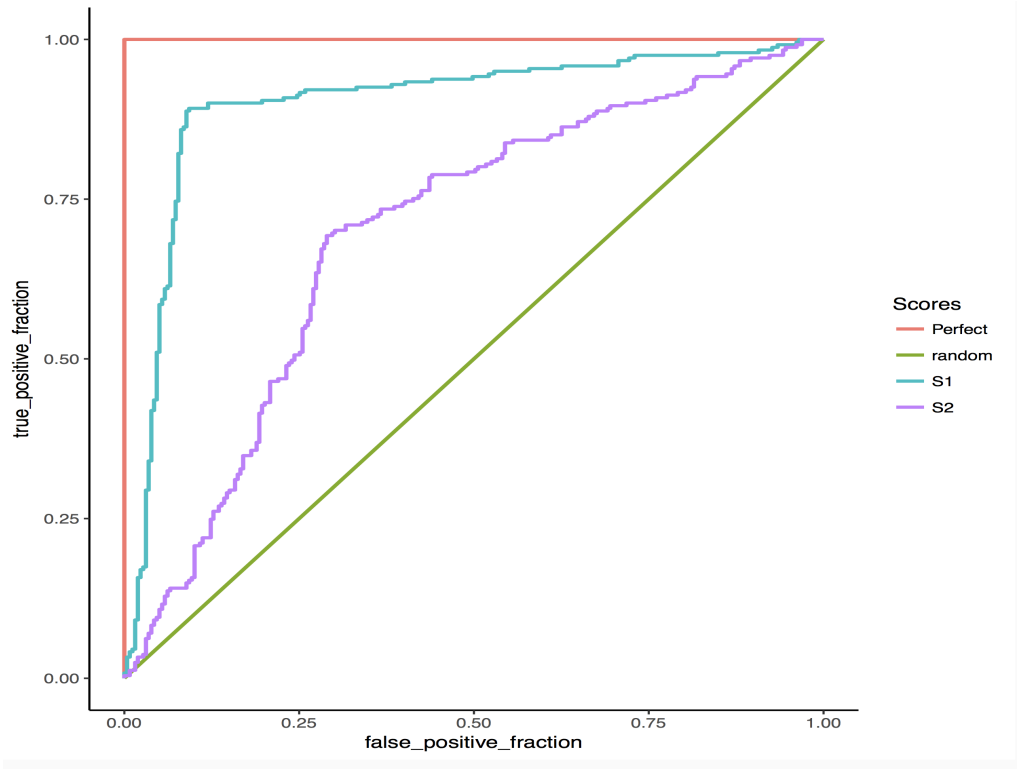


Figure: We measure performance of a score by its ability to approach the line $y = 1$ as fast as possible.

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.

Area Under ROC II

Computations of AUC corresponding to the curves in Figure 2 leads to the following:

```
 > df1 %>% group_by(Scores) %>% summarize(auc(D,M))
1  Perfect    1.0000000
2  random   0.5000000
3      S1       0.8999824
4      S2       0.6957177
```

AUC(S) can be seen as a cost function for a score$S$;
Question: does there exist an optimal score $S^\star$ for this cost function?

Area Under ROC III

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

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

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

As previously, the distribution of (X,Y) being generally unknown, one cannot have access to $S^\star(x)$ and then should find a good estimate S_n of $S^\star(x) = \mathbb{P}(Y = 1|X = x)$. Here is a summary of this section.

Summary

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

Summary

To the empirical risk

We are given

- an n-sample $\mathcal{D}_n = (X_1, Y_1), \ldots, (X_n, Y_n)$, where the (X_i, Y_i)'s are i.i.d. copies of (X, Y),
- a class of potential classifiers $\mathcal{C}$.

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 machine $f_n(x) = f_n(x, D_n)$ closed to the optimal machine $f^\star$ 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

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

- The estimation error is random, and 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 ERM

- $\mathcal{C}$ 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.

e.g. consider 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)

Lemma

The following holds

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: blackboard

Controlling the quantity $\sup_{f \in \mathcal{C}} |\widehat{\mathcal{R}}_n(f) - \mathcal{R}(f)|$ allows to

(i) the sub-optimality of $f_n^\star$ in $\mathcal{C}$

(ii) the error $|\widehat{\mathcal{R}}_n(f_n^\star) - \mathcal{R}(f_n^\star)|$ that we make by estimating the true risk $\mathcal{R}(f_n^\star)$ of the chosen machine with its empirical version.

So next, for the ERM

Focus on

$$\sup_{f \in \mathcal{C}} |\widehat{\mathcal{R}}_n(f) - \mathcal{R}(f)|.$$

Remark

Since $\mathcal{D}_n$ is used to construct the machine f_n, the law of large numbers (LLN) does not apply. In general, $\widehat{\mathcal{R}}_n(f_n)$ underestimates $\mathcal{R}(f_n)$. In practice, one solution to this problem can be the cross-validation or bootstrap approaches.

Summary

The case of a finite class

- Setting: binary classification
- Recall that 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

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

- 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

Having the previous Lemma in mind, we want to control:

$$\sup_{f \in \mathcal{C}} |\widehat{\mathcal{R}}_n(f) - \mathcal{R}(f)|.$$

Therefore, one needs uniform deviation of $\widehat{\mathcal{R}}_n(f)$ from its expectation $\mathcal{R}(f)$.

1. **Preliminary** Given $f \in \mathcal{C}$, what is the distribution of $n\widehat{\mathcal{R}}_n(f)$?

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

Therefore, we need uniform deviations of binomial r.v. from their expectations.

2. Tools for deviation

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 $t > 0$,

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

and

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

In particular,

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

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 $s > 0$,

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

Proof of Hoeffding's inequality:

$$\mathbb{P}\left(\sum_{i=1}^{n} X_i - \mathbb{E}\sum_{i=1}^{n} X_i \geqslant t\right) \overset{Markov}{\leqslant} \frac{\mathbb{E}e^{s\sum_i X_i}}{e^{st}} \overset{Lemma11}{\leqslant} e^{st}\prod_{i=1}^{n} e^{s^2(b_i^2-a_i^2)}$$

$$= e^{st}e^{s^2\sum_i(b_i^2-a_i^2)/8},$$

$$= e^{-2t^2/\sum_{i=1}^{n}(b_i-a_i)^2},$$

by choosing $s = 4t/\sum_i(b_i - a_i)^2$.

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 $t \geqslant 0$,

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

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 $t > 0$,

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

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

- 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 $t > 0$,
$\mathbb{P}(Z \geqslant t) \leqslant Ce^{-2nt^2}$. *Then,*

$$\mathbb{E}Z \leqslant \sqrt{\frac{\log(Ce)}{2n}}.$$

Lemma $\mathbb{P}$ to $\mathbb{E}$ and the previous thm lead to

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

and

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

ERM with a finite class

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

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

Summary

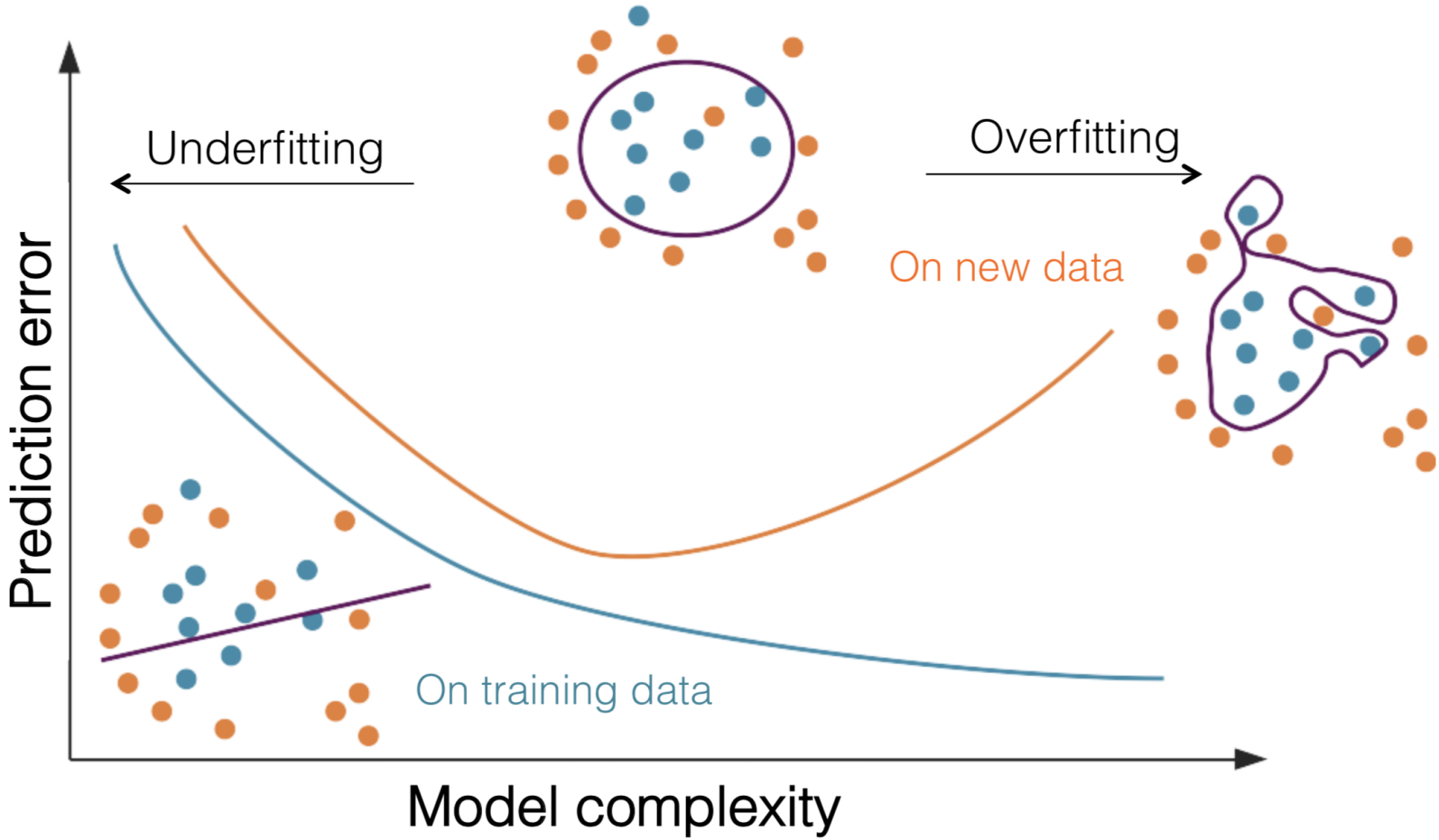
Model complexity

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!

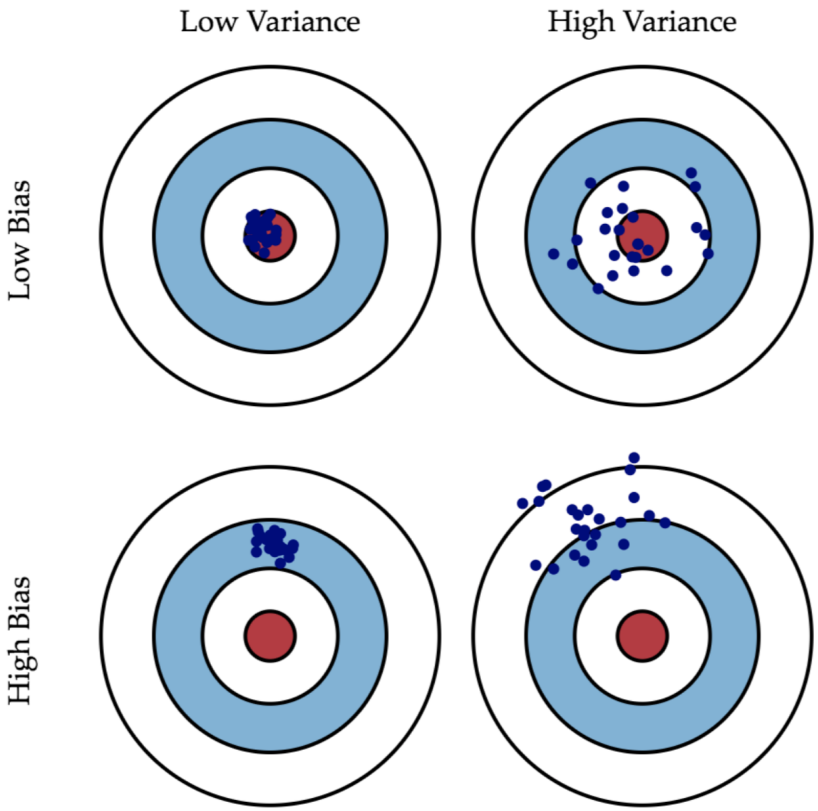
$$\text{Var}(\hat{f}(x)) = \mathbb{E}\left[\left(\hat{f}(x) - \mathbb{E}(\hat{f}(x))\right)^2\right]$$

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

Ideally we want to optimize both of them.

Bias-Variance tradeoff

3



- The center of the target is a model that perfectly predicts the correct values!
- We can repeat our entire model building process to get a number of separate hits on the target!
  $\rightsquigarrow$ Each hit represents an individual realization of the model!

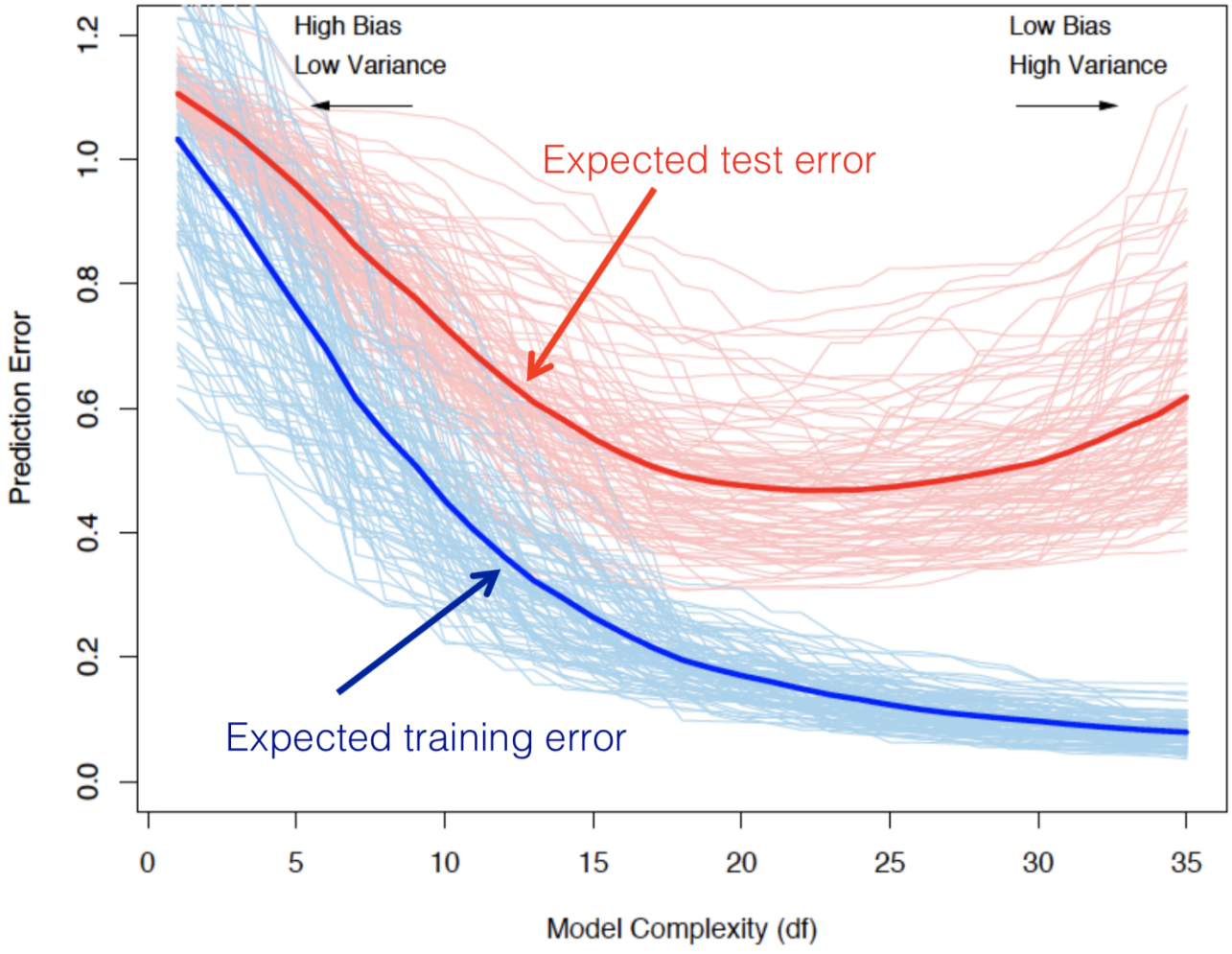
- Bias measures how far are in general these models' predictions are from the correct value!
- Variance is how much the predictions for a given point vary between different realizations of the model!

[3] taken from `http://scott.fortmann-roe.com/docs/BiasVariance.html`

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!

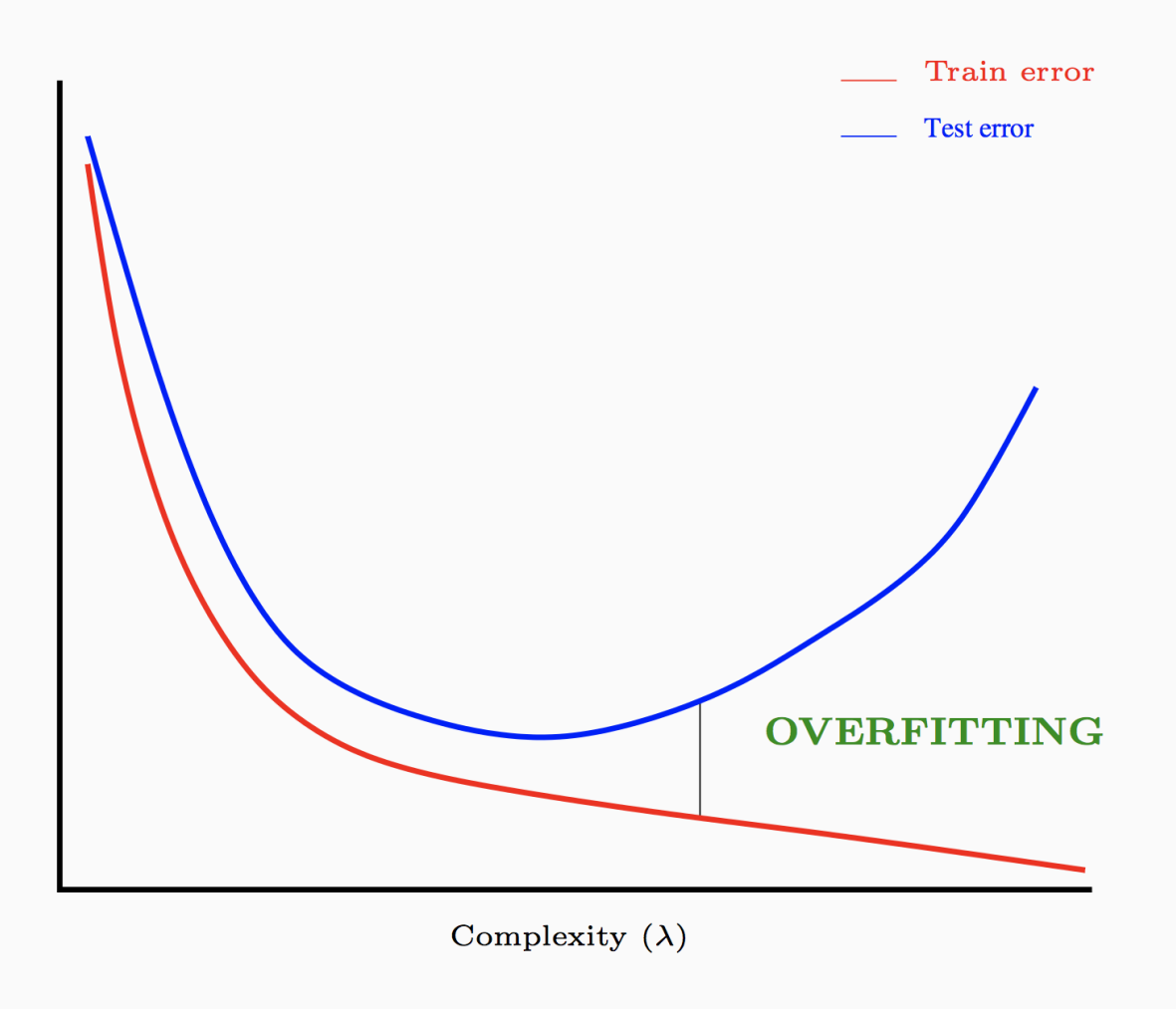


Figure: Overfitting can be detected by an increase in the test error.

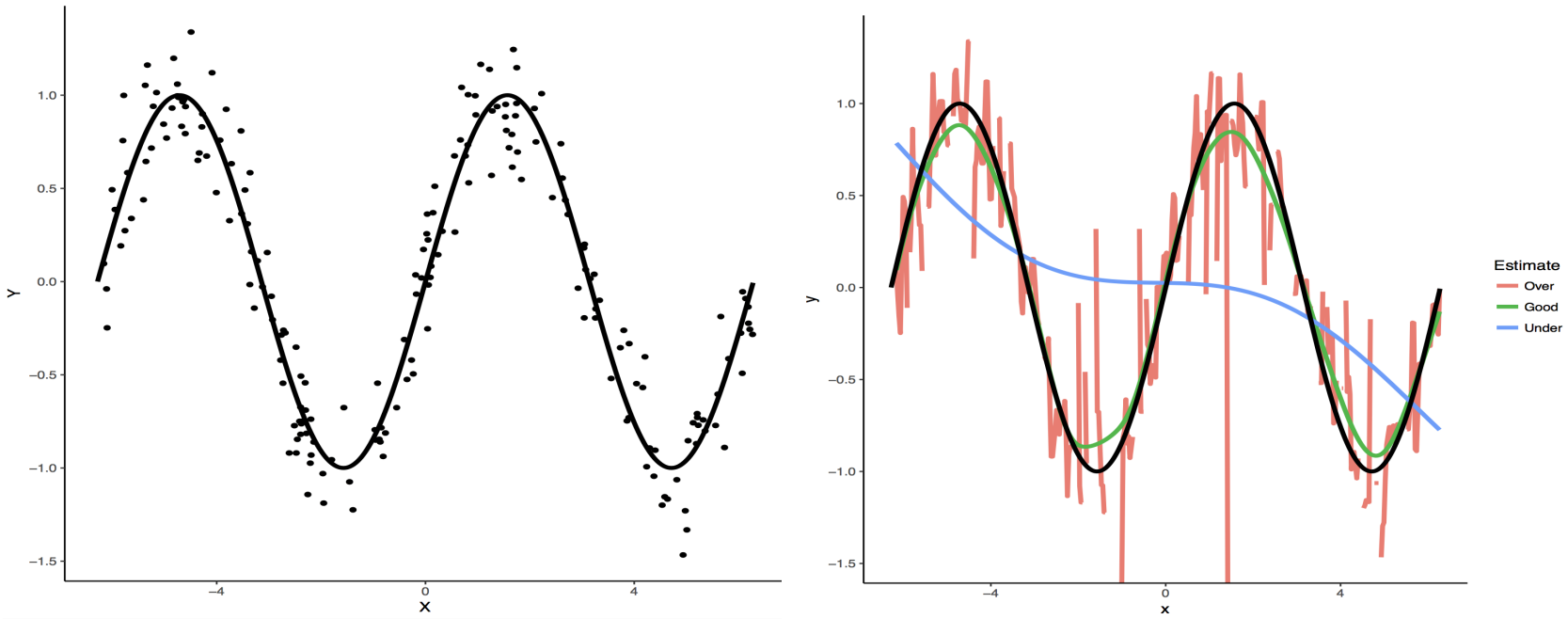


Figure: Illustration of overfitting in regression. Here, λ controls the regressor smoothness.

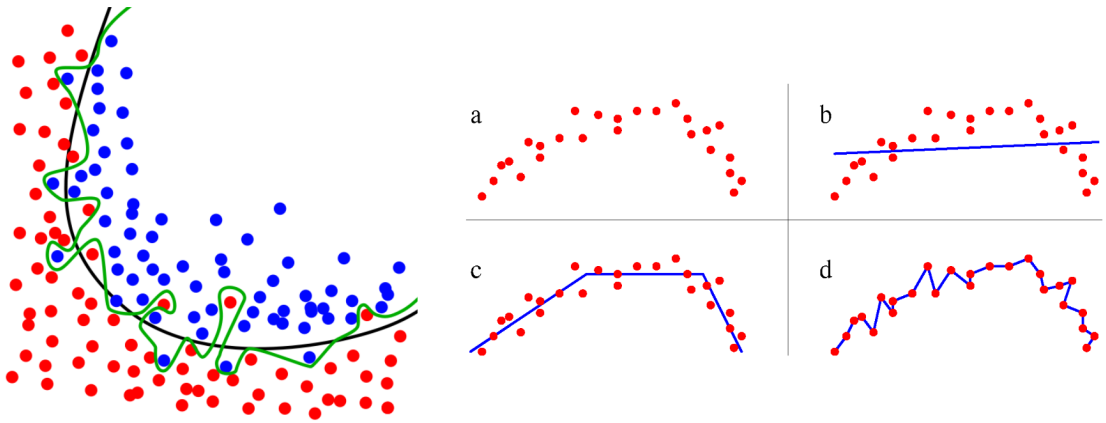


Figure: Illustration of overfitting in detection. Here, λ controls the frontier smoothness.

A way to avoid overfitting

Penalization

Overfitting can also be prevented by penalization.

For instance, in linear regression, computing

$$\hat{\beta}, \hat{e} \in \text{argmin}_{\beta,e} \frac{1}{n} \sum_{i=1}^{n} \ell(Y_i, X_i^T \beta + e)$$

could lead to bad classifier. Minimize instead

$$\hat{\beta}, \hat{e} \in \text{argmin}_{\beta,e} \frac{1}{n} \sum_{i=1}^{n} \ell(Y_i, X_i^T \beta + e) + \lambda \text{pen}(\beta)$$

where pen is a penalization function, it forbids β to be "too complex". $\lambda > 0$ is a tuning or smoothing parameter, that balances goodness-of-fit and penalization.

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!

Any idea?

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!

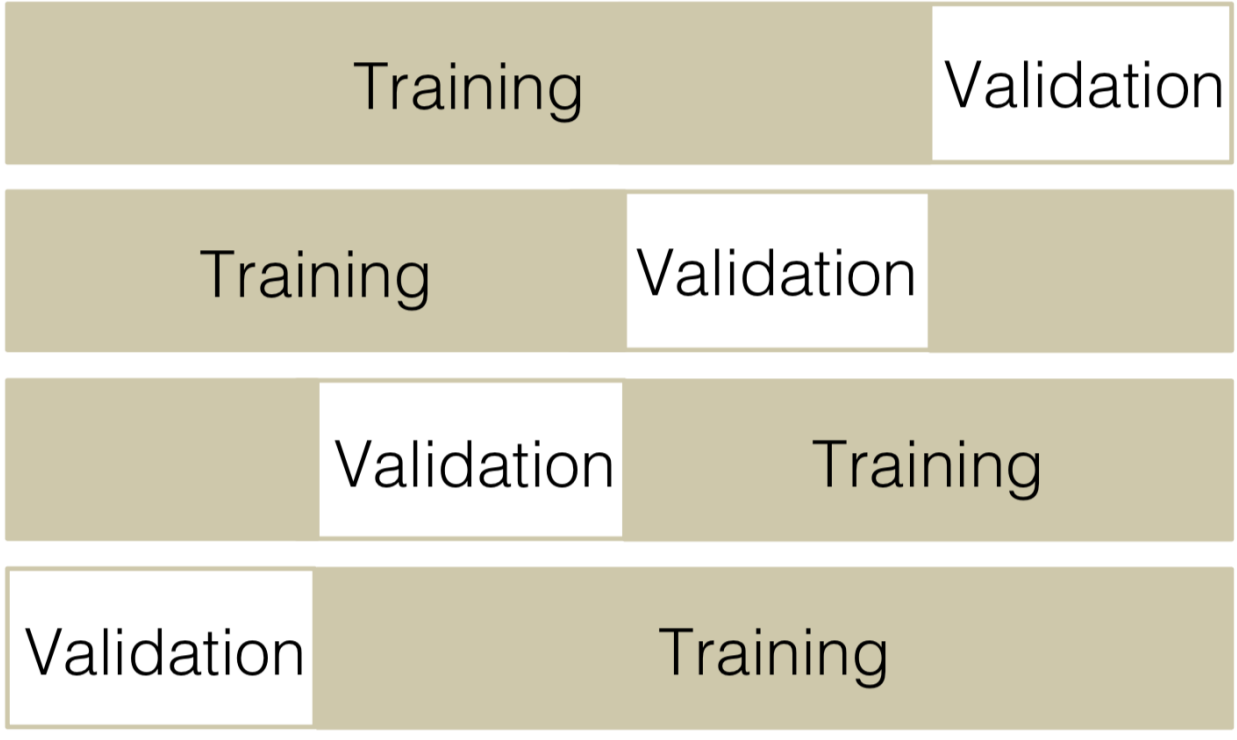
- Model selection: pick the best model!
- Model assessment: estimate its prediction error on new data!

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