MAP 534
Introduction to machine learning
Basic concepts and linear models for regression

Alain Durmus

Outline

What is a data scientist?

"A data scientist is someone who knows more statistics than a computer scientist and more computer science than a statistician."

- Josh Blumenstock -

"Data Scientist = statistician + programmer + coach + storyteller + artist."

- Shlomo Aragmon -

- Machine learning (ML):
  - mix of statistics, computer science and applied mathematics;
  - gathers all scientific methods to adress issues related to data.

The Data Science Process

- Ask an interesting question
  - scientific/business goal?
  - is there any obvious answer?
- Get the Data
- Explore the Data
- Model the Data and make inference
- Communicate/Visualize the Results

The Data Science Process

- Ask an interesting question
- Get the Data
  - How were the data sampled?
  - Which data are relevant?
  - Are there privacy issues?
- Explore the Data
- Model the Data and make inference
- Communicate/Visualize the Results

The Data Science Process

- Ask an interesting question
- Get the Data
- Explore the Data: still performed by human
  - Plot the data
  - Are there anomalies or outliers?
  - Are there patterns?
  - Useful library: pandas (see the course website)
- Model the Data and make inference
- Communicate/Visualize the Results

The Data Science Process

- Ask an interesting question
- Get the Data
- Explore the Data
- Model the Data and make inference: use of machine learning!
  - Build a model
  - Train the model
  - Validate the model
- Communicate/Visualize the Results

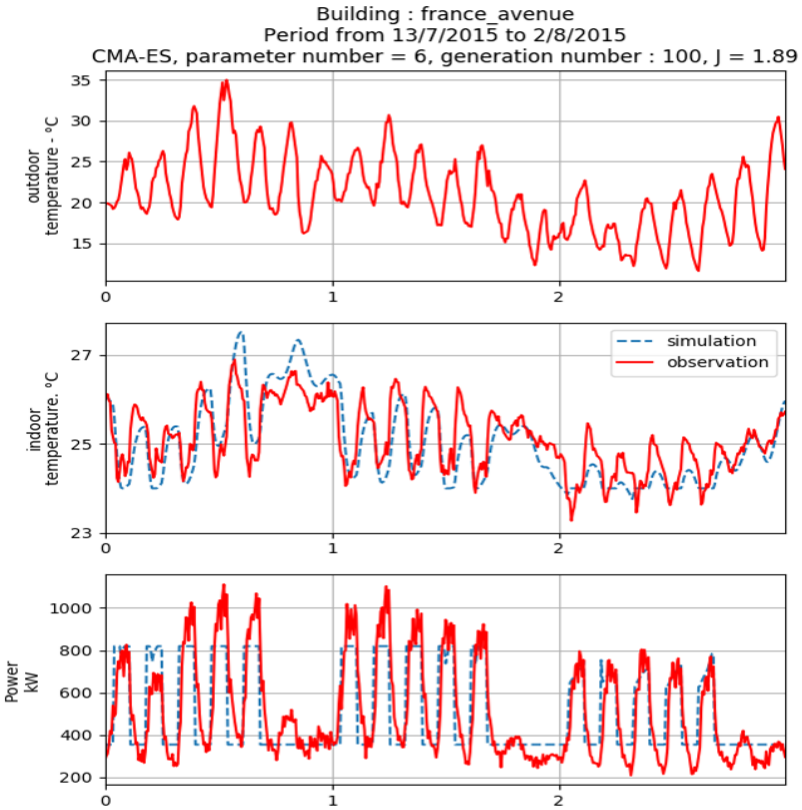
The Data Science Process

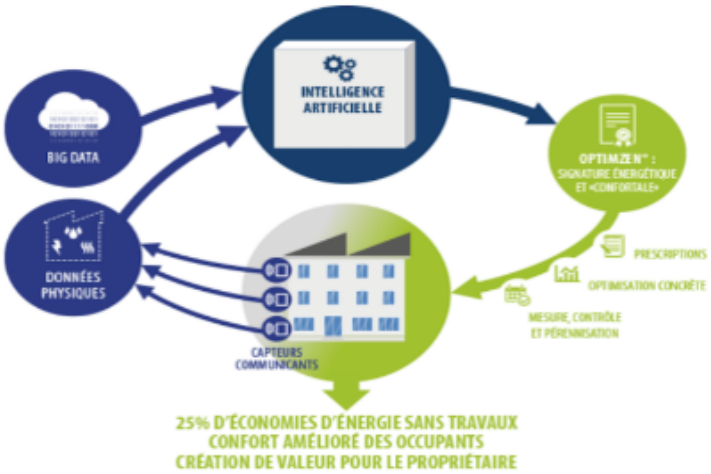
- Ask an interesting question
- Get the Data
- Explore the Data
- Model the Data and make inference: use of machine learning!
- Communicate/Visualize the Results
    - Use Seaborn for example (see the course website)

Motivations: object recognition

- Task: say if an object is present or not in the image.
- Experience: set of previously seen labeled images.

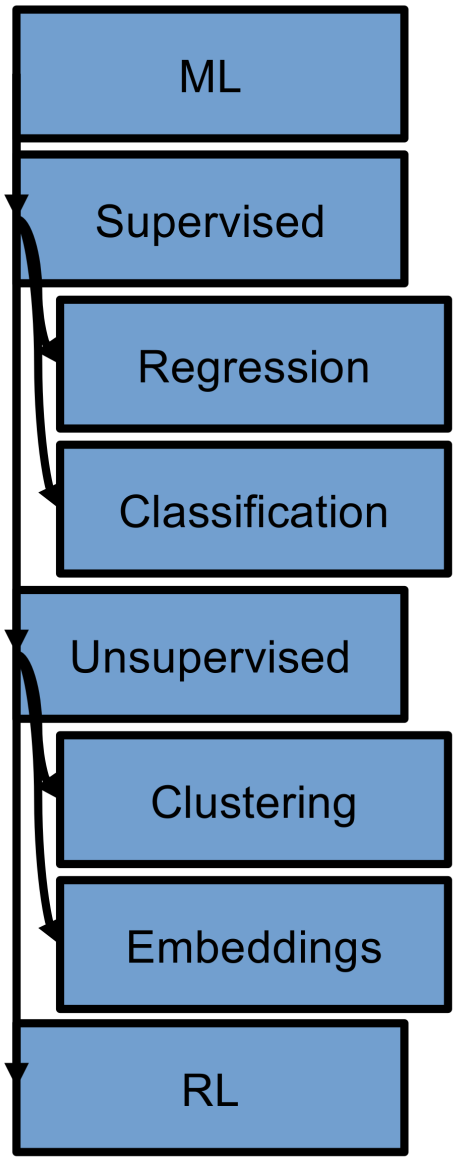
Motivations: prediction of time series



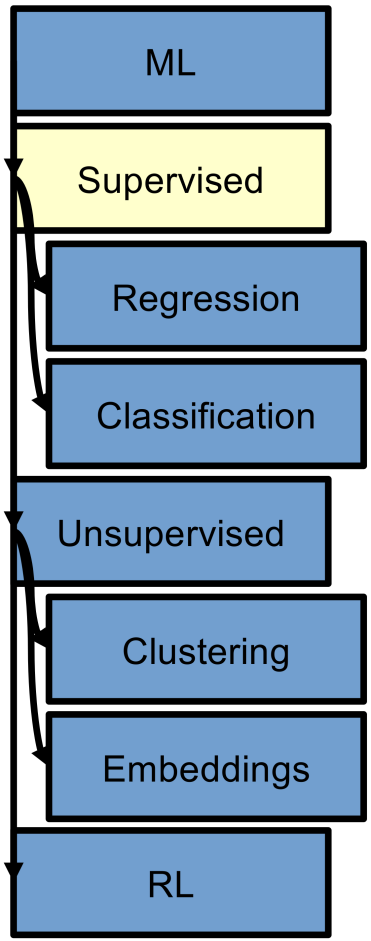


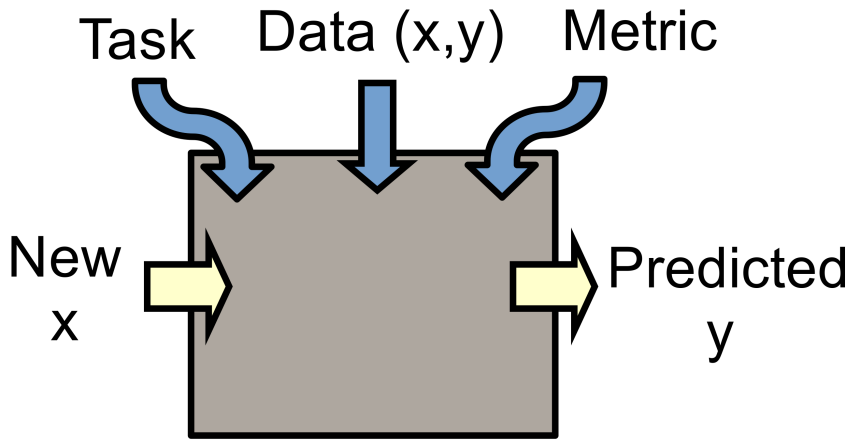
- Task: predict future consumptions, gas emissions and temperatures.
- Experience: millions of observations obtained from hundreds of sensors.

Taxonomy of ML problems



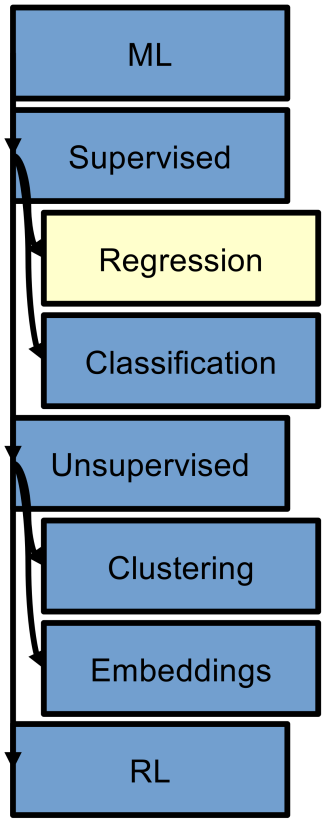
Supervised learning

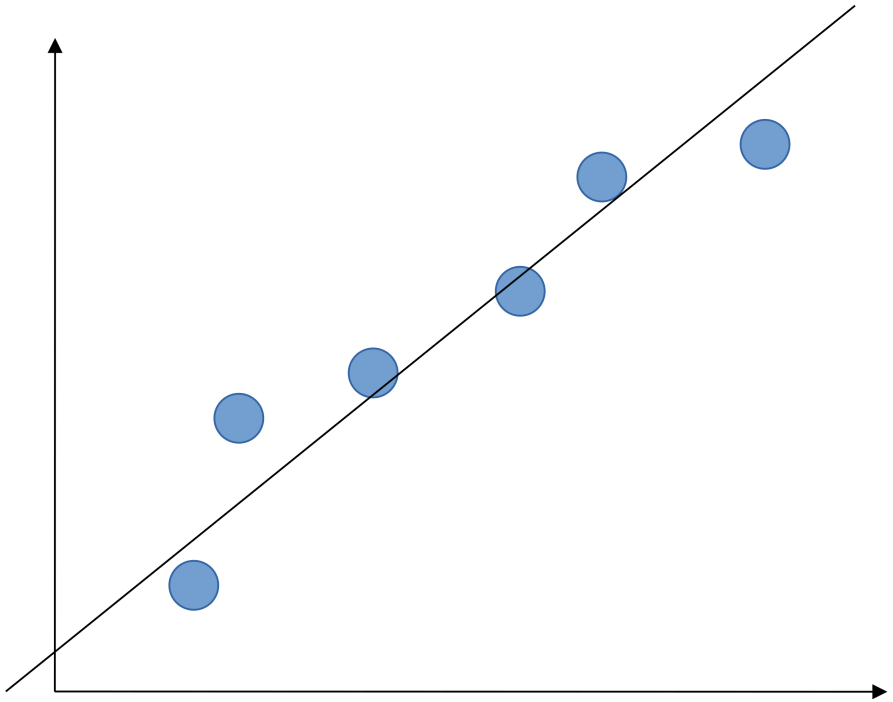




- Supervised/predictive learning: $\mathcal{D} = \{(x_i, y_i)\}_{i \in [N]}$
  - $x_i \in \mathrm{X}$ explanatory variables/covariates/features such as images, sentences, email messages, time series, molecular shapes, graphs...
  - $y_i \in \mathrm{Y}$ labels/classes
    such as sex/species (finite sets) or income level/temperature (infinite set)...

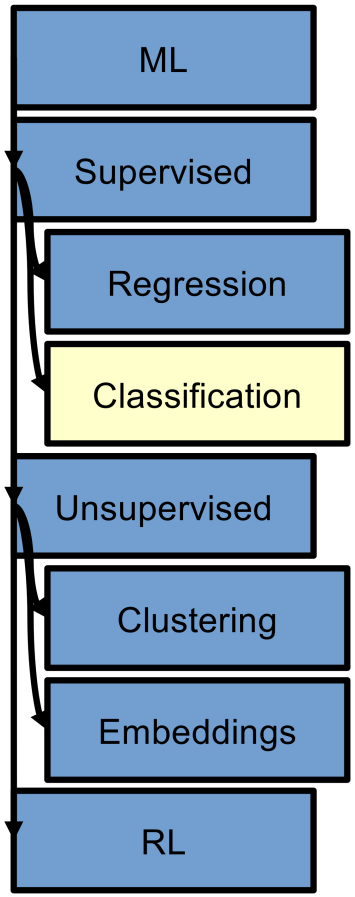
Regression

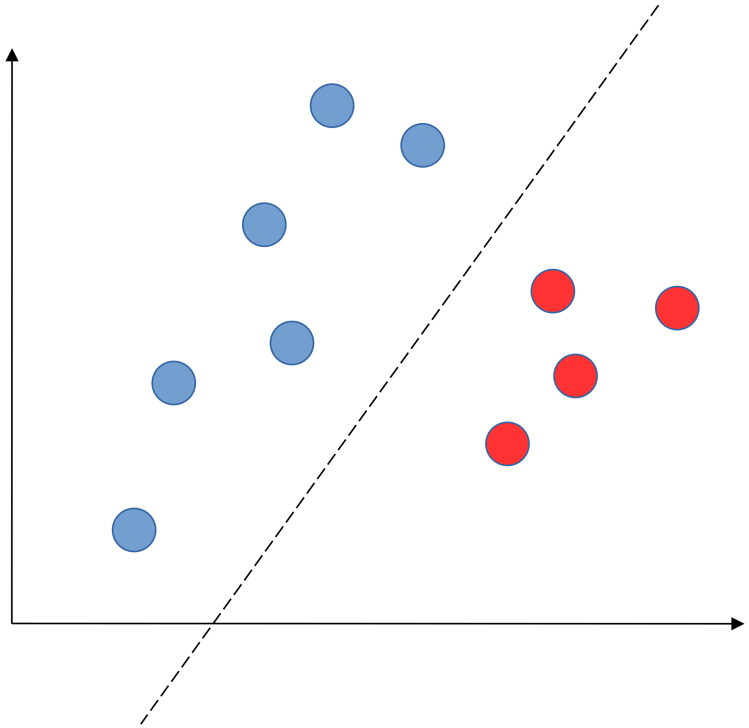




- $\mathsf{Y} = \mathbb{R}$ or $\mathsf{Y} = \mathbb{R}^p$
- Example: time series prediction

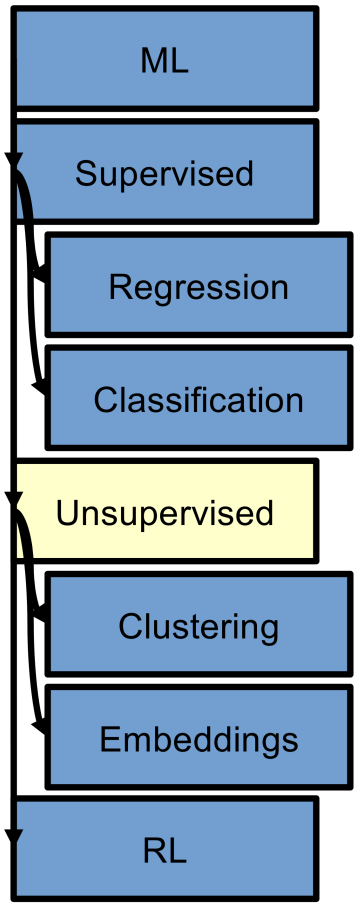
Classification

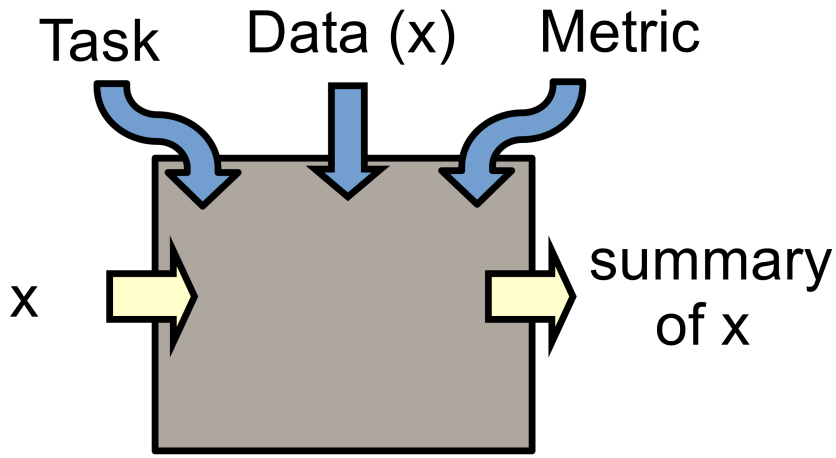




- Y is a finite set
- Example: object recognition

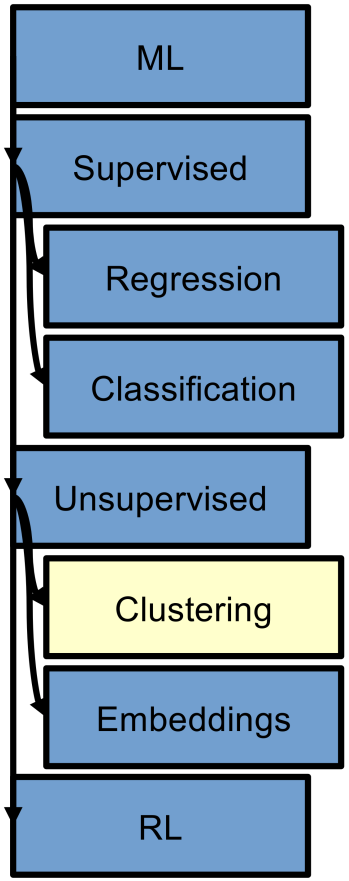
Unsupervised learning

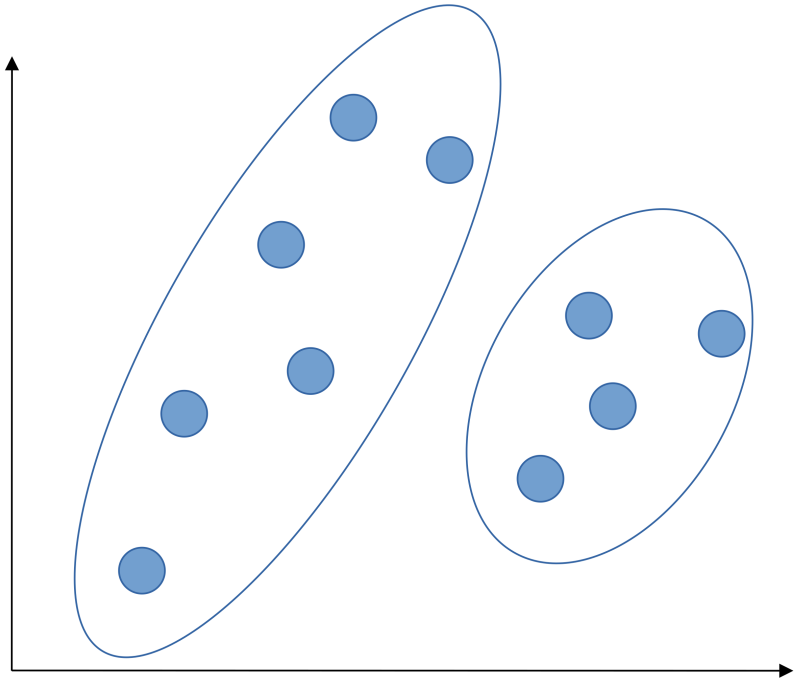




- Unsupervised learning: $\mathcal{D} = \{x_i\}_{i \in [N]}$
  - Goal: find "interesting patterns" in the data
  - Not so well-defined mathematically
  - No obvious performance criterion, depends on the applications

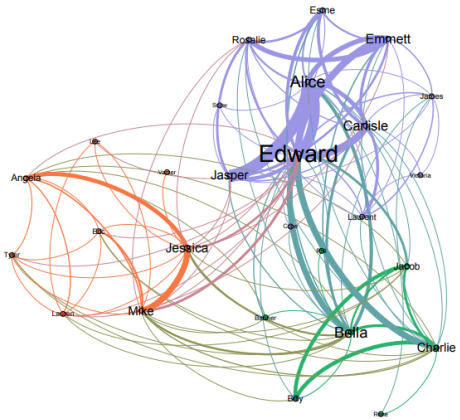
Clustering

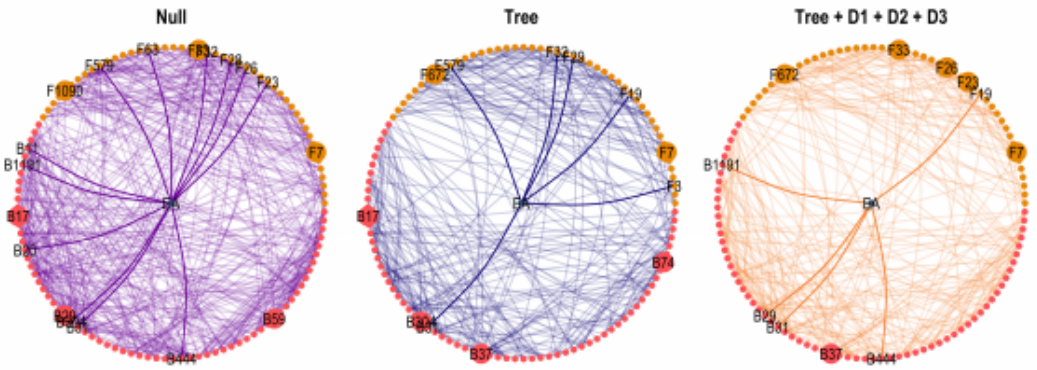




- Clustering: $\mathcal{D} = \{x_i\}_{i \in [N]}$
  - Given a number of class K
  - Goal: find a partition the dataset into K classes
  - Not so well-defined mathematically
  - No obvious performance criterion, depends on the applications

Examples - clustering

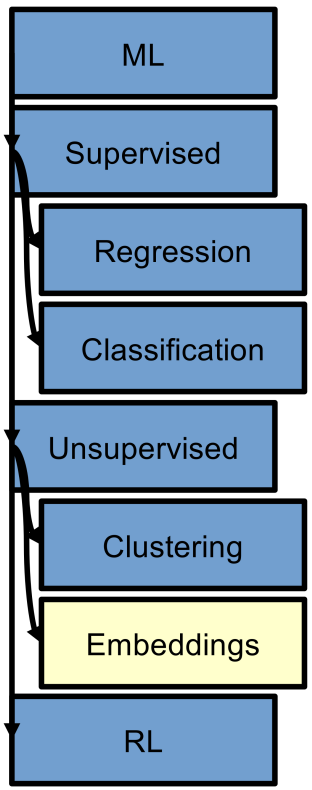


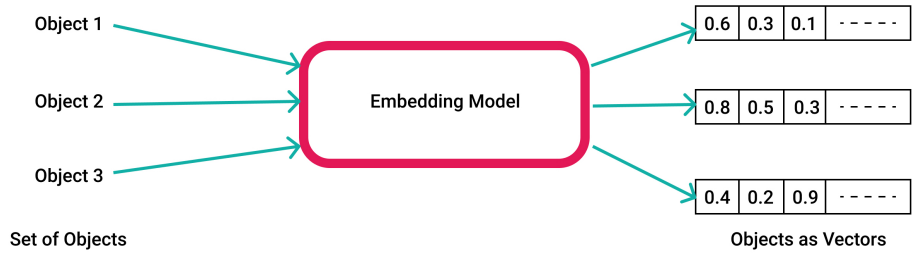


- Task: group characters/news in texts/journals, infer an ecological network[1].
- Data: number of occurrences of two key names both within a specified distance in the text / abundance of all species in all sites.

[1] **Tree-based Reconstruction of Ecological Network from Abundance Data**, Momal, R. et al., ***ArXiv:1905.02452***

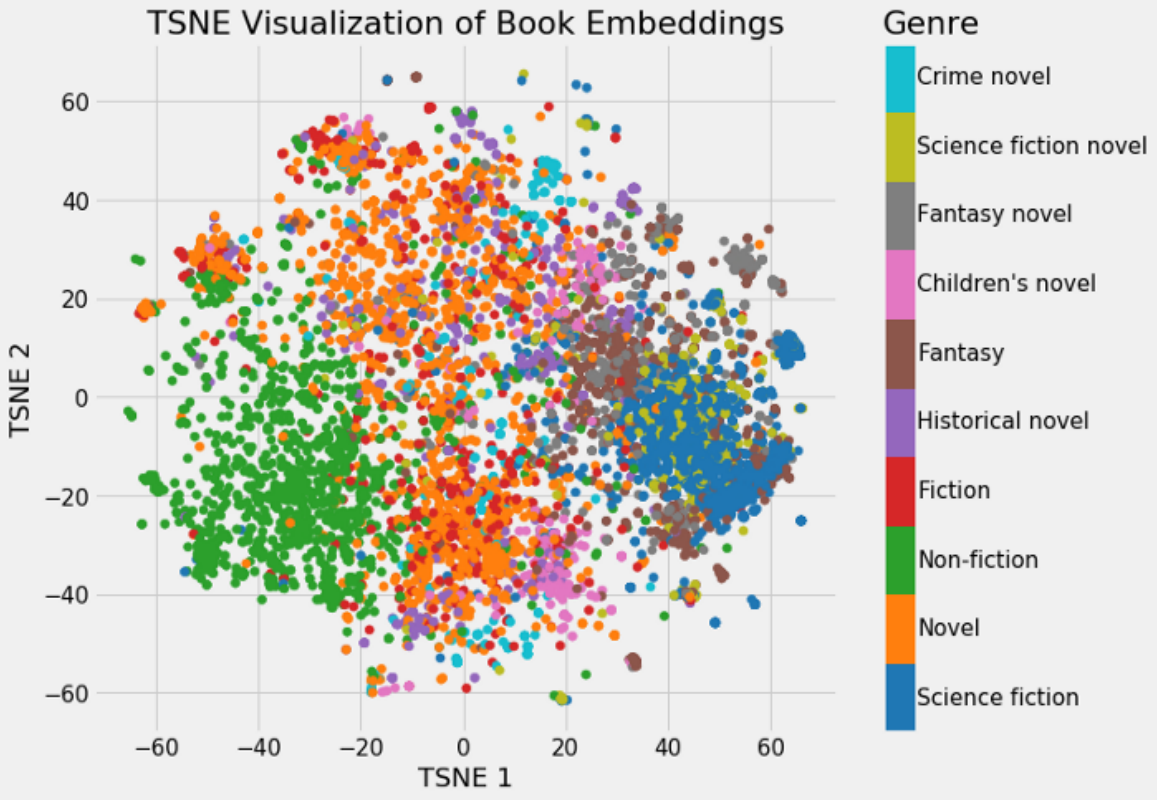
Embedding





- Clustering: $\mathcal{D} = \{x_i\}_{i \in [N]}$
  - Goal: Learn an embedding model
  - i.e., a function $f^\star : \mathsf{X} \to \mathbb{R}^d$ that best represents the data
  - as a bonus, we may require that this representation is sparse, i.e., only $f^\star(x) \in \mathbb{R}^d$ has only few non-zero components
  - Again not so well-defined mathematically and no obvious performance criterion, depends on the applications

Example - embedding



- Task: Word embedding for book recommendation
- Data: (Book 1, Book 2, Association)

Organization of the course

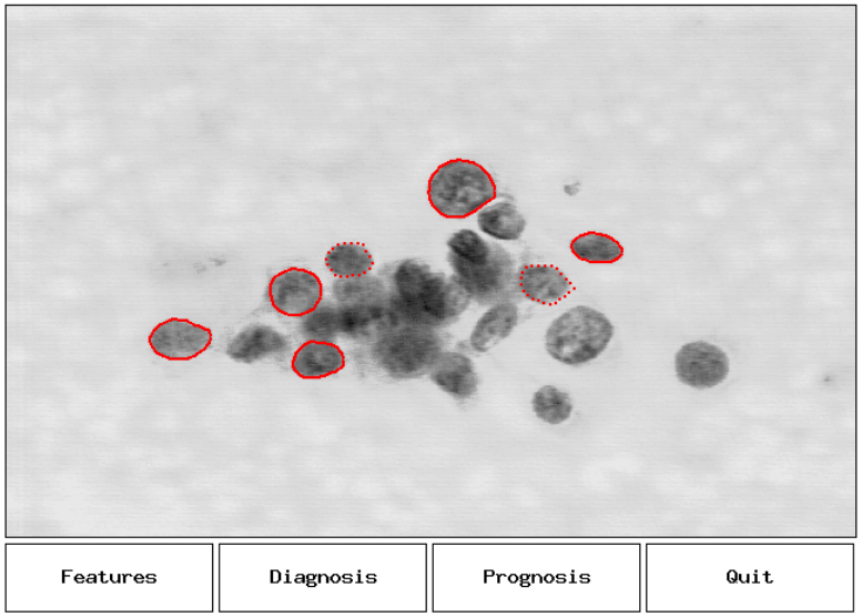
- Regression
- Bayesian statistics/machine learning
- Classification
- Neural Networks
- Kernel methods and SVM
- and a last lecture not decided yet

Organization

- Other professors for the course: Edouard Oyallon and Maxime Sangnier.
- Always bring your laptop for the practical sessions (Exercises + Labs)!
- Evaluation: 3 assignments (50%)+ a final exam (50%).
- The course will follow [Bis07] (see the moodle to download the book).
- Slides + exercises + Labs + further reading on the moodle.

Outline

A more concrete example of supervised learning



- Cell samples were taken from tumors in breast cancer patients before surgery, and imaged.
- Tumors were excised.
- Patients were followed to determine whether or not the cancer recurred, and how long until recurrence or disease free.

A more concrete example of supervised learning

- 30 real-valued variables per tumor.
- 2 variables that can be predicted:
  1. Outcome (R=recurrence, N=non-recurrence)
  2. Time (until recurrence, for R, time healthy, for N).

tumor size	texture	perimeter	...	outcome	time
18.02	27.6	117.5		N	31
17.99	10.38	122.8		N	61
20.29	14.34	135.1		R	27

Terminology

tumor size	texture	perimeter	...	outcome	time
18.02	27.6	117.5		N	31
17.99	10.38	122.8		N	61
20.29	14.34	135.1		R	27

- Columns are called covariates/input variables/features or attributes.
- The outcome and time (which we are trying to predict) are called output variables/targets/labels/classes.
- A row in the table is called training example/instance/sample or observations.
- The whole table is called (training) data set.
- The problem of predicting the recurrence is called (binary) classification.
- The problem of predicting the time is called regression.

Terminology

x			y	
tumor size	texture	perimeter ...	outcome	time
18.02	27.6	117.5	N	31
17.99	10.38	122.8	N	61
20.29	14.34	135.1	R	27

- A training example has the form: (x_i, y_i) (i corresponds to the index of the row we considered),
  - with $x_i = (x_i^{(1)}, \dots x_i^{(d_x)})$ (here $d_x = 30$)
  - and $y_i = (y_i^{(1)}, \dots, y_i^{(d_y)})$ (here $d_y = 2$).
- The training set $\mathcal{D}$ consists of n training examples.

Mathematical description of a regression task and its modelization

- Supervised/predictive learning: consider N observations $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ such that

$$x_i \in \mathrm{X} \text{ and } y_i \in \mathrm{Y} \ . \tag{1}$$

  - $x_i \in \mathrm{X}$ explanatory variables/covariates/features/inputs
  - $y_i \in \mathrm{Y}$ labels/classes/outputs
- Goal: find $f^\star$ to predict the label associated with new covariates x_{new}

$$y_{\text{pred}} = f^\star(x_{\text{new}}) \ . \tag{2}$$

- Question: how to find $f^\star$?
- We need to make some guess on how $f^\star$ looks like?
- It corresponds to the modelization step we mention before!
- It boils down choosing a family of functions $\mathcal{F} \subset \{f : \mathrm{X} \to \mathrm{Y}\}$
- In most cases, $\mathcal{F}$ is parametric and parametrized by $w \in \mathbb{R}^d$.
- Vocabulary:
  - $\mathcal{F}$: class of hypothesis or model class;
  - $f \in \mathcal{F}$: hypothesis.

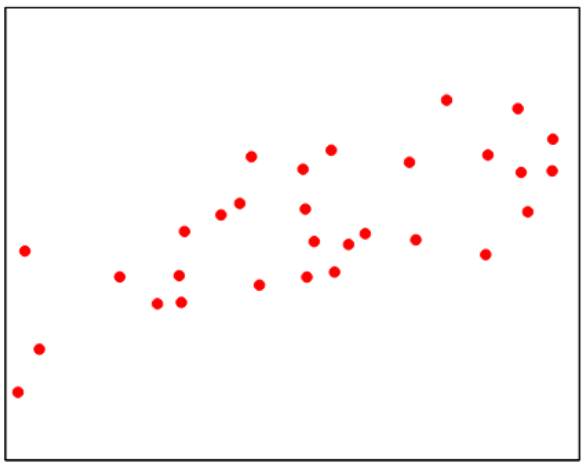
Steps to solving a supervised learning problem

- Decide what the input-output pairs are.
- Decide how to encode inputs and outputs. This defines the input space X, and the output space Y.
- Choose a class of hypotheses/representations $\mathcal{F}$.
- ? which class?

Hypothesis class

- $\mathcal{F} \subset Y^X$ such that $f \in \mathcal{F}$ satisfies conditions justified by the problem.
- Example of condition: smoothness, integrability...
- So... Is a larger class always better?
- How can we control the size of the class?

Example: What hypothesis class should we pick?



x	y
0.86	2.49
0.09	0.83
−0.85	−0.25
0.87	3.10
−0.44	0.87
−0.43	0.02
−1.10	−0.12
0.40	1.81
−0.96	−0.83
0.17	0.43

(3)

Linear hypothesis

- Example for $\mathsf{Y} = \mathbb{R}$:

$$\mathcal{F} = \left\{ x \mapsto \sum_{i=1}^{d} w_i \tilde{x}^{(i)} = w^{\mathrm{T}} x \right\}$$

  for

$$x = (\tilde{x}^{(1)}, \ldots, \tilde{x}^{(d)})^{\mathrm{T}} \in \mathsf{X} \subset \mathbb{R}^d \, .$$

- $w = (w_1, \ldots, w_d)^{\mathrm{T}} \in \mathbb{R}^d$ are called parameters or weights.
-  : here notations are not consistent between w and x!
- Reason: x_i is the i-th observation...

Linear hypothesis

- Example for $\mathsf{Y} = \mathbb{R}$:

$$\mathcal{F} = \left\{ x \mapsto \sum_{i=1}^{d} w_i \tilde{x}^{(i)} = w^{\mathrm{T}} x \right\}$$

  for

$$x = (\tilde{x}^{(1)}, \ldots, \tilde{x}^{(d)})^{\mathrm{T}} \in \mathsf{X} \subset \mathbb{R}^d \ .$$

- $w = (w_1, \ldots, w_d)^{\mathrm{T}} \in \mathbb{R}^d$ are called parameters or weights.
- To simplify notation, we can concatenate an attribute $\tilde{x}_i^{(0)} = 1$ to all n observations ($\tilde{x}^{(0)}$ is called bias term or intercept term):

$$x_i \leftarrow \begin{pmatrix} \tilde{x}_i^{(0)} \\ x \end{pmatrix} \in \mathbb{R}^{n+1} \ . \tag{4}$$

- Then, we obtain the new class:

$$\mathcal{F} = \left\{ x \mapsto \sum_{i=0}^{d} w_i \tilde{x}^{(i)} = w^{\mathrm{T}} x \right\}$$

  where w now is vectors of size $n + 1$ as x.
- How should we pick w?

Error minimization!

- Consider a general class parametrized by w: $\{f_w : \mathsf{X} \to \mathsf{Y} \ : \ w \in \mathbb{R}^d\}$ to get the best prediction:

$$y_{\mathsf{pred}} = f^\star(x_{\mathsf{new}}) \ . \tag{5}$$

- Intuitively, we should find w such that $f_w(x_i) \approx y_i$...
- Hence, we will define an error function or cost function to measure how much our prediction differs from the "true" answer.
- We will pick w such that the error function is minimized
- How should we choose the error function?

Linear regression

- Main idea: we should find w such that $f_w(x_i) \approx y_i$...
- Example in the case $\mathrm{Y} = \mathbb{R}$, we define the error function or empirical risk by

$$E(w) = \frac{1}{2} \sum_{i=1}^{m} \left(f_w\left(x_i\right) - y_i\right)^2 \text{ (the } 1/2 \text{ is just for convenience...)}$$

- This framework is called the linear regression.
- E admits a unique minimizer
- Solution $\hat{w} \in \operatorname{argmin} E(w)$.
- Here, we measure the difference between the prediction on the observation using $\ell(y, y') = (y - y')^2$.
- We could use other function ℓ: $\ell(y, y') = |y - y'|^p$...
- ℓ is referred to as the cost or loss.

Steps to solving a supervised learning problem

- Decide what the input-output pairs are.
- Decide how to encode inputs and outputs. This defines the input space X, and the output space y.
- Choose a class of hypotheses/representations $\mathcal{F} = \{f_w : \mathsf{X} \to \mathsf{Y} \ : \ w \in \mathbb{R}^d\}$.
- Choose a loss function ℓ.
- Define error function

$$E_\ell(w) = n^{-1} \sum_{i=1}^{n} \ell(y_i, f_w(x_i)) \ . \tag{6}$$

- Choose an algorithm to solve

$$\text{minimize } E_\ell \ . \tag{7}$$

- How to do so: vanish the gradient or gradient descent (see later)...

Reminder on the gradient operator

- Gradient: Multivariate generalization of the derivative.
- Consider a function $J(u_1, u_2, \ldots, u_d) : \mathbb{R}^d \mapsto \mathbb{R}$ (e.g., an error function).
- The partial derivative w.r.t. u_i is denoted:

$$\frac{\partial}{\partial u_i} J(u_1, u_2, \ldots, u_n) : \mathbb{R}^d \mapsto \mathbb{R}$$

  The partial derivative is the derivative along the u_i axis, keeping all other variables fixed.
- The gradient $\nabla J(u_1, u_2, \ldots, u_n) : \mathbb{R}^d \mapsto \mathbb{R}^d$ is a function which outputs a vector containing the partial derivatives. That is:

$$\nabla J = \left(\frac{\partial}{\partial u_1} J, \frac{\partial}{\partial u_2} J, \ldots, \frac{\partial}{\partial u_n} J \right)^{\mathrm{T}} .$$

Reminder on the gradient operator

- Gradient: Multivariate generalization of the derivative.
- Points such that $\nabla J(u) = 0$ are called stationary points.
- If $\hat{w}$ minimizes/maximizes J locally:

$$\nabla J(u^\star) = 0 \ . \tag{8}$$

- To optimize J:
  1. find u such that $\nabla J(u) = 0$.
  2. find the one which are global minimizers/maximizers...
  3. In the case J is convex, u such that $\nabla J(u) = 0$ is necessarily a minimizer!
- If finding u such that $\nabla J(u) = 0$ is not analytical $\rightarrow$ optimization algorithms!
- In the case $J = E$ for the linear regression, optimization can be explicitly done!

Outline

Finding the linear regression estimator

- Going back to

$$E(w) = \frac{1}{2}\sum_{i=1}^{n}(f_w(x_i) - y_i)^2 = \|\boldsymbol{X}w - y\|^2 \ , \quad \text{since } f_w(x_i) = x_i^{\mathrm{T}} w \ , \quad (9)$$

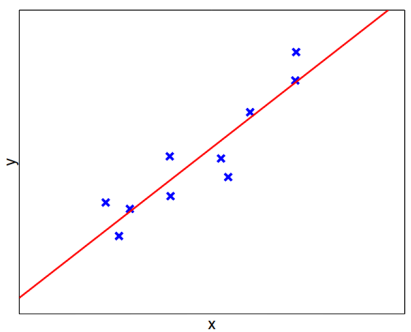
where

$$y = (y_1, \ldots, y_n)^{\mathrm{T}} \in \mathbb{R}^n \ , w = (w_1, \ldots, w_d)^{\mathrm{T}} \in \mathbb{R}^d \ , \quad (10)$$

$$\boldsymbol{X} = (x_1, \ldots, x_n)^{\mathrm{T}} = \begin{pmatrix} x_1^{(1)} & \ldots & x_1^{(d)} \\ . & . & . \\ . & . & . \\ . & . & . \\ x_n^{(1)} & \ldots & x_n^{(d)} \end{pmatrix} \in \mathbb{R}^{n \times d} \ . \quad (11)$$

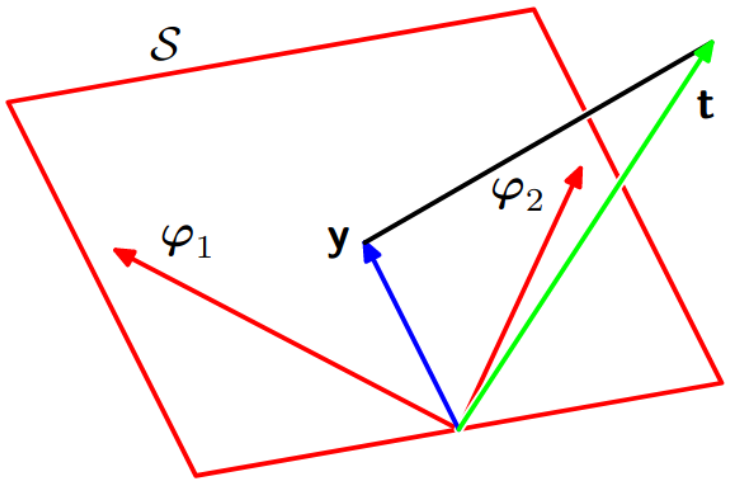
- If the columns of $\boldsymbol{X}$ are linearly independent we can show:

$$\hat{w} = (\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{T}} y \ . \quad (12)$$



Geometric interpretation

- $\hat{w} = (\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{T}}y$ is the orthogonal projection onto the span generated by the d column of $\boldsymbol{X}$.



Is linear regression enough?

- Linear regression is too simple for most realistic problems...
- But it should be the first thing you try for real-valued outputs!
- One solutions: transform the data:
  - Add cross-terms, higher-order terms.
  - More generally, apply a transformation of the inputs from X to some other space Z, then do linear regression in the transformed space.

Linear models for regression

- Idea: apply a transformation of the inputs from X to some other space $\tilde{X}$, then do linear regression in the transformed space.
- We consider some functions $\{\phi_j\}_{j=1}^d$ such that $\phi_j : \mathsf{X} \to \mathbb{R}$.
- They are referred to as basis functions or feature maps.
- The function space that we consider now is

$$\mathcal{F}_\phi = \left\{ f_w : x \mapsto \sum_{i=1}^{d} w_j \phi_j(x) \right\} . \tag{13}$$

- The model is said to be linear because for any x, $w \mapsto f_w(x)$ is linear.
- Example $\mathsf{X} \subset \mathbb{R}^{d_x}$:
  - $\phi_j : x \in \mathbb{R}^d \mapsto \tilde{x}^{(j)}$, where $\tilde{x}^{(j)}$ stands for the j-th coordinate, we get back the common linear regression model.
  - Gaussian basis function: $\phi_j(x) = \exp(-\left\|x - \mu_j\right\|^2 / (2\sigma_j^2))$.
  - Sigmoidal basis function: $\phi_j(x) = \mathrm{logit}((x_j - \mu_j)/\sigma_j)$, where

$$\mathrm{logit}(t) = \mathrm{e}^t/(1+\mathrm{e}^t) . \tag{14}$$

  - In signal processing: $\phi_j(x)$ j-th coefficient of an image in a dictionary (Fourier, wavelets...).

Least square estimation

- Here we can use the same logic than for the common linear regression and consider the error function based on $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$:

$$E(w; \mathcal{D}, \phi) = \frac{1}{2} \sum_{i=1}^{n} \left\{ y_i - \sum_{j=1}^{d} w_j \phi_j(x_i) \right\}^2 . \tag{15}$$

- The LSE is:

$$\hat{w}_\phi \in \operatorname{argmin} E(w; \mathcal{D}, \phi) . \tag{16}$$

- Note that we can set $\tilde{z}_i^{(j)} = \phi_j(x_i)$ which gives a common linear regression problem associated with $\{(y_i, z_i)\}_{i=1}^n$, $z_i = (\tilde{z}_i^{(1)}, \ldots, \tilde{z}_i^{(d)})^{\mathrm{T}}$.
- This framework is referred to as least square estimation...

Least square estimator bis

- Here we can use the same logic than for the common linear regression and consider the error function based on $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$:

$$E(w; \mathcal{D}, \phi) = \frac{1}{2} \sum_{i=1}^{n} \left\{ y_i - \sum_{j=1}^{d} w_j \phi_j(x_i) \right\}^2 . \tag{17}$$

- The LSE is:

$$\hat{w}_\phi \in \operatorname{argmin} E(w; \mathcal{D}, \phi) . \tag{18}$$

- Note that we can set $\tilde{z}_i^{(j)} = \phi_j(x_i)$ which gives a common linear regression problem associated with $\{(y_i, \tilde{z}_i)\}_{i=1}^n$.
- We easily deduce that

$$\hat{w}_\phi = (\Phi_x^{\mathrm{T}} \Phi_x)^{-1} \Phi_x^{\mathrm{T}} y , \tag{19}$$

where as previously $y = (y_1, \ldots, y_n)^{\mathrm{T}}$ and

$$\Phi_x = \begin{pmatrix} \phi_0(x_1) & \phi_1(x_1) & \cdots & \phi_{d-1}(x_1) \\ \phi_0(x_2) & \phi_1(x_2) & \cdots & \phi_{d-1}(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(x_N) & \phi_1(x_N) & \cdots & \phi_{d-1}(x_N) \end{pmatrix} \tag{20}$$

- Φ_x is called the design matrix.

Link with statistics: Gaussian regression and maximum likelihood estimation

- The statistical model associated with the LSE is:

$$Y_i \overset{\text{iid}}{\sim} \sum_{i=1}^{d} w_j \phi_j(X_i) + \sigma^2 Z_i \ , \quad i \in \{1, \dots, N\} \ , \tag{21}$$

  where
  - $Z_i \overset{\text{iid}}{\sim} \mathrm{N}(0, 1)$;
  - X_i are i.i.d random variables with unknown distribution;
  - w is the parameter to infer.
- Then, if we are just interested in inferring w, the log-likelihood is

$$\ell(w) = ? \tag{22}$$

  where the observations are $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$.

Link with statistics: Gaussian regression and maximum likelihood estimation

- The statistical model associated with the LSE is:

$$Y_i \overset{\text{iid}}{\sim} \sum_{i=1}^{d} w_j \phi_j(X_i) + \sigma^2 Z_i \ , \quad i \in \{1, \dots, N\} \ , \tag{23}$$

  where
  - $Z_i \overset{\text{iid}}{\sim} \mathrm{N}(0, 1)$;
  - X_i are i.i.d random variables with unknown distribution;
  - w is the parameter to infer.

- Then, if we are just interested in inferring w, the log-likelihood is

$$\ell(w) = -\frac{1}{2\sigma} \sum_{i=1}^{N} \left\{ y_i - \sum_{i=1}^{d} w_j \phi_j(x_i) \right\}^2 \ , \tag{24}$$

  where the observations are $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$.

- Therefore, maximizing the log-likelihood leads to the same solution as minimizing the error function.

Outline

A simple highlighting example: polynomial curve fitting

- Consider N observations $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ such that

$$x_i \in \mathsf{X} = [0,1] \text{ and } y_i \in \mathsf{Y} = \mathbb{R} \,. \tag{25}$$

- We consider here that

$$\mathcal{P}_d = \left\{ f_w(x) = \sum_{i=1}^{d} w_i x^i \right\} \,. \tag{26}$$

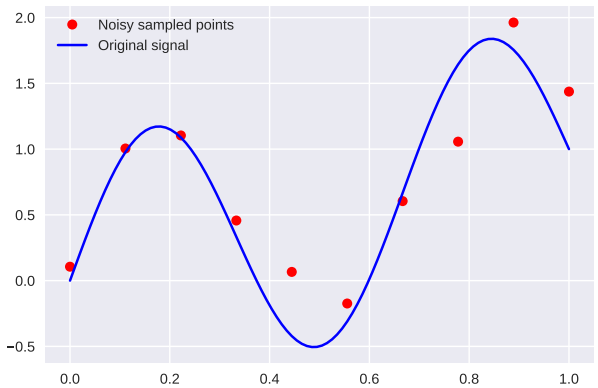
- This corresponds to the choice of basis function $\phi_j(x) = x^j$.
- Justification: polynomials can approximate continuous any function on $[0,1]$.
- LSE:

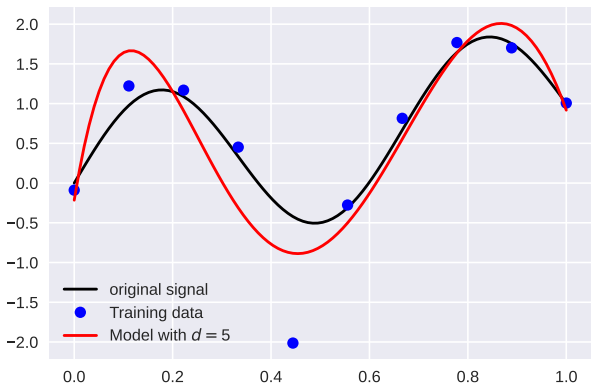
$$\hat{w} \in \operatorname*{argmin}_w E(w) \,, \quad E(w) = \frac{1}{2} \sum_{j=1}^{N} \left\{ y_j - \sum_{i=1}^{d} w_i x_j^i \right\}^2 \,. \tag{27}$$

- Prediction:

$$\hat{y}_{\mathsf{pred}} = f_{\hat{w}}(x_{\mathsf{new}}) \,. \tag{28}$$

Figures and examples





- Here the observations $\{(x_i, y_i)\}_{i=1}^N$, $N = 10$, satisfies

$$y_i = f^\star(x_i) + \sigma_i \varepsilon_i \ , \quad \{\varepsilon_i\}_{i=1}^N \overset{\text{iid}}{\sim} \mathrm{N}(0,1) \ , \quad \{\sigma_i\}_{i=1}^N \subset (0, 0.5) \ . \tag{29}$$

- Function to recover: $f^\star : x \mapsto \sin(2\pi x) + (1/2)\sin(4\pi x)$.
- $f^\star$ is not a polynomial but remember that the choice of $\mathcal{P}_d$ is a model!
- Justification: any function on $[0, 1]$ can be approximated by a sequence of polynomials.

Dependence with respect to $\mathcal{D}$ and d

- Solution associated with $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$:

$$\hat{w}(\mathcal{D}, d) \in \operatorname*{argmin}_w E(w|\mathcal{D}, d) \,, \quad E(w|\mathcal{D}, d) = \frac{1}{2} \sum_{j=1}^{N} \left\{ y_j - \sum_{i=1}^{d} w_i x_j^i \right\}^2 . \tag{30}$$

- The function E and therefore the solution $\hat{w}$ depends on
  - the degree d;
  - the dataset set $\mathcal{D}$.
- Problem: how to compare two solutions with
  - different degreess d_1 and d_2?
  - two different sets of observations $\mathcal{D}_1$, $\mathcal{D}_2$?

Model selection

- Solution associated with $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$:

$$\hat{w}(\mathcal{D}, d) \in \operatorname*{argmin}_w E(w|\mathcal{D}, d) \ , \quad E(w|\mathcal{D}, d) = \frac{1}{2} \sum_{j=1}^{N} \left\{ y_j - \sum_{i=1}^{d} w_i x_j^i \right\}^2 . \tag{31}$$

- Problem: how to compare two solutions with
  - different degrees d_1 and d_2?
- Consider the corresponding solutions $\hat{w}(d_1, \mathcal{D})$ and $\hat{w}(d_2, \mathcal{D})$.
- As usual in many fields (computer science, decision theory (see next)), we would like "a metric" to compare these two solutions.
- More precisely, a function $\mu : \mathcal{P}_d \to \mathbb{R}$ which
  - tries to measure the distance between a model $f \in \mathcal{P}_d$ and the sought function;
  - or at least indicate that which solution $\hat{w}(d_1, \mathcal{D})$ and $\hat{w}(d_2, \mathcal{D})$ should be preferred.

How to discriminate two solutions?

- Solution associated with $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$:

$$\hat{w}(\mathcal{D}, d) \in \operatorname*{argmin}_w E(w|\mathcal{D}, d) \ , \quad E(w|\mathcal{D}, d) = \frac{1}{2} \sum_{j=1}^{N} \left\{ y_j - \sum_{i=1}^{d} w_i x_j^i \right\}^2 . \tag{32}$$

- Problem: how to compare two solutions with
  - different degrees d_1 and d_2?
- Consider the corresponding solutions $\hat{w}(d_1, \mathcal{D})$ and $\hat{w}(d_2, \mathcal{D})$
- and a function $\mu : \mathcal{P} = \cup_{d \in \mathbb{N}^*} \mathcal{P}_d \to \mathbb{R}$ which tries to measure the distance between a model $f \in \mathcal{P}$ and the sought function:

$$\text{Ideally: } \mu_\infty(f) = \sup_{x \in [0,1]} |f(x) - f^\star(x)| \ . \tag{33}$$

- But quite hard to study even theoretically!
- Practically unfeasible since $f^\star$ is unknown.

How to discriminate two solutions?

- Solution associated with $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$:

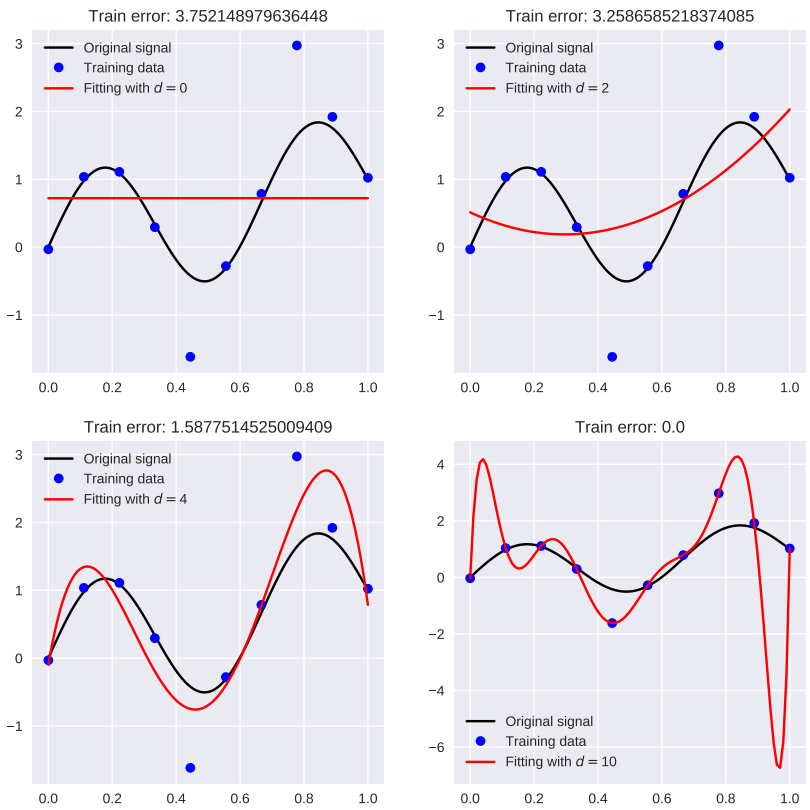
$$\hat{w}(\mathcal{D}, d) \in \operatorname*{argmin}_{w \in \mathbb{R}^d} E(w|\mathcal{D}) \ , \quad E(w|\mathcal{D}, d) = \frac{1}{2} \sum_{j=1}^{N} \left\{ y_j - \sum_{i=1}^{d} w_i x_j^i \right\}^2 . \tag{34}$$

- Problem: how to compare two solutions with
  - different degrees d_1 and d_2?
- Consider the corresponding solutions $\hat{w}(d_1, \mathcal{D})$ and $\hat{w}(d_2, \mathcal{D})$
- Practical function $\mu : \mathcal{P} = \cup_{d \in \mathbb{N}^*} \mathcal{P}_d \to \mathbb{R}$ which tries to measure the distance between a model $f_w \in \mathcal{P}_d$ and the sought function:

$$\text{First practical idea: } \mu_{\mathcal{D}}(f_w) = \sum_{i=1}^{N} \{y_i - f_w(x_i)\}^2 = E(w|\mathcal{D}, d) \ , \tag{35}$$

  if $f = f_w$ for $w \in \mathbb{R}^d$.
- Recall the notation $f_w(x) = \sum_{i=1}^d w_i x^i$.
- $\mu_{\mathcal{D}}$ is called the training error.
- Is it a good idea?

Over and under-fitting



- If d is too large, then $\mu_{\mathcal{D}}(f_w) = 0$.
- However, we can see that $d = 10$ is not necessarily the best choice a priori...
- This phenomenom is called over-fitting.
- The degree of the polynomial is so large that we make no mistake on the training data: $\mu_{\mathcal{D}}(f_w) = 0$.
- The problem of defining a performance metric remains...
- Solution: consider a test data set $\mathcal{D}_{\text{test}}$ disjoint from $\mathcal{D}$!

Evaluation of a model: the root mean-square error

- In practice, $\mathcal{D}$ is in fact divided into two parts: $\mathcal{D} = \mathcal{D}_{\text{test}} \sqcup \mathcal{D}_{\text{train}}$:

$$\mathcal{D}_{\text{train}} = \{(x_i, y_i)\}_{i=1}^{N} , \quad \mathcal{D}_{\text{test}} = \{(\tilde{x}_i, \tilde{y}_i)\}_{i=1}^{\tilde{N}} . \tag{36}$$

- Then, for a fixed degree d, we compute

$$\hat{w}(\mathcal{D}_{\text{train}}, d) \in \underset{w \in \mathbb{R}^d}{\operatorname{argmin}} E(w|\mathcal{D}_{\text{train}}) , \quad E(w|\mathcal{D}_{\text{train}}, d) = \frac{1}{2} \sum_{j=1}^{N} \{y_j - f_w(x_j)\}^2 . \tag{37}$$

- Then, we define the square root mean-square error associated with $\mathcal{D}_{\text{test}}$ for $f_w \in \mathcal{P}$, $w \in \mathbb{R}^d$:

$$\hat{\mu}_{\text{test}}(f_w) = \sqrt{\frac{E(w|\mathcal{D}_{\text{test}}, d)}{N}} = (2\tilde{N})^{-1/2} \left[\sum_{j=1}^{\tilde{N}} \{\tilde{y}_j - f_w(\tilde{x}_j)\}^2 \right]^{1/2} . \tag{38}$$

- Division by $N^{1/2}$ to compare different sizes of data sets on an equal footing.

Example of square root mean-square error

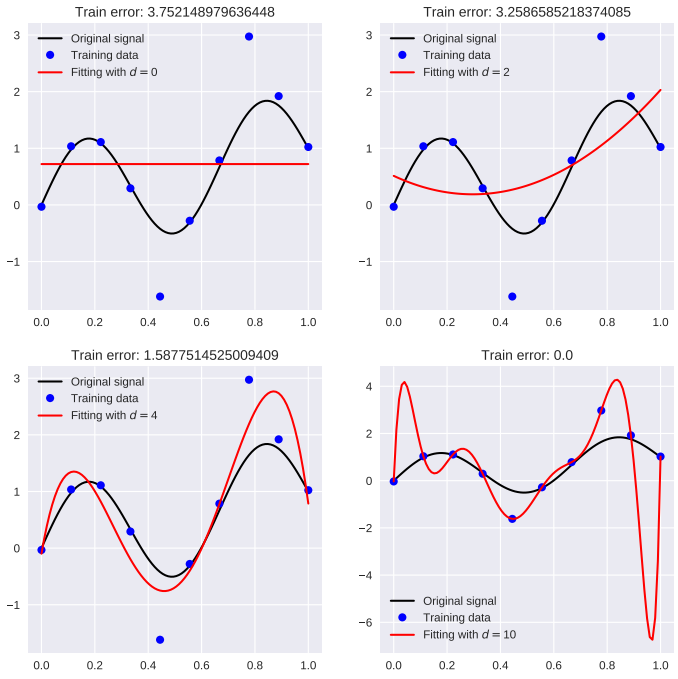


Figure 1: Train errors

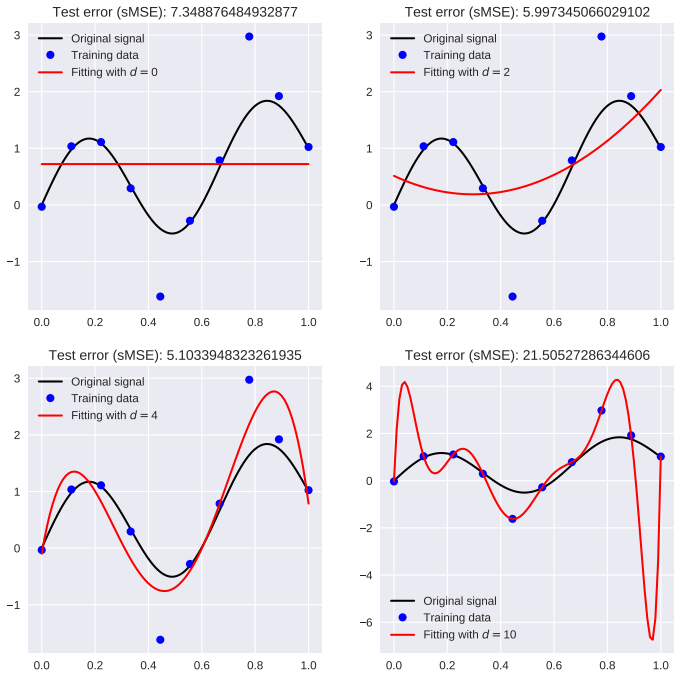


Figure 2: Test errors

- Here $\tilde{N} = 100$.
- Data generation for the test data set is the same than for the train data set.

Inference with multiple datasets?

- Solution associated with $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$:

$$\hat{w}(\mathcal{D}, d) \in \operatorname*{argmin}_w E(w|\mathcal{D}, d) \,, \quad E(w|\mathcal{D}, d) = \frac{1}{2} \sum_{j=1}^{N} \left\{ y_j - \sum_{i=1}^{d} w_i x_j^i \right\}^2 . \tag{39}$$

- Problem: how to compare two solutions with
  - two different sets of observations $\mathcal{D}_1$, $\mathcal{D}_2$?
- Answer:
  - Consider $\mathcal{D}_{1,2} = \mathcal{D}_1 \cup \mathcal{D}_2$ and the solution $\hat{w}(\mathcal{D}_{1,2}, d)$ for this dataset;
  - the more data the better!
  - Constraints: having access to $\mathcal{D}_1$ and $\mathcal{D}_2$.
  - If that is not the case, be extremely cautious in what you do!

Example of square root mean-square error

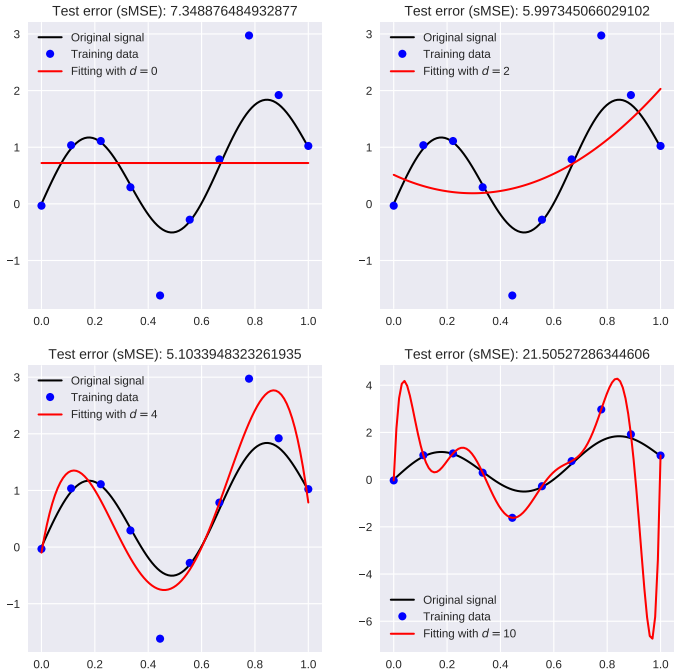


Figure 3: $N = 10$

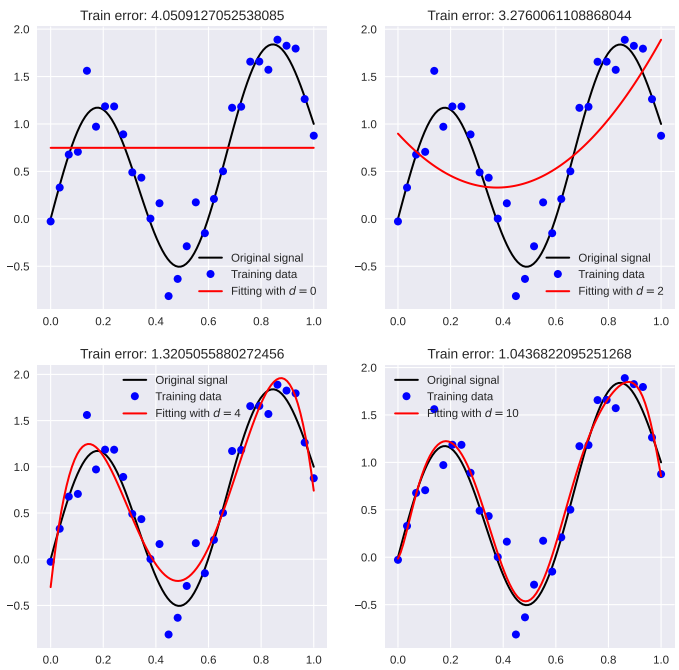


Figure 4: $N = 30$

- Training with $N = 30$
- Here $\tilde{N} = 100$.
- Data generation for the test data set is the same than for the train data set.
- Over-fitting problem becomes less severe.

Digging into the over-fitting problem

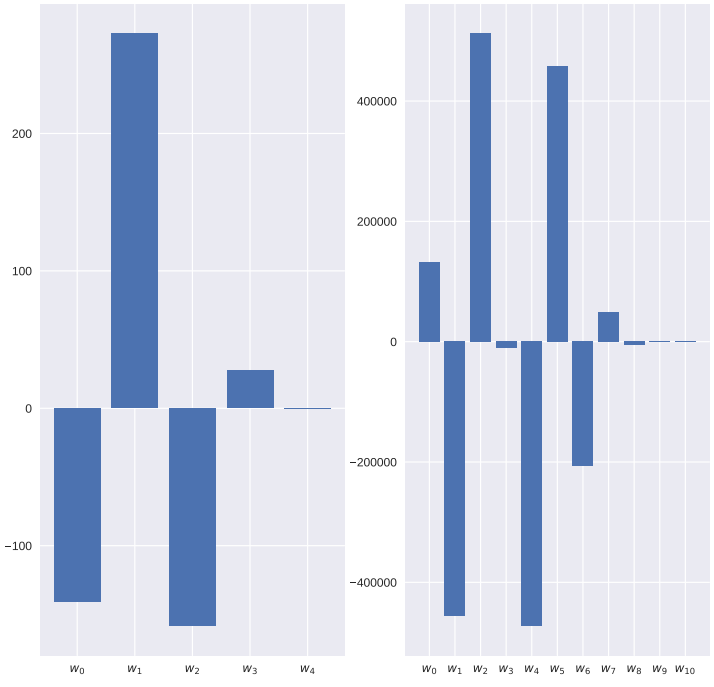


Figure 5: Train errors

- Going back to the over-fitting problem: can it be avoided?
- Let's look the values for $\hat{w}(4, \mathcal{D})$ and $\hat{w}(10, \mathcal{D})$.
- For higher order degree models, coefficients are huge!
- This explains the high variations that we observe in the plots before.
- How to temper this variation?
- Answer: penalize high value for $\|w\|$.

Regularized/penalized polynomial fitting

- We want to impose that $\|w\|$ is "not too large".
- We consider the new error function based on the training data $\mathcal{D}_{\text{train}} = \{(x_i, y_i)\}_{i=1}^N$,

$$E_{\text{reg}}(w|d, \mathcal{D}, \lambda) = E(w|d, \mathcal{D}) + (\lambda/2)\|w\|^2 = \frac{1}{2}\sum_{i=1}^{N}\{y_i - f_w(x_i)\}^2 + \frac{\lambda}{2}\|w\|^2 \ , \tag{40}$$

  where $\lambda \geq 0$ is a regularization parameter.
- E_{reg} is also called the structural risk in opposition with E called the population risk.
- The term structural is coined because we impose some structure on the solution by imposing smaller and larger variations depending on if we use large/small λ.

Regularized/penalized polynomial fitting

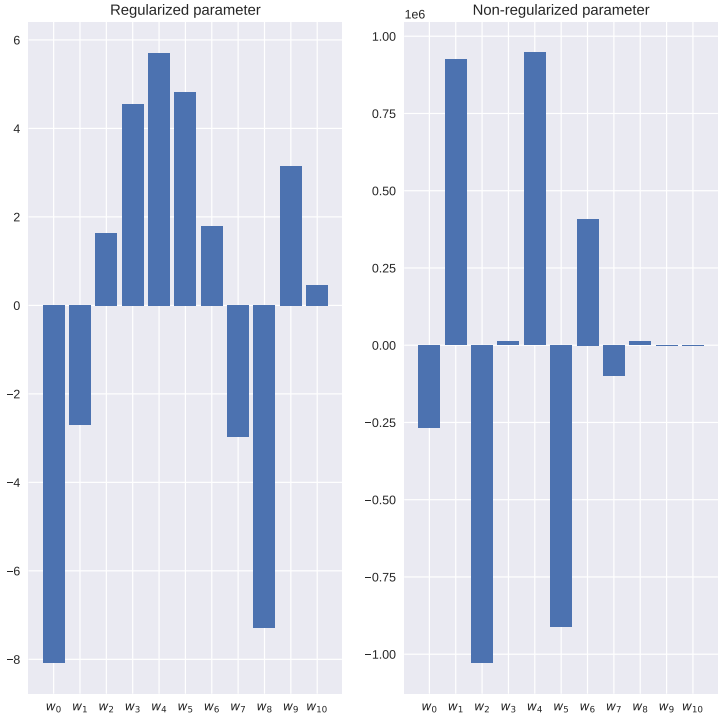
- We want to impose that $\|w\|$ is "not too large".
- We consider the new error function based on the training data $\mathcal{D}_{\text{train}} = \{(x_i, y_i)\}_{i=1}^N$,

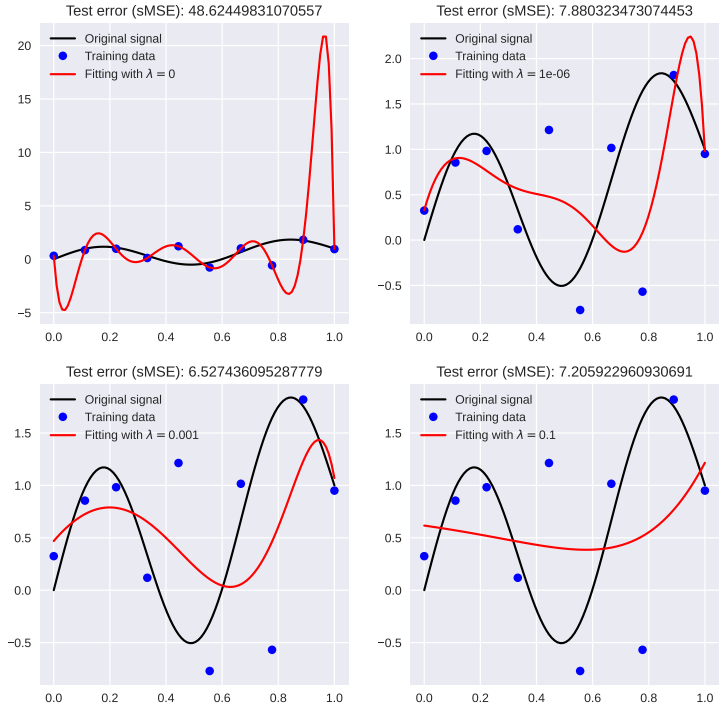
$$E_{\text{reg}}(w|d, \mathcal{D}, \lambda) = E(w|d, \mathcal{D}) + (\lambda/2)\|w\|^2 = \frac{1}{2}\sum_{i=1}^{N}\{y_i - f_w(x_i)\}^2 + \frac{\lambda}{2}\|w\|^2 \ , \tag{41}$$

  where $\lambda \geq 0$ is a regularization parameter.
- E_{reg} is a quadratic function of w therefore admits a unique minimizer.
- The solution of the regularized polynomial fitting problem associated with $\lambda > 0$ is

$$\hat{w}(d, \mathcal{D}, \lambda) \in \operatorname*{argmin}_w E_{\text{reg}}(w|d, \mathcal{D}, \lambda) \ . \tag{42}$$

Benefits of regularized polynomial fitting





- Question how to find the best λ?

Regularity of the problem

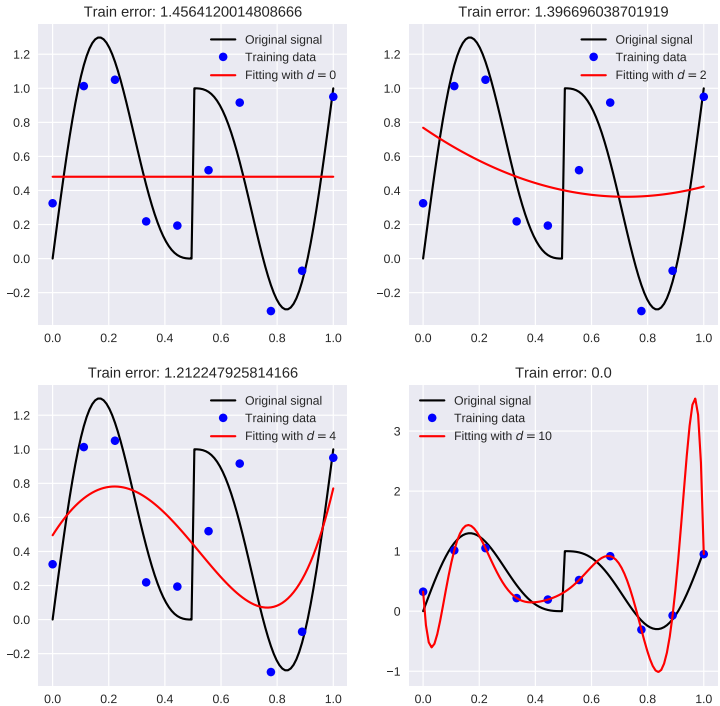


Figure 6: Train errors

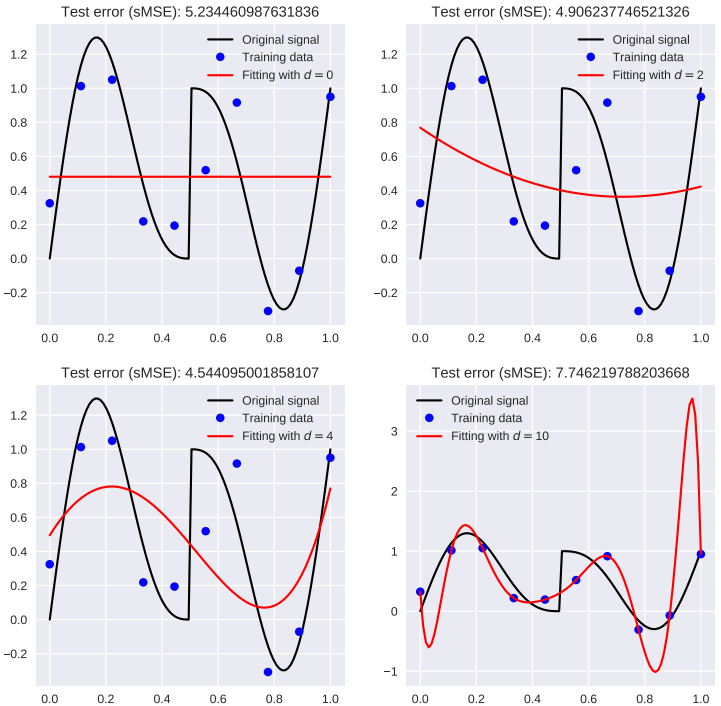


Figure 7: Test errors

- We studied a pretty smooth function $f^\star$.
- What about rougher function?
- In general the problem become much harder!

General linear models: over/under-fitting

- The problem of over/under-fitting still occurs in the general linear models and the least-square estimation problem associated with:

$$E(w; \mathcal{D}, \phi) = \frac{1}{2} \sum_{i=1}^{n} \left\{ y_i - \sum_{j=1}^{d} w_j \phi_j(x_i) \right\}^2 . \tag{43}$$

  and LSE is $\hat{w}_\phi \in \operatorname{argmin} E(w; \mathcal{D}, \phi)$.

- As in the example, we consider the penalized error function/structural risk:

$$E_{\text{reg}}(w; \mathcal{D}, \phi) = \frac{1}{2} \sum_{i=1}^{n} \left\{ y_i - \sum_{j=1}^{d} w_j \phi_j(x_i) \right\}^2 + \lambda \|w\|^2 / 2 \tag{44}$$

$$= \frac{1}{2} [\|y - \Phi_x w\|^2 + \|w\|^2] . \tag{45}$$

- Here we use the notations introduced previously.

- $w \mapsto E_{\text{reg}}(w; \mathcal{D}, \phi)$ is a quadratic function which admits as unique minimizer:

$$\hat{w}(\phi, \mathcal{D}) = (\lambda \mathrm{I}_d + \Phi_x^{\mathrm{T}} \Phi_x)^{-1} \Phi_x^{\mathrm{T}} y . \tag{46}$$

Further reading

- Section 1.4 in [Bis07] about The Curse of Dimensionality
- Section 3.1.3 in [Bis07] about Sequential learning.
- Section 3.1.4 in [Bis07] about other regularizations.
- Section 3.1.5 in [Bis07] about multiple outputs.

Bibliography i

Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. 1st ed. Springer, 2007. ISBN: 0387310738.