

Machine/Statistical Learning Introduction

Florent Chatelain

SICOM, M2 Sigma

2022-23

Organization

Volume

- ▶ $2 \times 2\text{h}$ lecture session
- ▶ $6 \times 2\text{h}$ lesson + Lab session
- ▶ Evaluation : 75% exam (2h) + 25% lab reports

Objectives

- ▶ model/algorithm analysis for supervised/unsupervised learning
- ▶ assess the quality of predictions and inferences
- ▶ application of these algorithms on different datasets (geosciences, bioinformatics, vision, etc ...)

Program

Lectures

- S1 Introduction and fundamental ML tradeoff
- S2 ML Basics

Labs

- S3 ML basics, k-NN, model assesment
- S4 Discriminant Analysis
- S5 Linear models
- S6 Support Vector Machine and Kernel methods
- S7 Unsupervised classification : clustering
- S8 Decision Trees and Random Forests

The material

- ▶ Slides (pdf) and notebooks available here :
<https://gricad-gitlab.univ-grenoble-alpes.fr/chatelaf/ml-sicom3a/>
- ▶ Jupyter notebooks are available to illustrate concepts and methods in Python (.ipynb files)
- ▶ Jupyter-hub servers or Binders are also available to run them remotely and interactively (no need to install Python and its dependencies, see README.md)

Homeworks

- ▶ check the instructions for the next session : [see README.md file](#) on the gitlab repo (updated after each session)
<https://gricad-gitlab.univ-grenoble-alpes.fr/chatelaf/ml-sicom3a>
- ▶ prepare the course at home by reading the materials according to instructions
- ▶ during the session : short recap - Q/A - quiz + illustrations and examples on machines (labs)

Reference books



Trevor Hastie, Robert Tibshirani et Jerome Friedman (2009)
The Elements of Statistical Learning (2nd Edition)
Springer Series in Statistics



Christopher M. Bishop (2007)
Pattern Recognition and Machine Learning *Springer*



Kevin P. Murphy (2012)
Machine Learning. A Probabilistic Perspective *MIT Press*

Supplementary materials, datasets, online courses, ...



<http://www-stat.stanford.edu/~tibs/ElemStatLearn/>



<https://www.coursera.org/course/ml> *very popular MOOC (Andrew Ng)*



<https://work.caltech.edu/telecourse.html> *more involved MOOC (Y. Abu-Mostafa)*

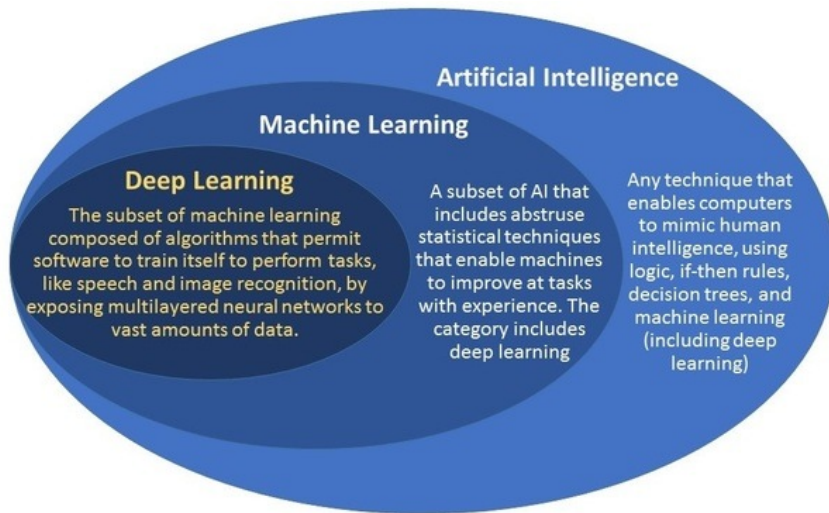


https://scikit-learn.org/stable/auto_examples/index.html *Examples from the sklearn library*



<http://chamilo.grenoble-inp.fr/courses/PHELMA5PMSAST6/> *webpage for this course*

Machine Learning \subset Artificial Intelligence



Data Science Objective

How to extract knowledge or insights from data?

Learning problems are at the cross-section of several applied fields and science disciplines

- ▶ *Machine learning* arose as a subfield of

- ▶ Artificial Intelligence,
- ▶ Computer Science.

Emphasis on large scale implementations and applications : **algorithm centered**

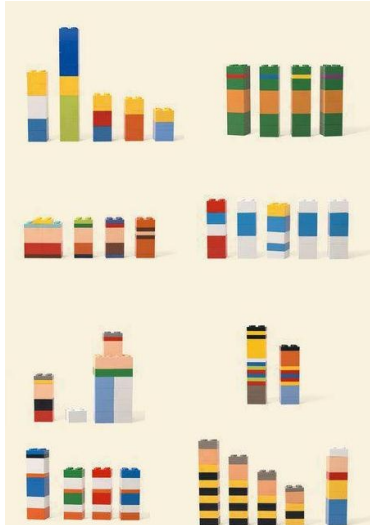
- ▶ *Statistical learning* arose as a subfield of

- ▶ Statistics,
- ▶ Applied Maths,
- ▶ Signal Processing, ...

Emphasizes models and their interpretability : **model centered**

👉 There is much overlap : **Data Science**

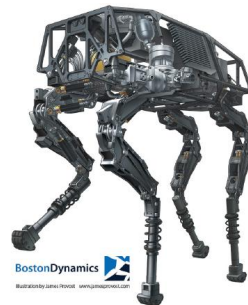
Learning problem



Learning : human vs machine

The learning of a child

- ▶ walking : 1 year
- ▶ speaking : 2 years
- ▶ reasoning : the rest of the time



Definitions of Learning

Machine Learning in Computer Science

Tom Mitchell (The Discipline of Machine Learning, 2006)

A computer program CP is said to learn from **experience E** with respect to some class of **tasks T** and **performance measure P**, if its performance at tasks in T, as measured by P, improves with experience E

Key points

- ▶ Experience E : data and statistics
- ▶ Performance measure P : optimization
- ▶ tasks T : utility
 - ▶ automatic translation
 - ▶ playing Go
 - ▶ ... doing what human does

Definitions of Learning

Machine Learning in Computer Science

Tom Mitchell (The Discipline of Machine Learning, 2006)

A computer program CP is said to learn from **experience E** with respect to some class of **tasks T** and **performance measure P**, if its performance at tasks in T, as measured by P, improves with experience E

Key points

- ▶ Experience E : **data and statistics**
- ▶ Performance measure P : **optimization**
- ▶ tasks T : utility
 - ▶ automatic translation
 - ▶ playing Go
 - ▶ ... doing what human does

Definitions of Learning

Machine Learning in Computer Science

Tom Mitchell (The Discipline of Machine Learning, 2006)

A computer program CP is said to learn from **experience E** with respect to some class of **tasks T** and **performance measure P**, if its performance at tasks in T, as measured by P, improves with experience E

Key points

- ▶ Experience E : **data and statistics**
- ▶ Performance measure P : **optimization**
- ▶ tasks T : utility
 - ▶ automatic translation
 - ▶ playing Go
 - ▶ ... doing what human does

Definitions of Learning

Machine Learning in Computer Science

Tom Mitchell (The Discipline of Machine Learning, 2006)

A computer program CP is said to learn from **experience E** with respect to some class of **tasks T** and **performance measure P**, if its performance at tasks in T, as measured by P, improves with experience E

Key points

- ▶ Experience E : **data and statistics**
- ▶ Performance measure P : **optimization**
- ▶ tasks T : utility
 - ▶ automatic translation
 - ▶ playing Go
 - ▶ ... doing what human does

Experience E : the data !

Type of data : qualitatives / ordinales / quantitatives variables

text strings

speech time series

images/videos 2/3d dependences

networks graphs

games interaction sequences

...

Big data (volume, velocity, variety, veracity)

Data are available without having decided to collect them !

- ▶ importance of preprocessings (cleaning up, normalization, coding,...)
- ▶ importance of a good representation : from raw data to vectors

Objective and performance measures P

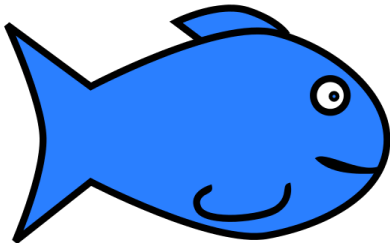
Generalize

- ▶ Perform well (minimize P) on **new data** (fresh data, i.e. unseen during learning)
- ▶ “Statistical learning” : derive good (P /error rate) prediction functions

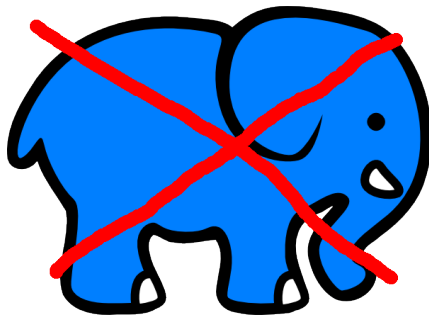
Objective and performance measures P

Generalize

- ▶ Perform well (minimize P) on **new data** (fresh data, i.e. unseen during learning)
- ▶ “Statistical learning” : derive good (P/error rate) prediction functions



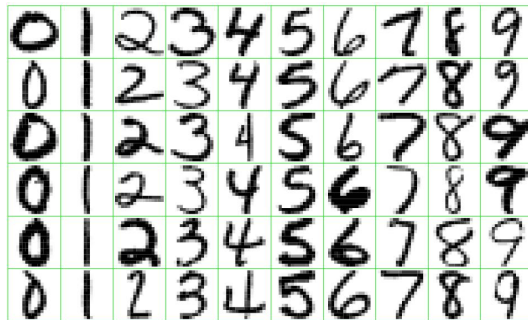
A fish



A fish

Examples of Tasks

Recognition of handwritten digits (US postal envelopes)



- 🔊 Predict the class (0,...,9) of each sample from an image of 16×16 pixels, with a pixel intensity coded from 0 to 255
- ▶ Low error rate to avoid wrong allocations of mails!

Supervised classification

Examples of Tasks

Spams Recognition

Spam

WINNING NOTIFICATION
We are pleased to inform you of the result
of the Lottery Winners International
programs held on the 30th january 2005.
[...] You have been approved for a lump sum
pay out of 175,000.00 euros.
CONGRATULATIONS!!!

No Spam

Dear George,
Could you please send me the report #1248 on
the project advancement?
Thanks in advance.

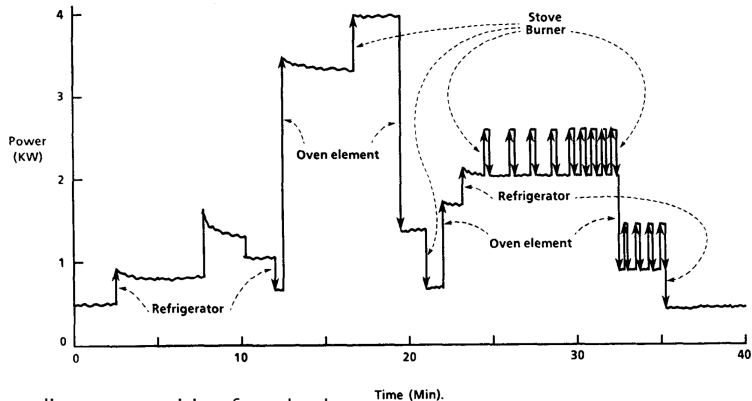
Regards,
Cathia

- 👉 Define a model to predict whether an email is spam or not
- ▶ Low error rate to avoid deleting useful messages, or filling the mailbox with useless emails

supervised classification

Examples of Tasks

Disaggregation/Prediction of appliance's, or industrial, load



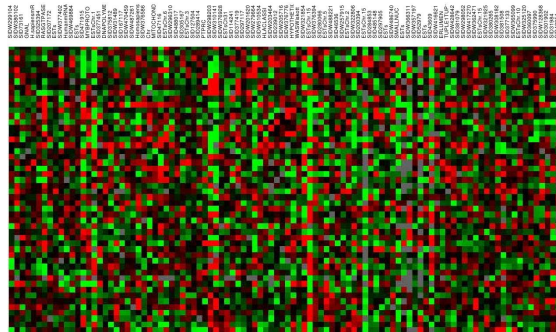
👉 Individual appliance recognition from load curves

👉 Predict the energy consumption

supervised or unsupervised classification or regression

Examples of Tasks

DNA-microarrays

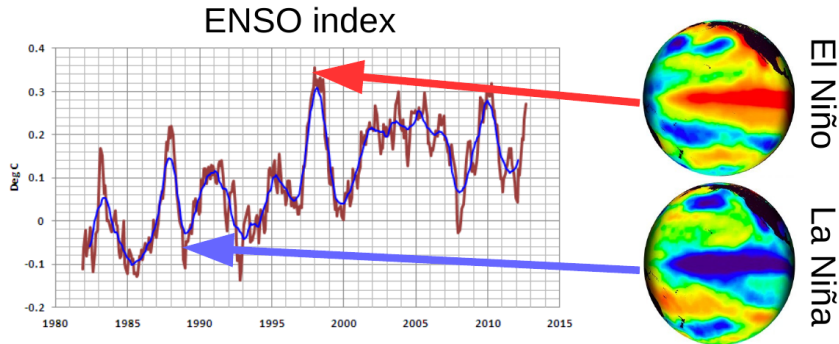


- ▶ Genes expression dataset for several thousand individual genes (columns) and tens of samples (rows)
- ▶ Classification of genes (resp. samples) with similar expression profiles across samples (resp. genes)

unsupervised classification

Examples of Tasks in Geosciences

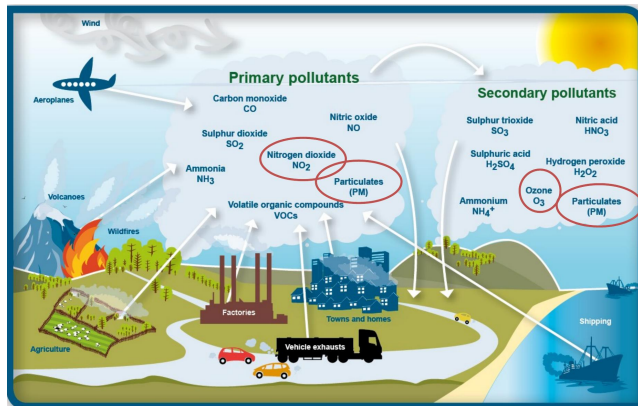
Prediction of El Niño southern oscillation



- 👉 Predict, 6 months in advance, the intensity of an El Niño Southern Oscillation (ENSO) event from ocean-atmosphere datasets (sea level pressure, surface wind components, sea surface temperature, surface air temperature, cloudiness...)

supervised regression

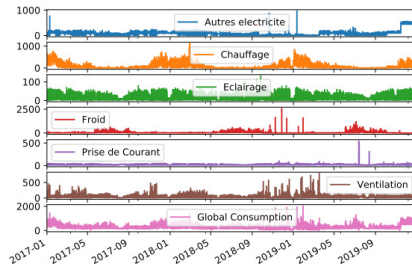
Prediction of pollutant concentrations



- ✎ Predict pollutant concentrations (O_3 , NO_2 , PM_{10} , $PM_{2.5}$) at time $D_0+1, +2, +3$ from hourly measures timeseries + weather data + chemistry based forecasting models

supervised regression (pollutant concentration prediction) / classification (pollution alert or not)

Prediction of Green-ER electrical consumption



- Predict electrical consumption (heating, air conditioning, electrical outlets, or global consumption...) at various time horizons from hourly measures timeseries + weather data + possible Green-ER occupancy data

supervised regression

Definitions

Variable terminology

- ▶ observed data referred to as *input* variables, *predictors* or *features* \leftarrow usually denoted as X
- ▶ data to predict referred to as *output* variables, or *responses* \leftarrow usually denoted as Y

Type of prediction problem : regression vs classification

Depending on the type of the *output* variables

- ▶ when Y are **quantitative** data (continuous variables, e.g. ENSO intensity index values) \leftarrow **regression**
- ▶ when Y are **categorical** data (discrete qualitative variables, e.g. handwritten digits $Y \in \{0, \dots, 9\}$) \leftarrow **classification**

Two very close problems.

Prediction problem

Assumptions

- ▶ couples of input and output variables (X_i, Y_i) are i.i.d.
- ▶ input variables X_i are vectors in \mathbb{R}^p :

$$X_i = (X_{i,1}, \dots, X_{i,p})^T \in \mathcal{X} \subset \mathbb{R}^p$$

- ▶ output variables Y_i take values :
 - ▶ in $\mathcal{Y} \subset \mathbb{R}$ (regression)
 - ▶ in a finite set \mathcal{Y} (classification)

Prediction rule

function of prediction / rule of classification \equiv function $\hat{f} : \mathcal{X} \rightarrow \mathcal{Y}$ that estimate the true link function f to get predictions of new elements Y given X

$$\hat{Y} = \hat{f}(X)$$

Supervised or unsupervised learning

Training set \equiv available sample \mathcal{T} to learn the prediction rule f

For a sized n training set, different cases :

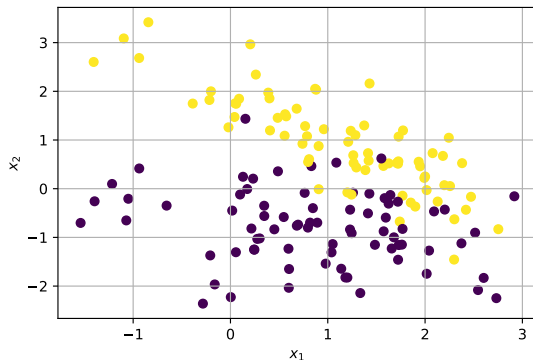
- ▶ **Supervised learning** : $\mathcal{T} \equiv ((X_1, Y_1), \dots, (X_n, Y_n))$ input/output couples are available to learn the prediction rule f
- ▶ **Unsupervised learning** : $\mathcal{T} \equiv (X_1, \dots, X_n)$ only the inputs are available
- ▶ **Semi-supervised** : mixed scenario (often encountered in practice, but less information than in the supervised case)

During this course :

- ▶ most courses and labs devoted to supervised learning (more interpretable results, abundant literature)
- ▶ S6 devoted to unsupervised learning, namely *clustering*

Binary classification

Toy 2D data set (two features X_1 and X_2) for binary classification (two classes)



- ▶ each sample (X_1, X_2) in the dataset is plotted as a 2D point where the two features X_1 and X_2 are displayed along the abscissa and ordinate axes respectively
- ▶ the binary class label Y is displayed as a color mark (e.g., yellow or purple)

Simple linear model for classification

We seek a prediction model based on the linear regression of the outputs $Y \in \{-1, 1\}$:

$$Y = \beta_1 X_1 + \beta_2 X_2 + \epsilon,$$

where $\beta = (\beta_1, \beta_2)^T$ is a 2D unknown parameter vector

Learning problem \Leftrightarrow Estimation of β

Least Squares Estimator $\hat{\beta} = (\hat{\beta}_1, \hat{\beta}_2)^T$: minimize the training error rate (quadratic cost sense)

$$RSS(\beta) = \sum_{i=1}^N (Y_i - \beta_1 X_{i,1} - \beta_2 X_{i,2})^2$$

Classification rule based on least squares regression

$$\hat{f}(X) = \begin{cases} 1 & \text{if } \hat{Y} = \hat{\beta}_1 X_1 + \hat{\beta}_2 X_2 \geq 0, \\ -1 & \text{otherwise} \end{cases}$$

Notebook `N1_Linear_Classification.ipynb`

Model complexity

Most of methods have a complexity related to their *effective* number of parameters

Linear classification : model order p

E.g. d th degree polynomial regression : $p = d + 1$ parameters β_k s.t.

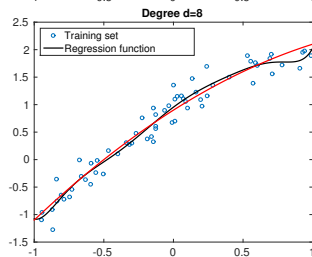
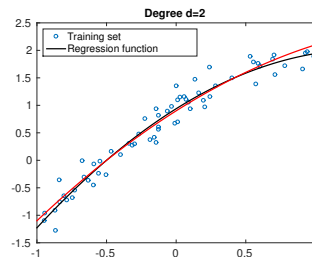
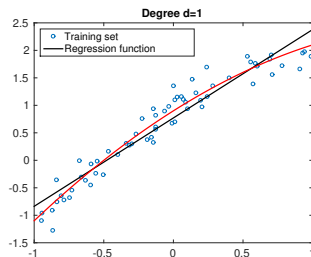
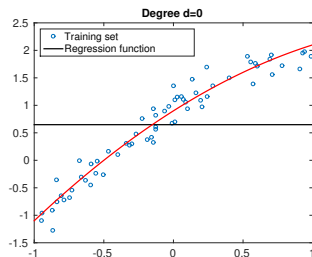
$$\begin{aligned} Y &= \beta_0 + \beta_1 x + \beta_2 x^2 + \dots + \beta_d x^d + \epsilon, \\ &= \mathbf{X}_d \boldsymbol{\beta}_d + \epsilon, \end{aligned}$$

where

$$\begin{aligned} \mathbf{X}_d &= \begin{bmatrix} 1, & x, & x^2, & \dots, & x^d \end{bmatrix}, \\ \boldsymbol{\beta}_d &= [\beta_0, \beta_1, \beta_2, \dots, \beta_d]^T. \end{aligned}$$

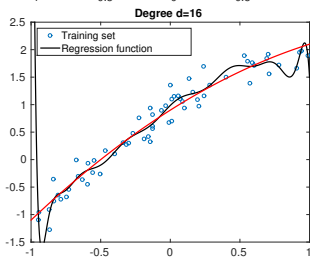
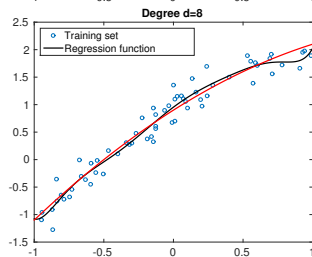
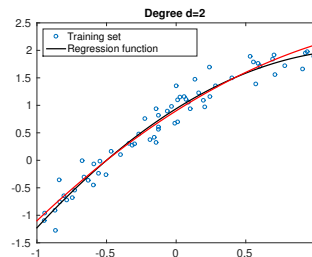
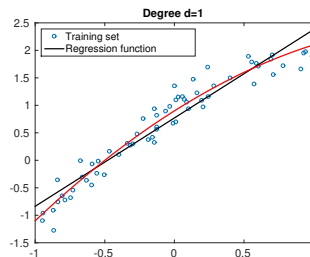
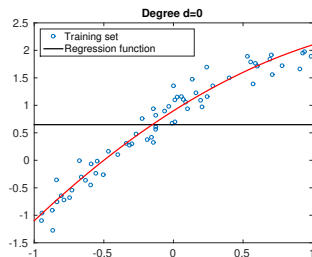
Linear regression : complexity vs stability

Polynomial degree d influence ← over-fitting issue



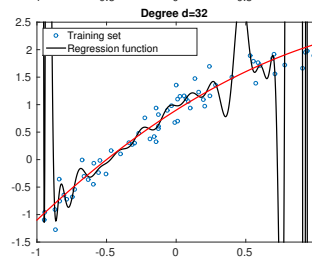
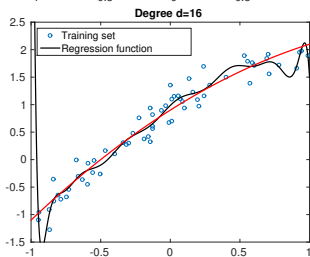
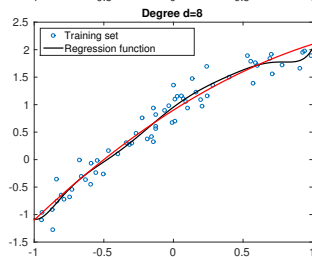
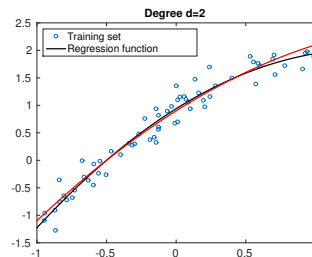
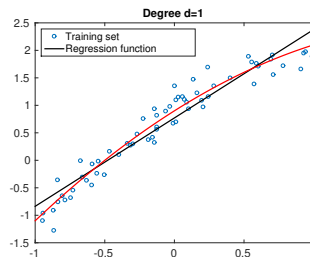
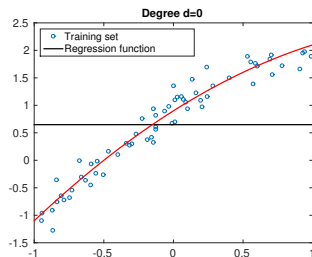
Linear regression : complexity vs stability

Polynomial degree d influence ← over-fitting issue



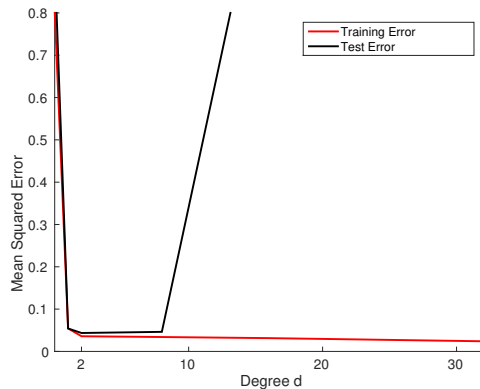
Linear regression : complexity vs stability

Polynomial degree d influence ← over-fitting issue



Linear Regression : Test error vs Train Error

Error rate vs polynomial order d

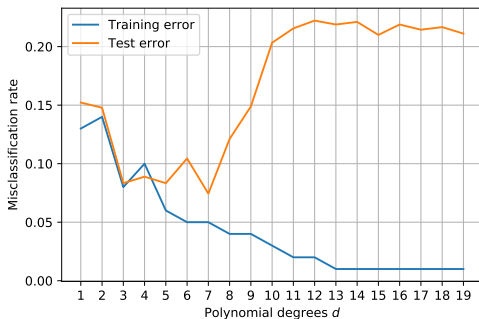


- ▶ True error rate (i.e. error rate for test data not used for learning) minimized when $d = 2 \dots$
- ▶ ... true generative model : order $d = 2$ polynomial (+ white noise)

👉 Training error always decrease with the model complexity. **Can't use alone to select the model !**

Binary classification : Test error vs Train Error

Notebook : `N2_Polynomial_Classification_Model_Complexity.ipynb`



Error rate vs polynomial order d (see Notebook)

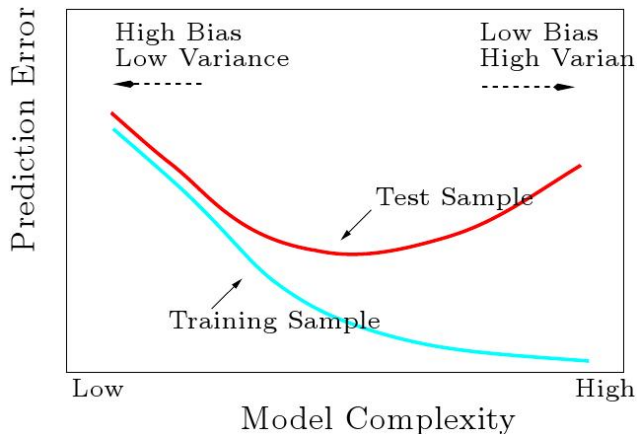
- ▶ Training error rate (i.e. error rate for train data used for learning) minimized when $d = 19$
- ▶ Test error rate (i.e. error rate for test data not used for learning) minimized when $3 \leq d \leq 7$...

👉 Training error always decrease with the model complexity. **Can't use alone to select the model !**

Model Selection

Fundamental trade-off

- ▶ too simple model (high bias) → **under-fitting**
- ▶ too complex model (high variance) → **over-fitting**



Fundamental Bias-Variance trade-off

if the true model is

$$Y = f(X) + \epsilon,$$

then for any prediction rule $\hat{f}(X)$, Mean Squared Error (MSE) expresses as

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

- ▶ $\text{Var} [\epsilon]$ is the *irreducible* part
- ▶ as the flexibility of $\hat{f} \nearrow$, its variance \nearrow and the bias \searrow
👉 overfitting/underfitting trade-off

Overview of Bias-Variance

