Introduction

Machine learning, 2021

Mansoor Rezghi
Mrezghi.ir
Department of Computer science, TMU

Ref:KM, CB

References

- R. O. Duda, P.E. Hart, D. G. Stork, Pattern Classification, Second Edition, Wiley, 2001.(DU)
- K. Murphy Machine Learning: A Probabilistic Perspective, MIT Press, 2012.(KM)
- A. Webb, Statistical machine learning, 2011, Wiley.(AW)

  A. Geron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2019.
- Ian Goodfellow and Yoshua Bengio and Aaron Courville, Deep Learning, MIT press(IG)
- M. Zaki, Data Mining and Machine Learning: Fundamental Concepts and Algorithms (MZ)

- C. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.(CB)

Grading

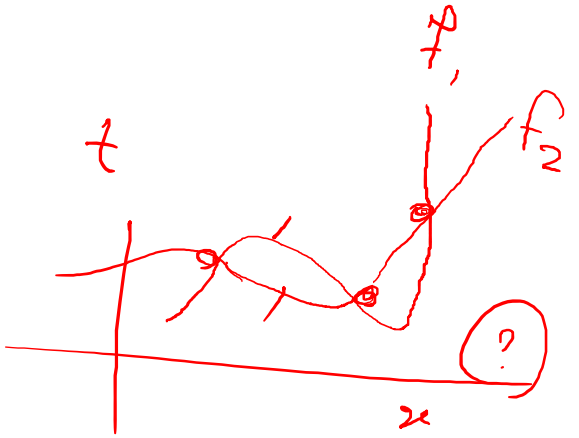
- Homework's (%35)
- Quiz and Mid-Term Exam(%25)
- Final Exam(%40)
- Final Project(Grade from -2 to 4)
- Mid term: آذر 9Nov30

HomeWorks must be submitted on time

- Programming

  Default: Python

Machine Learning



$f_1(x_i) = f_2(x_i)$

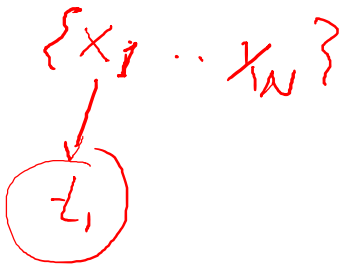
- ML is a set of methods that can use features in data to detect patterns in data and then use the discovered patterns to predict the future or make other decisions under uncertainty.

$\{x_1 \dots x_N\}$ $t_1 \dots t_N$ → $f(x_i) \sim t_i$

x_i : ith sample

$x_i \in \mathbb{R}^n$

$x_i = \begin{bmatrix} x_i' \\ \vdots \\ x_n' \end{bmatrix}$ → feature. → property.



Machine Learning

- Predictive (supervised Learning)

  classification, regression
- Descriptive (Unsupervised Learning)

  clustering, …
- Reinforcement Learning

- Parametric

  SVM, generative methods
- Nonparametric

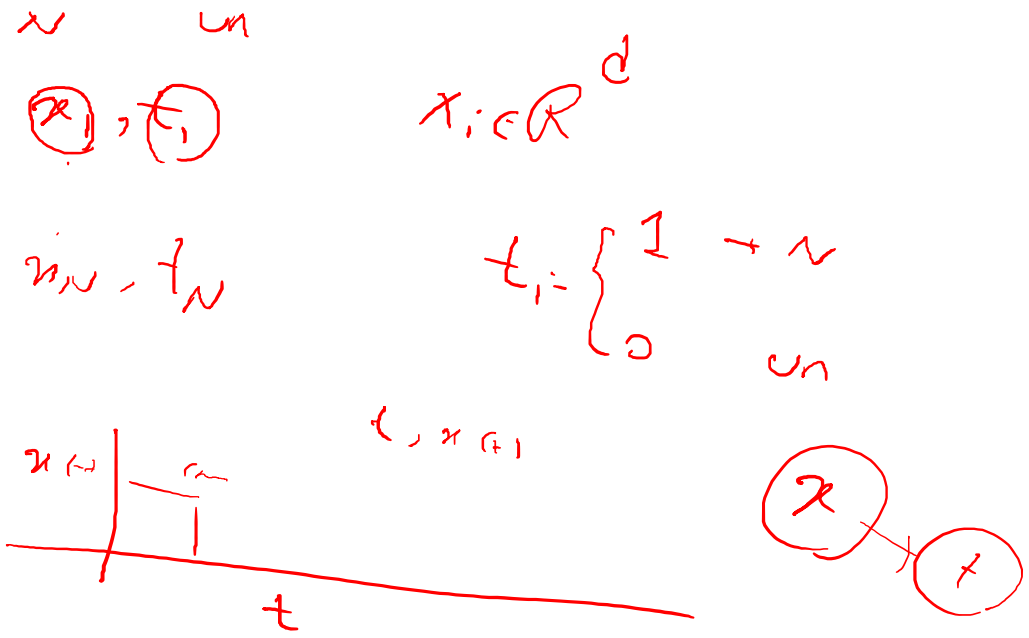
  KNN classifier, K-means

Supervised Learning

Given : {(x_i,y_i), i=1,…,N}

x_i: features of the i 'th sample

y_i: Target Value



Goal: Learn Mapping y=f(x) by using given data and predict the value of y for new data x

Supervised: Classification

$y = f(x)$

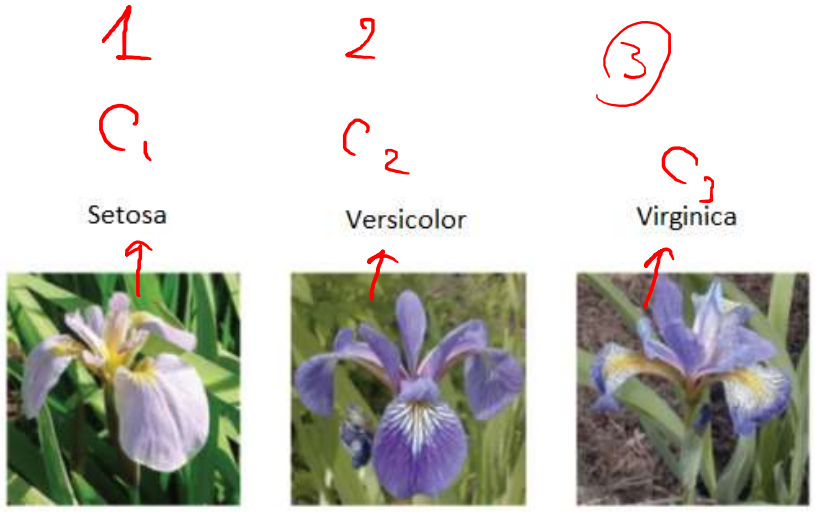
- Classification

  Train Data $x_1, \ldots\ldots, x_N$

  Labels $y_1, \ldots, y_N \in \{1, \ldots, C\}$

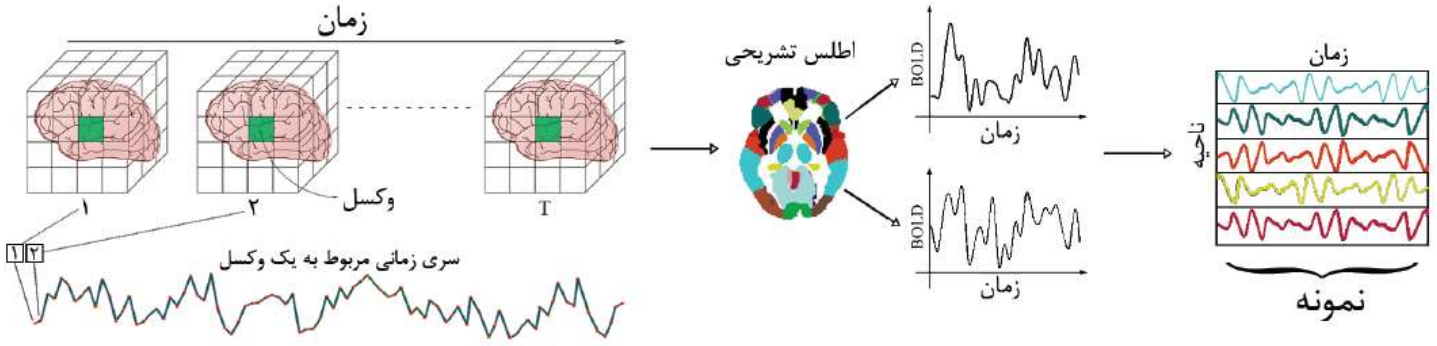
  - Test

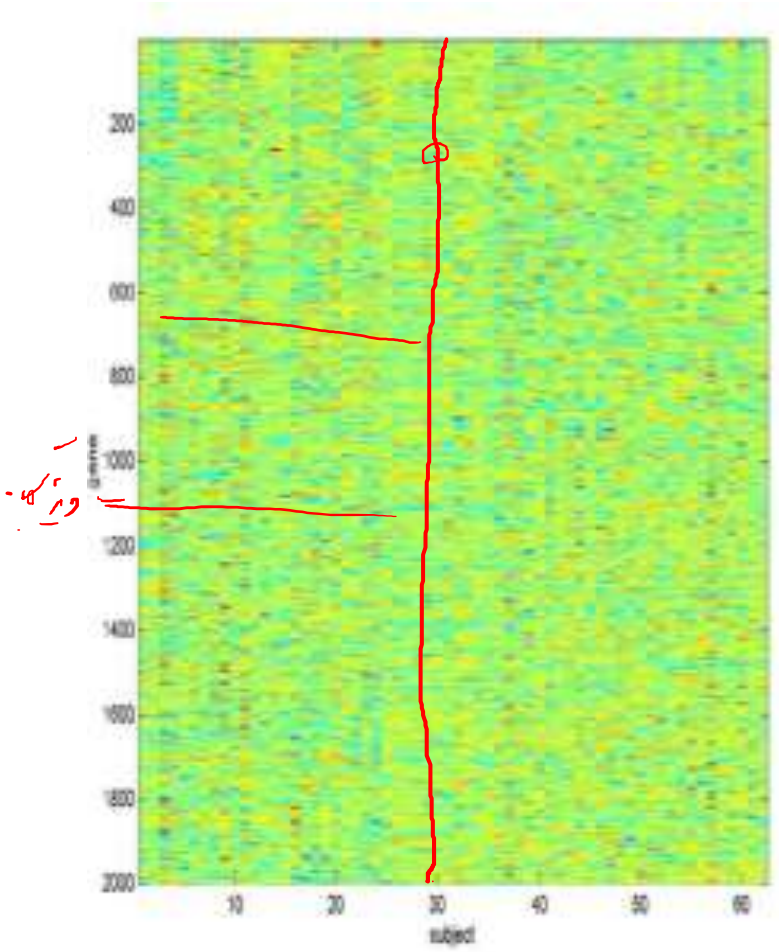
    For test data x predict its label
    $y \in \{1, \ldots, C\}$



Classification :Applications

- Document classification(email spam filtering)
- Face recognition
- Diagnosis
- ...





Task: Classify gene expression datasets into different categories, e.g., normal v.s. cancer

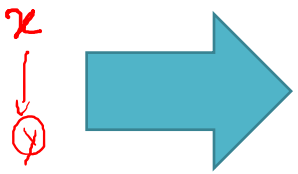
Challenge: Thousands of genes measured in the micro-array data. Only a small subset of genes are probably correlated with the classification task.

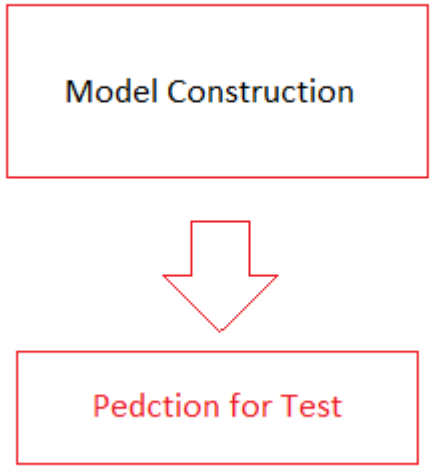
Supervised: Regression

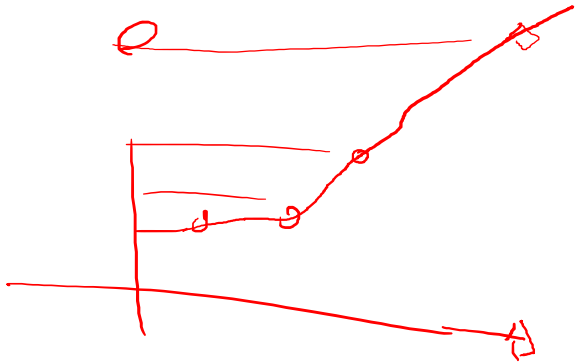
Train Data $x_1, \ldots\ldots, x_N$

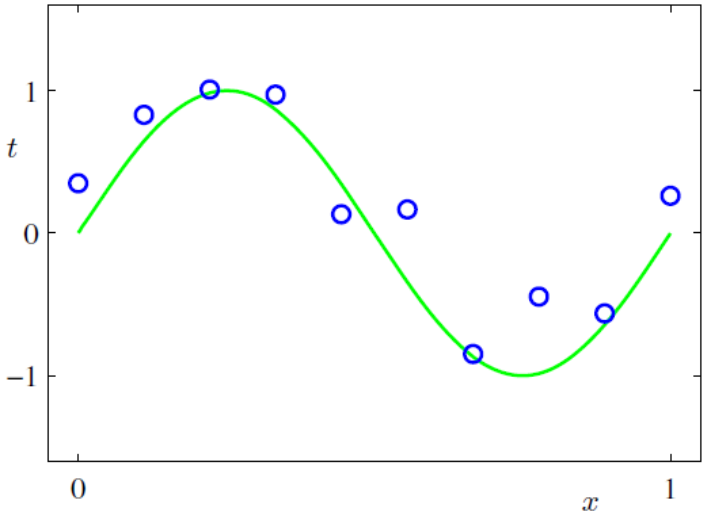
Target values $y_1, \ldots\ldots, y_N$

What's the target y of test data x?





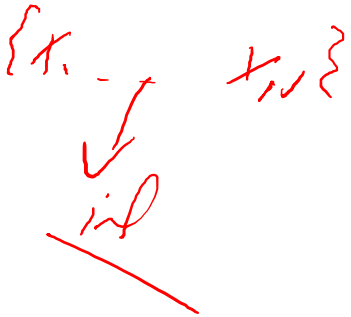




Regression: Applications

- Predict tomorrow's stock market price given current market conditions and other possible side information.
- Predict the age of a viewer watching a given video on YouTube.
- Predict the location in 3d space of a robot arm end effector, given control signals (torques) sent to its various motors.
- Predict the amount of prostate specific antigen (PSA) in the body as a function of a number of different clinical measurements.
- Predict the temperature at any location inside a building using weather data, time, door sensors, etc.

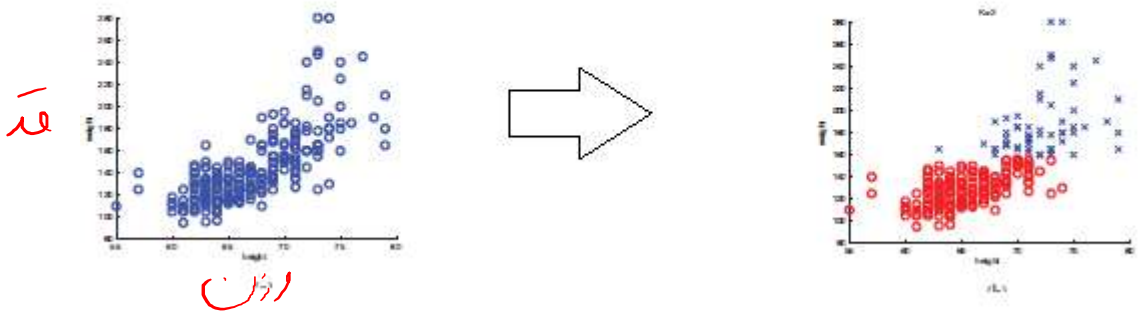
Unsupervised Learning



Unsupervised learning is arguably more typical of human and animal learning. It is also more widely applicable than supervised learning, since it does not require a human expert to manually label the data. Labeled data is not only expensive to acquire[6], but it also contains relatively little information, certainly not enough to reliably estimate the parameters of complex models.

When we're learning to see, nobody's telling us what the right answers are — we just look. Every so often, your mother says "that's a dog", but that's very little information. You'd be lucky if you got a few bits of information — even one bit per second — that way. The brain's visual system has 10^{14} neural connections. And you only live for 10^9 seconds. So it's no use learning one bit per second. You need more like 10^5 bits per second. And there's only one place you can get that much information: from the input itself. — Geoffrey Hinton, 1996 (quoted in (Gorder 2006)).

Unsupervised: Clustering



- In astronomy, the **autoclass** system (Cheeseman et al. 1988) discovered a new type of star, based on clustering astrophysical measurements.
- In e-commerce, it is common to cluster users into groups, based on their purchasing or web-surfing behavior, and then to send customized targeted advertising to each group (see e.g., (Berkhin 2006)).
- In biology, it is common to cluster flow-cytometry data into groups, to discover different sub-populations of cells (see e.g., (Lo et al. 2009)).

Unsupervised: Other applications

- Latent factors
- Graph structure
- Matrix completion
- Image Inpainting (image super resolution)
- Recommender systems
- Market basket analysis (frequent itemset learning)
- Text Mining
- Web mining
- Bioinformatics

Complex networks

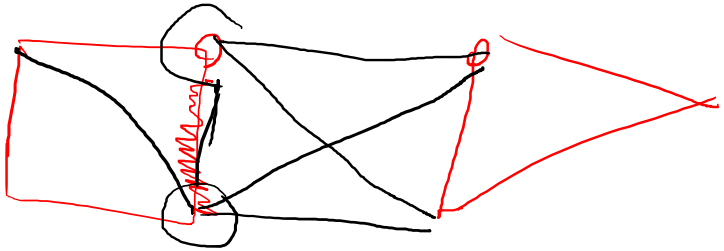
Community detection

Diffusion

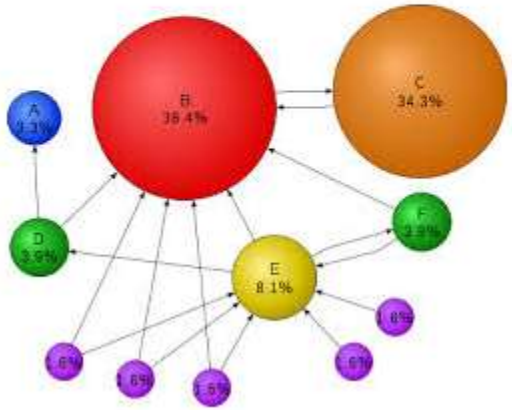
Centrality

Link Prediction

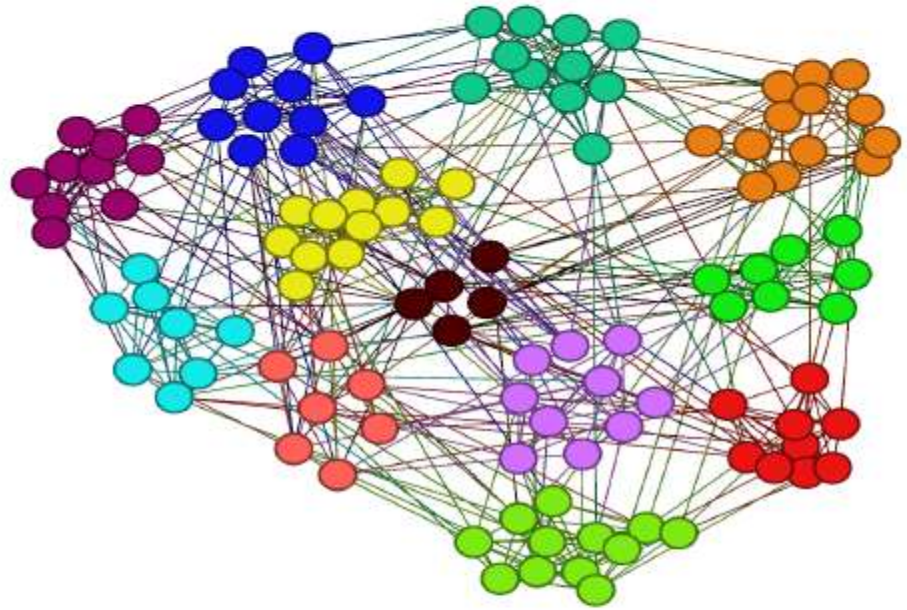
Ranking



page Rank



Social Network Analysis



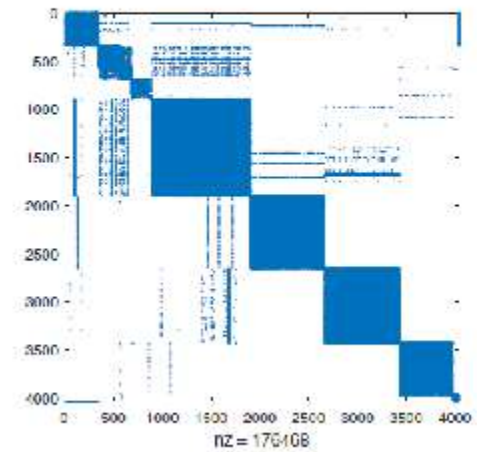
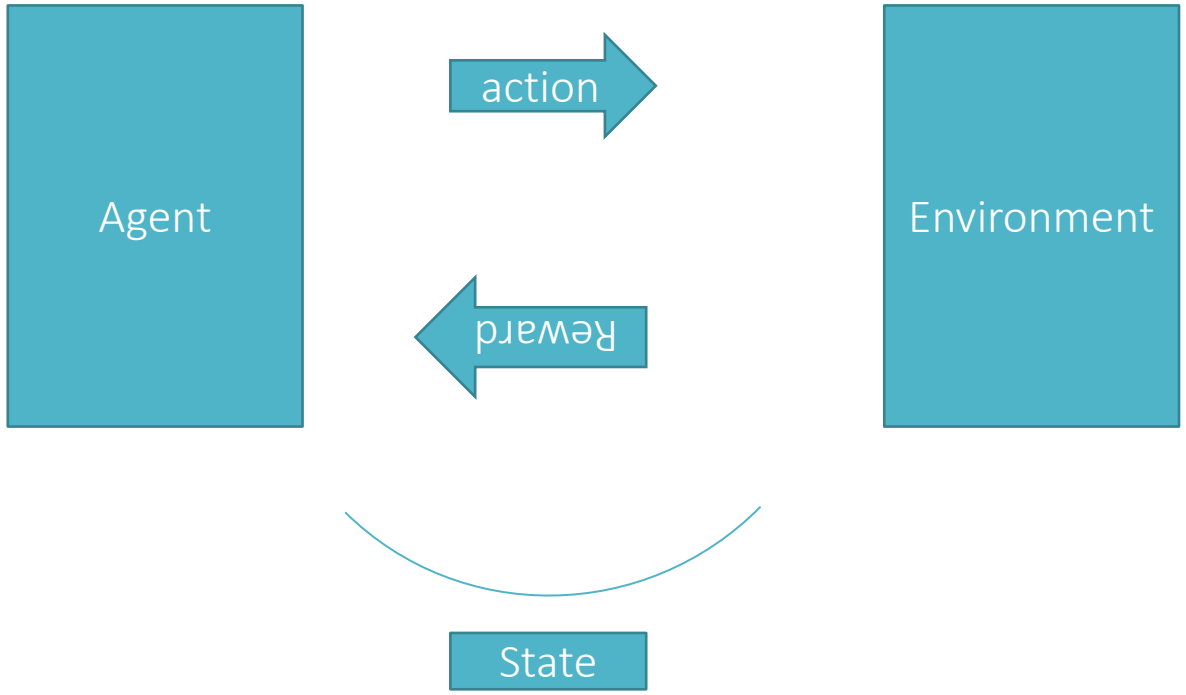


Figure 11: Facebook social network with 4039 nodes

$\{x_i, y_i\}$

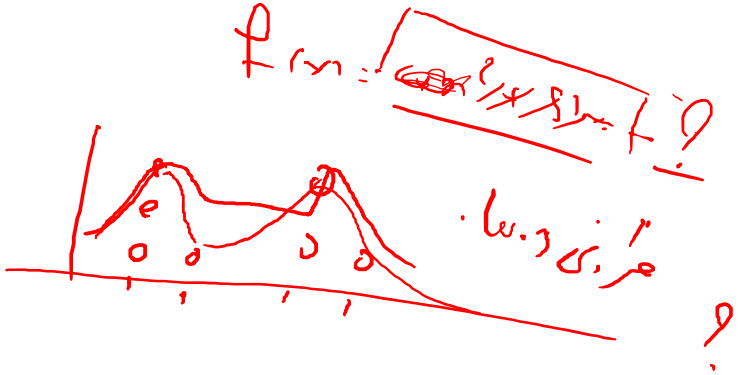
Reinforcement Learning



Parametric VS Nonparametric

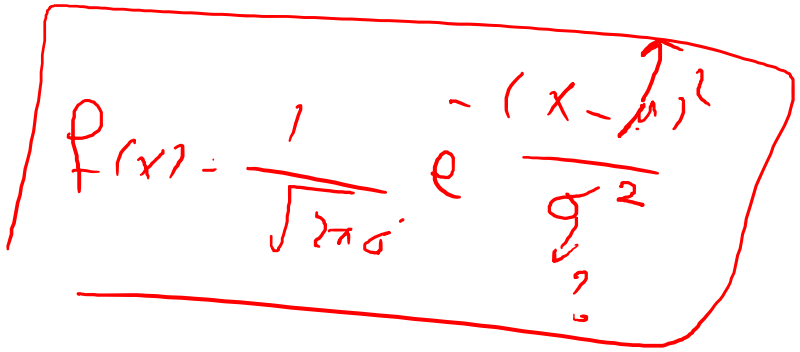
- Parametric:

  Example: KNN classifier



- Non-Parametric:

  Example: SVM classifier



Nonparametric: KNN-classifier

(2)

$$p(y = c|\mathbf{x}, \mathcal{D}, K) = \frac{1}{K} \sum_{i \in N_K(\mathbf{x}, \mathcal{D})} \mathbb{I}(y_i = c)$$

$$\mathbb{I}(e) = \begin{cases} 1 & \text{if } e \text{ is true} \\ 0 & \text{if } e \text{ is false} \end{cases}$$

3 NN

MAP estimate $\hat{y}(\mathbf{x}) = \text{argmax}_c(y = c|\mathbf{x}, \mathcal{D})$.

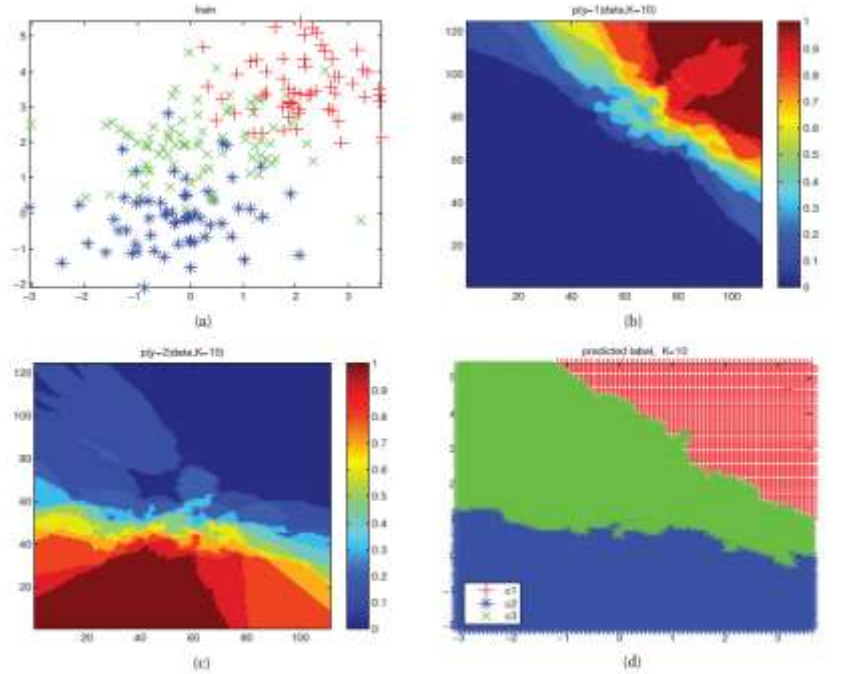


Figure 1.15 (a) Some synthetic 3-class training data in 2d. (b) Probability of class 1 for KNN with $K = 10$. (c) Probability of class 2. (d) MAP estimate of class label. Figure generated by `knnClassifyDemo`.

Parametric Classifier(Example)

$$g(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + w_0$$

with decision rule

$$\boldsymbol{w}^T\boldsymbol{x} + w_0 \begin{cases} > & 0 \\ < & 0 \end{cases} \Rightarrow \boldsymbol{x} \in \begin{cases} \omega_1 \text{ with corresponding numeric value, } y_i = +1 \\ \omega_2 \text{ with corresponding numeric value, } y_i = -1 \end{cases}$$

Parametric statistical method

$$p(y|\mathrm{x}, \theta) = \mathcal{N}(y|\mu(\mathrm{x}), \sigma^2(\mathrm{x}))$$

$\{x_1 \dots x_N\}$

$t_1 \dots t_N$

Train

$x \downarrow t$

$y_i = f(x_i)$ → model.

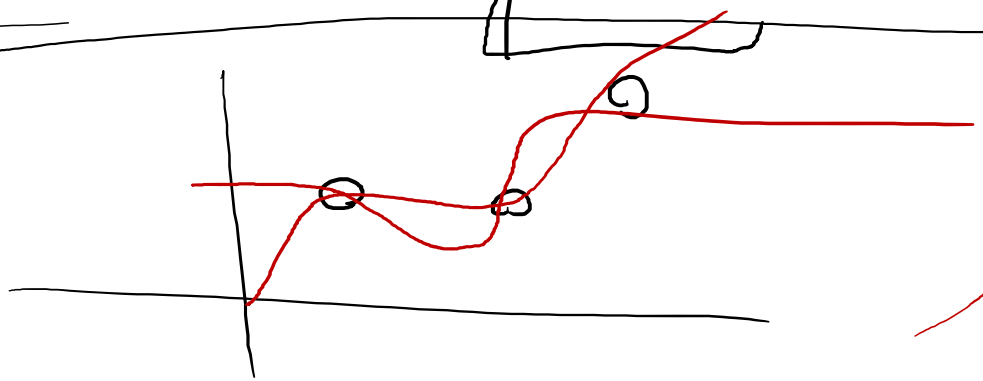
$\min_\theta \sum_{i=1}^{N} \text{loss}(f(x_i), y_i)$

?

Constant

Non-constant

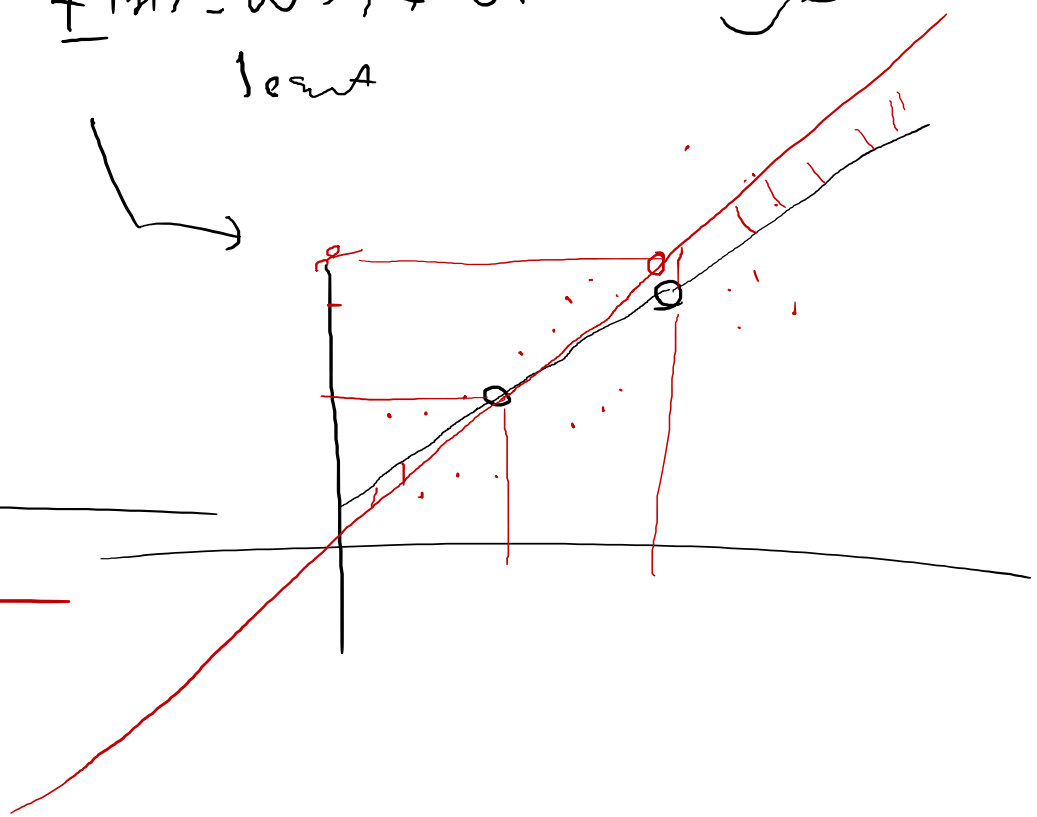
$f(x_i) = y_i$

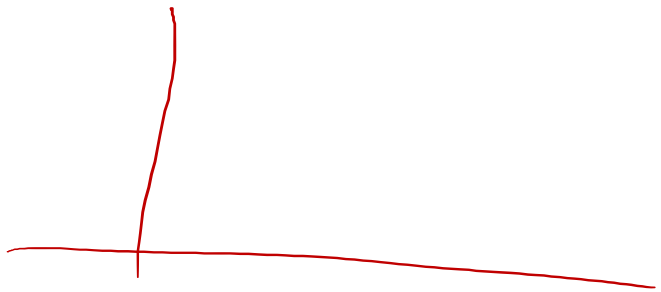


$f(x) = w^T x + w_0$

linear

نمو





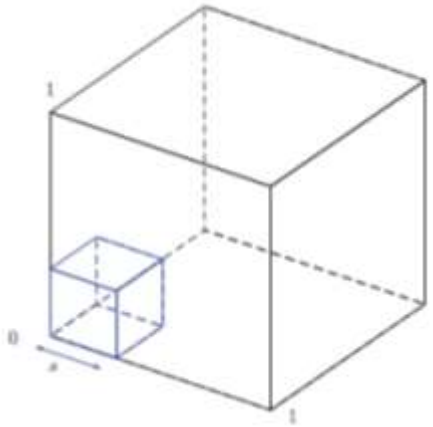
Learning systems Performance

- Error rate (Prob. of misclassification)
- Speed (throughput)
- Cost
- Robustness
- Reject option

Challenges: Curse of dimensionality

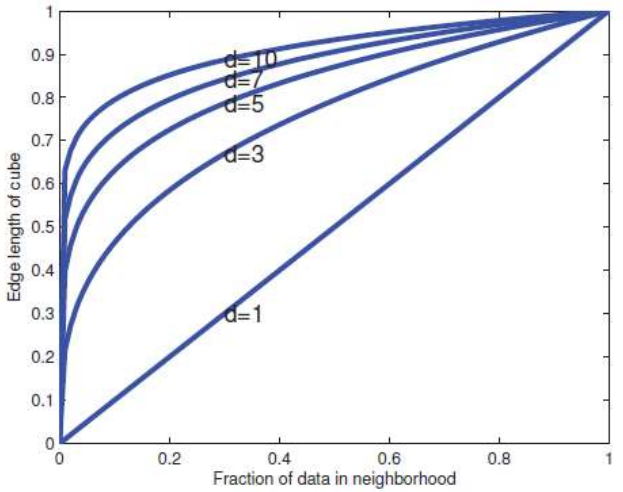
Example 1: Locality

- KNN classifier on Input data from distributed Uniformly On D-dimensional Cube
- The density of class labels around test point x by “growing” a hypercube around x until contains a desired fraction f of the data points



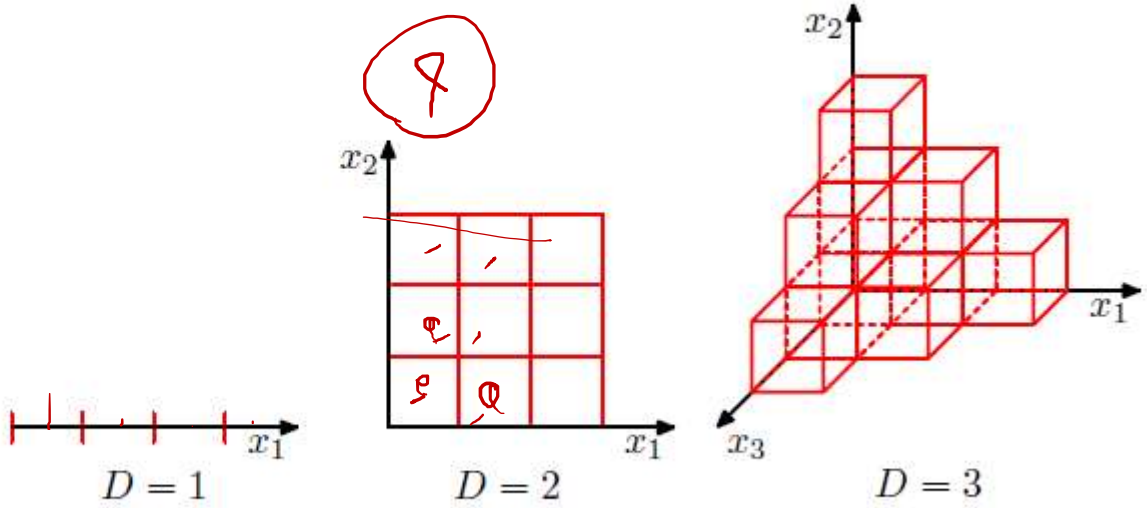
The expected edge length of this cube will be $e_D(f) = f^{1/D}$

$e_{10}(0.1) = 0.8$, $e_{10}(0.01) = 0.63$



Other viewpoint: Sparsity

Illustration of the curse of dimensionality, showing how the number of regions of a regular grid grows exponentially with the dimensionality D of the space. For clarity, only a subset of the cubical regions are shown for $D = 3$.



$x \in R$ $\quad (x_i, y_i) \rightarrow$ $\underline{Y} \sim a_i$

$$p(m) = a_0 + a_1 x + a_2 x^2 + \boxed{a_3 x^3}$$

$$m_i = \sum \phi_{pss}(p(x_i, y_i))$$

a_0, a_1, a_2 ~~a_3~~ (3)

$x \in R^2$

$$p(m) = a_0 + a_1 x_1 + a_2 x_2 + a_{12} x_1 x_2 + \boxed{a_{x_1 x_2}} + a_{x_2} x_2^2 + a_{x_1} x_1^2$$ (6)

Curse of dimensionality

- Curve Fitting

$$y(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{i=1}^{D} w_i x_i + \sum_{i=1}^{D}\sum_{j=1}^{D} w_{ij} x_i x_j + \sum_{i=1}^{D}\sum_{j=1}^{D}\sum_{k=1}^{D} w_{ijk} x_i x_j x_k.$$

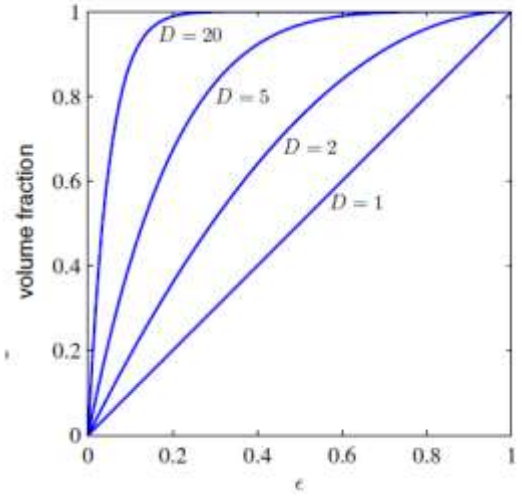
For data of dimesion D the number of parameters of a polynomial with degree M is D^M

- Example 3

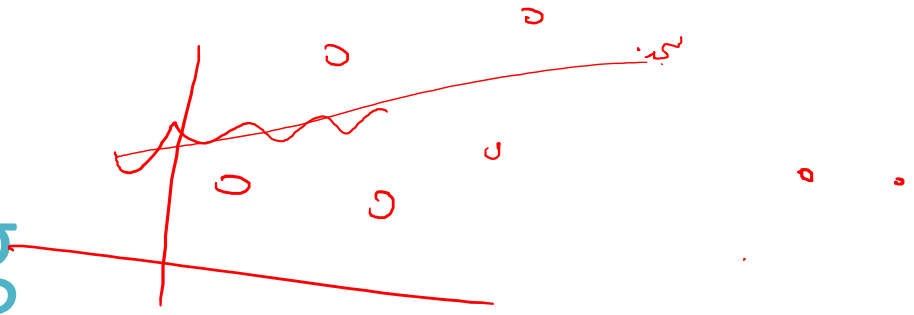
$$V_D(r) = K_D r^D$$

$$\frac{V_D(1) - V_D(1-\epsilon)}{V_D(1)} = 1 - (1-\epsilon)^D$$

Plot of the fraction of the volume of a sphere lying in the range $r = 1-\epsilon$ to $r = 1$ for various values of the dimensionality D.



Overfitting& Udefitting

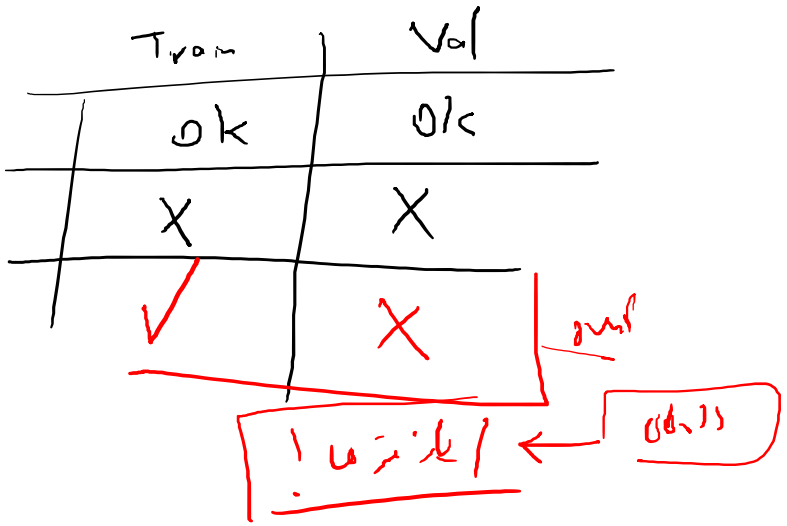


When we fit highly flexible models, we need to be careful that we do not **overfit** the data, that is, we should avoid trying to model every minor variation in the input, since this is more likely to be noise than true signal.

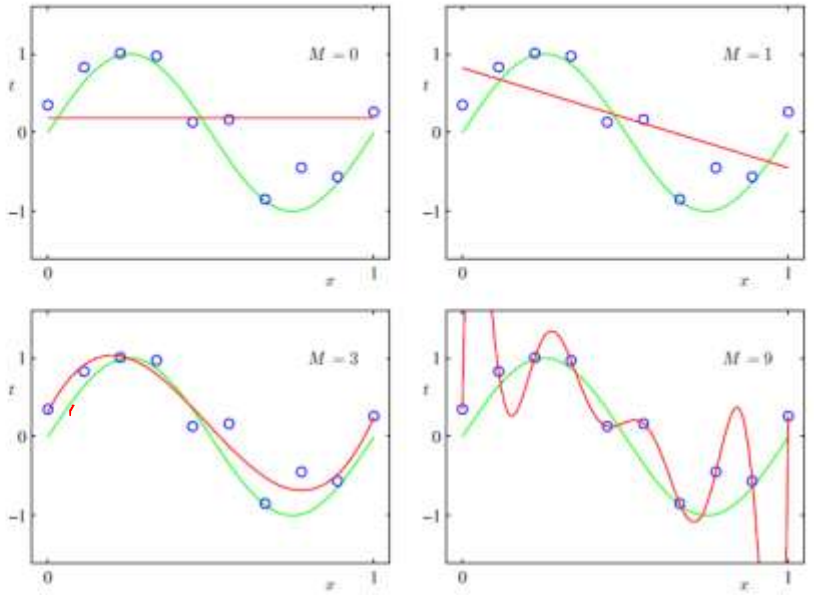
Example: Polynomial Curve Fitting

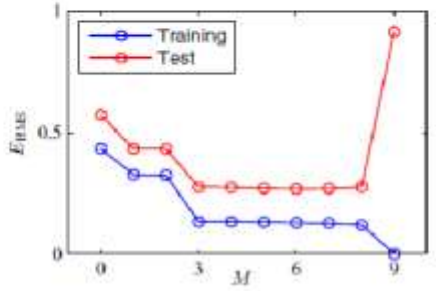
Model: $y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$

Objective Function $E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \{y(x_n, \mathbf{w}) - t_n\}^2$

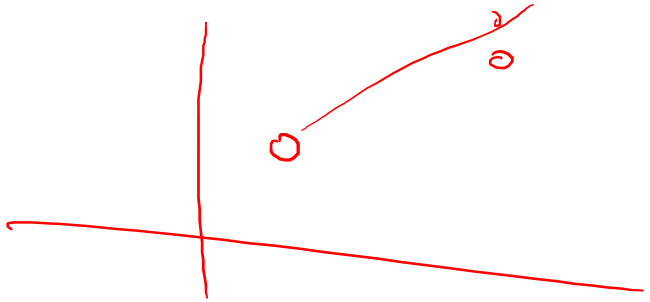


$E_{\mathrm{RMS}} = \sqrt{2E(\mathbf{w}^*)/N}$





Overfitting: Solution



- Large number of data
- Insert extra information
- …

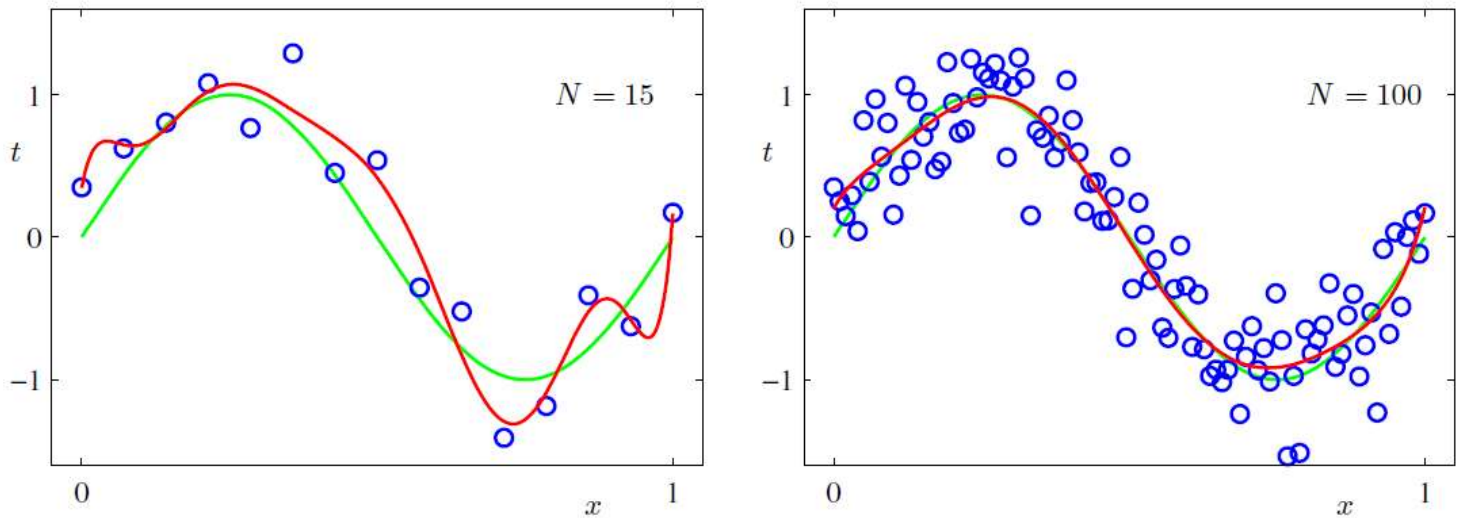


Figure 1.6 Plots of the solutions obtained by minimizing the sum-of-squares error function using the $M = 9$ polynomial for $N = 15$ data points (left plot) and $N = 100$ data points (right plot). We see that increasing the size of the data set reduces the over-fitting problem.

Large number of data

Table of the coefficients $\mathbf{w}^*$ for polynomials of various order. Observe how the typical magnitude of the coefficients increases dramatically as the order of the polynomial increases.

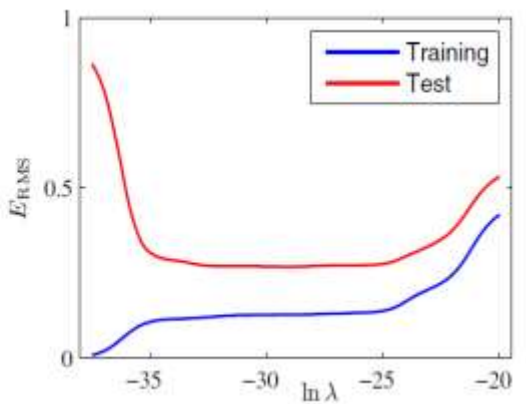
	$M=0$	$M=1$	$M=6$	$M=9$
w_0^*	0.19	0.82	0.31	0.35
w_1^*		-1.27	7.99	232.37
w_2^*			-25.43	-5321.83
w_3^*			17.37	48568.31
w_4^*				-231639.30
w_5^*				640042.26
w_6^*				-1061800.52
w_7^*				1042400.18
w_8^*				-557682.99
w_9^*				125201.43

Large order M ⇨ Large coefficients for inexact data

$$\widetilde{E}(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n,\mathbf{w}) - t_n\}^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2$$

	$\ln\lambda = -\infty$	$\ln\lambda = -18$	$\ln\lambda = 0$
$w_0^\star$	0.35	0.35	0.13
$w_1^\star$	232.37	4.74	-0.05
$w_2^\star$	-5321.83	-0.77	-0.06
$w_3^\star$	48568.31	-31.97	-0.05
$w_4^\star$	-231639.30	-3.89	-0.03
$w_5^\star$	640042.26	55.28	-0.02
$w_6^\star$	-1061800.52	41.32	-0.01
$w_7^\star$	1042400.18	-45.95	-0.00
$w_8^\star$	-557682.99	-91.53	0.00
$w_9^\star$	125201.43	72.68	0.01

Graph of the root-mean-square error (1.3) versus $\ln\lambda$ for the $M = 9$ polynomial.



Using extra information

Data

- Vector
- Array
- Graph
- Text(vector)
- ….

- features
- spatial
- Temporal
- Spatial-temporal
- …….
- Sequential

- nominal
- ordinal
- ….
- Real

Challenges of Data

- Missing values
- Noisy data
- Huge data
- Heterogeneous data
- …..