Optimization Methods for Machine Learning (OMML)

Laura Palagi
Department of Computer, Control, and Management Engineering
Antonio Ruberti

SAPIENZA
UNIVERSITÀ DI ROMA

2nd lecture
September 25, 2018

Machine Learning and Statistics

Statistical Inference (V. Vapnick)

Given a collection of empirical data originating from some functional dependency, infer this dependency

There are two main approaches

- parametric (particular) inference, which aims to create simple methods of inference to be use to solve specific real –life problems
- general inference which aims to create one (induction) method for any problem of statistical inference

Parametric Inference

Beginning 1930. Golden age '30-'60

- Assume to know the problem, e.g.
  - the physical law that generates the stochastic properties of data
  - and the function to be found up to a finite number of parameters.
- the essence of the inference problem stays in estimating parameters and using data to verify reliabilty of it
- To find these parameters, using information about the statistical law and the target function one adopts the maximum likelihood method

Parametric Inference

- Inference models are quite simple and they were suitable for the computational resources available in the sixties.
- These models are based on three main principles
  - The Weierstrass Theorem: *any continuous function on a finite interval can be approximated by a polynomials (i.e. a linear function in the parameters) to any degree of accuracy*
  - The central limit theorem: (roughly) *the distribution of the sum (or average) of a large number of independent, identically distributed variables will be approximately normal*, regardless of the underlying distribution.
  - the maximum likelihood method is a good tools to estimate the parameters

The end of parametric inference

- Curse of dimensionality (R. Bellman, ca 1960): increasing the number of factors to be taken into account requires **exponentially increasing** the amount of computational resources. For ex: if the function is not sufficiently smooth to obtain the given degree of accuracy one needs an exponential number of terms in the polynomial (and hence of variables)

- (Tukey ca 1960) statistical components of real-life problems cannot be described by classical distribution functions

- the maximum likelihood method may not be a good one even for very simple cases (James and Stein)

Beyond parametric inference

- General statistical inference: ones does not have a priori information about the statistical law underlying the problem or about the function to be approximated.
  - Look for a method that infers an approximating function from examples (inductive method)
  - data used to define the model itself
  - non linear models in the parameters

data analysis/data mining

Data Mining and Big Data

Report| McKinsey Global Institute

Big data: The next frontier for innovation, competition, and productivity - May 2011

"The amount of data in our world has been exploding, and analyzing large data sets—so-called big data—will become a key basis of competition, underpinning new waves of productivity growth, innovation, and consumer surplus, according to research by MGI and McKinsey's Business Technology Office……. Leading companies are using data collection and analysis to conduct controlled experiments to make better management decisions; …… sophisticated analytics can substantially improve decision-making……."

http://www.mckinsey.com/insights/business_technology/big_data_the_next_frontier_for_innovation

- The subject of **data mining** is the extraction of patterns and knowledge from large amount of data using automatic or semi-automatic methods and the operative use of this information.
- Exponential growth of tools and techinques to collect and store huge amount of data

Story

- In 1958 Rosenblatt (a fisiology) proposed a learning machine (namely a program) called Perceptron to solve a simple classification problem. The perceptron reproduced some neurofiologic learning model. The perceptron was able to generalize (it learns !).
- 1958-1992: Feedforward Neural Networks (shallow)
- (1992-) back to the general statistical inference: other learning machines have been proposed which do not have any similarity with the biological neuron.

Does an inductive inference principle

exist in common to all these machines ?

- (2010 -) Deep Learning (FFN deep) and beyond

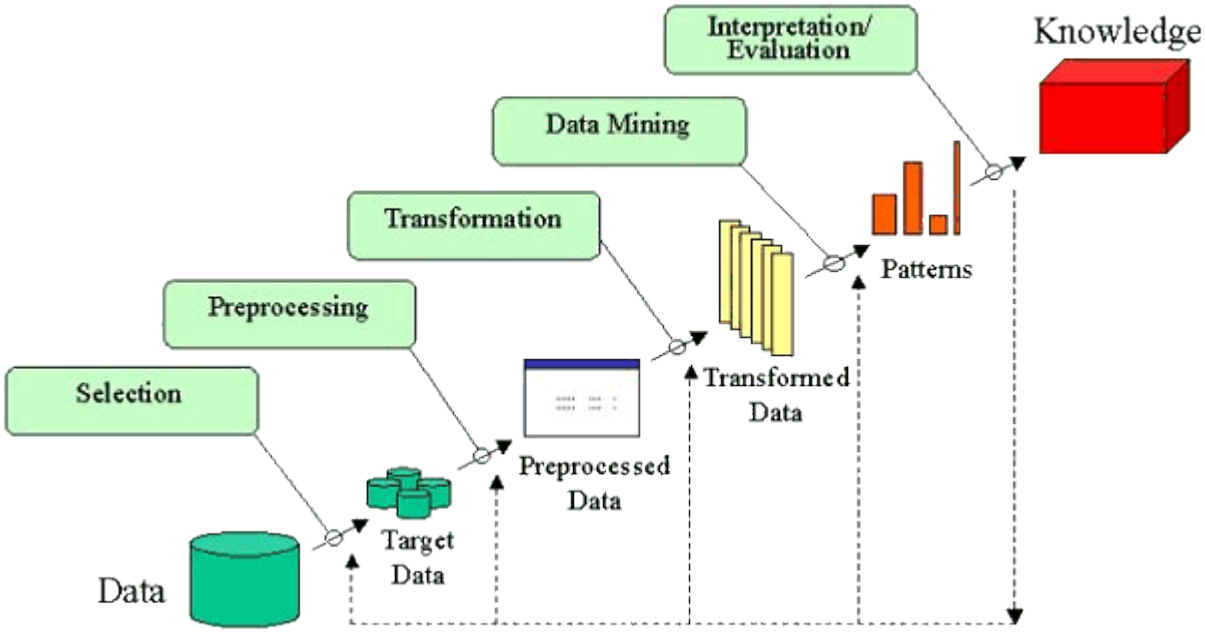
SAPIENZA
UNIVERSITÀ DI ROMA

What is Data Mining ?

The core of Knowledge Discovery in Databases (KDD)

The term **KDD**, denotes the full research knowledge process from data, namely the techniques to help decision manager in the process of extraction of knowledge in a clever and automatic way. The KDD process includes

- Formulation of the problems
- Data collection
- Data Cleaning and preprocessing
- Data mining
- Analysis of the results produced by the model



Non-trivial extraction of implicit, previously unknown and potentially useful information from data

Exploration & analysis, by automatic or semi-automatic means, of large quantities of data in order to discover meaningful patterns

However the *data mining* (DM) step constitutes a so important phase in the overall KKD process to be often identified with the full KDD process

What is (not) Data Mining?

By Namwar Rizvi

- **Ad Hoc Query**: ad Hoc queries just examines the current data set and gives you result based on that. This means you can check what is the maximum price of a product but you can not predict what will be the maximum price of that product in near future.
- **Event Notification**: you can set different alerts based on some threshold values which will inform you as soon as that threshold will reach by actual transactional data but you can not predict when that threshold will reach.
- **Multidimensional Analysis**: you can find the value of an item based on different dimensions like Time, Area, Color but you can not predict what will be the value of the item when its color will be Blue and Area will be UK and Time will be First Quarter of the year
- **Statistics**: Statistics can tell you the history of price changes, moving averages, maximum values, minimum values etc. but it can not tell you how price will change if you start selling another product in the same season.

Data Mining tasks…

Define a learning model to be used in

- Prediction
  - Use data to predict unknown or future values of some variables.
- Description
  - Find human-interpretable patterns that describe the data;

- Classification [Predictive] …addressed in the course
- Regression [Predictive] …addressed in the course
- Clustering [Descriptive] …NOT addressed in the course

Rule for a "safe use"

Not everything is foreseeable or can be learned

In some cases developing refined mathematical models and/or increasing the tools' reliability may lead to predict phenomena which are not predictable nowadays

In other cases, although deterministic, no refined tools or model may produce a good prediction

In a dynamic system a small perturbation of the initial condition may lead to a totally different final state. (see e.g. the 1998 nice movie S*liding doors*)

Some process are «intrinsicly chaotic», e.g. social/economic phenomena which are characterized by the unpredictability and by personal choices

Learning model

The main focus of the course is on optimization tools for machine learning. In order to study mathematically, we need to formally define the learning problem.

Keep in mind:

1. A learning model should be **rich enough** to capture important aspects of the problem, but simple enough to be tackled mathematically.
2. As usual in mathematical modelling, **simplifying assumptions** are unavoidable.
3. A learning model should answer several questions:
   - How is the data being generated?
   - How is the data presented to the learner?
   - What is the goal of learning in this model?

What are Data ?

- A collection of objects (examples) and their attributes
- An attribute is a property or characteristic of an object
  - Examples: eye color of a person, temperature, etc.
  - Attribute is also known as variable, field, characteristic, or feature
  - **Attributes** are encoded as vectors in some vector space
- A collection of attributes describe an object (also called record, point, case, sample, entity, or instance)

Attributes

Instances

Education	Occupation	Home Owner	Cars	Commute Distance	Region	Age
Bachelors	Skilled Manual	Yes	0	0-1 Miles	Europe	42
Partial College	Clerical	Yes	1	0-1 Miles	Europe	43
Partial College	Professional	No	2	2-5 Miles	Europe	60
Bachelors	Professional	Yes	1	5-10 Miles	Pacific	41
Bachelors	Clerical	No	0	0-1 Miles	Europe	36
Partial College	Manual	Yes	0	1-2 Miles	Europe	50
High School	Management	Yes	4	0-1 Miles	Pacific	33
Bachelors	Skilled Manual	Yes	0	0-1 Miles	Europe	43
Partial High School	Clerical	Yes	2	5-10 Miles	Pacific	58
Partial College	Manual	Yes	1	0-1 Miles	Europe	48
High School	Skilled Manual	No	2	1-2 Miles	Pacific	54
Bachelors	Professional	No	4	10+ Miles	Pacific	36
Partial College	Professional	Yes	4	0-1 Miles	Europe	55
Partial College	Clerical	Yes	1	1-2 Miles	Europe	35
Partial College	Skilled Manual	No	1	0-1 Miles	Pacific	45

Learning paradigms

Supervised learning

There is a "teacher", namely one knows the right answer on the training instances

One of the attributes describing the instances is the **class**
attribute=(feature,class)

Find a model for identify the class attribute as a function of the values of other attributes.

Unsurpervised learning

no "teacher, output classes are not known and one wants to find similarity class

None of the attributes is the **class**.
attribute=feature

Find a model to define an assignment rule of an instance to a class

Supervised Classification

Attribute

Features

Class

Training Set
(features,class)

Education	Occupation	Home Owner	Cars	Commute Distance	Region	Age	High Value Customer
Bachelors	Skilled Manual	Yes	0	0-1 Miles	Europe	42	Yes
Partial College	Clerical	Yes	1	0-1 Miles	Europe	43	Yes
Partial College	Professional	No	2	2-5 Miles	Europe	60	Yes
Bachelors	Professional	Yes	1	5-10 Miles	Pacific	41	No
Bachelors	Clerical	No	0	0-1 Miles	Europe	36	Yes
Partial College	Manual	Yes	0	1-2 Miles	Europe	50	No
High School	Management	Yes	4	0-1 Miles	Pacific	33	No
Bachelors	Skilled Manual	Yes	0	0-1 Miles	Europe	43	Yes
Partial High School	Clerical	Yes	2	5-10 Miles	Pacific	58	No
Partial College	Manual	Yes	1	0-1 Miles	Europe	48	Yes

The goal: predict the class for the unknown

Test set

High School	Skilled Manual	No	2	1-2 Miles	Pacific	54	?
Bachelors	Professional	No	4	10+ Miles	Pacific	36	?
Partial College	Professional	Yes	4	0-1 Miles	Europe	55	?
Partial College	Clerical	Yes	1	1-2 Miles	Europe	35	?
Partial College	Skilled Manual	No	1	0-1 Miles	Pacific	45	?

Attributes

Features

Label

Supervised learning

Data set
(features,label)

Education	Occupation	Home Owner	Cars	Commute Distance	Region	Age	High Value Customer
Bachelors	Skilled Manual	Yes	0	0-1 Miles	Europe	42	Yes
Partial College	Clerical	Yes	1	0-1 Miles	Europe	43	Yes
Partial College	Professional	No	2	2-5 Miles	Europe	60	Yes
Bachelors	Professional	Yes	1	5-10 Miles	Pacific	41	No
Bachelors	Clerical	No	0	0-1 Miles	Europe	36	Yes
Partial College	Manual	Yes	0	1-2 Miles	Europe	50	No
High School	Management	Yes	4	0-1 Miles	Pacific	33	No
Bachelors	Skilled Manual	Yes	0	0-1 Miles	Europe	43	Yes
Partial High School	Clerical	Yes	2	5-10 Miles	Pacific	58	No
Partial College	Manual	Yes	1	0-1 Miles	Europe	48	Yes

Attributes=Features

Unsupervised learning

Data set
(features)

Education	Occupation	Home Owner	Cars	Commute Distance	Region	Age
Bachelors	Skilled Manual	Yes	0	0-1 Miles	Europe	42
Partial College	Clerical	Yes	1	0-1 Miles	Europe	43
Partial College	Professional	No	2	2-5 Miles	Europe	60
Bachelors	Professional	Yes	1	5-10 Miles	Pacific	41
Bachelors	Clerical	No	0	0-1 Miles	Europe	36
Partial College	Manual	Yes	0	1-2 Miles	Europe	50
High School	Management	Yes	4	0-1 Miles	Pacific	33
Bachelors	Skilled Manual	Yes	0	0-1 Miles	Europe	43
Partial High School	Clerical	Yes	2	5-10 Miles	Pacific	58
Partial College	Manual	Yes	1	0-1 Miles	Europe	48

SAPIENZA
UNIVERSITÀ DI ROMA

Unsupervised Classification

Attribute=Features

Training set

Education	Occupation	Home Owner	Cars	Commute Distance	Region	Age
Bachelors	Skilled Manual	Yes	0	0-1 Miles	Europe	42
Partial College	Clerical	Yes	1	0-1 Miles	Europe	43
Partial College	Professional	No	2	2-5 Miles	Europe	60
Bachelors	Professional	Yes	1	5-10 Miles	Pacific	41
Bachelors	Clerical	No	0	0-1 Miles	Europe	36
Partial College	Manual	Yes	0	1-2 Miles	Europe	50
High School	Management	Yes	4	0-1 Miles	Pacific	33
Bachelors	Skilled Manual	Yes	0	0-1 Miles	Europe	43
Partial High School	Clerical	Yes	2	5-10 Miles	Pacific	58
Partial College	Manual	Yes	1	0-1 Miles	Europe	48

No Class

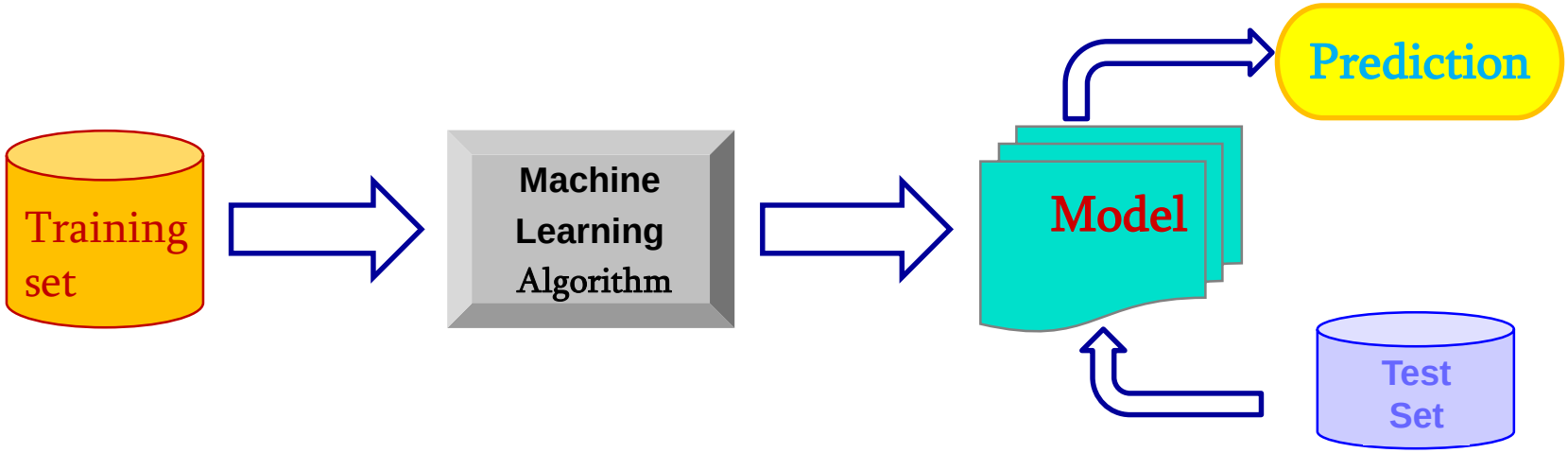
The goal: find the classes and predict for the unknown

Test set

High School	Skilled Manual	No	2	1-2 Miles	Pacific	54	?
Bachelors	Professional	No	4	10+ Miles	Pacific	36	?
Partial College	Professional	Yes	4	0-1 Miles	Europe	55	?
Partial College	Clerical	Yes	1	1-2 Miles	Europe	35	?
Partial College	Skilled Manual	No	1	0-1 Miles	Pacific	45	?

Goal of learning process

- **Given** a collection of instances (*training set*), each one containing a set of attributes
- **Define** a *model* (prediction rule) that returns the output as a function of the values of the attributes.
- **Goal**: previously unseen instances should be assigned a class as accurately as possible.



Goal of learning process

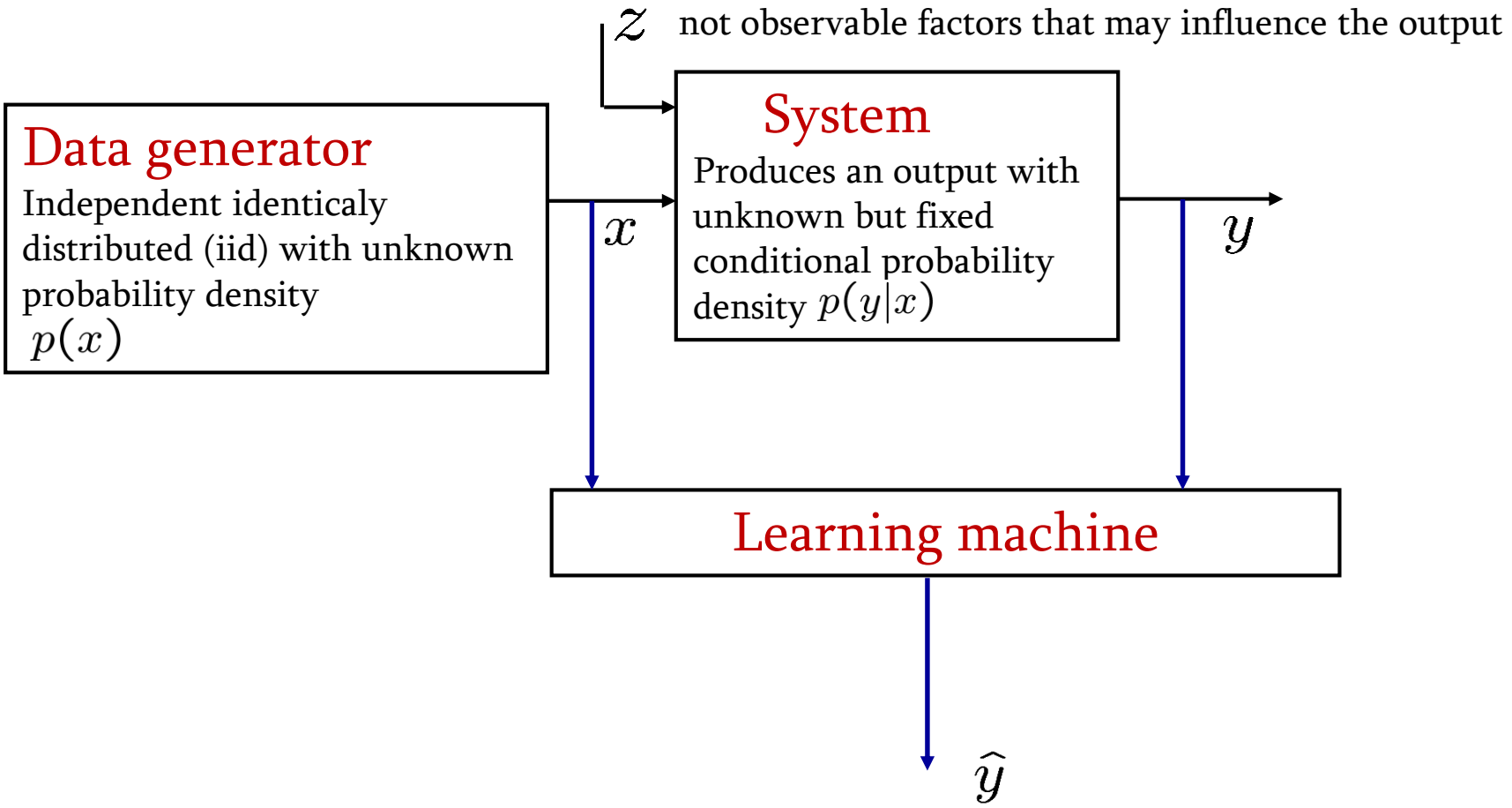
- In the learning process we have two main phases
  - **construction** of a model (learning) using a set of available data
  - **use** (prediction/description) the model on unseen data to check capability of giving the "right answer" on new instances (generalization).

Data in learning process

Available data (target set) may be used in different phases

- Training set: data used for the learning phase
  - incrementally (on-line learning): Data are obtained incrementally during the training process
  - batch (off-line) learning: Data of the training set are available in advance before entering the training process
- Test set: data used ins the 2nd phase for checking the accuracy
- Validation set: data used as testing in the learning phase

The learning system



The learning system

- The generator produces random vectors drawn independently from a fixed probability density
- The learning machine has NO control on the process of sampling generation
- The system produces an output with unknown but fixed conditional probability density
- More formally a learning machine is a function $f_\alpha \subseteq \mathcal{F}$ in a given class $\mathcal{F}$ which depends on the type of machine chosen; α represents a set of parameters which identifies a particular function within the class.
- The machine is deterministic.

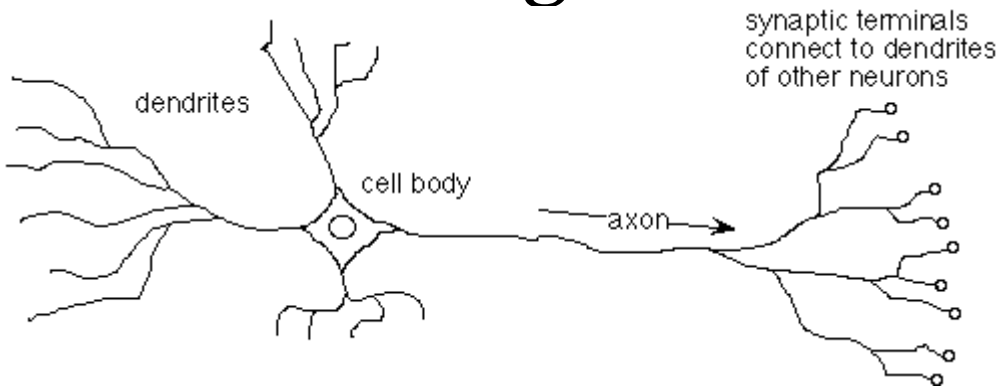
Learning Machine

- Nonlinear model in the parameters

$$f_\alpha = \sum_{i=1}^{\ell} g_i(x, \alpha)$$

- An example: the formal neuron (perceptron)

From the biological neurons ……



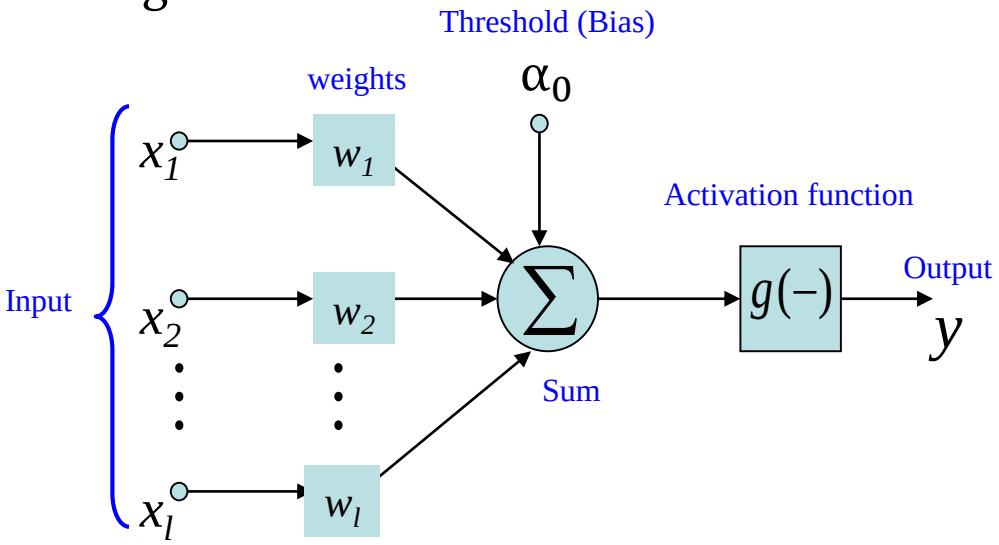
Neurons encode their activations (outputs) as a series of brief electrical pulses

- The neuron's ***cell body*** processes the incoming activations and converts them into output activations.
- ***Dendrites*** are fibres which emanate from the cell body and provide the receptive zones that receive activation from other neurons.
- ***Axons*** are fibres acting as transmission lines that send activation to other neurons.
- The junctions that allow signal transmission between the axons and dendrites are called ***synapses***

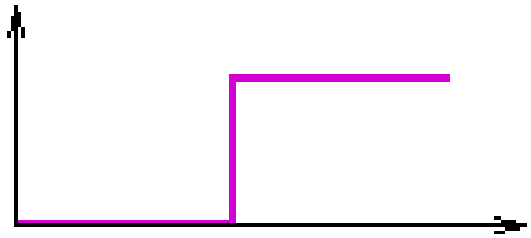
…..to the artificial neurons

The key components of neural signal processing are:

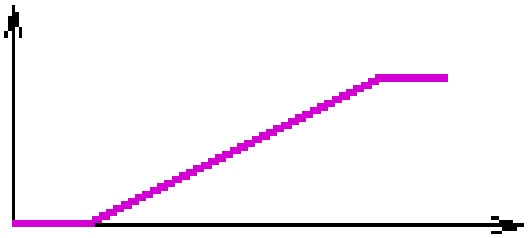
1. Signals from connected neurons are collected by the dendrites.
2. The cells body sums the incoming signals
3. When sufficient input is received (i.e. a threshold is exceeded), the neuron generates an action potential (i.e. it 'fires').
4. That action potential is transmitted along the axon to other neurons
5. If sufficient input is not received, the inputs quickly decay and no action potential is generated.



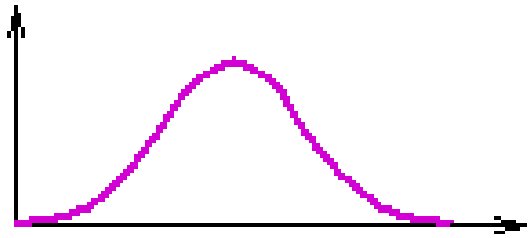
Examples of activation functions



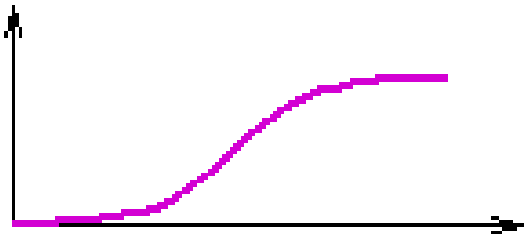
Threshold



Linear



Gaussian



Sigmoid

An example: the formal neuron

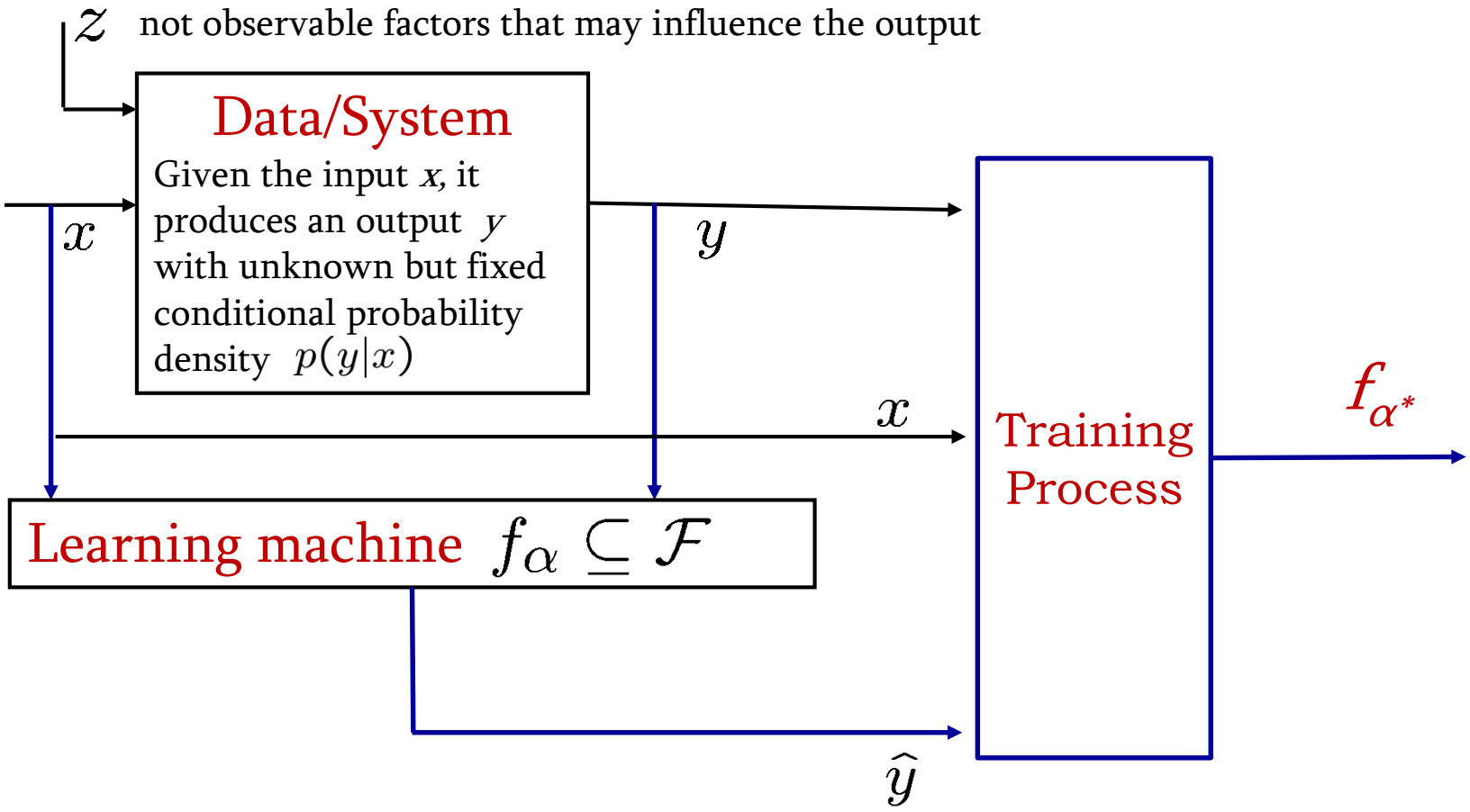
The formal neuron (perceptron) is a simple learning machine which implements the class of function

$$f_\alpha = g(\sum_{i=1}^{\ell} \alpha_i x^i - \alpha_0)$$

Inputs are multiplied by weights, representing di importance of the synaptic connection; their algebraic sum is compared with a threshold value. Output is 1 if the sum is greater than the threshold, -1 (or 0, Heaviside function) otherwise

$$f_\alpha = sgn(\sum_{i=1}^{\ell} \alpha_i x^i - \alpha_0)$$

The training process



Training process

Given a learning machine, namely given a class $\mathcal{F}$ of function

$$f_\alpha \subseteq \mathcal{F}, \text{ con } f_\alpha : R^n \rightarrow \mathcal{Y}$$

The training process consists in finding a particular value of the parameters α^* which selects a special function f_{α^*} in the chosen class.

The goal is modelling the process in a way that it is able to give right answer on instances never seen before (generalization property) rather than interpolating (="make no mistake") on the training set

Quality measure

In order to choose among all the possible function f_α of the parameter α one needs to define a *quality measure* to be optimized.

We introduce a Loss function

$$\mathcal{L}(y, f_\alpha(x)) = \mathcal{L}(y, x, \alpha)$$

Which is a function that measures the difference between the value returned by the machine $f_\alpha(x)$ and the true value y. By definition, the loss is nonnegative, hence high positive values indicates bad performance.

Given the values of α, the value of the loss function (depending only by x, y) measures the error on the realization of the pair (x, y)

Minimization of the "risk"

The "quality criterion" that drives the choice of the parameters α is the expected value of the error obtained using a given loss function

$$R(\alpha) = R(f_\alpha) = E_{\mathcal{P}}[\mathcal{L}(f_\alpha(x), y)]$$

The function $R(\alpha)$ is called the expected risk to be minimized over the α (namely choosing $f \in \mathcal{F}$)

$$\min_{\alpha} R(\alpha)$$

Learning is the process of estimating the function $f_\alpha(x)$ which minimizes the expected risk over the set of functions supported by the learning machine using only the training data

Examples of Loss functions

(two class) Classification problem

The output takes only two values, e.g. $y \in \{-1, 1\}$

The learning machine $f_\alpha(x) : R^n \rightarrow \{-1, 1\}$

$$\mathcal{L}(y, f_\alpha(x)) = \begin{cases} 0 & \text{se } f_\alpha(x) = y \\ 1 & \text{se } f_\alpha(x) \neq y \end{cases}$$

$$\mathcal{L}(y, f_\alpha(x)) = \frac{1}{2}|f_\alpha(x) - y|$$

Examples of Loss functions

Regression Problem

Estimating a real-value function $f_\alpha(x) : R^n \rightarrow R$

A loss function is the squared error

$$\mathcal{L}(y, f_\alpha(x)) = \frac{1}{2}(f_\alpha(x) - y)^2$$

Minimization of the "risk"

The expected risk depends on the distribution function underlying data and is given by

$$R(\alpha) = \int L(f_\alpha(x), y) p(x) p(x|y) dx\, dy$$

Any learning task can be solved my minimization of the expected risk if the densities $p(x, y) = p(x)p(x|y)$ were known.

But it is not !

Learning process

- Actually, finding the function f_{α^*} that minimizes the expected risk in the class of functions using a finite number of training data is an ill posed problem

- In a classical (parametric) setting, the model is given (specified) first and then its parameters are estimated from data using the Empirical Risk Minimization (ERM)

Empirical risk minimization

- We know ℓ observations of i.i.d random variables

$$\{(x^1, y^1), \ldots (x^\ell, y^\ell)\}$$

- We look for a function which approximates the expected risk using only the available data
- Given a class of function and a loss function, we define the empirical risk as

$$R_{emp}(f_\alpha) = \frac{1}{\ell} \sum_{i=1}^{\ell} \mathcal{L}(f_\alpha(x^i), y^i)$$

Empirical risk minimization

Empirical risk (training error) depends only on data and on the function f_α

Probability distribution does not enter the definition of empirical risk. Once the values $\alpha, \{x^i, y^i\}_{i=1,\dots,\ell}$ are fixed it has a given value.

In order to get a good generalization property on new examples (test), the ERM principle uses as a decision function the training error

$$\min_{\alpha} R_{emp}(f_\alpha)$$

"Occam's razor" principle

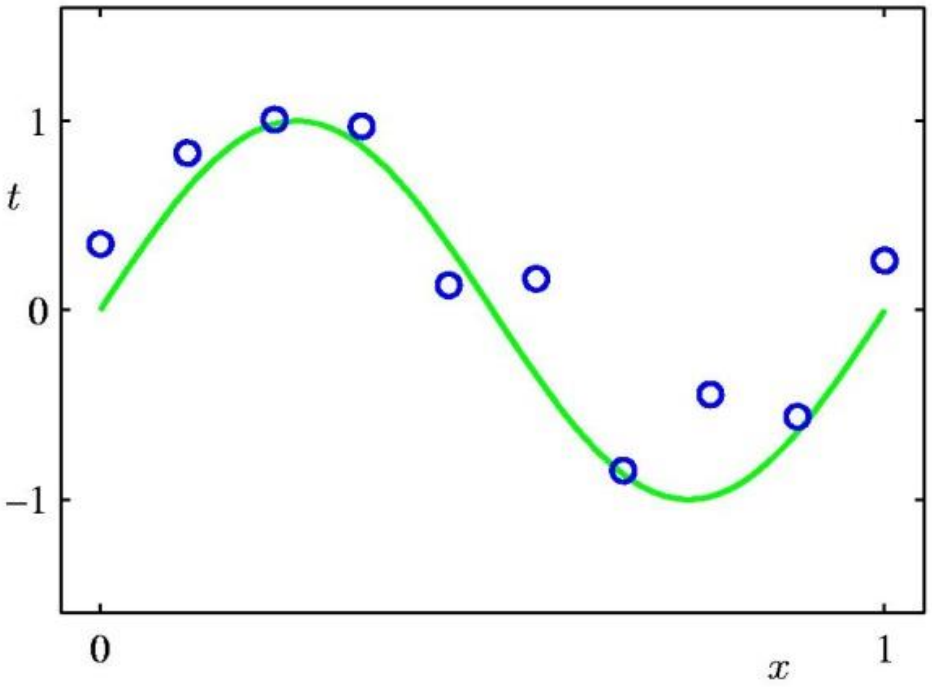
There is a general belief that for flexible learning methods with finite samples, the best prediction performance is provided by a model of optimum complexity.

According to Occam's razor principle, we should seek simpler models over complex ones and optimize the tradeoff between model complexity and the accuracy of model's description of the training data.

Models that are too complex (i.e., that fit the training data very well) or too simple (i.e., that fit the data poorly) provide poor prediction for future data.

Minimization of empirical risk: parametric regression

Let's consider N =10 data points, shown as blue circles, each comprising an observation of the input variable x along with the corresponding output (target) variable.
The green curve shows the function $\sin(2\pi x)$ used to generate the data (noise is added).

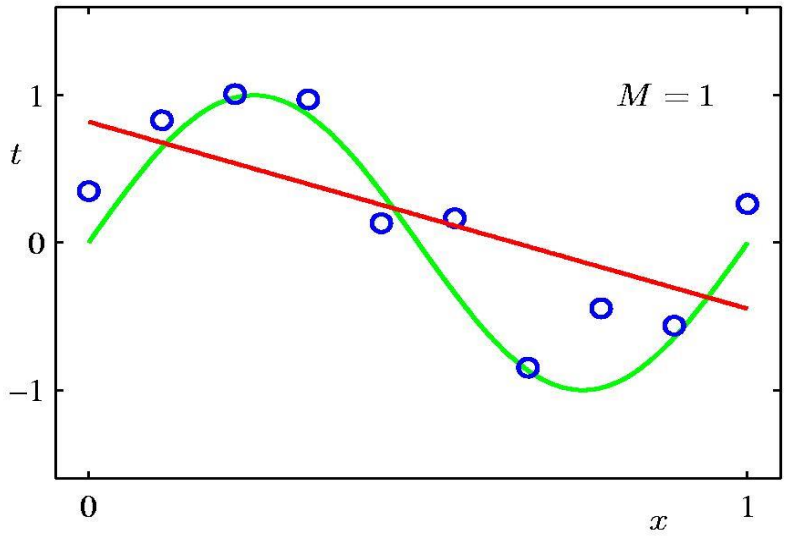


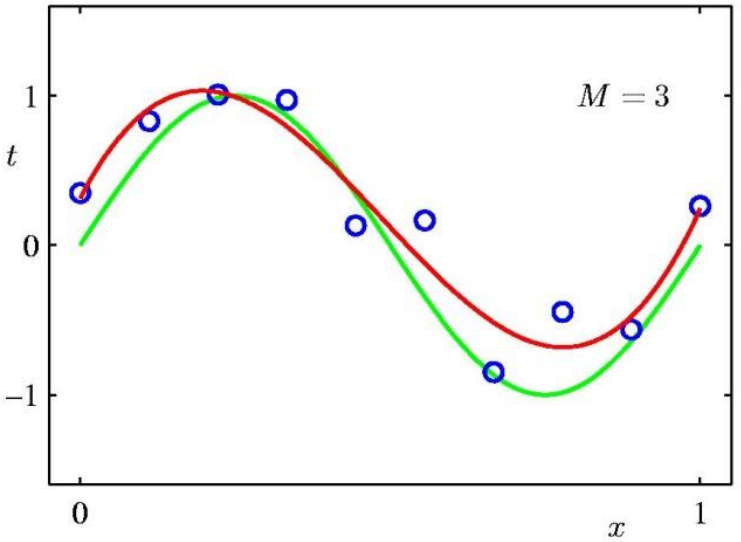
Our goal is to predict the target value for some new value of x, without knowledge of the green curve.

Minimization of empirical risk: parametric regression

We want to use as functions the polynomials of given degree M

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





Parametric Regression continue

Once the model is chosen (e.g. a polynomial of degree M)…

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

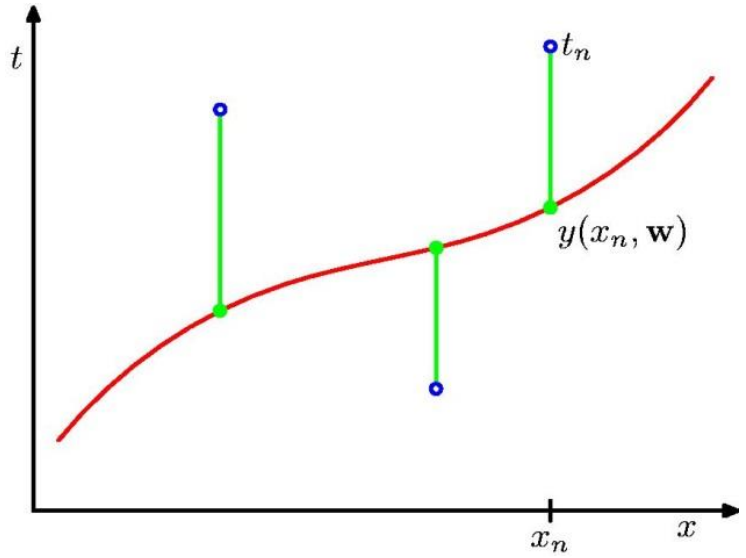
The values of the coefficients will be determined by fitting the polynomial to the training data. This can be done by minimizing an error function that measures the misfit between the function y(x,w), for any given value of w, and the training set
data points.

…it is possible to calculate the least square error; let (x_i, t_i) denote the training data we get:

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

Parametric Regression continue

Because the error function is a quadratic function of the coefficients w, the minimization of the error function has a unique solution, denoted by w*. The resulting polynomial is given by the function $y(x,\mathbf{w}^*)$.



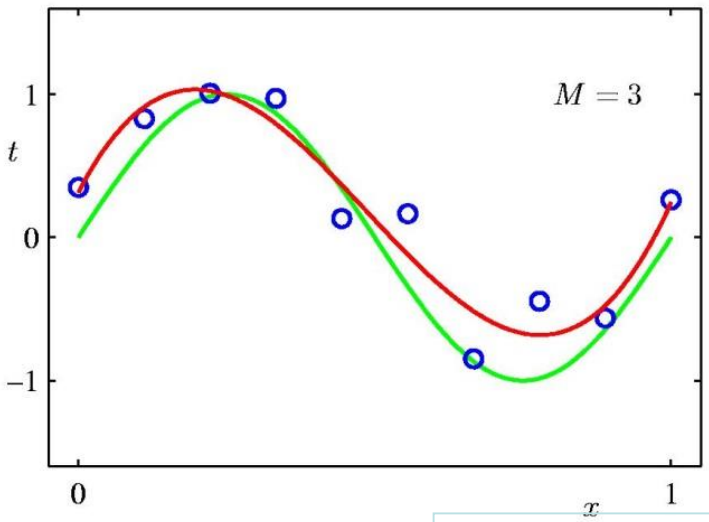
$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n,\mathbf{w}) - t_n\}^2$$

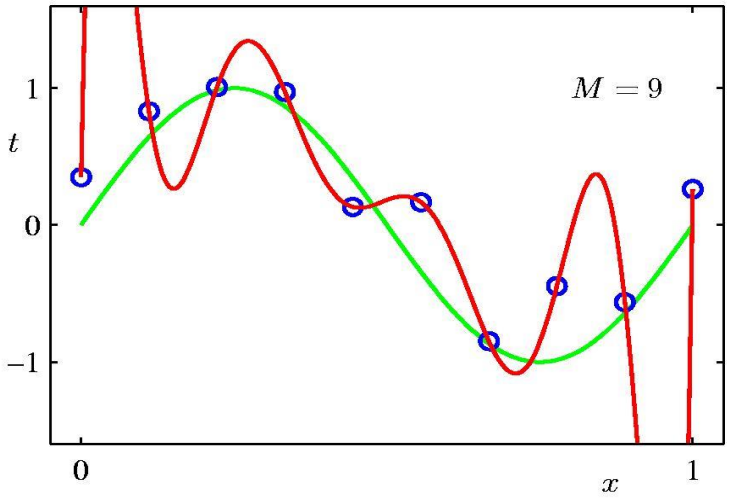
The error on the training data (the sum of the squares of the displacements (shown by the vertical green bars) of each data point from the function) may become zero, but what about the new data (test data)?

Parametric Regression

Let's increase the degree M from 3 to 9

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



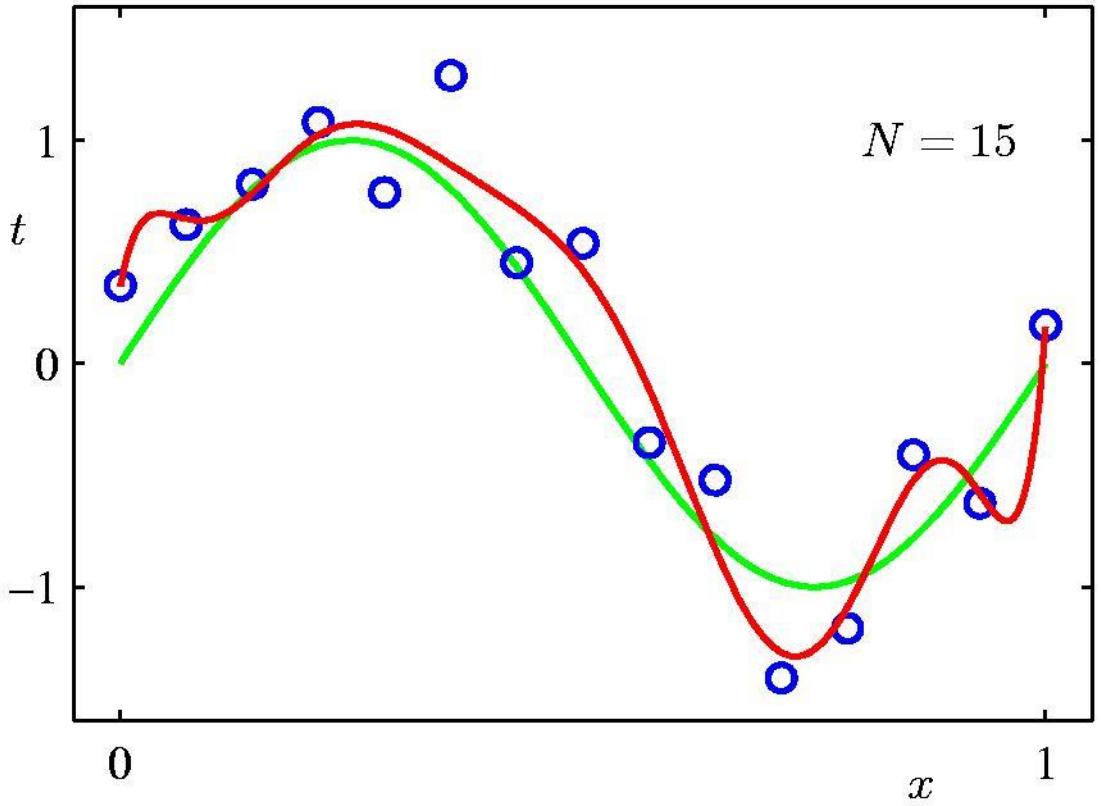


Which is the "better" one?

With M=9 the error is zero, but clearly is a worst approximation that M=3.

Parametric Regression

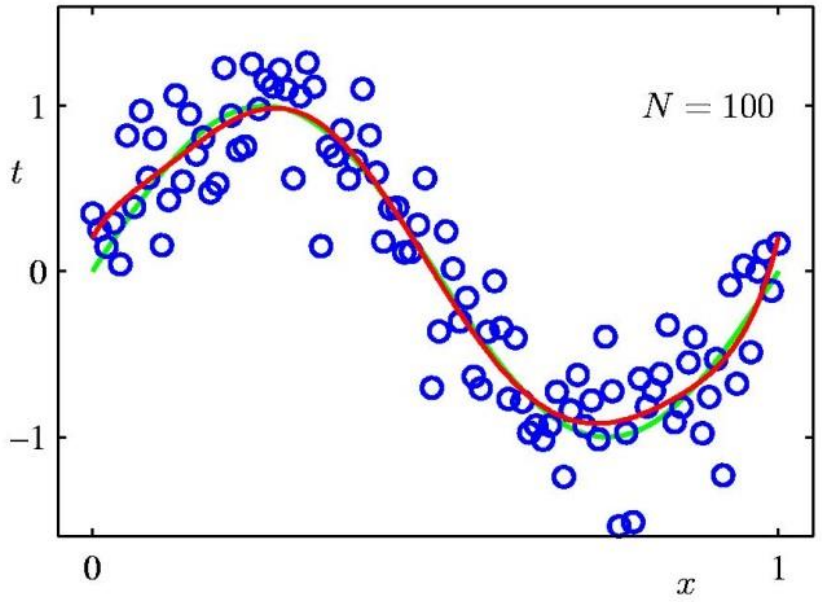
Indeed if the # of data instances increases to $N = 15$



Polynomial of degree M=9: better behaviour than before

Parametric Regression

Increasing the # of training data up to $N = 100$

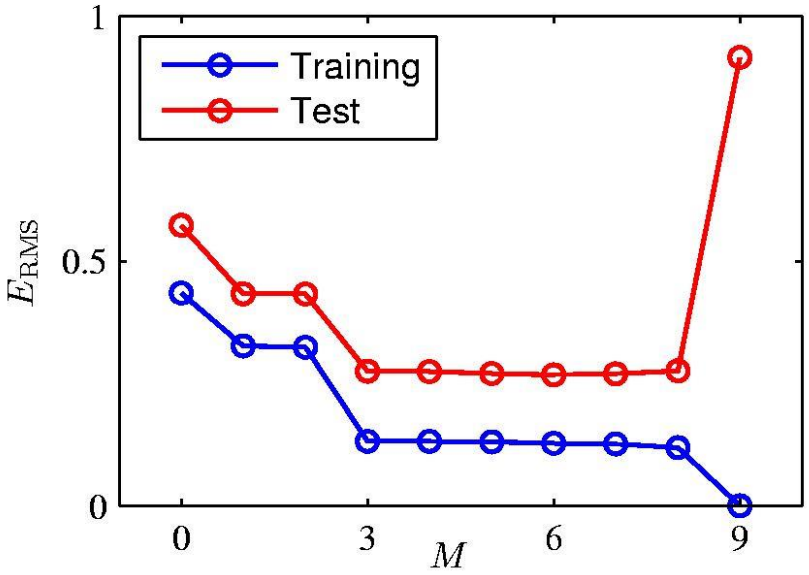


Degree 9 polynomial almost overlap the unknown function

Increasing the complexity of the machine (in this case the degree) is related with a better predictive use as a function of the number of training data

Error behaviour

If we draw the training error and the test error $E_{\mathrm{RMS}} = \sqrt{2E(\mathbf{w}^\star)/N}$



Driving the training error to zero may lead to big error on data test: **Over-fitting**

This is not an absolute rule !

Inductive principle

- The underlying model is not known, we seek for a general prescription for obtaining an estimate of f_{α^*} of the "true dependency" among a large (infinite) number of candidate models (i.e., approximating functions of a learning machine) using the available finite data
- The main issue here is choosing the candidate model of the right complexity to describe the training data
  - Empirical risk minimization with regularization (ERM)
  - Early stopping rules
  - Structural risk minimization (SRM)

Consistency of empirical risk

In general $R_{emp}(f_\alpha) \neq R(f_\alpha)$

Goal: find a relationship among the solutions of the two different optimization problems

$$R(\alpha^*) = \min_\alpha R(\alpha)$$ ⇒ imponderable

$$R_{emp}(\alpha^{**}) = \min_\alpha R_{emp}(\alpha)$$ ⇒ computable

Training error can give any information about probability of error on new data ?

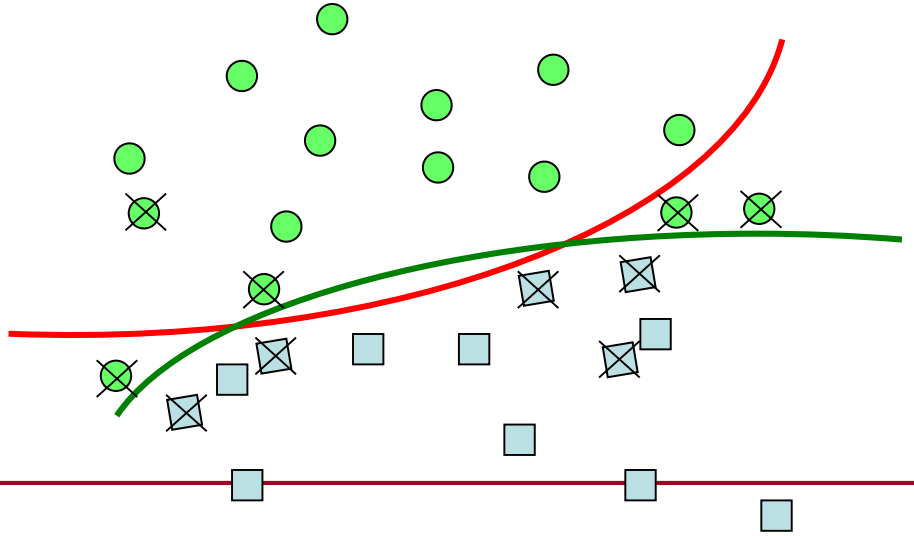
Empirical risk minimization

Whenever the value ℓ is finite minimizing the empirical risk cannot guarantee the minimization of the risk

The choice of a function which minimizes the empirical risk in not unique

Both functions have zero empirical risk (zero error on TS)

The risk on new data is different

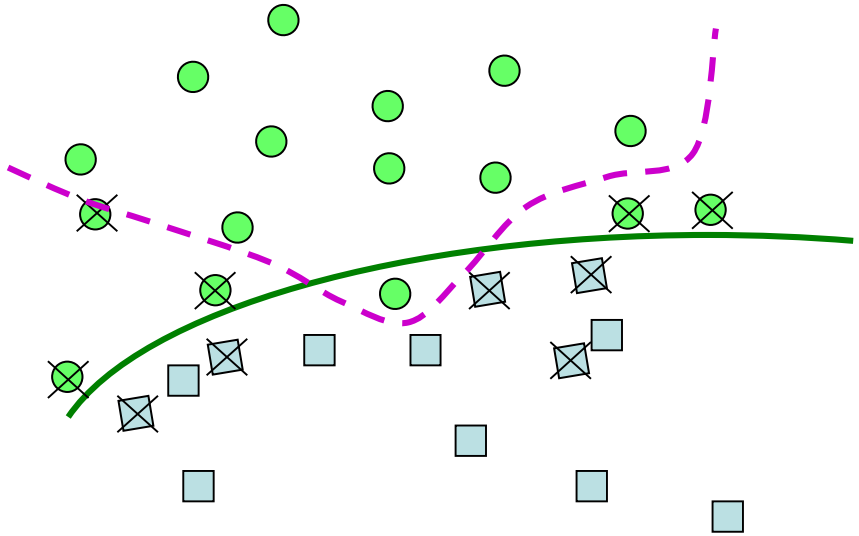


Complexity of the class

A very complex function may describe exactly the training data but not on new data (no generalization property)

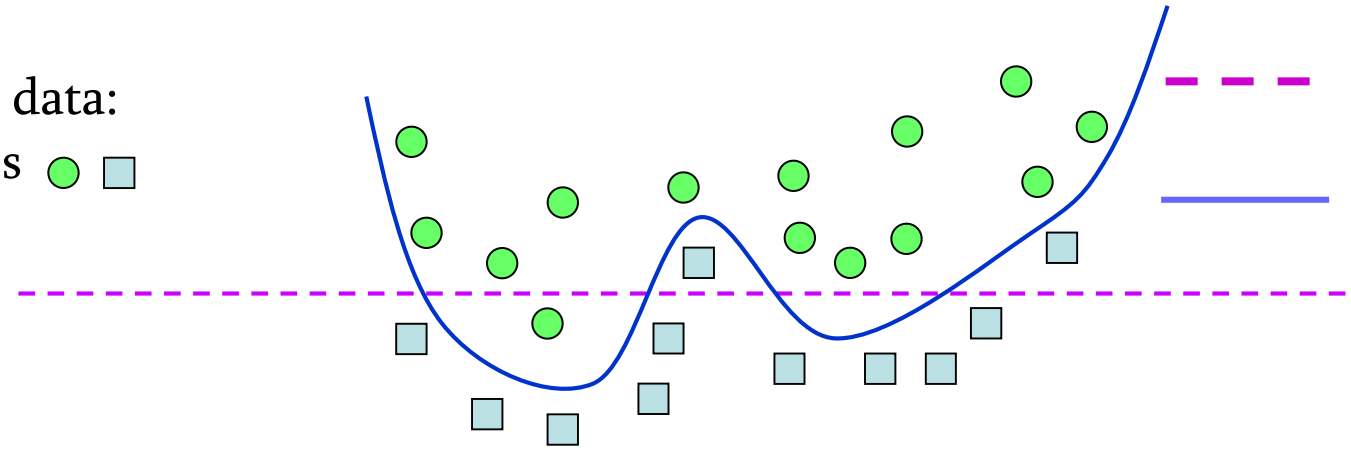
More complex

Simpler

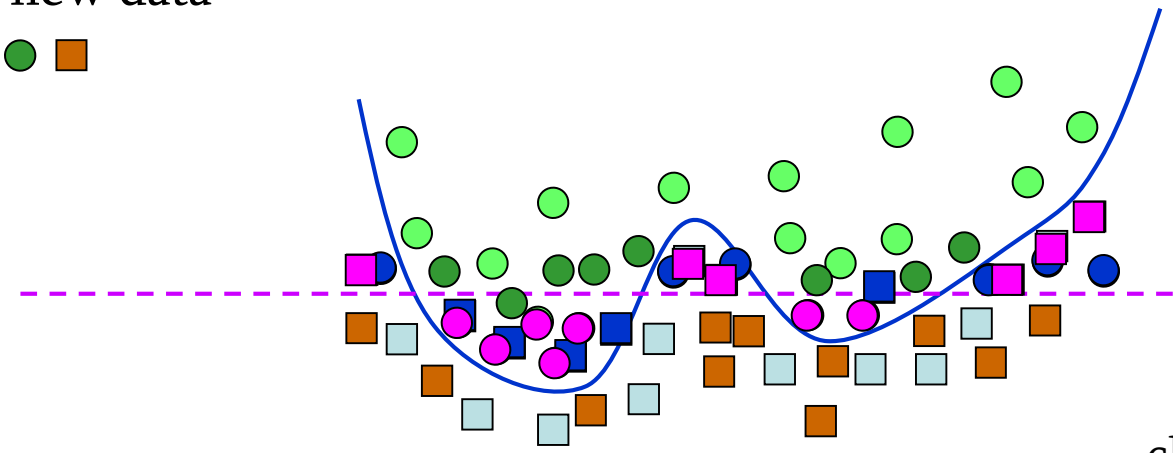


Over and under fitting

Trainig data:
2 classes



Add new data



underfitting
class f_α "too simple"

overfitting
class f_α "too complicated"