Optimization Methods for Machine Learning (OMML)

- 2nd lecture
- Prof. L. Palagi

References:
1. Bishop – *Pattern Recognition and Machine Learning*, Springer, 2006 (Chap 1)
2. V. Cherlassky, F. Mulier - *Learning from Data*, John Wiley and Sons, 2007 (chap.1 and 2)
3. Teaching Notes

Course at a glance

- You can find all info on the web site

  http://www.dis.uniroma1.it/~palagi

  following the path (not yet)

  didattica
  aa-2017-18
  optimization-methods-machine-learning

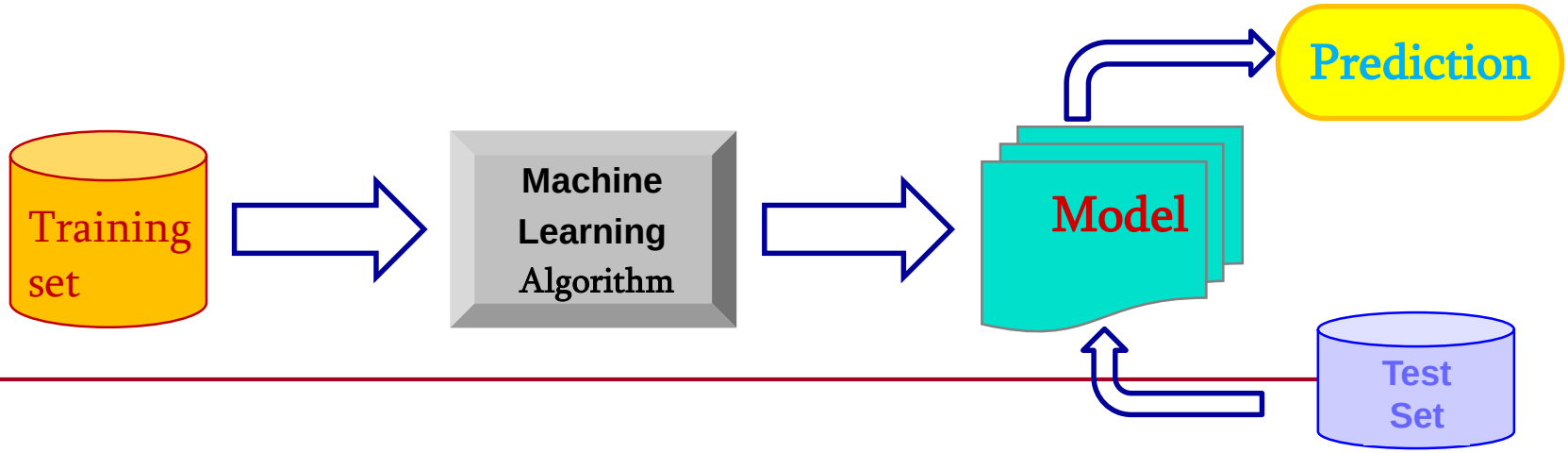
- Check Lectures schedule on the website

Join the Google Group "OMML_2017-18"

General scheme of learning process

Given a collection of instances (*training set*), each one containing a set of attributes

- Find a *model* (prediction rule) that return the output as a function of the values of the attributes.
- Goal: previously unseen instances should be assigned a class as accurately as possible.



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	?

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

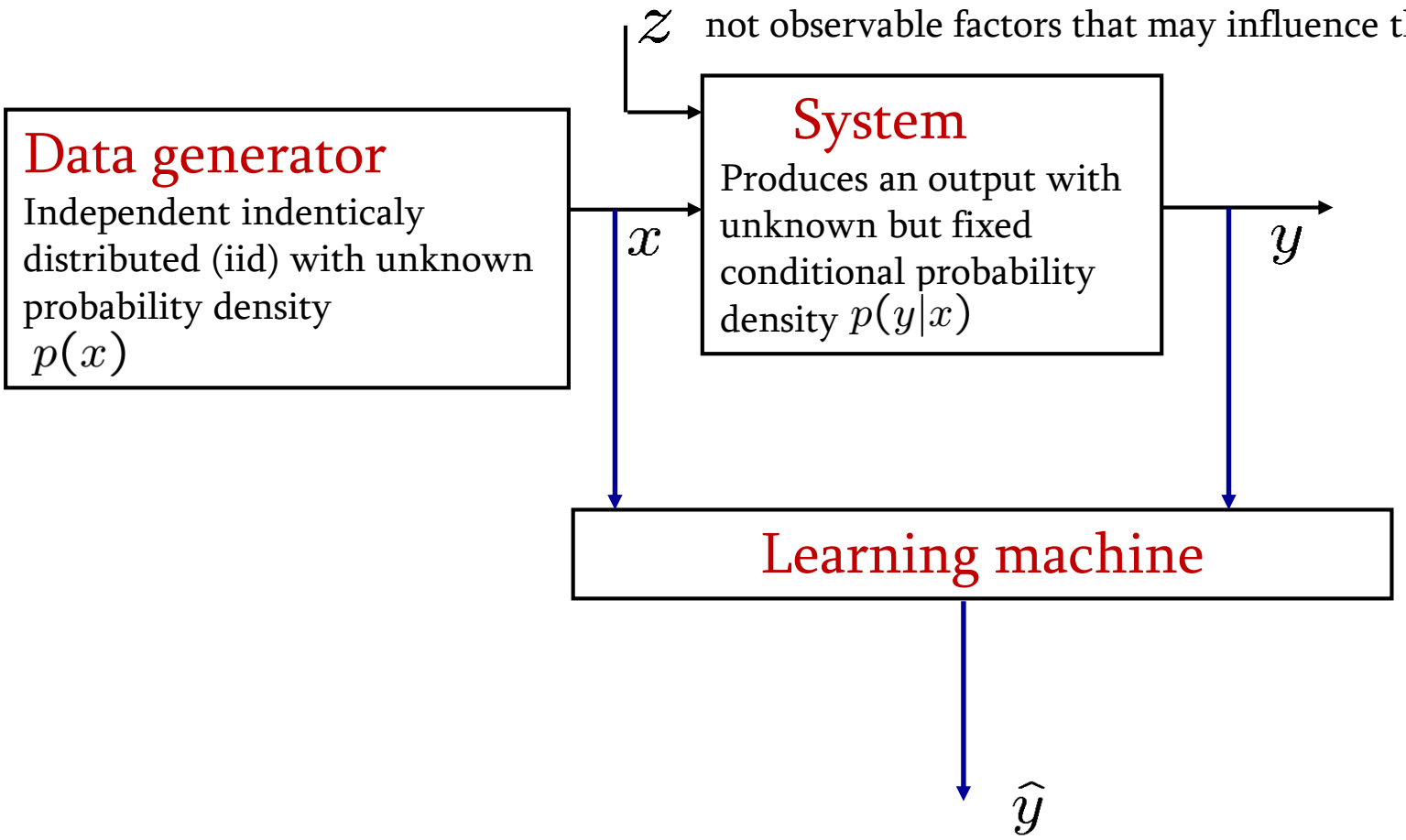
No Class

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	?

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)

An example: the formal neuron

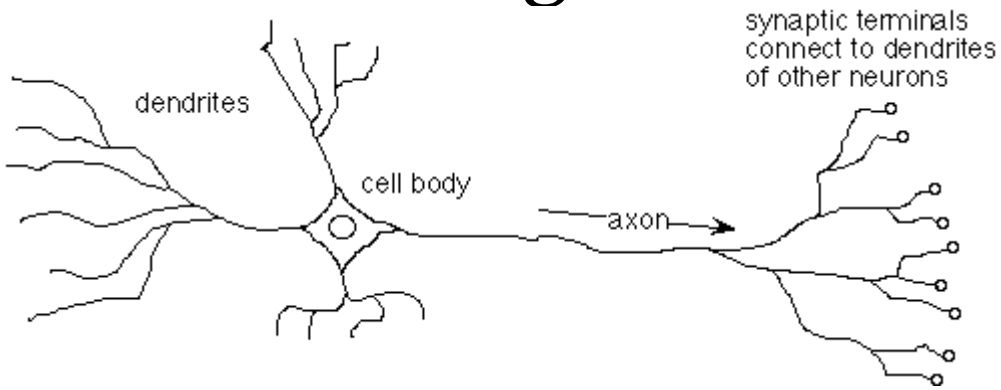
The formal neuron (perceptron) is a simple learning machine which implements the class of function

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

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

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

From the biological neurons ……



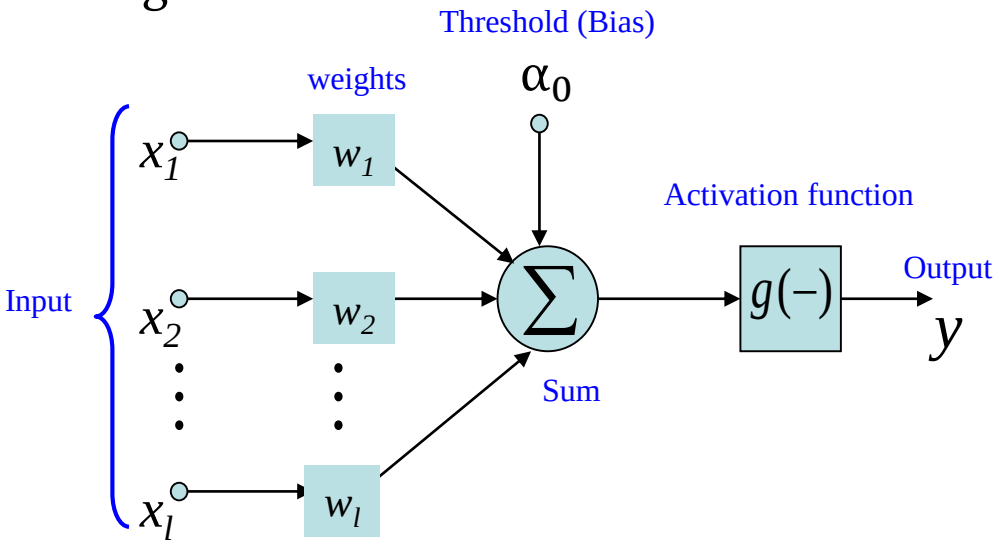
Neurons encode their activations (outputs) as a series of brief electrical pulses

- 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***

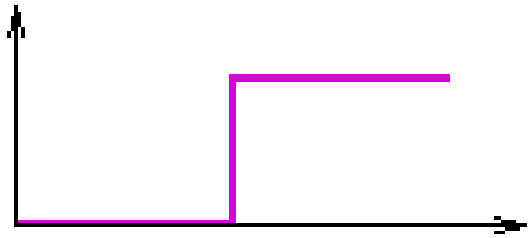
…..to the artificial neurons

The key components of neural signal processing are:

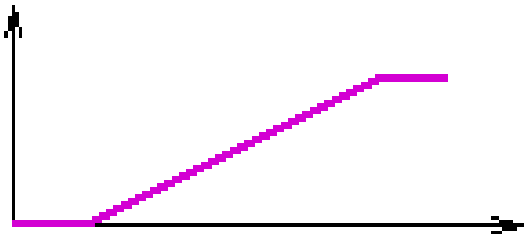
1. Signals from connected neurons are collected by the dendrites.
2. The cells body sums the incoming signals
3. When sufficient input is received (i.e. a threshold is exceeded), the neuron generates an action potential (i.e. it 'fires').
4. That action potential is transmitted along the axon to other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



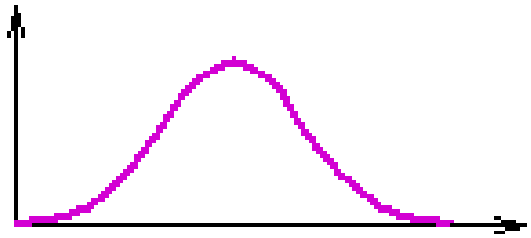
Examples of activation functions



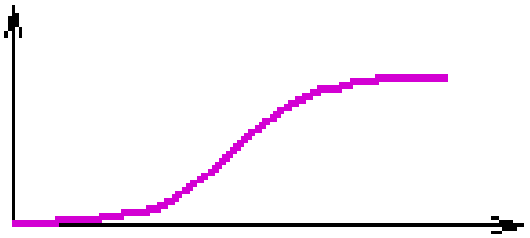
Threshold



Linear

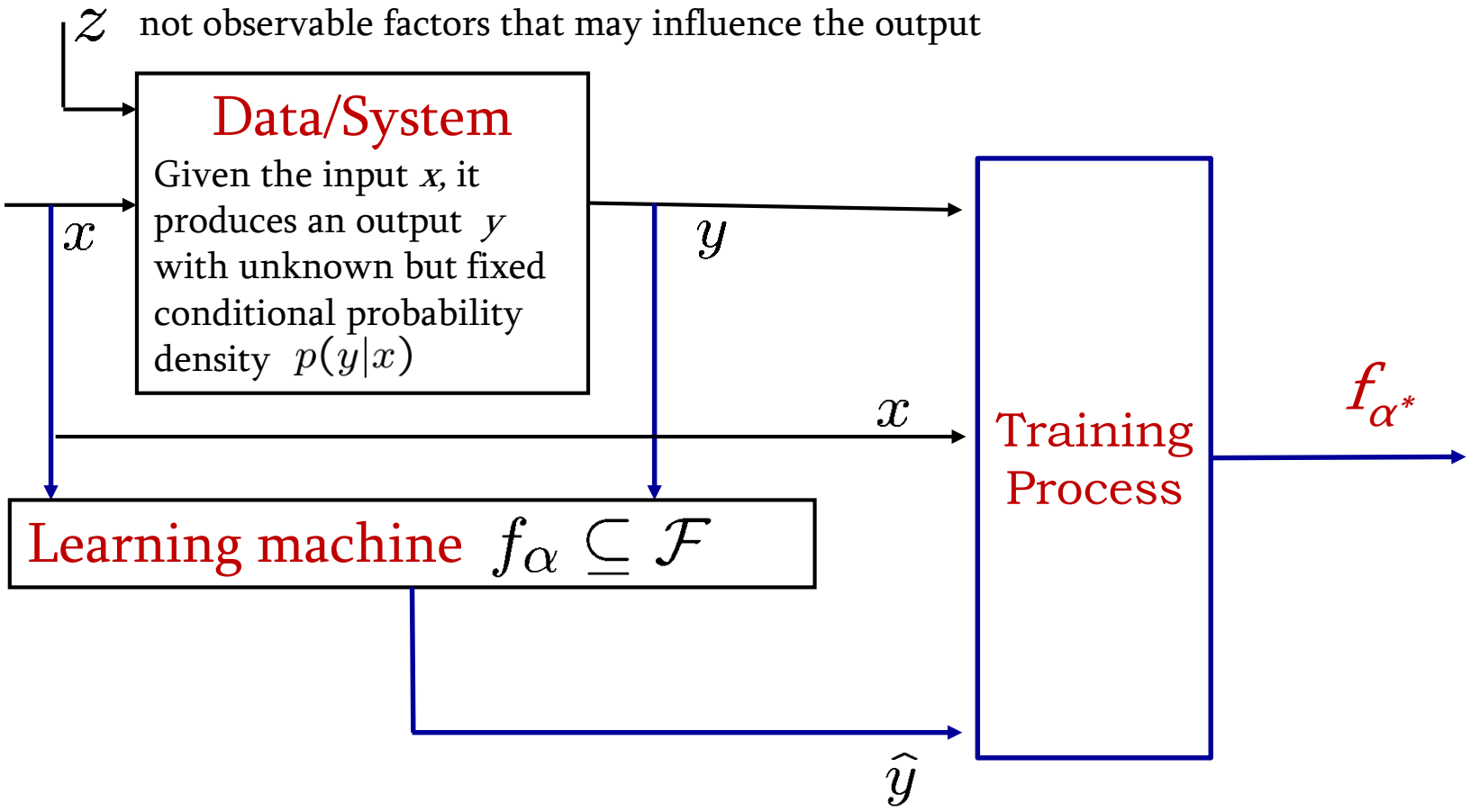


Gaussian



Sigmoid

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

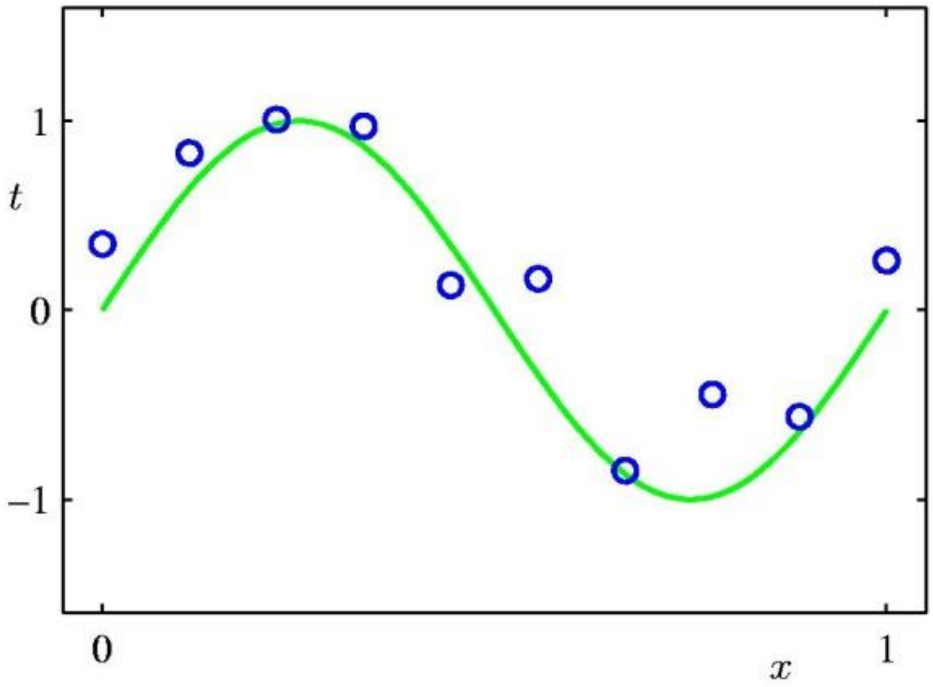
There is a general belief that for flexible learning methods with finite samples, the best prediction performance is provided by a model of optimum complexity.

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).

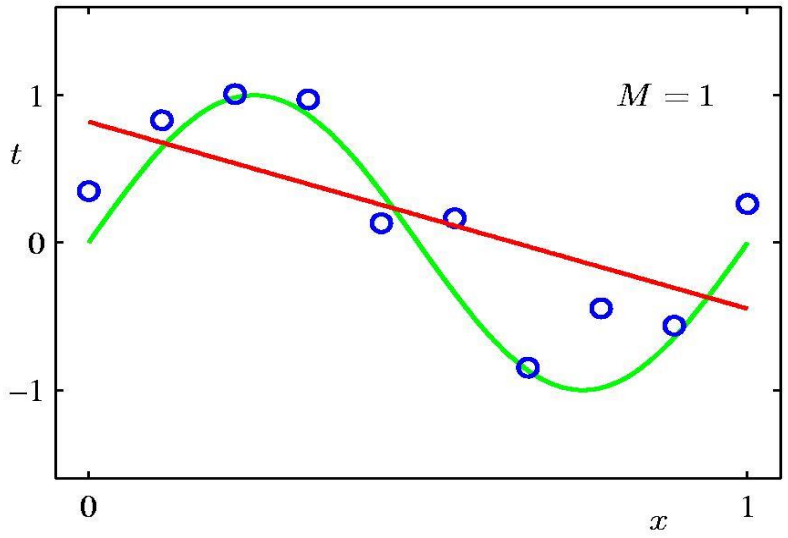


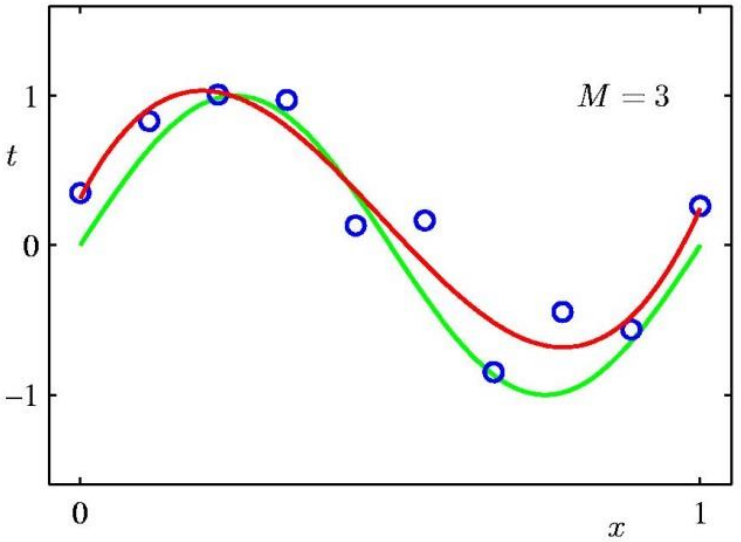
Our goal is to predict the target value for some new value of x, without knowledge of the green curve.

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)…

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

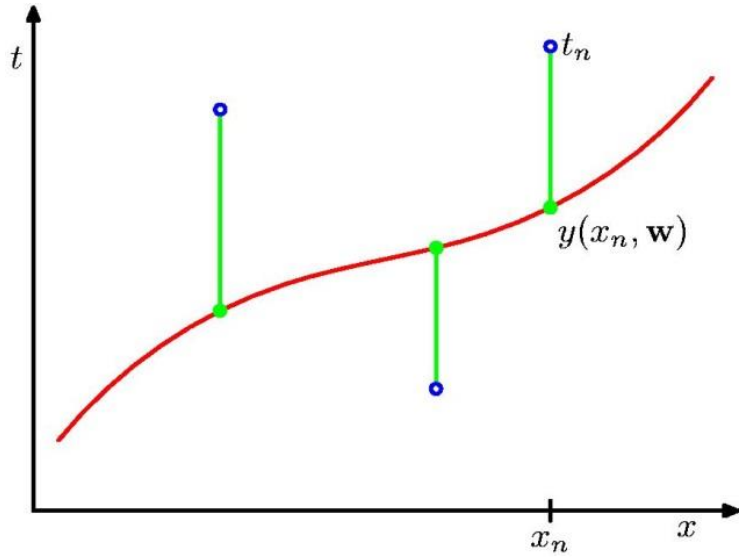
The values of the coefficients will be determined by fitting the polynomial to the training data. This can be done by minimizing an error function that measures the misfit between the function y(x,w), for any given value of w, and the training set
data points.

…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 $y(x,\mathbf{w}^*)$.



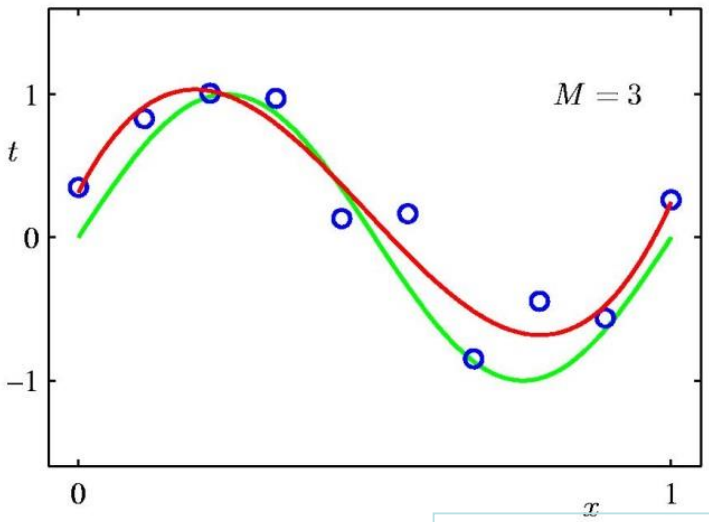
$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n,\mathbf{w}) - t_n\}^2$$

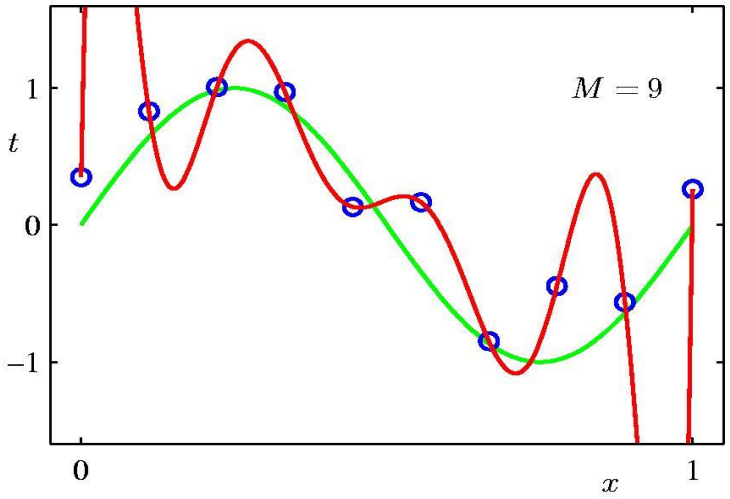
The error on the training data (the sum of the squares of the displacements (shown by the vertical green bars) of each data point from the function) may become zero, but what about the new data (test data)?

Parametric Regression

Let's increase the degree M from 3 to 9

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



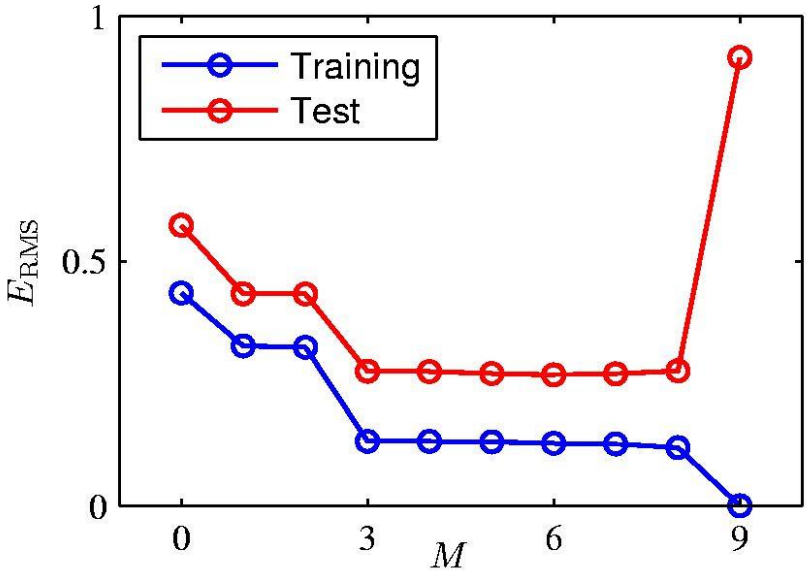


Which is the "better" one?

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

Error behaviour

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

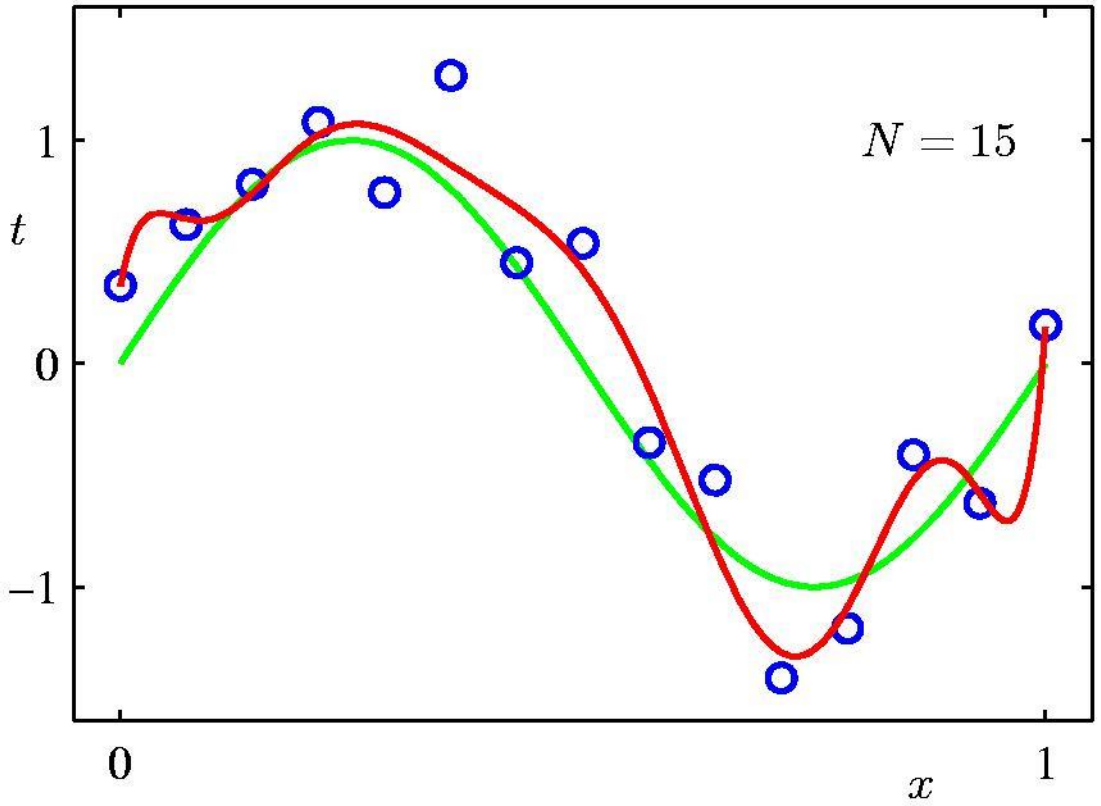


Driving the training error to zero may lead to big error on data test: **Over-fitting**

This is not an absolute rule !

Parametric Regression

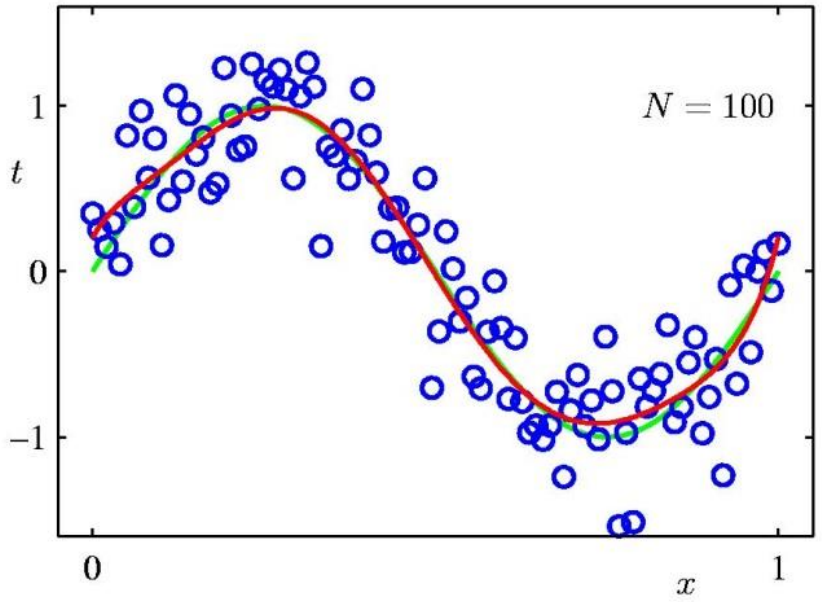
Indeed if the # of data instances increases to $N = 15$



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

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 ?

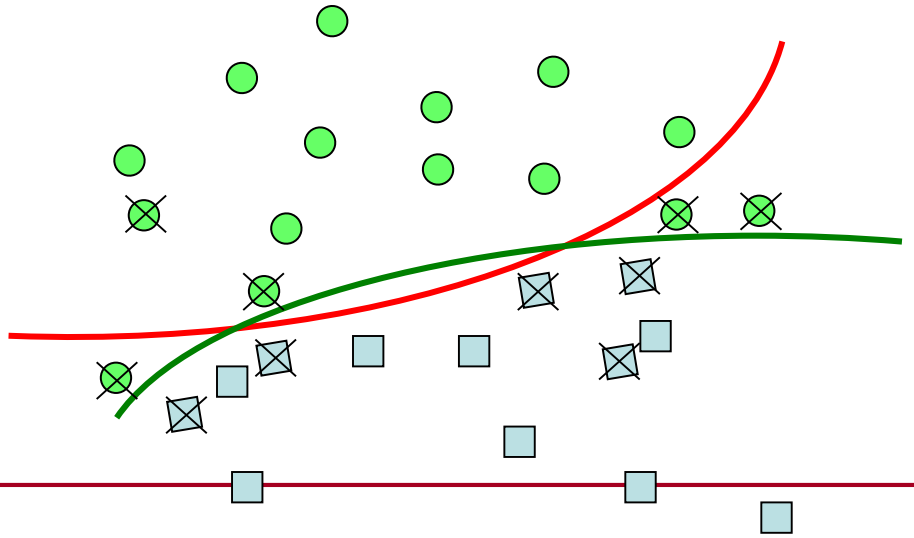
Empirical risk minimization

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

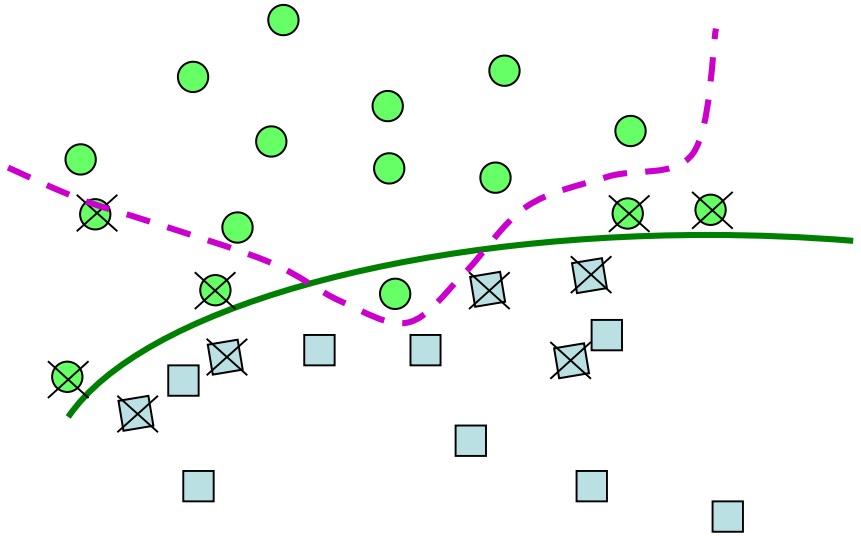


Complexity of the class

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

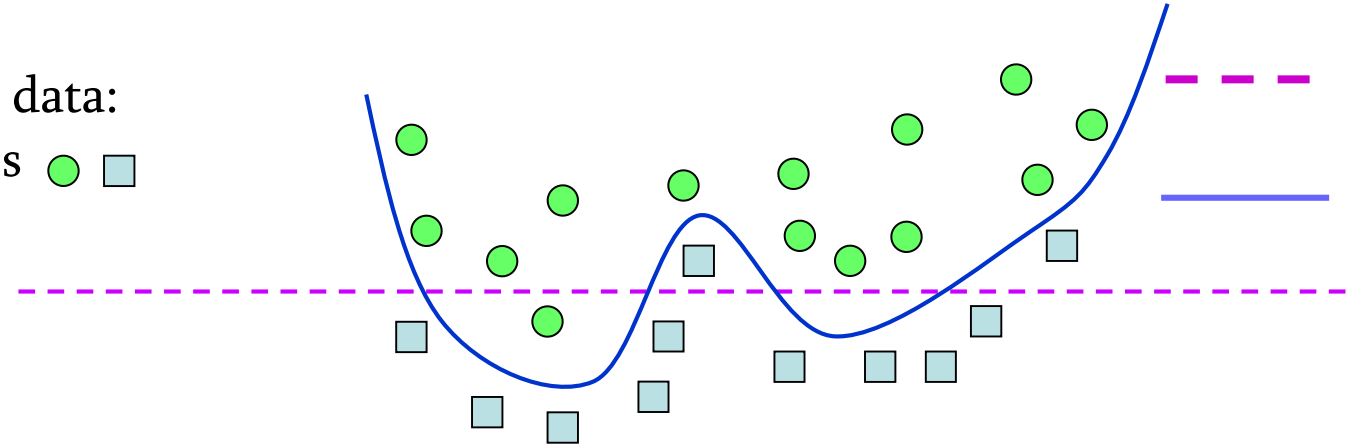
More complex

Simpler

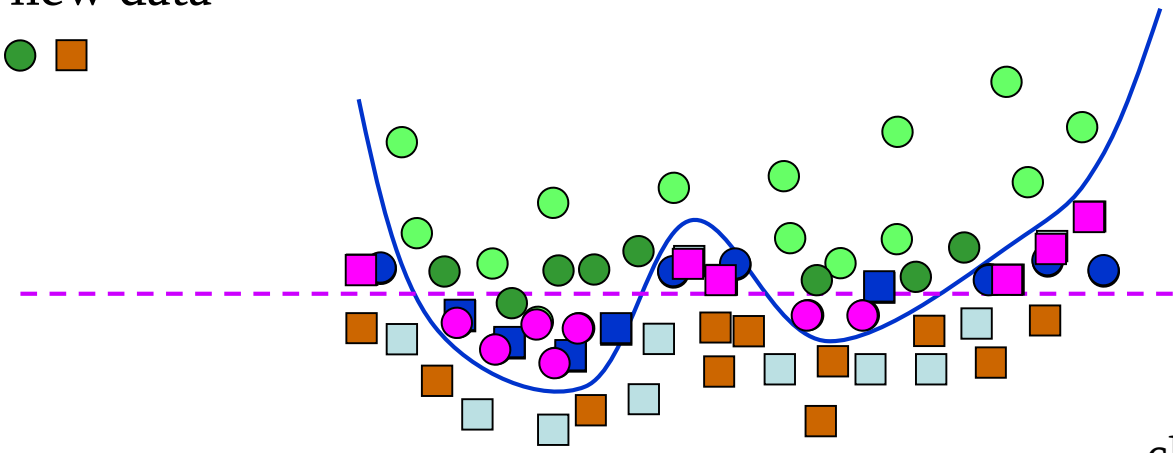


Over and under fitting

Trainig data:
2 classes



Add new data



underfitting
class f_α "too simple"

overfitting
class f_α "too complicated"

Beyond the ERM principle

It is possible to prove that

$$\Pr\{R(\propto) \leq R_{emp} + C_{VC}(\eta, l, h)\} = 1 - \eta$$

being $\eta \in (0,1)$.

h is a parameter which describes capacity of the class of functions. Capacity is a measure of complexity and measures the expressive power, richness or flexibility of a set of functions

This rule has been used to define a new inductive principle based on the "trade-off" between complexity of the class and minimization of the empirical risk

New inductive principle

Indeed the idea is to minimize the new functional

Penalization term on the complexity

$$\min_{f_\alpha} R_{emp} + C_{VC}(\eta, l, h)$$

- Vapnik Chervonenkis (VC) theory
  - the missing parameter is called VC dimension h
  - the second term $C_{VC}(\eta, l, h)$ is called VC confidence term

VC confidence

$$R(\alpha) \leq R_{emp}(\alpha) + C_{VC}(h, \ell, \eta)$$

VC dimension is the important parameter

$$C_{VC}(h, \ell, \eta) = \sqrt{\frac{h\left(\log\left(\frac{2\ell}{h}\right) + 1\right) - \log\left(\frac{\eta}{4}\right)}{\ell}},$$

The function to be minimized is

$$\min_{\alpha, f_\alpha} \; U(f_\alpha, \alpha) = R_{emp}(\alpha) + C_{VC}(h, \ell, \eta)$$

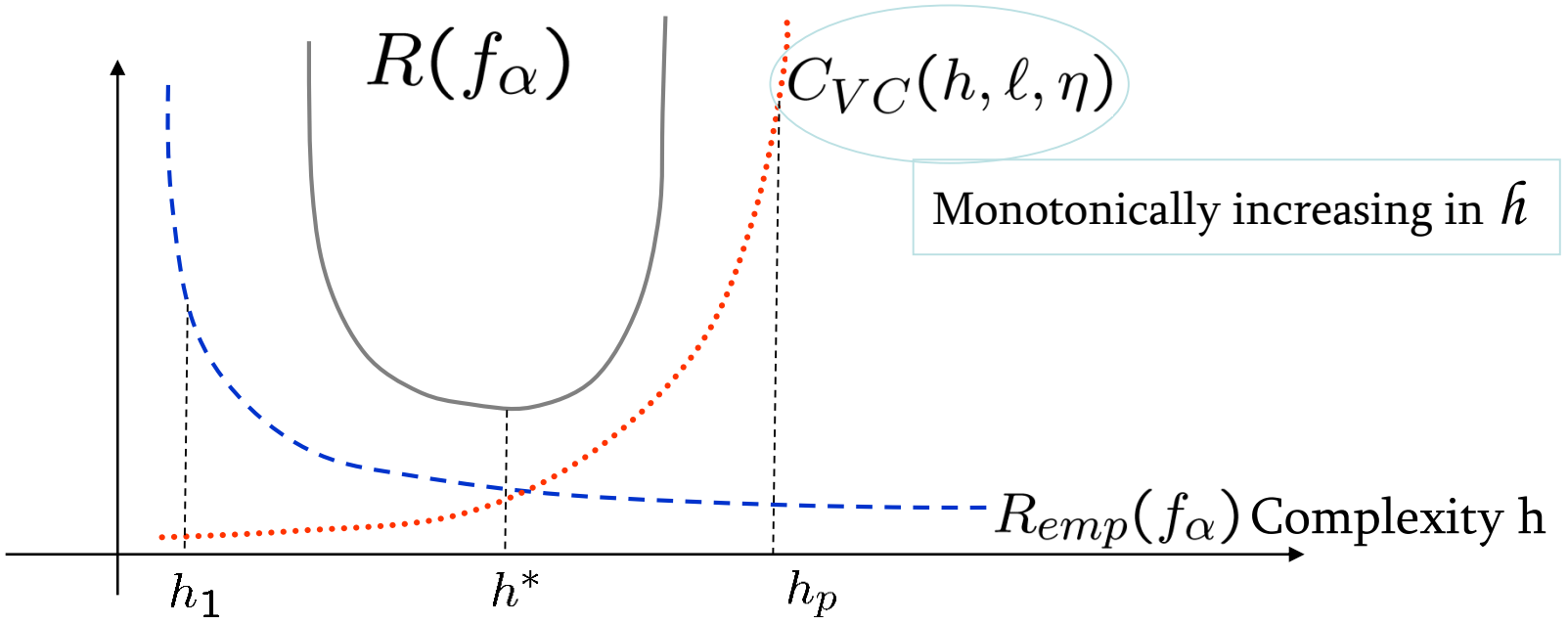
Minimization with respect to both the class and the parameters

Inductive principle SRM

The learning machine should be chosen by minimizing both

- The empirical risk
- The VC confidence

However the behaviors of the two terms as a function of h are opposite, hence the goal is to find the best "trade-off" between the minimization of the empirical risk and the VC confidence



Structural Risk Minimization principle

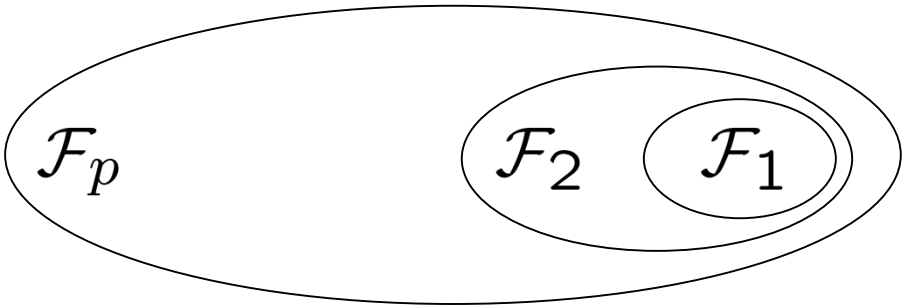
The VC confidence depends only on the class of function chosen and on the number of data, while the empirical risk depends of the particular values of the parameters chosen in the training phase that identifies one function in the class

Heuristic procedure to minimize $\min_{f_\alpha} U(f_\alpha)$

Choose an integer value for the VC dimension

Defined nesting class of function with NON Decreasing VC dimension value

$$\mathcal{F}_1 \subseteq \mathcal{F}_2 \subseteq \ldots \subseteq \mathcal{F}_p$$
$$h_1 \leq h_2 \leq \ldots < h_p$$



Structural Risk minimization principle

- For each class $\mathcal{F}^j$ with VC dimension h^j
- Find the optimal solution

$$\alpha^{j*} = \arg\min_{\alpha} R_{emp}(f_\alpha^j) \qquad f_\alpha^j \in \mathcal{F}^j$$

- Find the value of the upper bound

$$U^{j*} = U(\alpha^{j*}, f_\alpha^j) = R_{emp}(\alpha^{j*}) + C_{VC}(h^j, \ell, \eta)$$

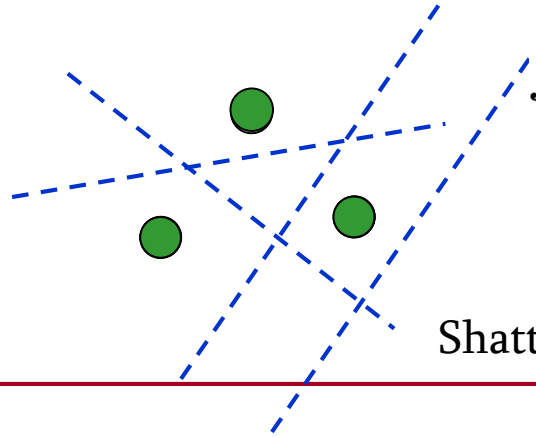
- Chose the class of functions which minimizes the upper bound

$$\mathcal{F}^* = \arg\min_{\mathcal{F}^j} U^{j*}$$

VC dimension

The Vapnik Chervonenkis dimension (VC dimension) h>0 is a measure of the capacity of the class of functions $\{f_\alpha\}$

The VC dimension h is equal to the maximum number of vectors x^i that can be shattered, namely that can be separated using this set of functions $\{f_\alpha\}$ into two different classes when labeled as ±1 in all the 2^h possible ways



Shatter set

$$f_\alpha(x) = sign(\alpha^T x + \alpha_0)$$

+1

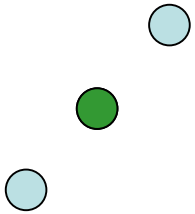
-1

these 3 vectors in R^2 can be seprated into two classes by an hyperplane

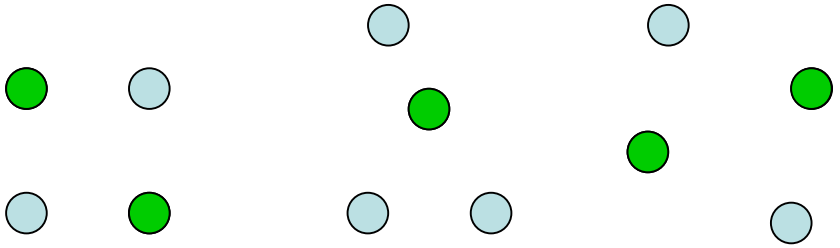
VC dimension

Stating that theVC dimensionof a class $\{f_\alpha\}$ is h means that we find at least a set of h points that can be shattered; it is not necessary that we be able to shatter any set of h points using $\{f_\alpha\}$

$$f_\alpha(x) = sign(\alpha^T x + \alpha_0)$$



This set of 3 points cannot be shattered in R^2



No set of 4 points in R^2 can be shattered by an hyperplane

TheVC dimension of the class $f_\alpha(x) = \alpha^T x + \alpha_0$ in R^2 is h=3

VC confidence

To obtain the VC confidence we need to know the value of the VC dimension h for a class of functions.

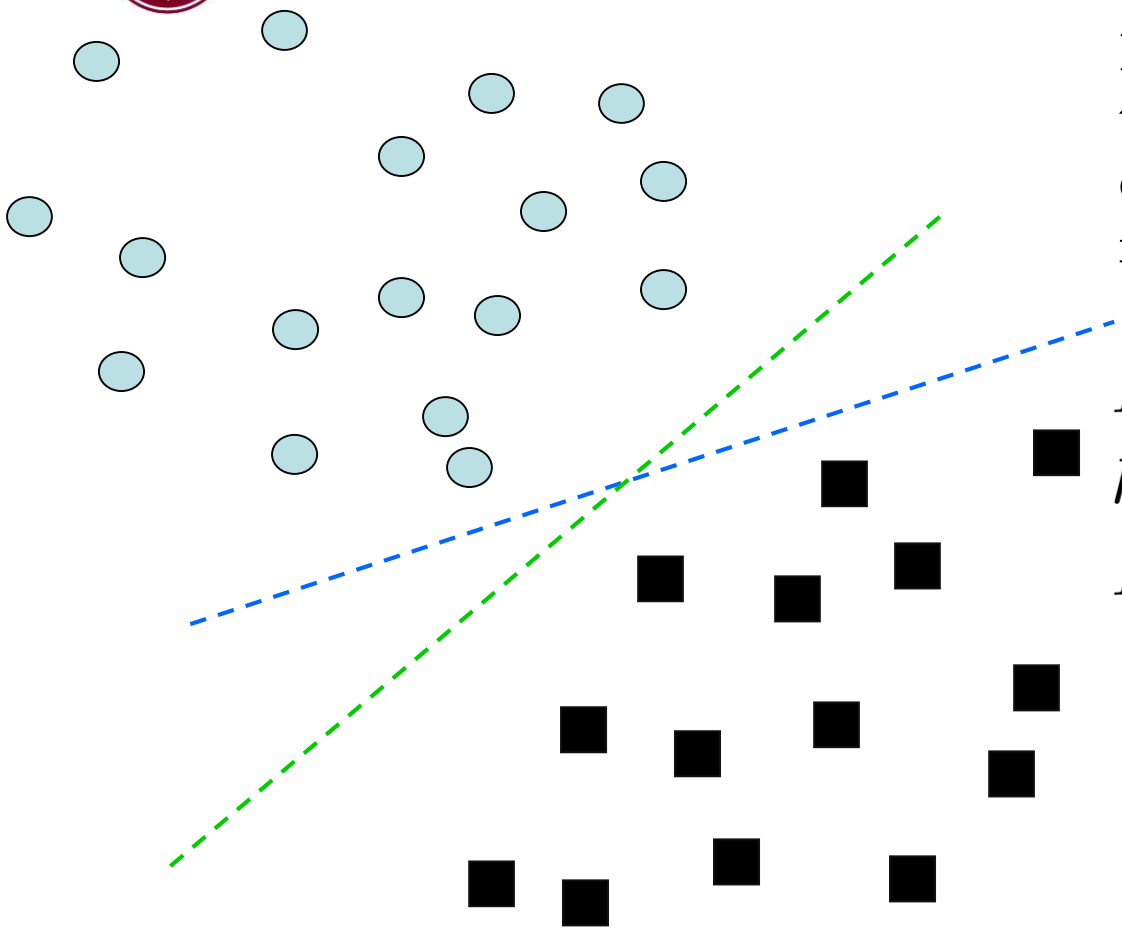
The numbers of parameters is NOT a useful information. Indeed h is not proportional to the # parameters

It is not true that learning machines with high number of parameters have high value of VC dimension. And vice versa learning machines with few parameter may not have low VC dimension.

h = is the maximum number of points x_i that can be can be classified for all possible assigned labels ±1

Linear separating function in R^2

$$f_{w,b}(x) = sign\{w^T x + b\}$$



In R^2 the VC dimension h=3; in this case we get the same value of the bound for both the functions in the class

$$R_{emp}(f_\alpha) = 0$$
$$h = 3$$
$$R(f_\alpha) \leq 0 + C_{VC}(3, \ell, \eta)$$

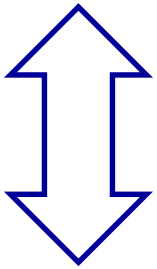
Is is true in general ?

VC dimension of hyperplans

Theorem: The VC dimension of hyperplanes in n dimensions is n+1.

To prove it we show that

$x^1, \ldots, x^p \in R^n$ are affinely independent



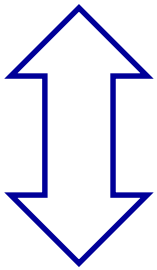
$x^1, \ldots, x^p \in R^n$ can be shattered by the class of function

$$f_{w,b} = w^T x + b$$

Definition

Let $\{x^0, x^1 .. x^p\}$ be points in R^n. These points are called affinely independent if there exist NO real numbers $\alpha_0, \alpha_1 ... \alpha_p$ not all zero such that $\sum^p_{i=0} \alpha_i x^i = 0$ and $\sum^p_{i=0} \alpha_i = 0$.

$x^1, \ldots, x^p \in R^n$ are affinely independent



$x^2 - x^1, \ldots, x^p - x^1$ are linearly independent

Hence the VC dimension of the hyperplane in R^n is the maximum number of affinely independent in R^n which is =n+1

Proof

On the blackboard……..

Correctly classified means:

$$w^T x^i + b > 0 \qquad y^i = 1$$
$$w^T x^i + b < 0 \qquad y^i = -1 \qquad\qquad x^i \in R^n$$
$$y^i(w^T x^i + b) > 0 \quad i = 1\ldots, \ell$$

Without loss of generality x_1=0

$$x^2, \ldots, x^m \text{ lin. ind.} \leftrightarrow \exists\ \bar{w} \in R^n, \bar{b} \in R$$
$$y^i\left(\bar{w}^T x^i + \bar{b}\right) > 0\ \forall\ y^i \in \{-1, 1\},\ \ i = 1, \ldots, m$$

Sketch of the proof

$(\rightarrow)$ The linear system $w^T x^i = y^i$ has full rank so that admits at least a solution $\bar{w} \in \mathbb{R}^n \ \forall i = 2, \ldots, m$ Posing

$$\bar{b} = \begin{cases} 1/2 & \text{if } \ y^1 = +1 \\ -1/2 & \text{if } \ y^1 = -1 \end{cases}$$

we get

$$y^1(w^T x^1 + \bar{b}) = y^1 \bar{b} > 0$$

$$y^i(\bar{w}^T x^i + \bar{b}) = 1 \pm \tfrac{1}{2} \qquad i = 2, \ldots, m$$

q.e.d

Sketch of the proof

($\leftarrow$) By contradiction assume that vectors x^i $i = 2, \ldots, m$ are linearly dep. so that

$$0 = \sum_{i=1^m} \lambda_i x^i = \sum_{\lambda_i > 0} \lambda_i x^i - \sum_{\lambda_i < 0} |\lambda_i| x^i,$$

Assign $y^i = \begin{cases} +1 & \text{if } \lambda_i > 0 \\ -1 & \text{if } \lambda_i < 0 \end{cases}$ $i = 2, \ldots, m$ and let $\bar{w} \in R^n$ and $\bar{b} \in R$ such that $y^i(\bar{w}^T x^i + \bar{b}) > 0 \qquad i = 1, \ldots, m.$

We get $0 = \sum_{i=1}^{m} \lambda_i \bar{w}^T x^i > \bar{b} \left(\sum_{\lambda_i < 0} |\lambda_i| - \sum_{\lambda_i > 0} \lambda_i \right) \neq 0,$

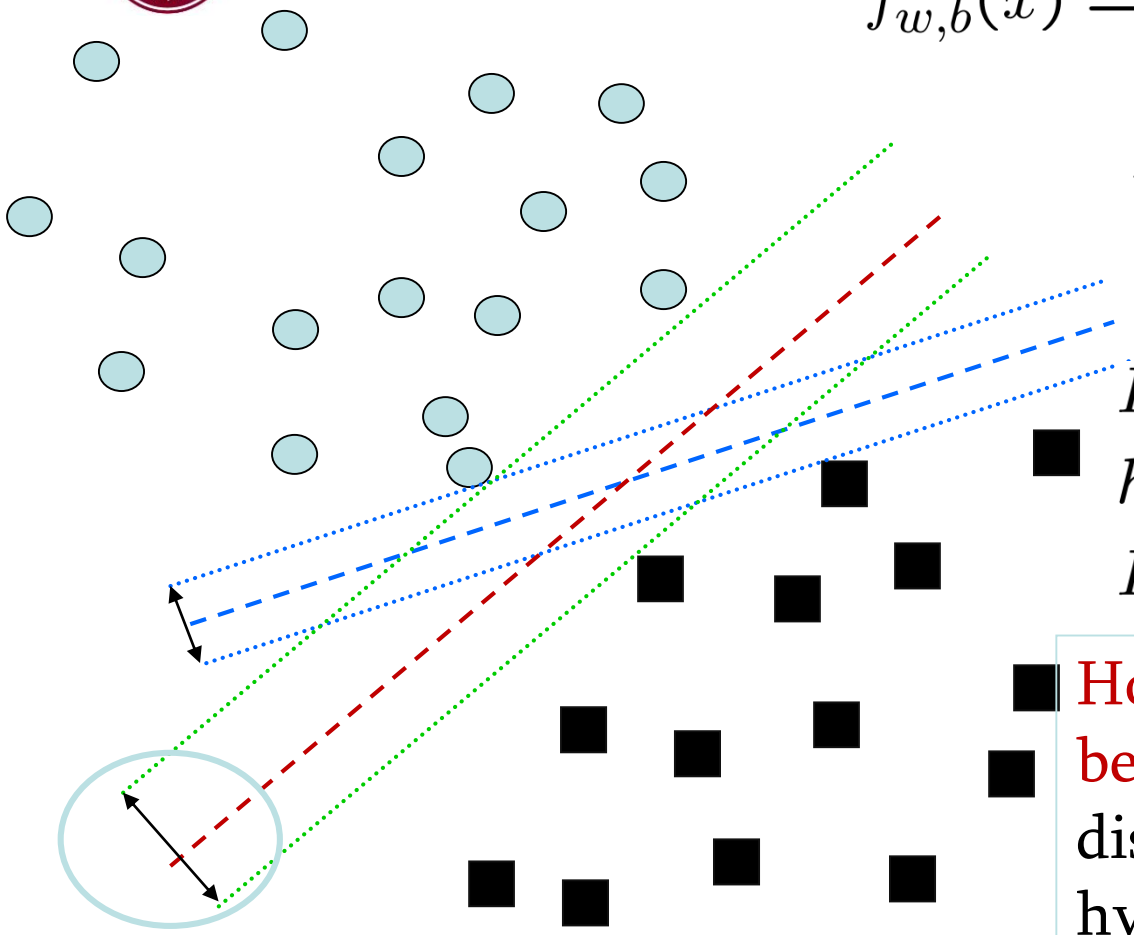
$$\left(\sum_{\lambda_i < 0} |\lambda_i| - \sum_{\lambda_i > 0} \lambda_i \right) \begin{cases} > 0 \text{ implies } \bar{b} < 0 \text{ impossibile for } y^1 = 1 \\ < 0 \text{ implies } \bar{b} > 0 \text{ impossibile for } y^1 = -1 \end{cases}$$

q.e.d

Computation of the VC dimension

The VC dimension of class of function "hyperplanes" is n+1

$$f_{w,b}(x) = sign\{w^T x + b\}$$



In this case we get the same value of the bound for both the functions in the class

$$R_{emp}(f_\alpha) = 0$$

$$h = 3$$

$$R(f_\alpha) \leq 0 + C_{VC}(3, \ell, \eta)$$

However the red one seems better: it "maximizes" the distance among the hyperplane and points in the two sets

Which one of the two is the best ?

Intuition: hyperplane with margin

- A hyperplane that passes too close to the training examples will be sensitive to noise and less likely to generalize well for new data

- Instead, it seems reasonable to expect that a hyperplane that is farthest from all training examples will have better generalization capabilities

Hyperplane with margin

Simple linear classification are useless as possible class of functions

Idea: restrict the choice within the class of linear classification may improve the VC dimension

Linear Classification with tolerance gap (≈margin)

Hyperplane with tolerance gap

Assume that data x^i stay in a sphere with diameter D=2R, namely $||x^i|| \leq R$

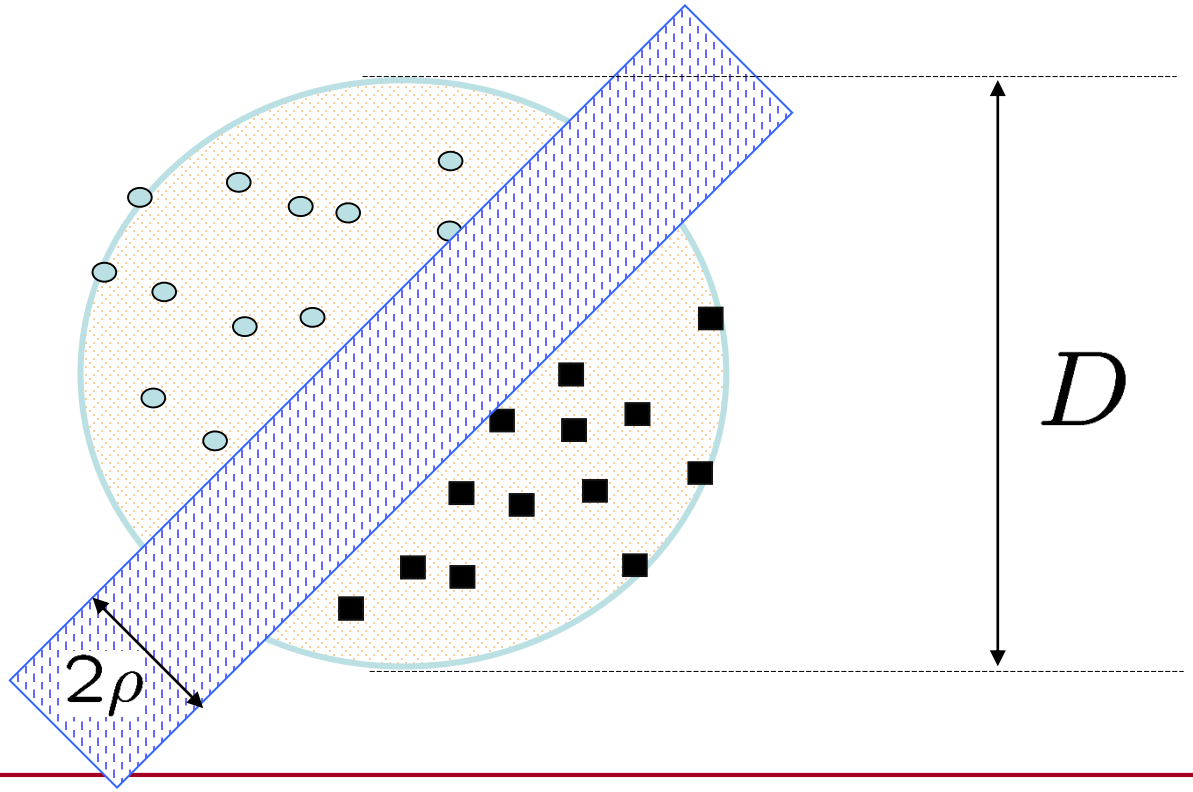
Consider the hyperplane $w^T x + b = 0$

$d(x^i; w, b)$ is the distance of x^i from the hyperplane

$$y^i = \begin{cases} 1 & \text{se } d(x^i; w, b) \geq \rho \\ -1 & \text{se } d(x^i; w, b) \leq -\rho \end{cases}$$

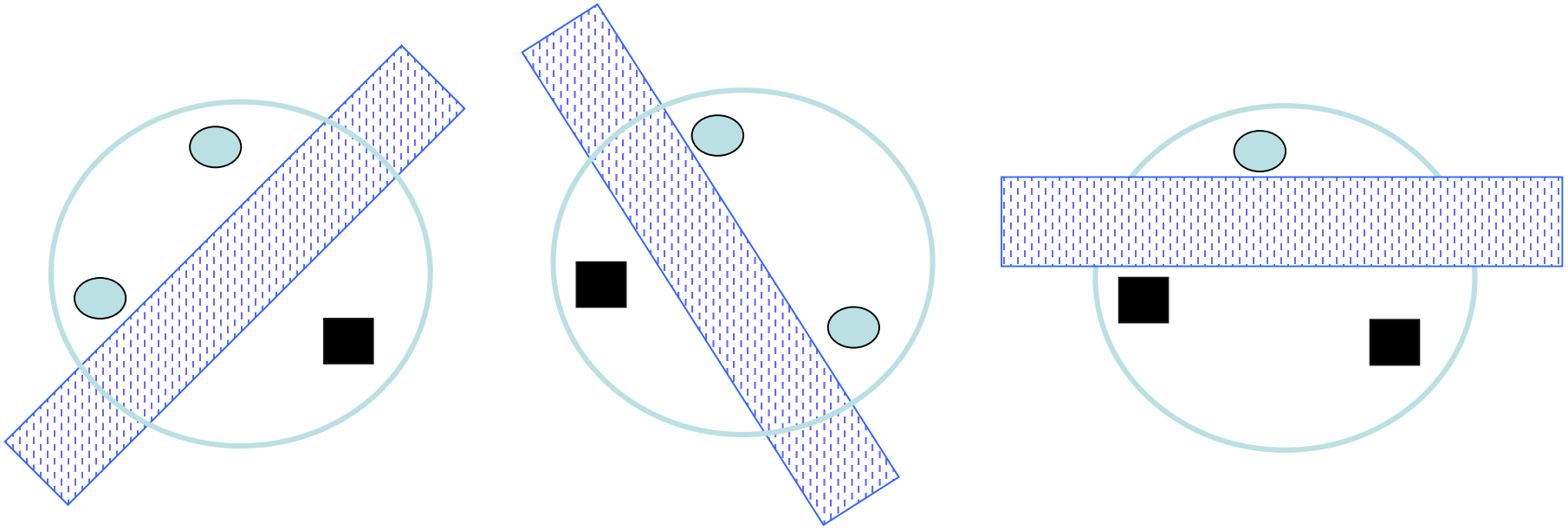
Hyperplane with tolerance gap

Hyperplane with tolerance gap classifies vectors within a sphere of diameter D and out of a tolerance 2ρ



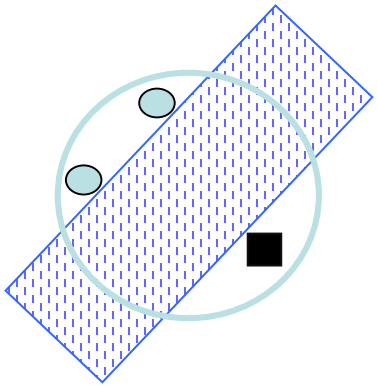
Hyperplane with tolerance gap

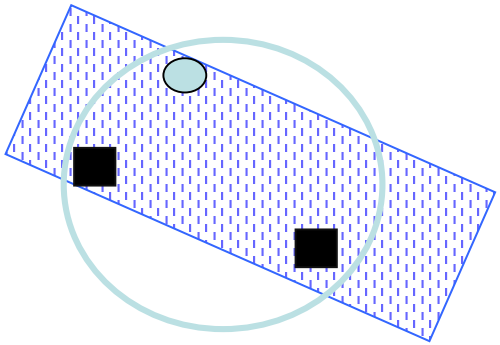
If ρ is small with respect to D, it is still possible to shatter 3 points in R^2



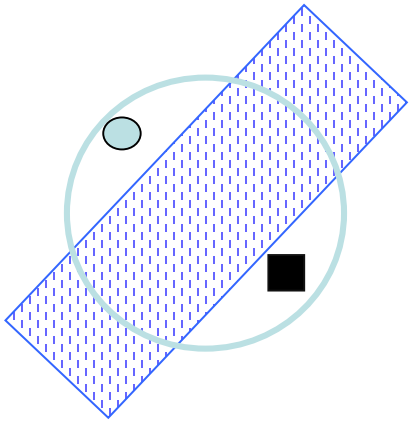
Hyperplane with tolerance gap

If ρ grows with respect to D, the number of points that can be shattered decreases



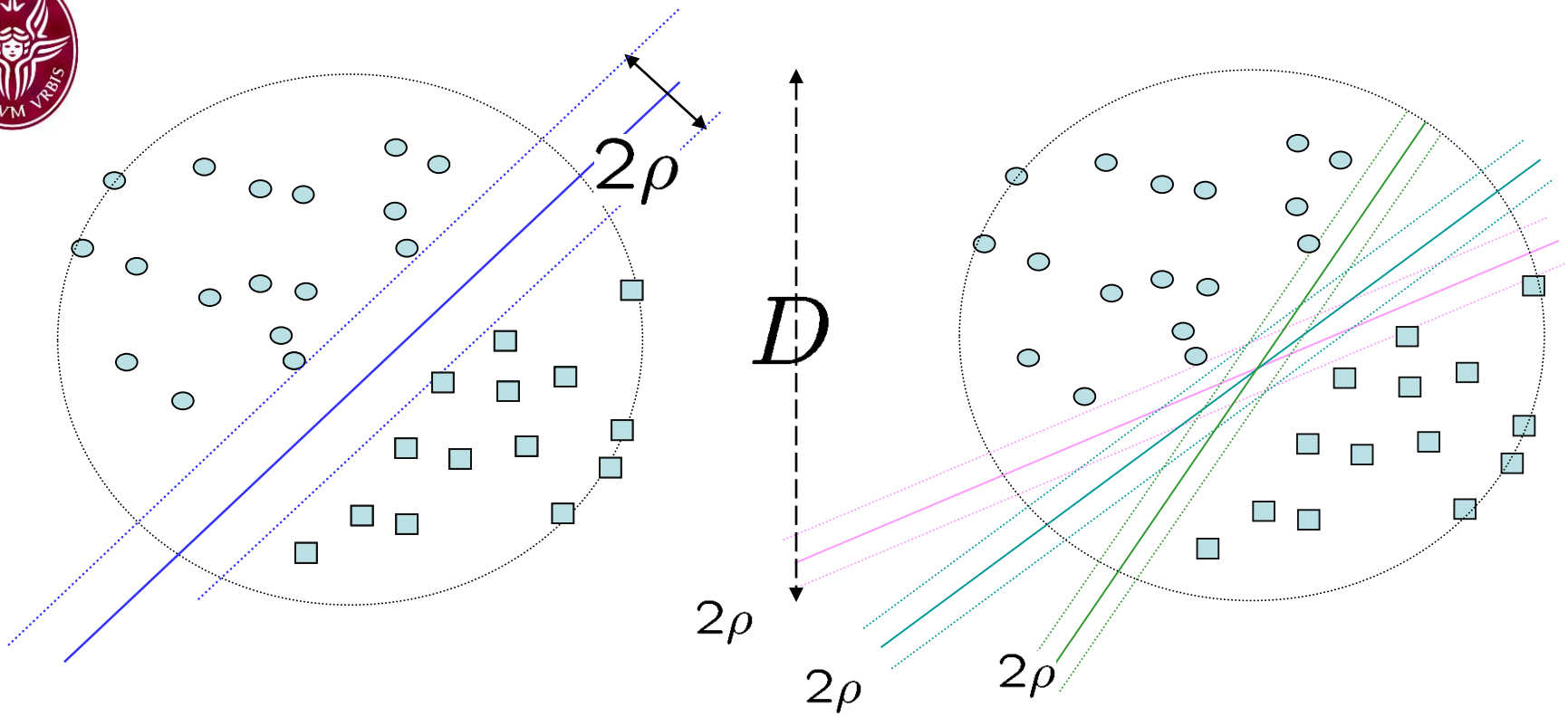


It is no more possible to shatter 3 points



It is still possible to shatter 2 points

Margin and VC confidence



The highest the margin the lowest the VC dimension h

$$h \leq \min\left\{\left\lceil \frac{D^2}{\rho^2} \right\rceil, n\right\} + 1 \leq n + 1$$

h hyperplane with tolerance gap ≤ h hyperplane

Structural risk minimization using hyperplane with tolerance gap

$$\mathcal{F}_i = \{f : R^n \rightarrow R : f(x) = w^T x + b,\ \ \rho(w,b) \geq \rho_i\},$$
$$\rho_1 \geq \rho_2 \geq \ldots \geq \rho_p$$

- Minimizing the upper bound on the risk
  - minimizing the empirical risk
  - maximizing the margin (hence minimizing the VC confidence)
- for each function f^j in the class
  - obtain a bound on h^j
  - minimize the empirical risk R^j_{emp}
  - compute the bound on the risk
- Chose the class with minimum bound

METODI

Minimization of the structural risk

- Support Vector Machines
  - empirical risk fixed to a given value
  - minimize VC confidence
- Neural Network
  - fixed the architecture (hence the complexity and VC confidence)
  - minimize the empirical risk