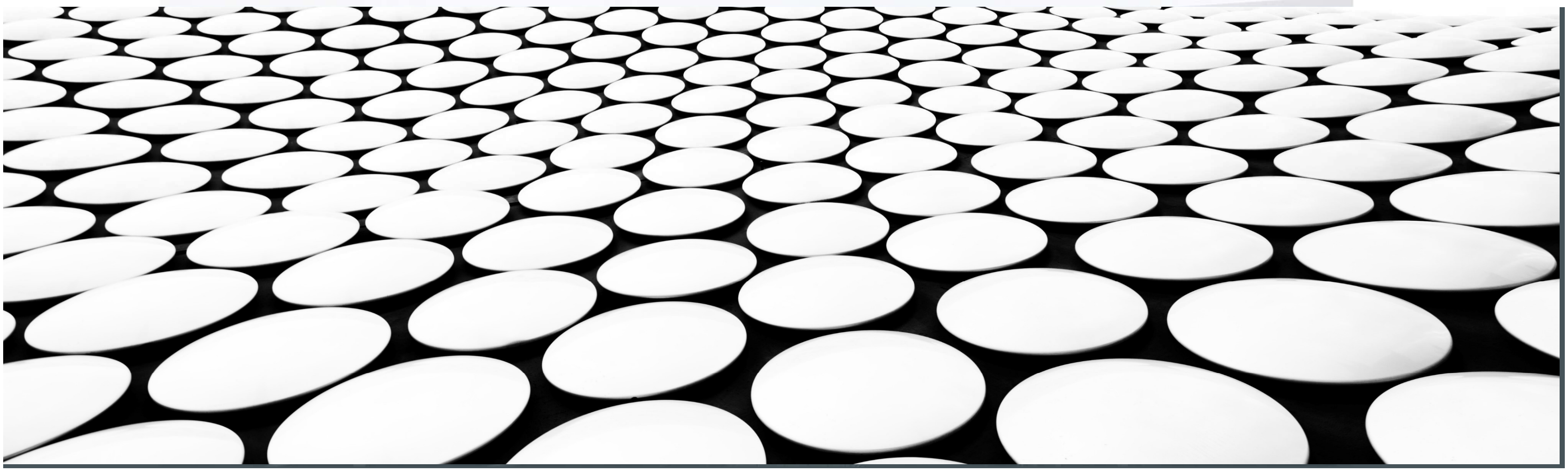
INTRODUCTION TO STATISTICAL LEARNING THEORY

ROBERTO BASILI, UNIVERSITÀ DI ROMA, TOR VERGATA

DEEP LEARNING, MARCH 2025



OUTLINE

- Statistical Learning Theory
  - PAC learnability
  - VC dimension
  - Learning Machines
  - Model Optimization and Concept Class complexity
  - Model Optimization via Cross-Validation
  - Towards perceptrons and SVMs

FROM STATISTICAL LEARNING THEORY TO SVMS

LEARNİNG A CLASS FROM EXAMPLES

- Class C of a “family car”
  - Prediction Is car x a “family car”?
  - Knowledge extraction What do people expect from a family car?

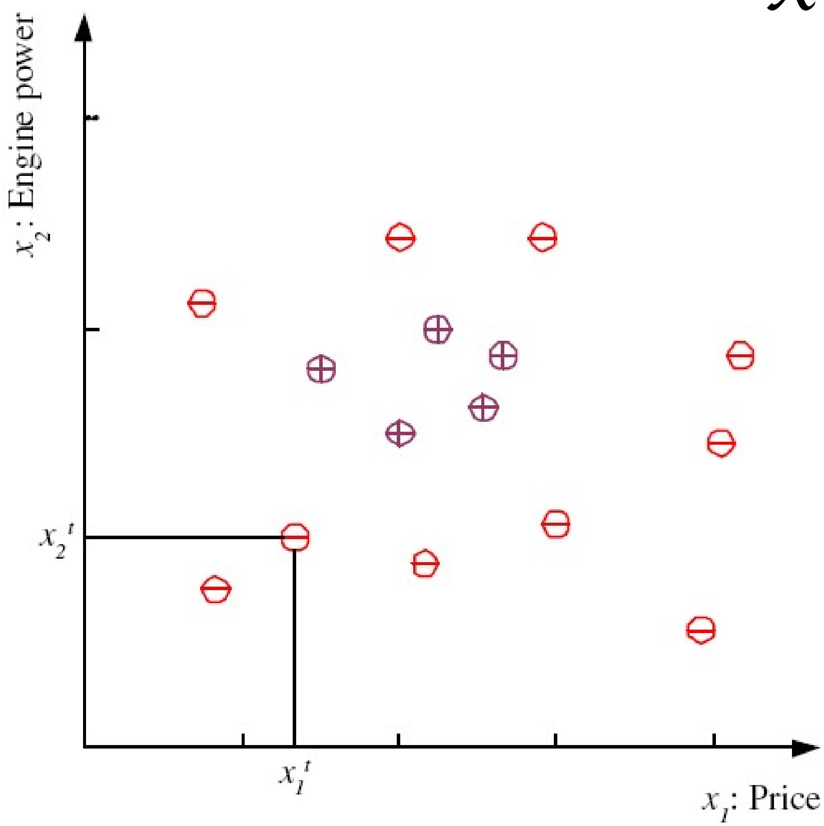
- Output:

  Positive (+) and negative (–) examples

- Input representation:

  x_1: price, x_2 : engine power

TRAİNİNG SET $\mathcal{X}$



$$\mathcal{X} = \{\boldsymbol{x}^t, r^t\}_{t=1}^N$$

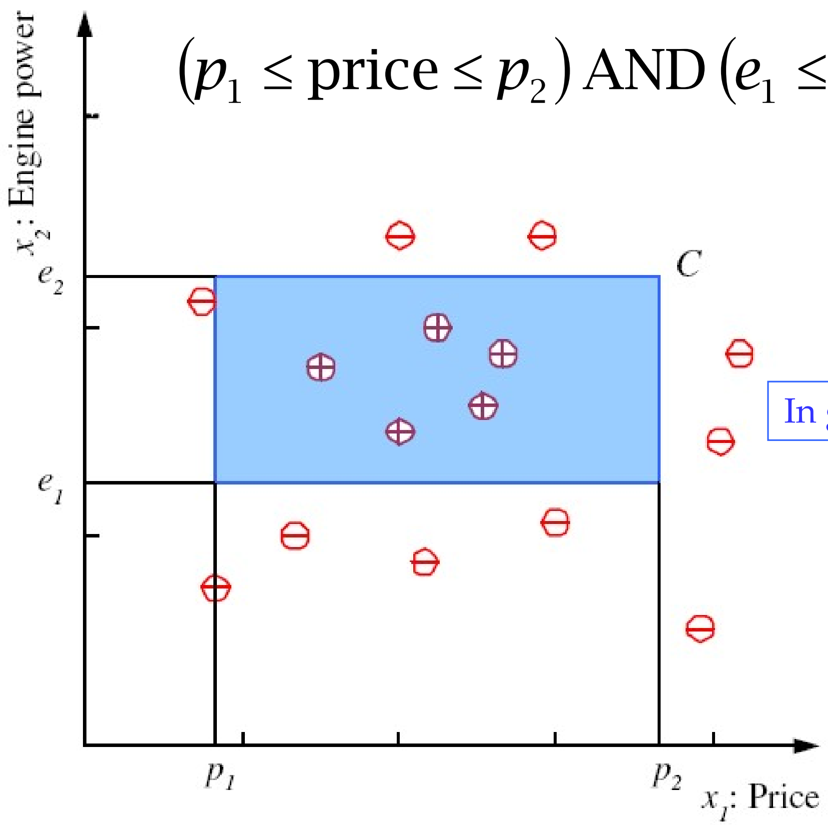
label

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$r = \begin{cases} 1 \text{ if } \boldsymbol{x} \text{ is positive} \\ 0 \text{ if } \boldsymbol{x} \text{ is negative} \end{cases}$$

CLASS C

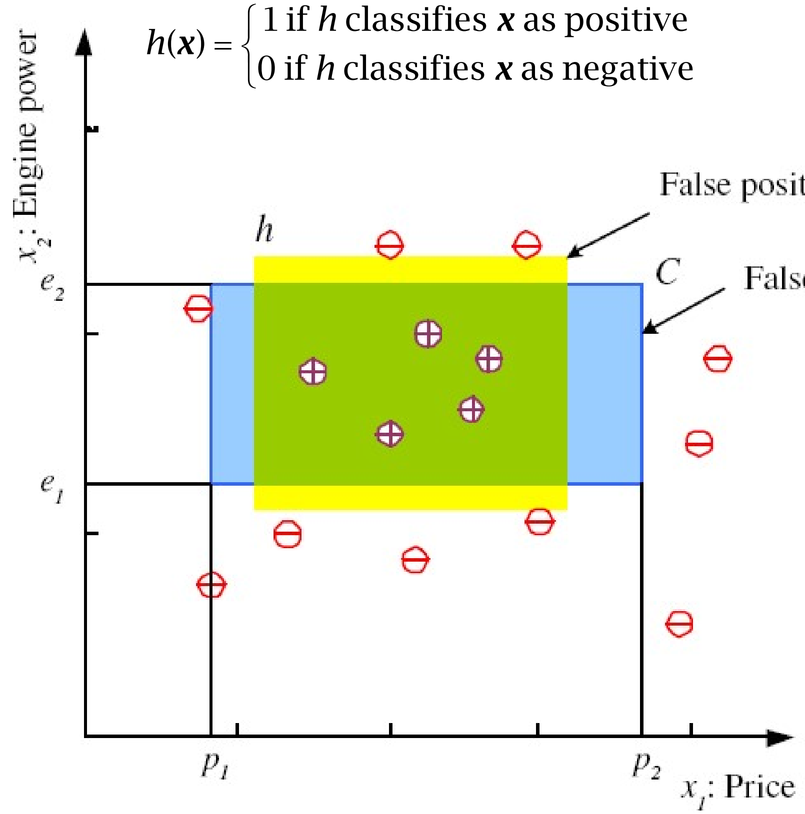
$$(p_1 \leq \text{price} \leq p_2) \text{ AND } (e_1 \leq \text{engine power} \leq e_2)$$



In general we do not know C(x).

HYPOTHESİS CLASS $\mathcal{H}$

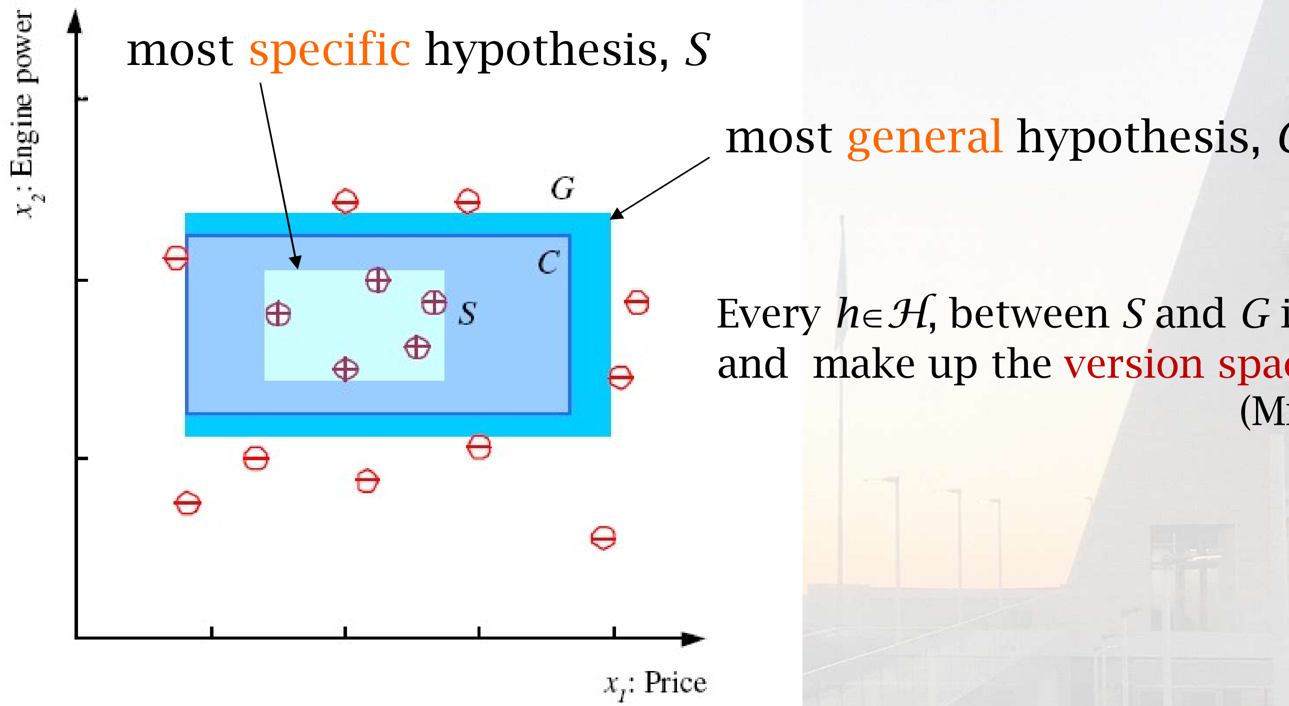
$$h(\boldsymbol{x}) = \begin{cases} 1 \text{ if } h \text{ classifies } \boldsymbol{x} \text{ as positive} \\ 0 \text{ if } h \text{ classifies } \boldsymbol{x} \text{ as negative} \end{cases}$$



Empirical error:

$$E(h \mid \mathcal{X}) = \sum_{t=1}^{N} 1\left(h\left(\mathbf{x}^t\right) \neq r^t\right)$$

S, G, AND THE VERSİON SPACE



Every $h \in \mathcal{H}$, between S and G is consistent and make up the version space

(Mitchell, 1997)

PROBABLY APPROXİMATELY CORRECT (PAC) LEARNİNG

- How many training examples are needed so that the tightest rectangle S which will constitute our hypothesis, will probably be approximately correct?
  - We want to be *confident* (*above a level*) that 1-δ
  - … the *error probability is bounded* by some value ε

- A concept class *C* is called *PAC-learnable* if there exists a PAC-learning algorithm such that, for any $\varepsilon>0$ and $\delta>0$, there exists a fixed sample size such that, for any concept $c \in C$ and for any probability distribution on X, the learning algorithm produces a probably-approximately-correct hypothesis *h*
- a (PAC) *probably-approximately-correct hypothesis h* is one that has error at most ε with probability at least 1-δ.

PROBABLY APPROXİMATELY CORRECT (PAC) LEARNİNG

- In PAC learning, given a class C and examples drawn from some unknown but fixed distribution p(x), we want to find the number of examples N, such that with probability at least 1-δ, h has error at most ε ? (Blumer et al., 1989)

$$P(C\Delta h \leq \varepsilon) \geq 1- \delta$$

- where $C\Delta h$ is (area of the) “the region of difference between C and h”, and $\delta>0$, $\varepsilon>0$.

PAC LEARNİNG

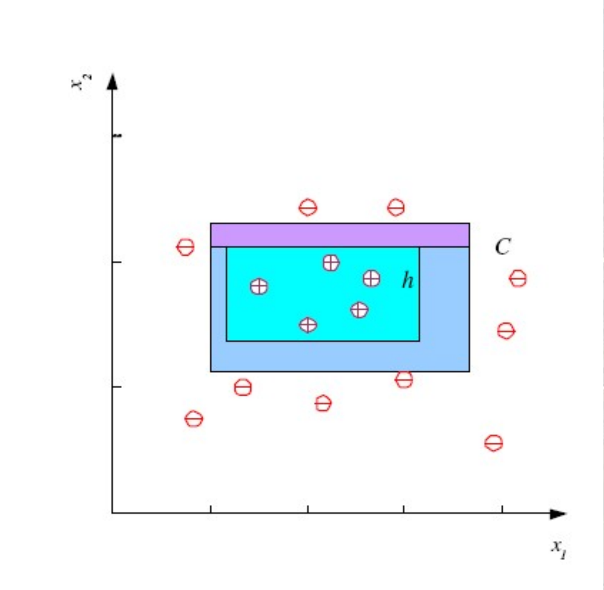
How many training examples m should we have, such that with probability at least $1 - \delta$, h has error at most ε ? (Blumer et al., 1989)

- Let prob. of a +ex. in each strip be at most $\varepsilon/4$
- Pr that a random ex. misses a strip: $1- \varepsilon/4$
- Pr that m random instances miss a strip:
  $(1 - \varepsilon/4)^m$
- Pr that m random instances instances miss 4 strips:
  $4(1 - \varepsilon/4)^m$
- We want $1-4(1 - \varepsilon/4)^m \geq 1-\delta$ or $4(1 - \varepsilon/4)^m \leq \delta$
- Using $1-x \leq e^{-x}$ an even stronger condition is:

[$(1-\varepsilon/4) \leq exp(-\varepsilon/4)$ so $(1-\varepsilon/4)^m \leq exp(-\varepsilon/4)^m = exp(-\varepsilon m/4)$]

$4e^{-\varepsilon m/4} \leq \delta$ OR

- Divide by 4, take ln... and show that $m \geq (4/\varepsilon)ln(4/\delta)$



PROBABLY APPROXİMATELY CORRECT (PAC) LEARNİNG

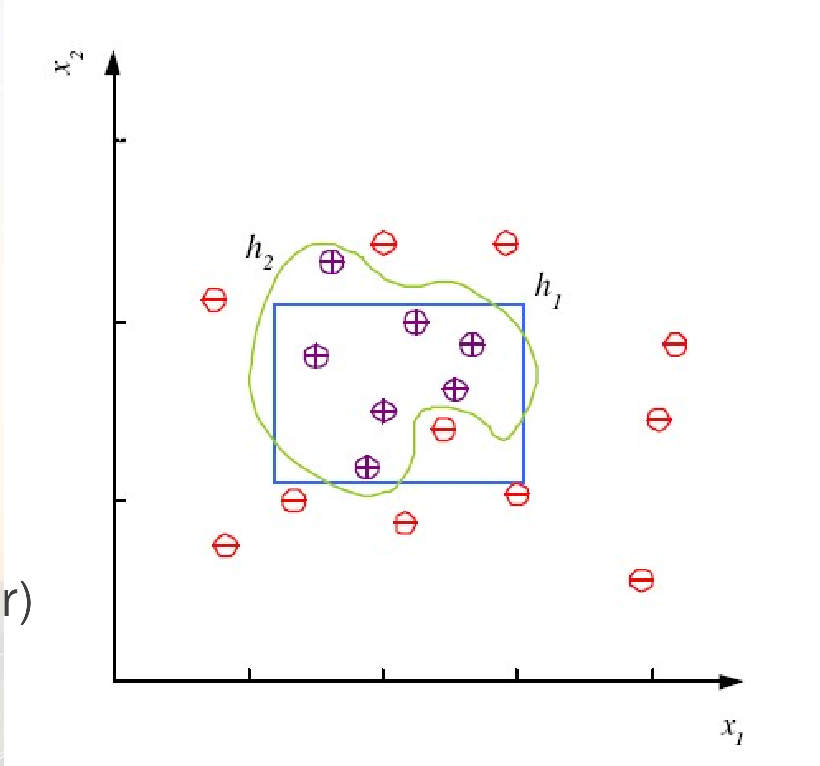
How many training examples m should we have, such that with probability at least 1 - δ, our hypothesis *h* has error at most ε ? (Blumer et al., 1989)

$$m \geq (4/\varepsilon) ln(4/\delta)$$

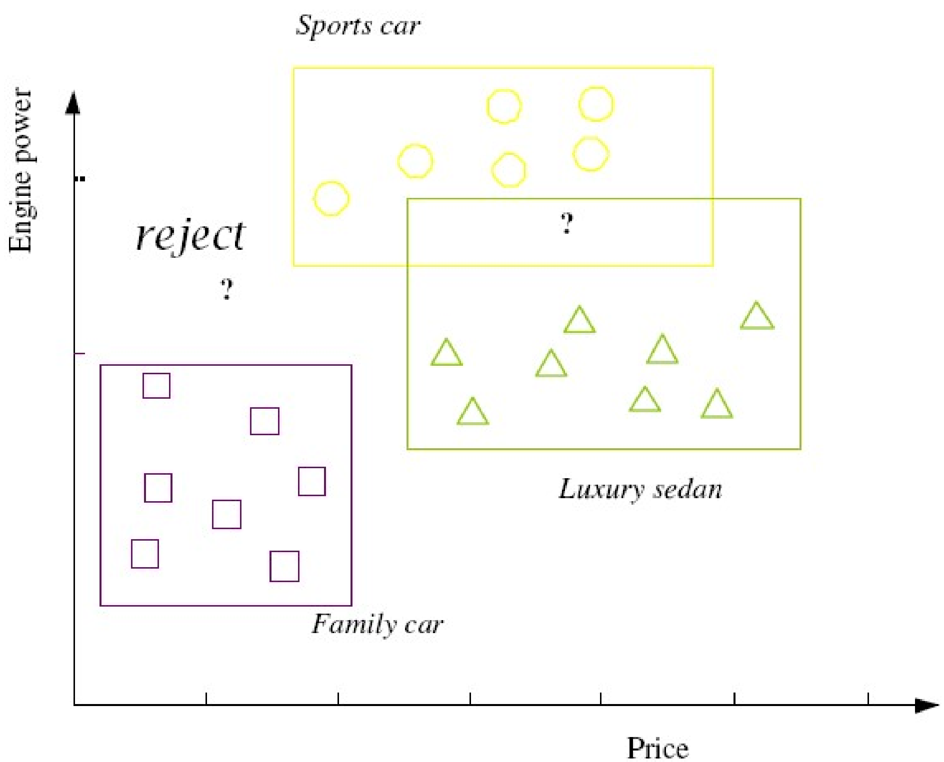
- *m* increases slowly with $1/\varepsilon$ and $1/\delta$
- Say ε=1% with confidence 95%, pick $m \geq 1752$
- Say ε=10% with confidence 95%, pick $m \geq 175$

MODEL COMPLEXİTY VS. NOISE

- Use the simpler one because
- Simpler to use (lower computational complexity)
- Easier to train (lower space complexity)
- Easier to explain (more interpretable)
- Generalizes better (lower variance – Occam’s razor)



MULTIPLE CLASSES, C_i i=1,...,K



$$\mathcal{X} = \{\boldsymbol{x}^t, r^t\}_{t=1}^N$$

$$r_i^t = \begin{cases} 1 \text{ if } \boldsymbol{x}^t \in C_i \\ 0 \text{ if } \boldsymbol{x}^t \in C_j, j \neq i \end{cases}$$

Train hypotheses
$h_i(\boldsymbol{x})$, $i = 1,...,K$:

$$h_i(\boldsymbol{x}^t) = \begin{cases} 1 \text{ if } \boldsymbol{x}^t \in C_i \\ 0 \text{ if } \boldsymbol{x}^t \in C_j, j \neq i \end{cases}$$

REGRESSİON

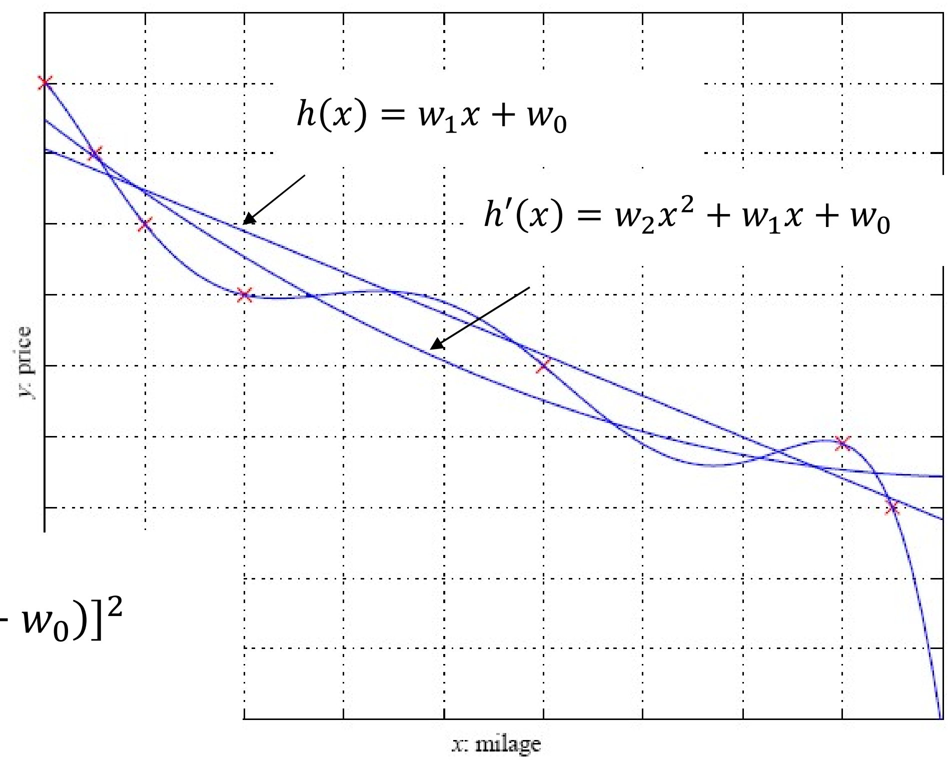
$$X = \{x^t, r^t\}_{t=1}^{N}$$

$$r^t \in \Re$$

$$r^t = f(x^t) + \varepsilon$$

$$E(h'|X) = \frac{1}{N}\sum_{t=1}^{N}[r^t - h'(x^t)]^2$$

$$E(h|X) = E(w_1, w_0|X) = \frac{1}{N}\sum_{t=1}^{N}[r^t - (w_1 x^t + w_0)]^2$$



VC-DIMENSION

- Are the finite samples that are sufficient to learn a concept even if it insists on a infinte domain

VC (VAPNİK-CHERVONENKİS) DİMENSİON

- N points can be labeled in 2^N ways as +/–
- $\mathcal{H}$ shatters N if there exists a set of N points such that $h \in \mathcal{H}$ is consistent with all of these possible labels:
  - Denoted as: $VC(\mathcal{H}) = N$
  - Measures the capacity of H

- Any learning problem definable by N examples can be learned with no error by a hypothesis drawn from H

What is the VC dimension of axis-aligned rectangles?

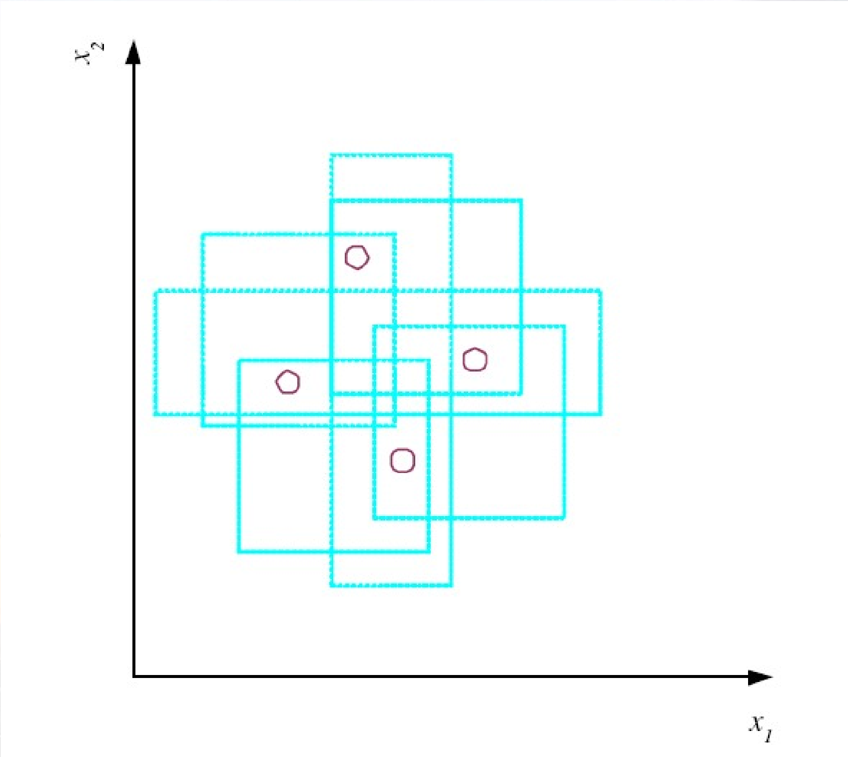
FORMAL DEFİNİTİON

- DEFINITION. The VC dimension of a set of functions $\mathcal{H}=\{h(\underline{x},\alpha)\}$ is d if and only if there exists a set of points $\{x_i\}$, with $i=1....d$, such that these points can be labeled in all the possible 2^d configurations, and for each labeling, a member of the set $\mathcal{H}$ can be found which correctly assigns those labels, but no set $\{x_i\}$, with $i=1....q$, where $q>d$, can be found satisfying this property.

VC (VAPNİK-CHERVONENKİS) DİMENSİON

- $\mathcal{H}$ shatters N if there exists N points and $h \in \mathcal{H}$ such that h is consistent for any labelings of those N points.

- VC($axis\ aligned\ rectangles$) = 4



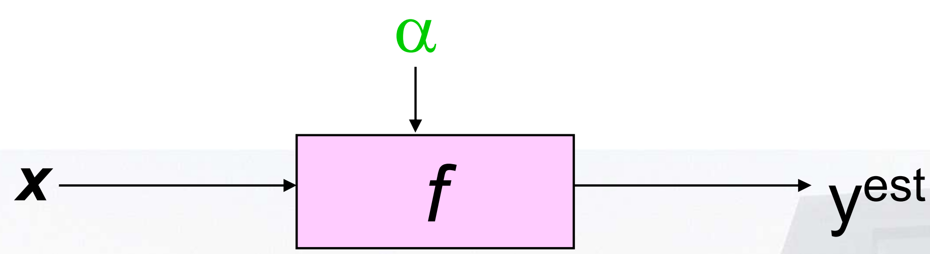
VC (VAPNİK-CHERVONENKİS) DİMENSİON

- *What does this say about using rectangles as our hypothesis class?*

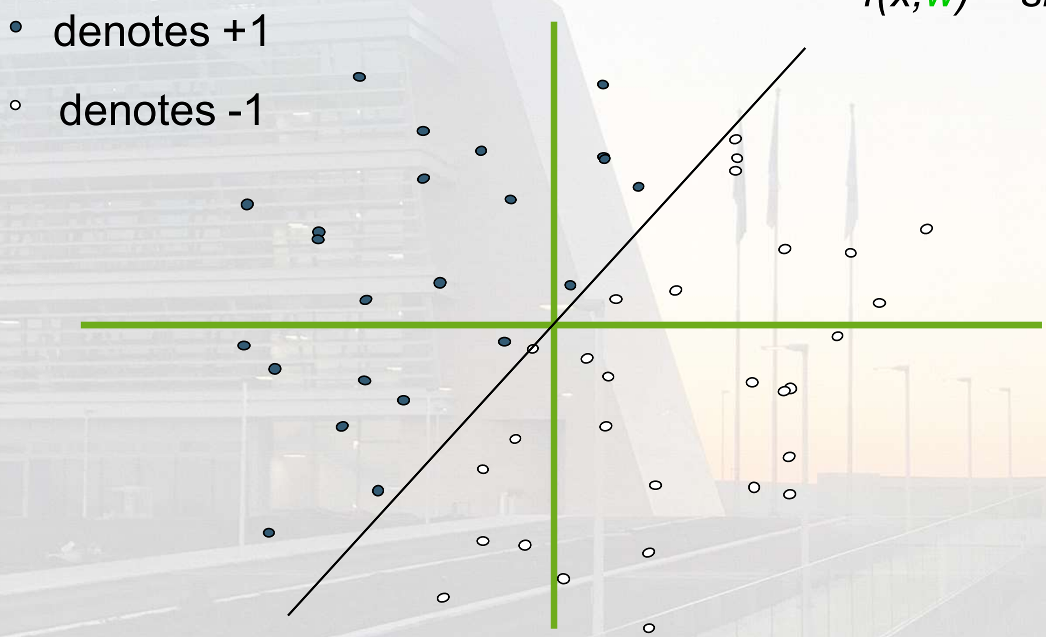
- VC dimension is pessimistic: in general we do not need to worry about all possible labelings

- It is important to remember that one can choose the arrangement of points in the space, but then the hypothesis must be consistent with all possible labelings of those fixed points.

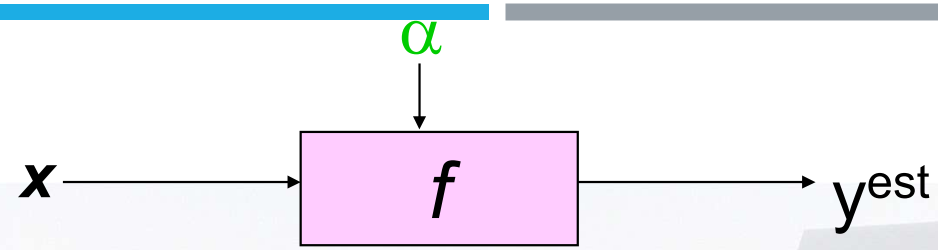
EXAMPLES



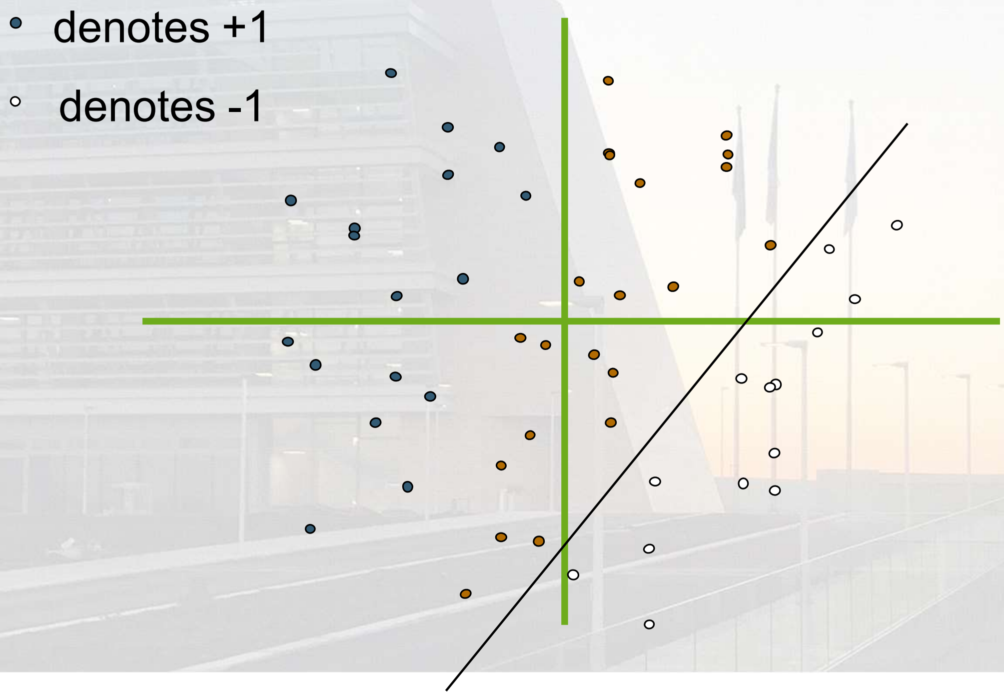
$f(x,w) = sign(x.w) = \sum_{I=1}^{n} w_i x_i$



EXAMPLES

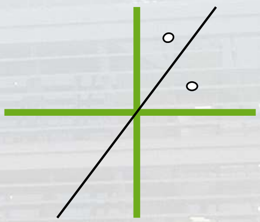


$f(x,w,b) = sign(x.w+b)$



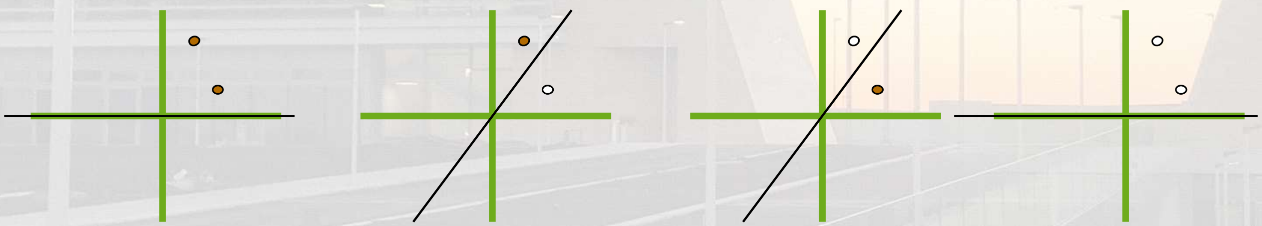
SHATTERING

- Question: Can the following *f* shatter the following points?



f(x,w) = sign(x.w)

Answer: Yes. There are four possible training set types to consider:



w=(0,1) w=(-2,3) w=(2,-3) w=(0,-1)

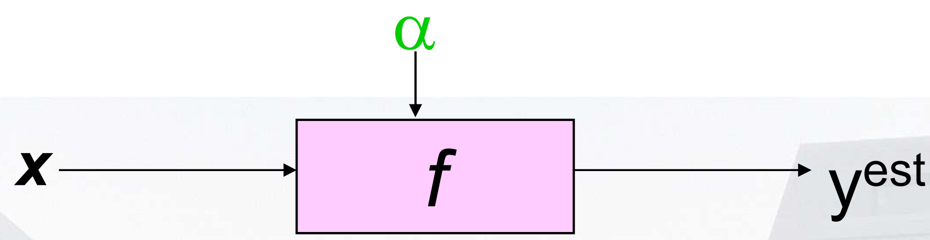
VC DIM OF LINEAR CLASSIFIERS IN M-DIMENSIONS

If input space is *m-dimensional* and if f is *sign(w.x-b)*, what is the VC-dimension?

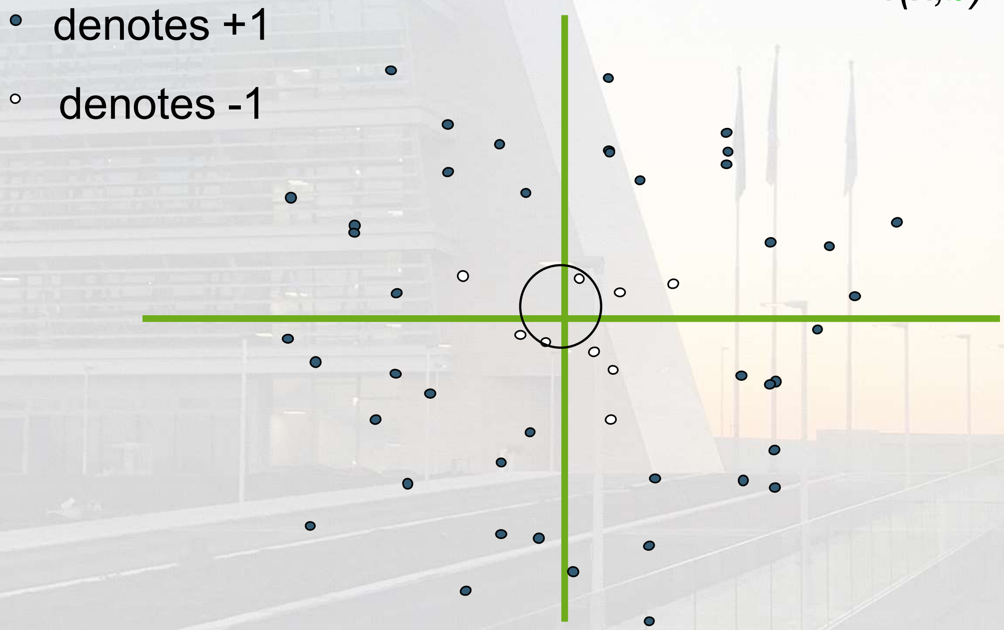
h=m+1

- Lines in 2D can shatter 3 points
- Planes in 3D space can shatter 4 points
- …

EXAMPLES



$f(x, b) = sign(x.x - b)$

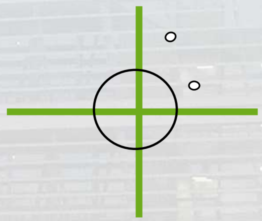


Diapositiva 25

rb1 roberto basili; 20/03/2023

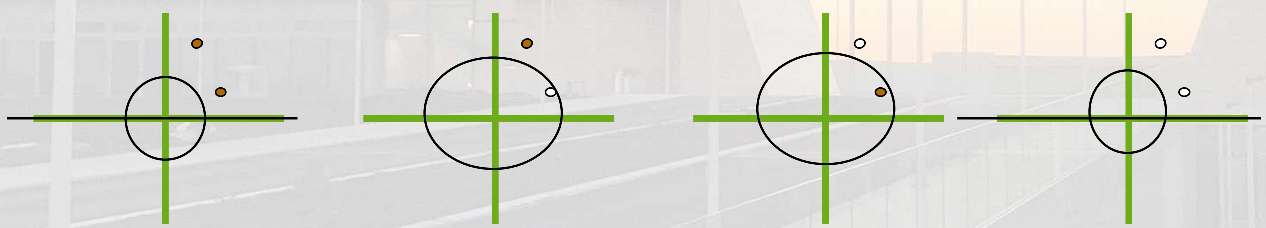
SHATTERING

- Question: Can the following *f* shatter the following points?



$$f(x,b) = sign(x.x-b)$$

Answer: Yes. Hence, the VC dimension of circles on the origin is at least 2.



MODEL SELECTİON & GENERALİZATİON

- Learning is an ill-posed problem; data is not sufficient to find a unique solution
- The need for inductive bias, assumptions about $\mathcal{H}$
- Generalization: How well a model performs on new data
- Different machines have different amounts of “power”.

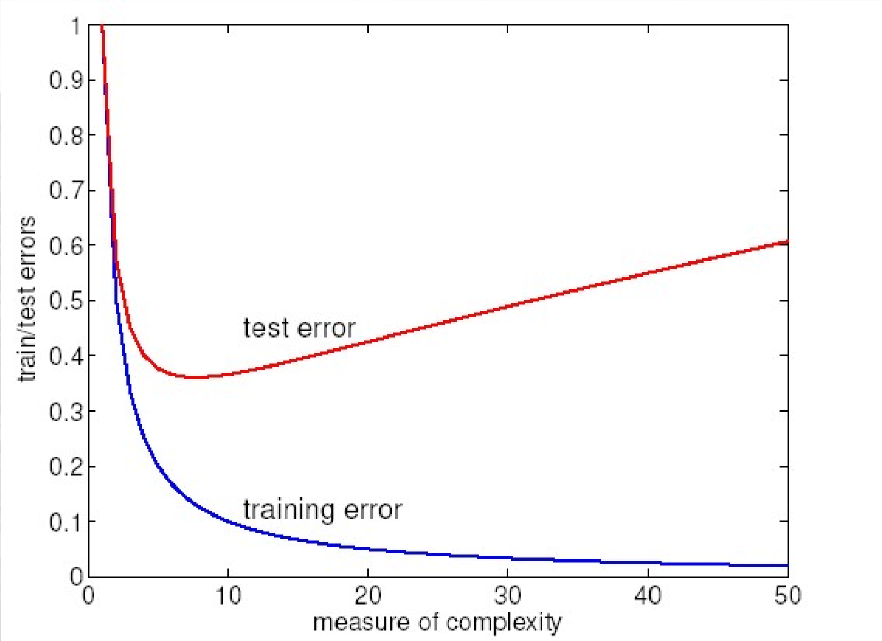
  Tradeoff between:
  - More power: Can model more complex classifiers but might overfit.
  - Less power: Not going to overfit, but restricted in what it can model.
  - Overfitting: $\mathcal{H}$ more complex than C or f
  - Underfitting: $\mathcal{H}$ less complex than C or f

TRIPLE TRADE-OFF

- There is a trade-off between three factors (Dietterich, 2003):
  1. Complexity of $\mathcal{H}$, $c(\mathcal{H})$,
  2. Training set size, N,
  3. Generalization error, E, on new data

- As $N\uparrow$, $E\downarrow$
- As $c(\mathcal{H})\uparrow$, first $E\downarrow$ and then $E\uparrow$

WHY CARE ABOUT COMPLEXITY?



- A quantitative measure of complexity is useful to determine the relationship between the training error (that we can observe during training) and the test error (which we want to minimize)

COMPLEXİTY

- “Complexity” is a measure of a family of classifiers, not of any specific (fixed) classifier

- There are many possible measures for complexity
  - degrees of freedom (e.g. number of parameters in polinomials)
  - description length
  - Vapnik-Chervonenkis (VC) dimension
  - etc.

EXPECTED AND EMPİRİCAL ERROR

$$\hat{\mathcal{E}}_n(i) = \frac{1}{n}\sum_{t=1}^{n} \overbrace{\text{Loss}(y_t, h_i(\mathbf{x}_t))}^{=0,1} = \text{empirical error of } h_i(\mathbf{x})$$

$$\mathcal{E}(i) = E_{(\mathbf{x},y)\sim P}\left\{\text{Loss}(y, h_i(\mathbf{x}))\right\} = \text{expected error of } h_i(\mathbf{x})$$

LEARNING AND THE VC DIMENSION

- Let d_{VC} be the VC-dimension of our set of classifiers F.

  Theorem: With probability at least $1-\delta$ over the choice of the training set, for all $h \in F$

  $$\mathcal{E}(h) \leq \hat{\mathcal{E}}_n(h) + \epsilon(n, d_{VC}, \delta)$$

  where

  $$\epsilon(n, d_{VC}, \delta) = \sqrt{\frac{d_{VC}(\log(2n/d_{VC}) + 1) + \log(1/(4\delta))}{n}}$$

MODEL SELECTION

- We try to find the model with the best balance of complexity and the fit to the training data
- Ideally, we would select a model from a nested sequence of models of increasing complexity (VC-dimension)

  Model 1 d_1
  Model 2 d_2
  Model 3 d_3

  where $d_1 \leq d_2 \leq d_3 \leq \ldots$
- The model selection criterion is: find the model class that achieves the lowest upper *bound* on the expected loss

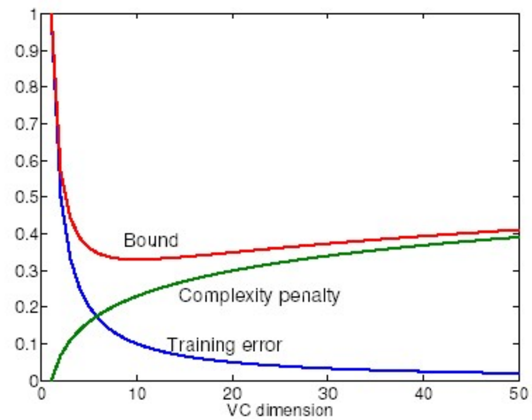
  Expected error $\leq$ Training error $+$ Complexity penalty

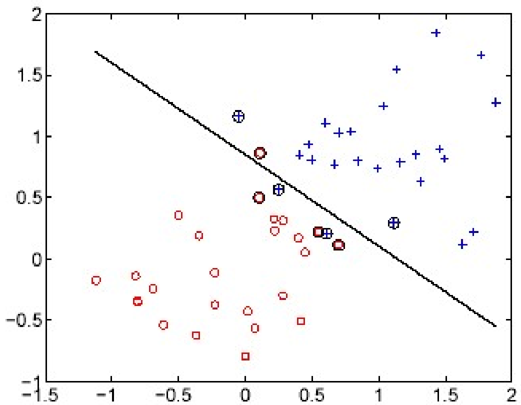
VC DIMENSION AND STRUCTURAL RISK MINIMIZATION

- We choose the model class F_i that minimizes the upper bound on the expected error:

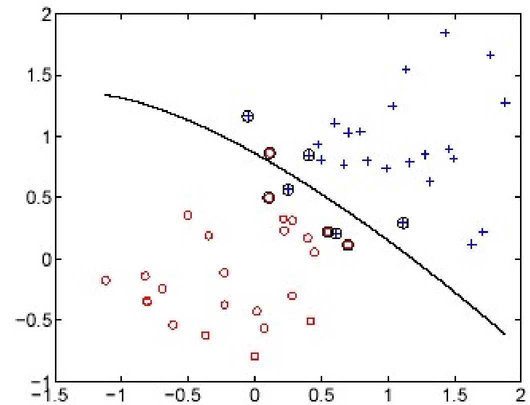
$$\mathcal{E}(\hat{h}_i) \leq \hat{\mathcal{E}}_n(\hat{h}_i) + \sqrt{\frac{d_i(\log(2n/d_i) + 1) + \log(1/(4\delta))}{n}}$$

where $\hat{h}_i$ is the best classifier from F_i selected on the basis of the training set.

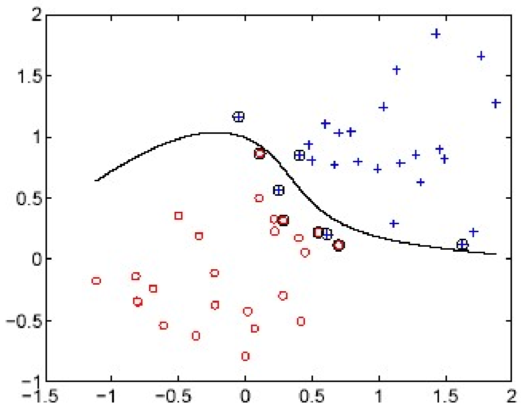




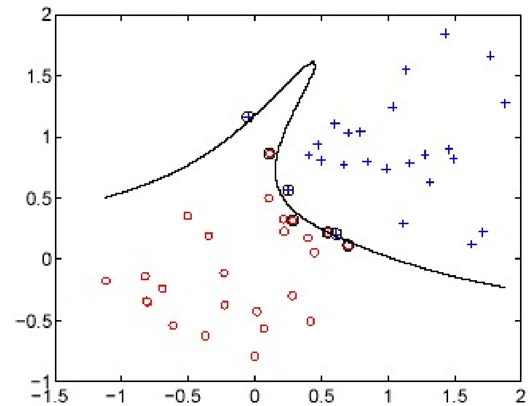
linear



2^{nd} order polynomial



4^{th} order polynomial



8^{th} order polynomial

STRUCTURAL RISK MINIMIZATION

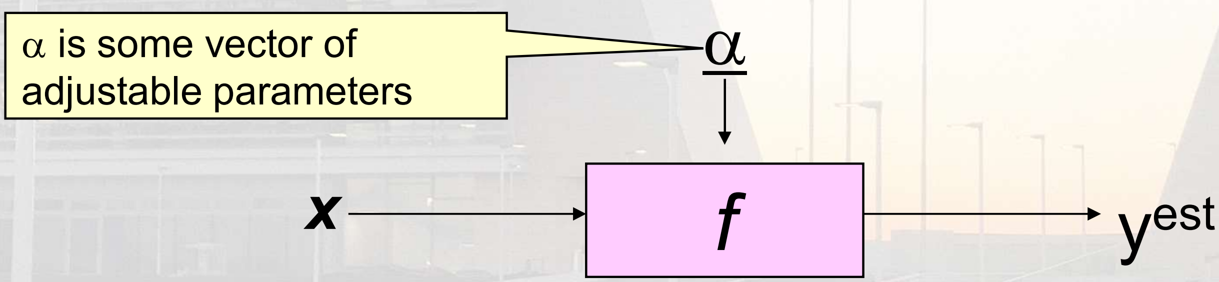
- Number of training examples $n = 50$, confidence parameter $\delta = 0.05$.

Model	d_{VC}	Empirical fit	$\epsilon(n, d_{VC}, \delta)$
1^{st} order	3	0.06	0.5501
2^{nd} order	6	0.06	0.6999
4^{th} order	15	0.04	0.9494
8^{th} order	45	0.02	1.2849

- Structural risk minimization would select the simplest (linear) model in this case.

SUMMARY: A LEARNING MACHINE

- A learning machine f takes an input x and transforms it, somehow using factors (as weights) $\underline{\alpha}$, into a predicted output y^{est} = +/- 1



VC-DIMENSION AS MEASURE OF COMPLEXİTY

$$\text{TESTERR}(\vec{\alpha}) \leq \text{TRAINERR}(\vec{\alpha}) + \sqrt{\frac{d_{VC}(\log(2n/d_{VC}) + 1) - \log(1/(4\delta))}{n}}$$

i	f_i	TRAINERR	VC-Conf	Probable upper bound on TESTERR	Choice
1	f_1				
2	f_2				
3	f_3				⍰
4	f_4				
5	f_5				
6	f_6				

USING VC-DIMENSIONALITY

- People have worked hard to find VC-dimension for …
  - Decision Trees
  - Perceptrons
  - Neural Nets
  - Decision Lists
  - Support Vector Machines
  - …and many many more

- All with the goals of
  - Understanding which learning machines are more or less powerful under which circumstances
  - Using Structural Risk Minimization for to choose the best learning machine

ALTERNATIVES TO VC-DIM-BASED MODEL SELECTION

Cross Validation

- To estimate generalization error, we need data unseen during training. We split the data as:
  - Training set (50%)   M1 M2   train(M2) < train(M1)
  - Validation set (25%) test(M1, Vs) = P1   test(M2, VS) = P2   P2>P1
  - Test (publication) set (25%)

- Resampling when there is few data
  - N-fold cross-validation: N-2 fold for training, 1 fold as validation set and 1 fold for testing (N*(N-1) tests)

ALTERNATIVES TO VC-DIM-BASED MODEL SELECTION

- What could we do instead of the scheme below?
  - Cross-validation

i	f_i	TRAINERR	10-FOLD-CV-ERR	Choice
1	f_1			
2	f_2			
3	f_3			(selected)
4	f_4			
5	f_5			
6	f_6			

EXTRA COMMENTS

- Further Readings:
  - An excellent tutorial on VC-dimension and Support Vector Machines: C.J.C. Burges. *A tutorial on support vector machines for pattern recognition*. Data Mining and Knowledge Discovery, 2(2):955-974, 1998.

    M. J. Kearns, U. V. Vazirani, *Introduction to Computational Learning Theory*, MIT Press, 1994 (Chapter 1).

WHAT YOU SHOULD KNOW

- Definition of PAC learning
- The definition of a learning machine: $f(x,\alpha)$
- The definition of Shattering
- Be able to work through simple examples of shattering
- The definition of VC-dimension
- Be able to work through simple examples of VC-dimension
- Structural Risk Minimization for model selection
- Awareness of other model selection methods