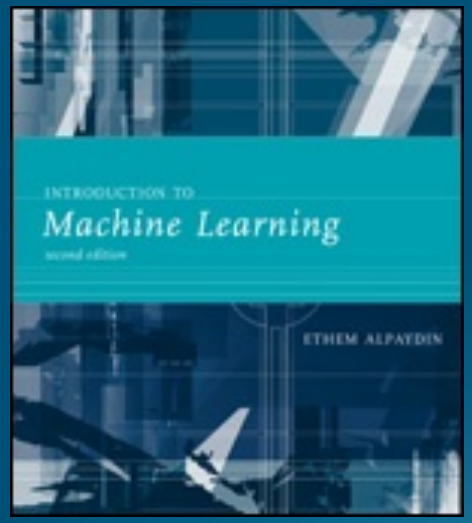


Lecture Slides for

INTRODUCTION TO

Machine Learning

ETHEM ALPAYDIN

alpaydin@boun.edu.tr
http://www.cmpe.boun.edu.tr/~ethem/i2ml2e

In preparation of these slides, I have benefited from slides prepared by:

E. Alpaydin (Intro. to Machine Learning),

D. Bouchaffra and V. Murino (Pattern Classification and Scene Analysis),

R. Gutierrez-Osuna (Texas A&M)

A. Moore (CMU)

CHAPTER 2:

Supervised Learning

Learning 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:

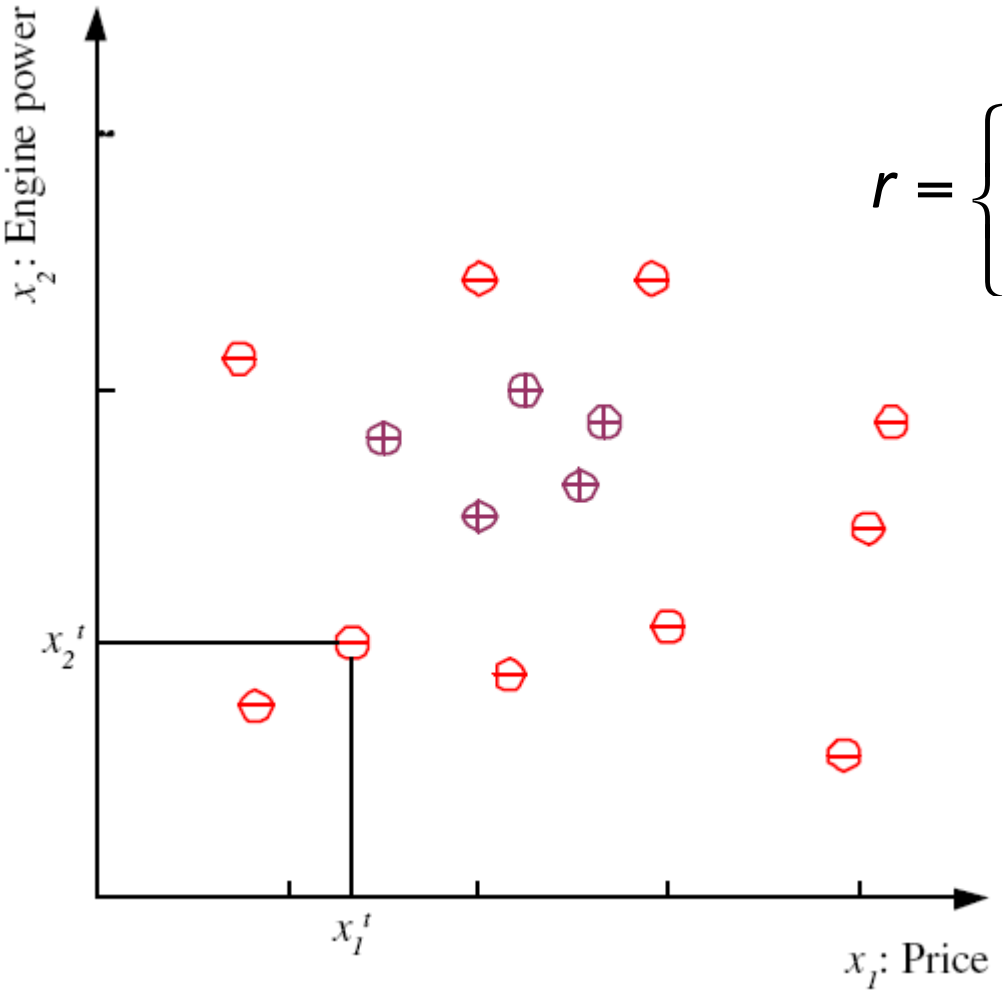
  x_1: price, x_2 : engine power

Training set $\mathcal{X}$

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

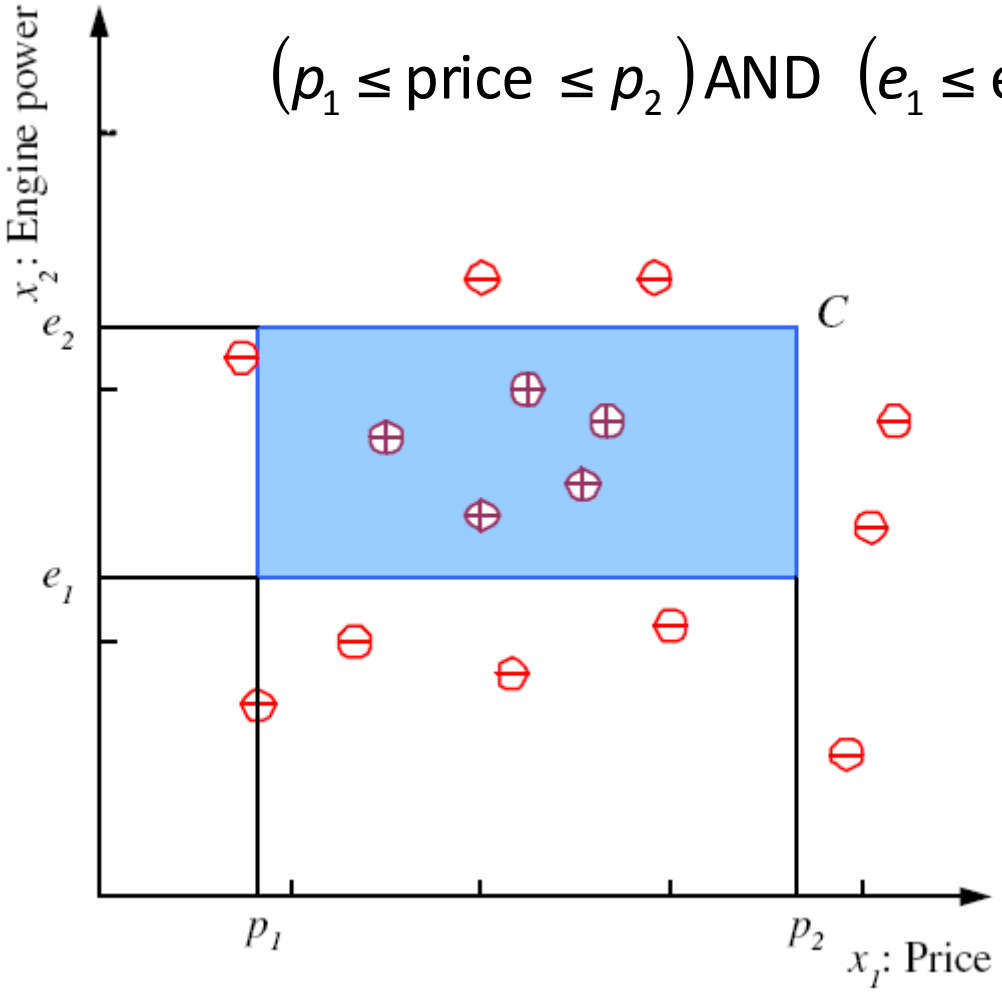
$$r = \begin{cases} 1 \text{ if } \mathbf{x} \text{ is positive} \\ 0 \text{ if } \mathbf{x} \text{ is negative} \end{cases}$$

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

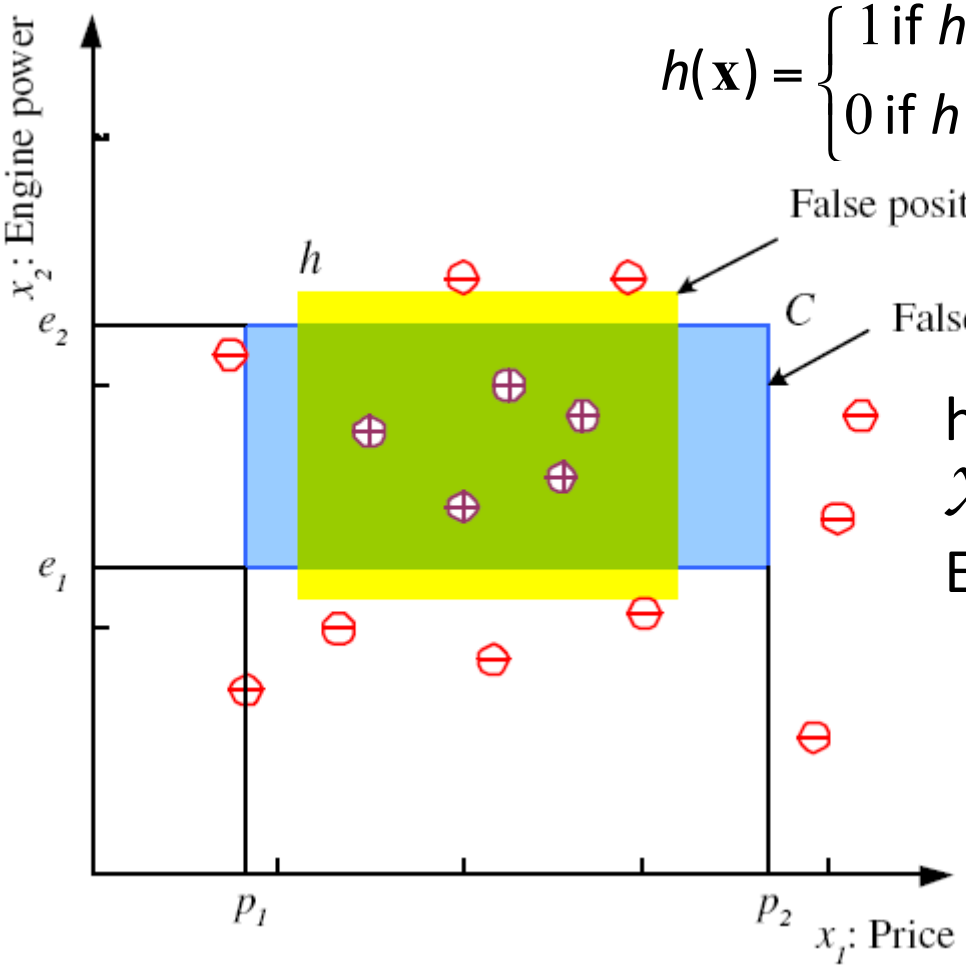


Class C

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



Hypothesis class $\mathcal{H}$



$$h(\mathbf{x}) = \begin{cases} 1 \text{ if } h \text{ says } \mathbf{x} \text{ is positive} \\ 0 \text{ if } h \text{ says } \mathbf{x} \text{ is negative} \end{cases}$$

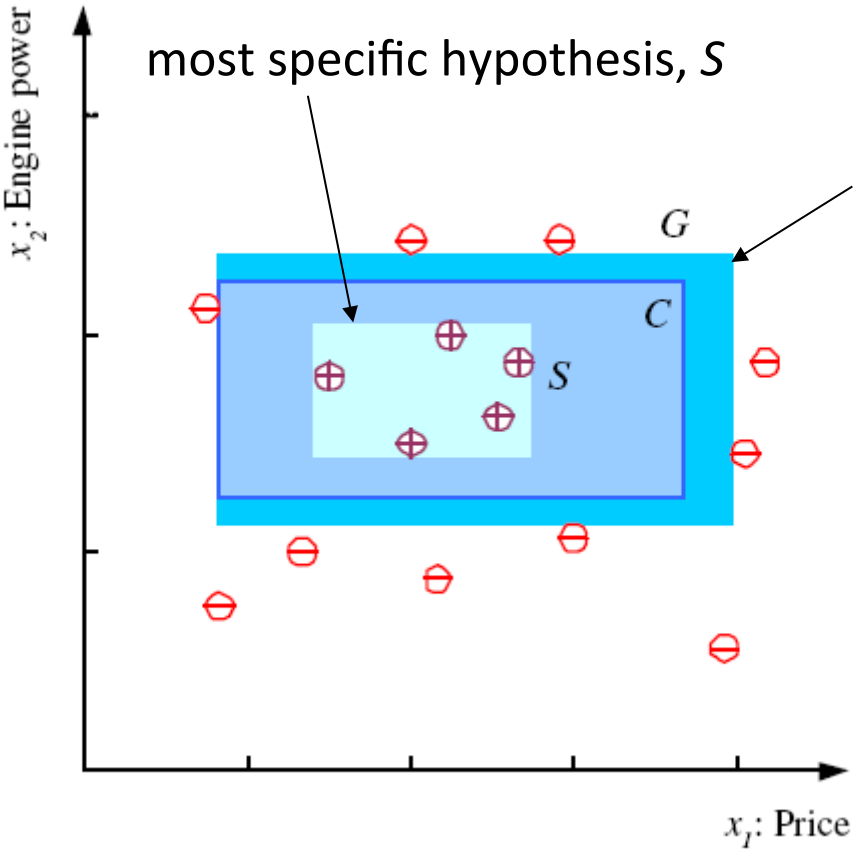
h: a hypothesis from $\mathcal{H}$

$\mathcal{X}$: training set

Error of h on $\mathcal{X}$

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

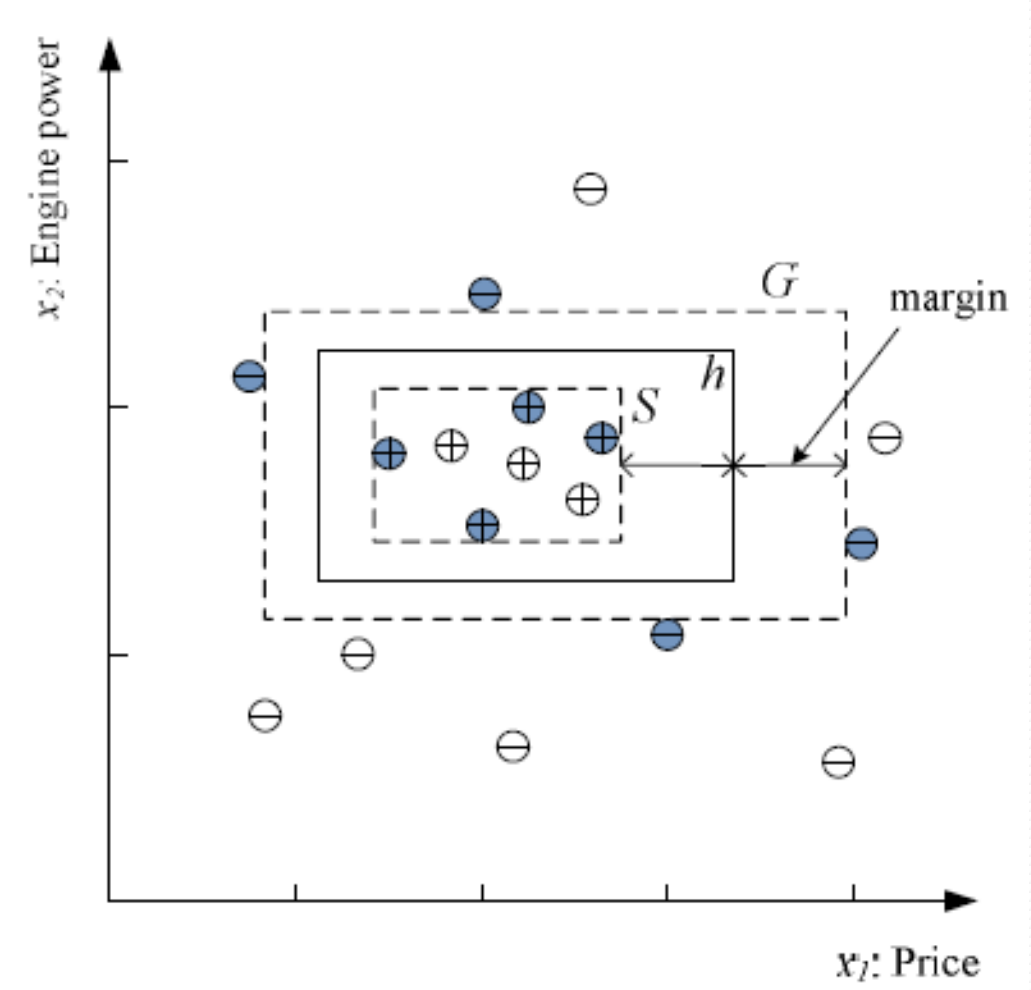
S, G, and the Version Space



$h \in$ H, between S and G is Consistent (error free) and make up the version space (Mitchell, 1997)

Margin

- Choose *h* with largest margin



VC Dimension and PAC Learning

Tools to analyze expected (test/generalization) error of hypothesis classes (i.e. classifiers)

- VC (Vapnik Chervonenkis) Dimension:
  - Given a hypothesis class (rectangles, circles, lines, neural networks) how many instances can it classify without any error.
  - Does not consider the input distribution

- PAC (Probably Approximately Correct) Learning
  - How many instances are needed to achieve a certain error with a certain probability?

VC Dimension

- N points can be labeled in 2^N ways as +/–

N=3 for this table

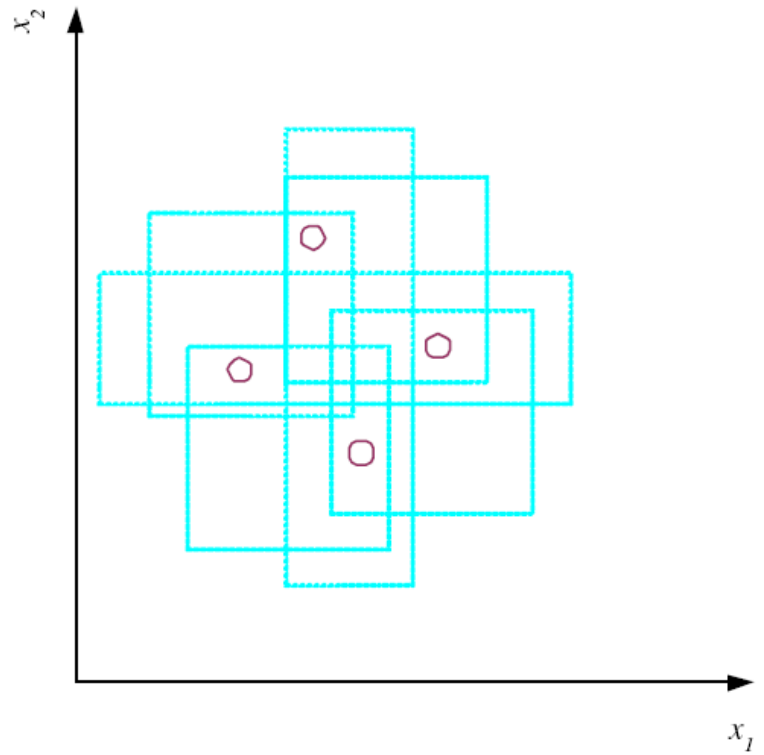
	R1	R2	R3	R4	R5	R6	R7	R8
x1	0	0	0	1	0	1	1	1
x2	0	0	1	0	1	0	1	1
x3	0	1	0	0	1	1	0	1

$\mathcal{H}$ shatters N if

there exists $h \in \mathcal{H}$ consistent

for any of these:

VC($\mathcal{H}$) = N

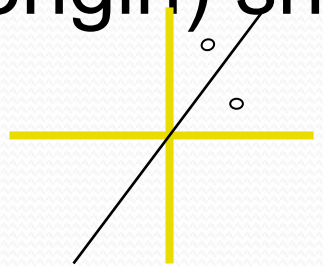


An axis-aligned rectangle shatters 4 points only !
Choose points in such a way that they can be shattered.
e.g. Do not put all of them at the same spot.

VC Dimension Examples

We use g to denote hypothesis (like h)

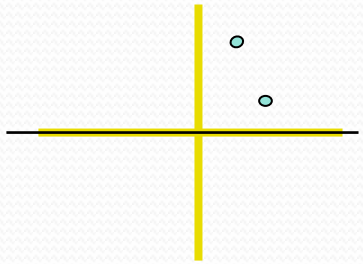
Question: Can the following g (line passing through origin) shatter the following points?



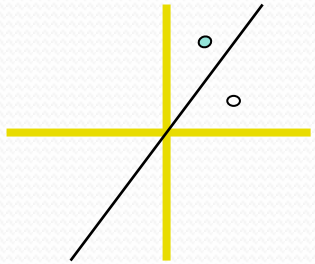
$g(x,\theta) = \text{sign}(x.\ \theta)=\text{sign}(x_1\theta_1+x_2\theta_2)$

Answer: No problem. There are four training sets to consider

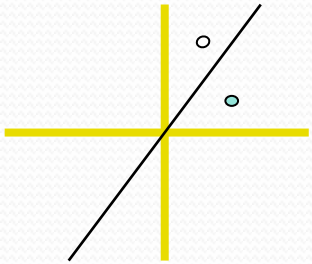
○ +



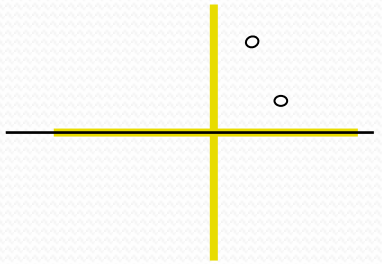
$\theta=(0,1)$



$\theta=(-2,3)$



$\theta=(2,-3)$



$\theta=(0,-1)$

by Andrew Moore

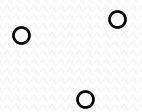
VC dim of line machine

Given machine g, the VC-dimension v is

The maximum number of points that can be arranged so that g shatter them.

Example: For 2-d inputs, what's VC-dim of g(x,θ,b) = sign(θ.x+b)?

Well, can g shatter these three points?



Yes, of course.

All -ve or all +ve is trivial

One +ve can be picked off by a line

One -ve can be picked off too.

by Andrew Moore

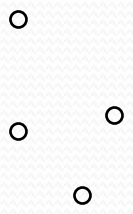
VC dim of line machine

Given machine ***g***, the VC-dimension *v* is

The maximum number of points that can be arranged so that ***g*** shatter them.

Example: For 2-d inputs, what's VC-dim of g(x,θ,b) = sign(θ.x+b)?

Well, can we find four points that ***g*** can shatter?



by Andrew Moore

VC dim of line machine

Given machine ***g***, the VC-dimension *v* is

The maximum number of points that can be arranged so that ***g*** shatter them.

Example: For 2-d inputs, what's VC-dim of $g(x,\theta,b) = \text{sign}(\theta.x+b)$?

Well, can g shatter these four points?

Can always draw six lines between pairs of four points.

by Andrew Moore

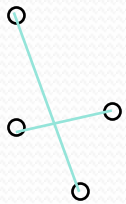
VC dim of line machine

Given machine ***g***, the VC-dimension *v* is

The maximum number of points that can be arranged so that ***g*** shatter them.

Example: For 2-d inputs, what's VC-dim of $g(x,\theta,b) = \text{sign}(\theta.x+b)$?

Well, can g shatter these four points?



Can always draw six lines between pairs of four points.

Two of those lines will cross.

by Andrew Moore

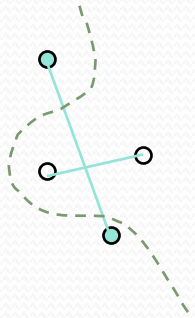
VC dim of line machine

Given machine ***g***, the VC-dimension *v* is

The maximum number of points that can be arranged so that ***g*** shatter them.

Example: For 2-d inputs, what's VC-dim of $g(x, \theta, b) = \text{sign}(\theta.x+b)$?

Well, can we find four points that ***g*** can shatter?



Can always draw six lines between pairs of four points.

Two of those lines will cross.

If we put points linked by the crossing lines in the same class they can't be linearly separated

So a line can shatter 3 points but not 4

So VC-dim of Line Machine is 3

If inputs are m dimensional, v=m+1(see A. Moore notes).

by Andrew Moore

Why VC Dimension

$$R^{emp}(\theta) = \text{TRAINERR}(\theta) = \text{Fraction Training Set misclassified}$$

$$R(\theta) = \text{TESTERR}(\theta) = \text{Probability of Misclassification}$$

- Given some machine **g**, let *v* be its VC dimension.
- *v* is a measure of **g**'s power (*v* does not depend on the choice of training set)
- Vapnik showed that with probability 1-η

$$\text{TESTERR}(\theta) \le \text{TRAINERR}(\theta) + \sqrt{\frac{v(\log(2N/v)+1) - \log(\eta/4)}{N}}$$

This bound is usually too big, so it may not be useful.

by Andrew Moore

$$\text{TESTERR}(\theta) \le \text{TRAINERR}(\theta) + \sqrt{\frac{v(\log(2N/v)+1) - \log(\eta/4)}{N}}$$

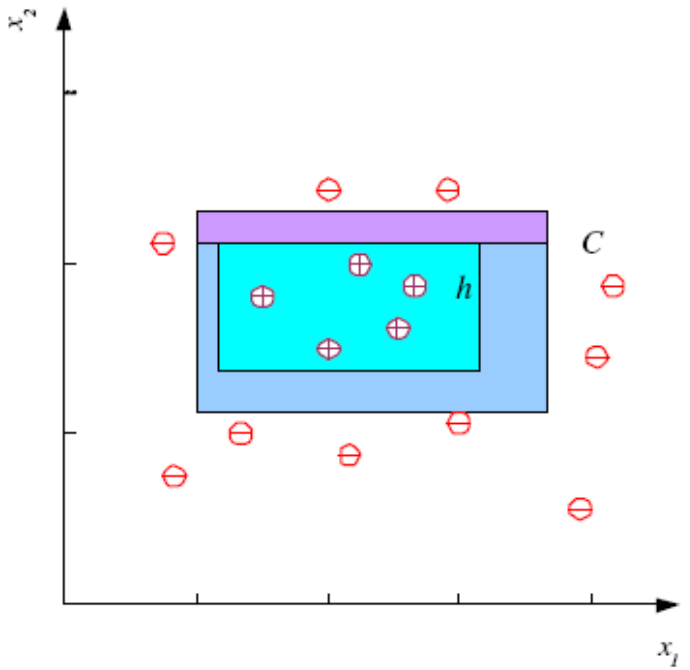
i	f_i	TRAINERR	VC-Conf	Probable upper bound on TESTERR	Choice
1	f_1				
2	f_2				
3	f_3				?
4	f_4				
5	f_5				
6	f_6				

Probably Approximately Correct (PAC) Learning

- How many training examples N should we have, such that with probability at least $1 - \delta$, h has error at most ε ?

  (Blumer et al., 1989)

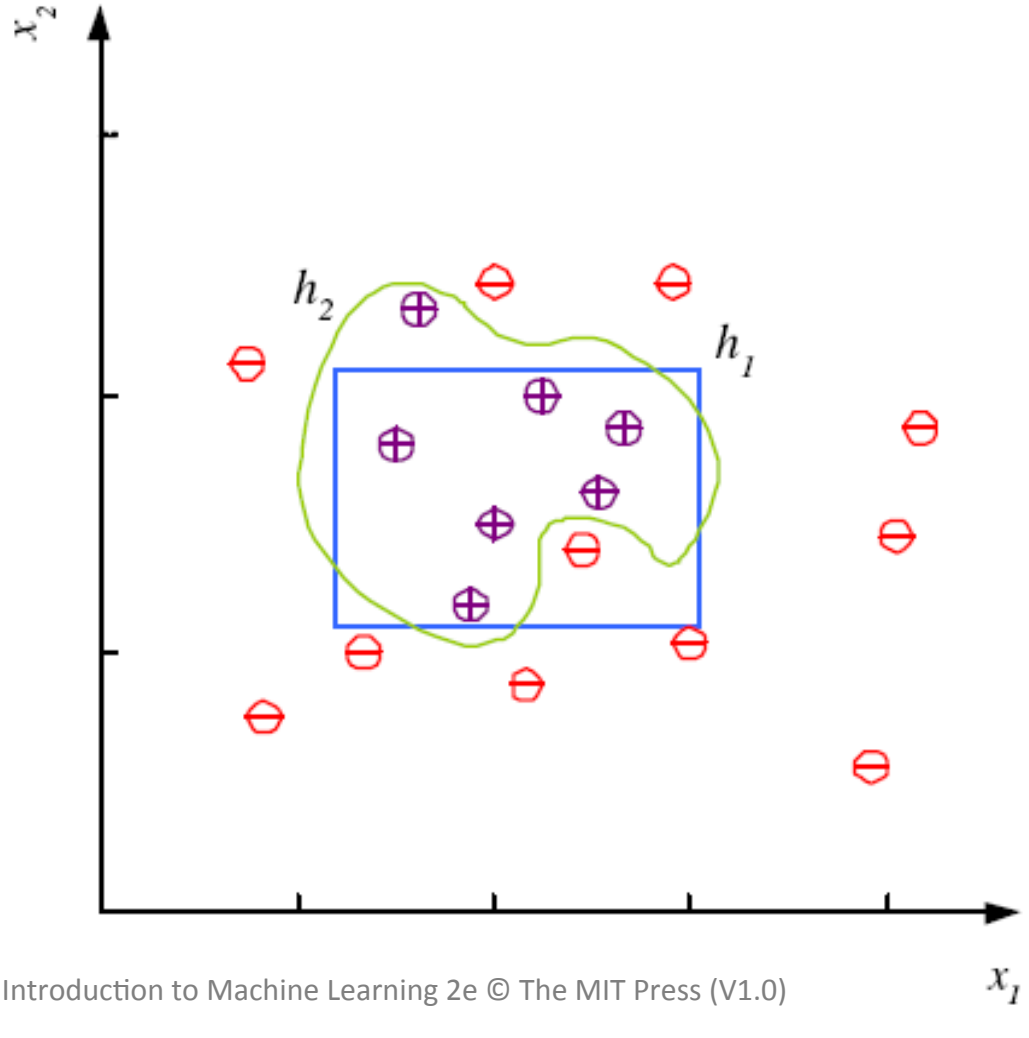
- Each strip is at most $\varepsilon/4$
- Pr that we miss a strip $1 - \varepsilon/4$
- Pr that N instances miss a strip $(1 - \varepsilon/4)^N$
- Pr that N instances miss 4 strips $4(1 - \varepsilon/4)^N$
- $4(1 - \varepsilon/4)^N \leq \delta$ and $(1 - x) \leq \exp(-x)$
- $4\exp(-\varepsilon N/4) \leq \delta$ and $N \geq (4/\varepsilon)\log(4/\delta)$



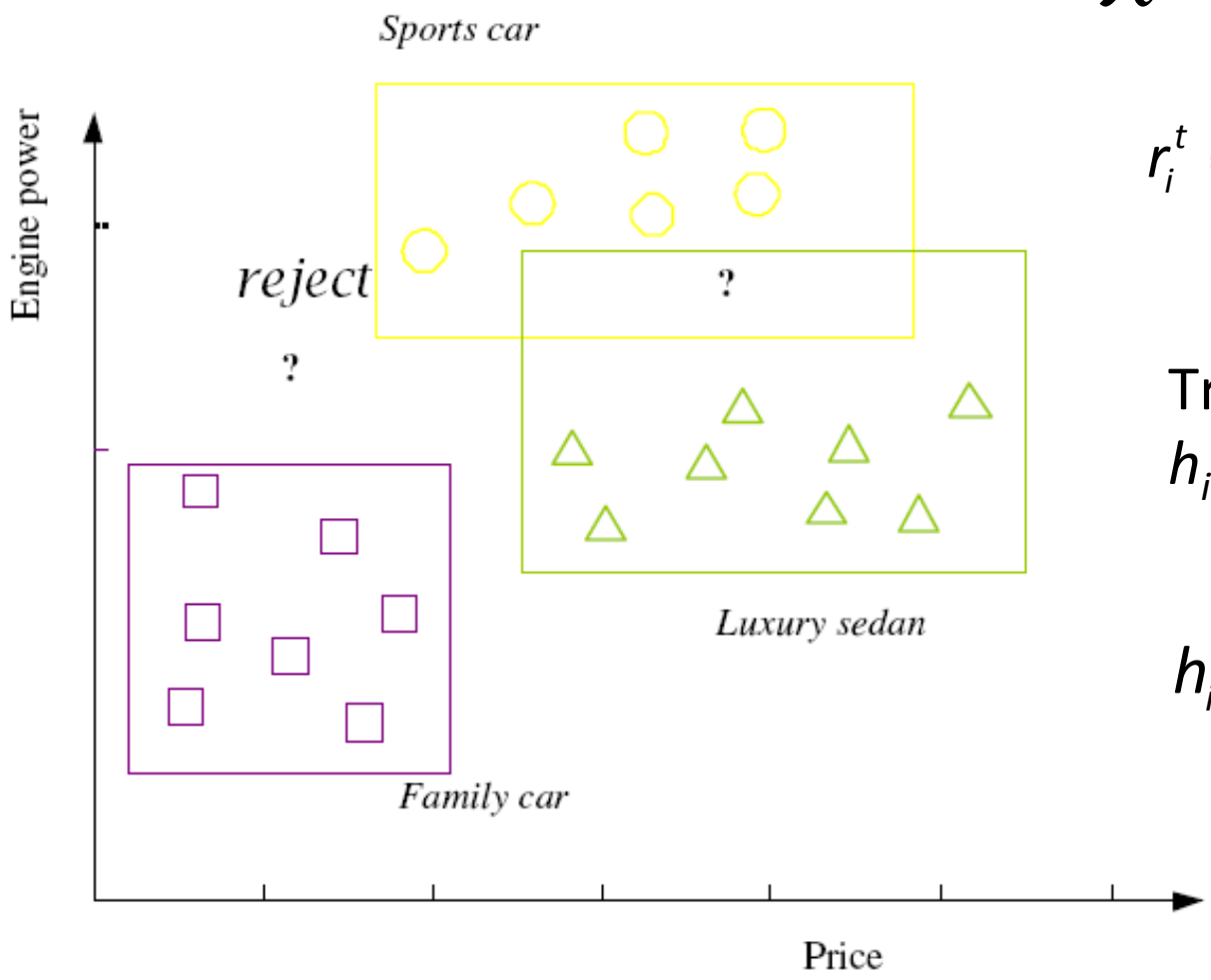
Noise and Model Complexity

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



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

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

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

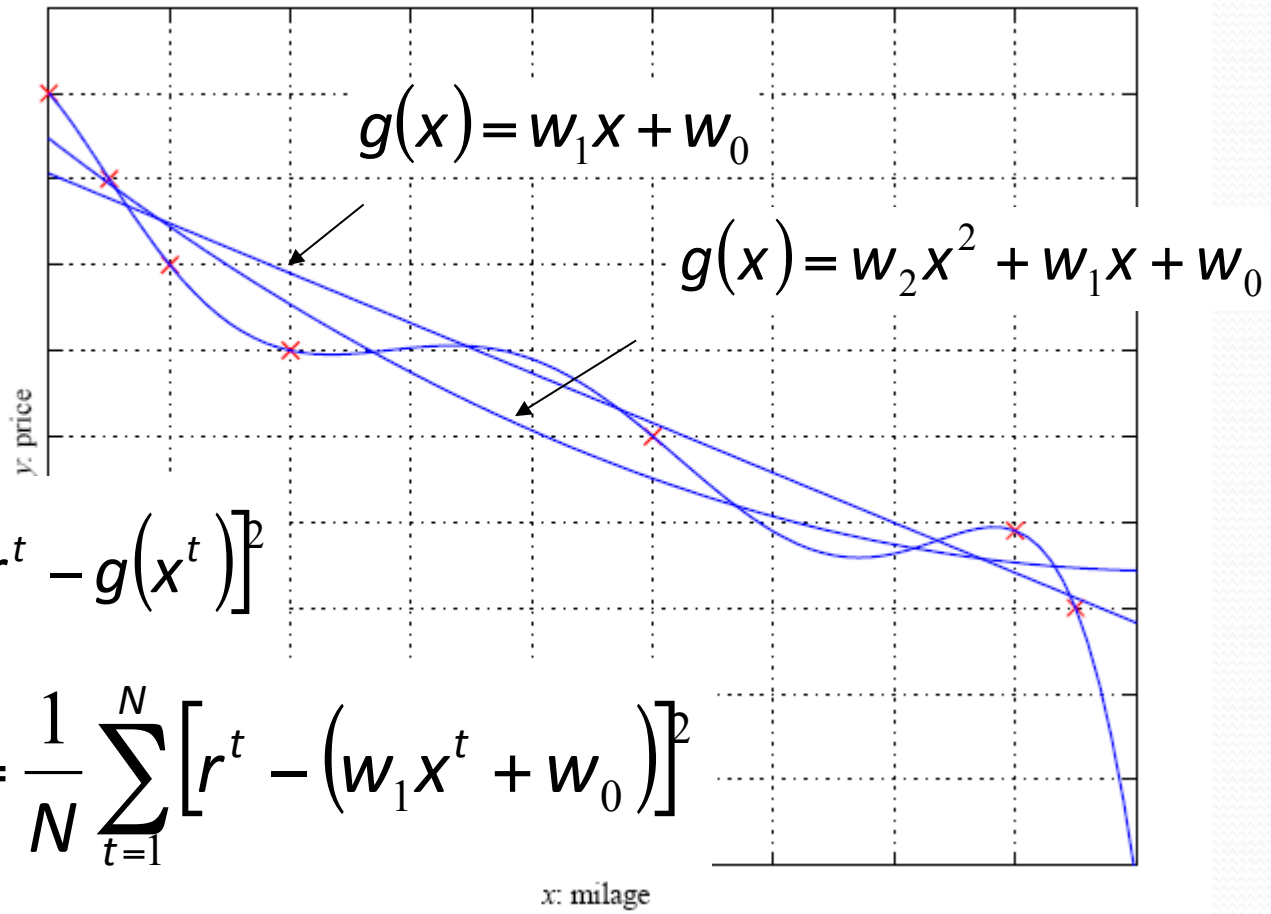
$$h_i\left(\mathbf{x}^t\right) = \begin{cases} 1 \text{ if } \mathbf{x}^t \in C_i \\ 0 \text{ if } \mathbf{x}^t \in C_j, j \neq i \end{cases}$$

Regression

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

$$r^t \in \Re$$

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



$$E(g \mid \mathcal{X}) = \frac{1}{N}\sum_{t=1}^{N}\left[r^t - g\left(x^t\right)\right]^2$$

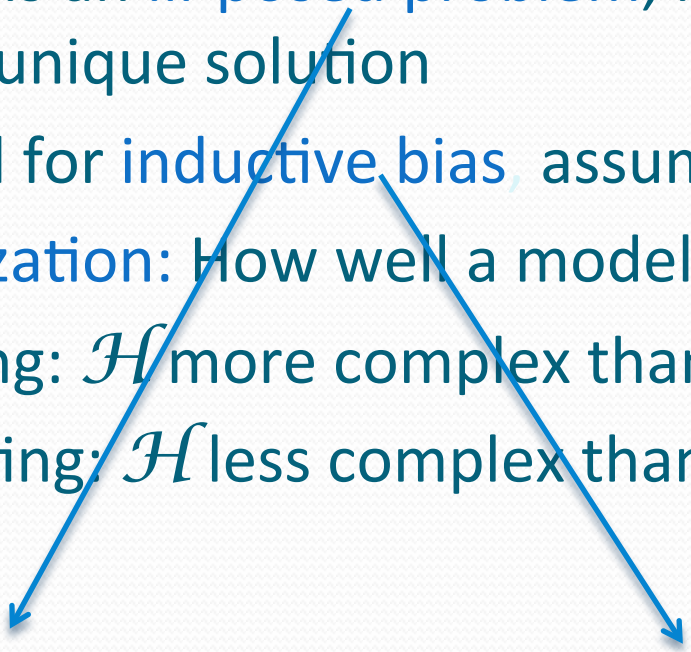
$$E(w_1, w_0 \mid \mathcal{X}) = \frac{1}{N}\sum_{t=1}^{N}\left[r^t - \left(w_1 x^t + w_0\right)\right]^2$$

Model Selection & Generalization

- Learning is an ill-posed problem, i.e. 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
- Overfitting: $\mathcal{H}$ more complex than C or f
- Underfitting: $\mathcal{H}$ less complex than C or f

Model Selection & Generalization

- Learning is an ill-posed problem, i.e. 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
- Overfitting: $\mathcal{H}$ more complex than C or f
- Underfitting: $\mathcal{H}$ less complex than C or f



Data not sufficient to find a unique solution

Set of assumptions we make to make learning possible.

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 $E\uparrow$

Cross-Validation

- To estimate generalization error, we need data unseen during training. We split the data as
  - Training set (50%)
  - Validation set (25%)
  - Test (publication) set (25%)
- Resampling (e.g. bootstrapping) when there is few data

Cross Validation

h more complex

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

Dimensions of a Supervised Learner

1. Model: $g(\mathbf{x}|\theta)$

2. Loss function: $$E(\theta|\mathcal{X}) = \sum_t L(r^t, g(\mathbf{x}^t|\theta))$$

3. Optimization procedure:

$$\theta^* = \arg\min_\theta E(\theta|\mathcal{X})$$