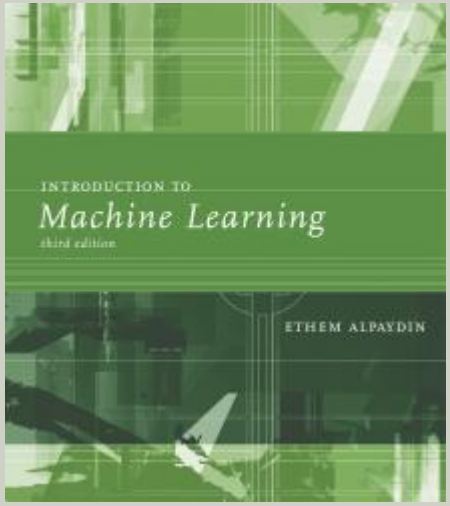


Lecture Slides for

INTRODUCTION TO MACHINE LEARNING

3RD EDITION

ETHEM ALPAYDIN

alpaydin@boun.edu.tr
http://www.cmpe.boun.edu.tr/~ethem/i2ml3e

CHAPTER 2:

SUPERVISED LEARNING

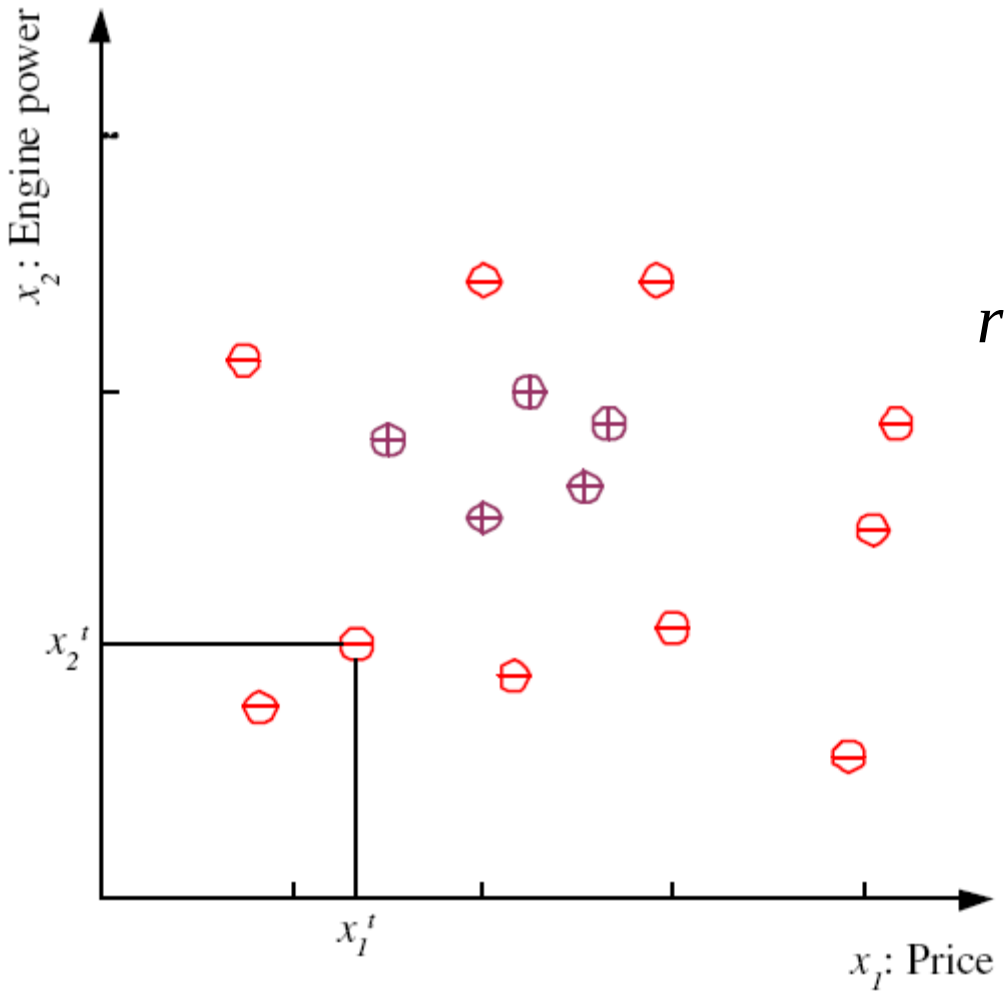
Learning a Class from Examples

- 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 X



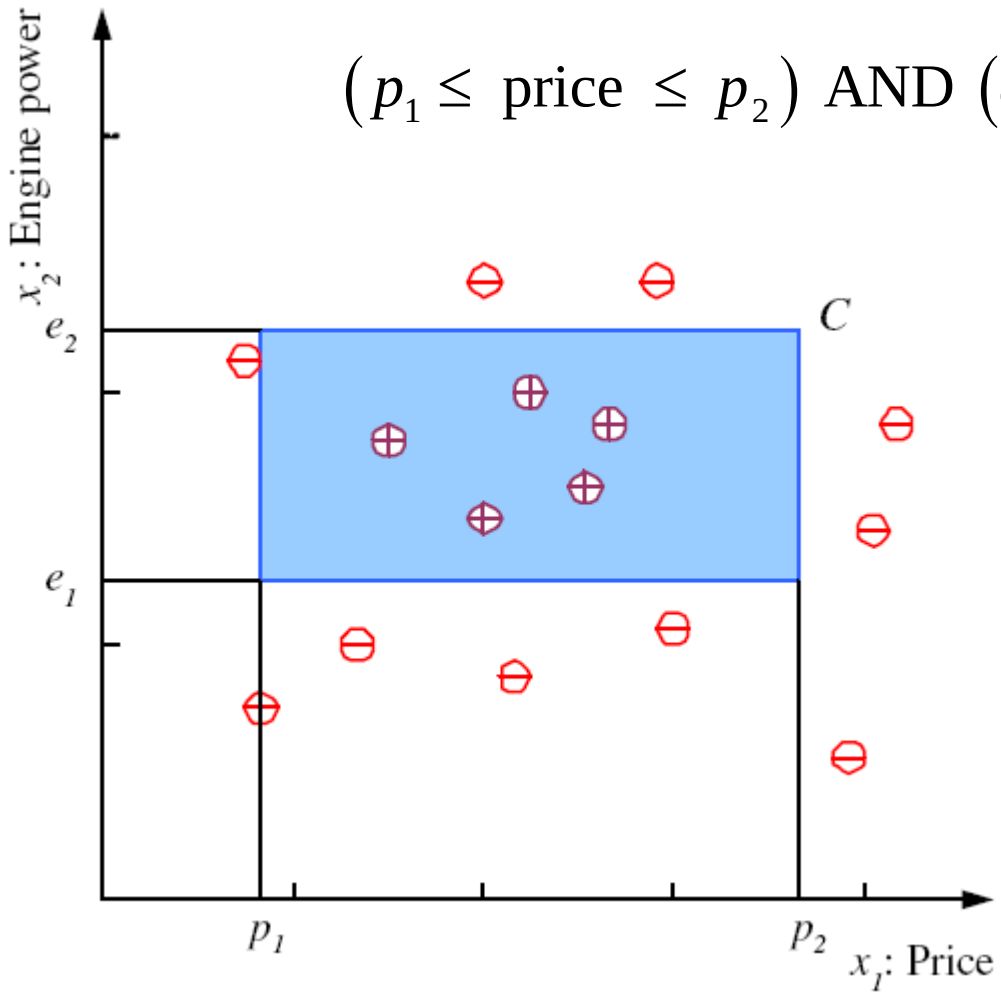
$$\mathcal{X} = \{\mathbf{x}^t, r^t\}_{t=1}^N$$

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

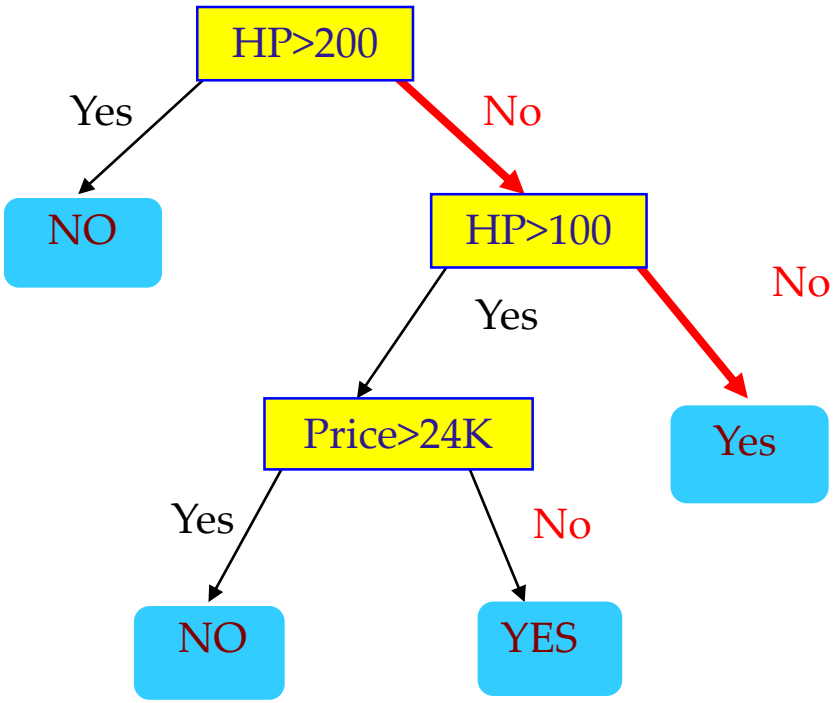
$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Class C

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



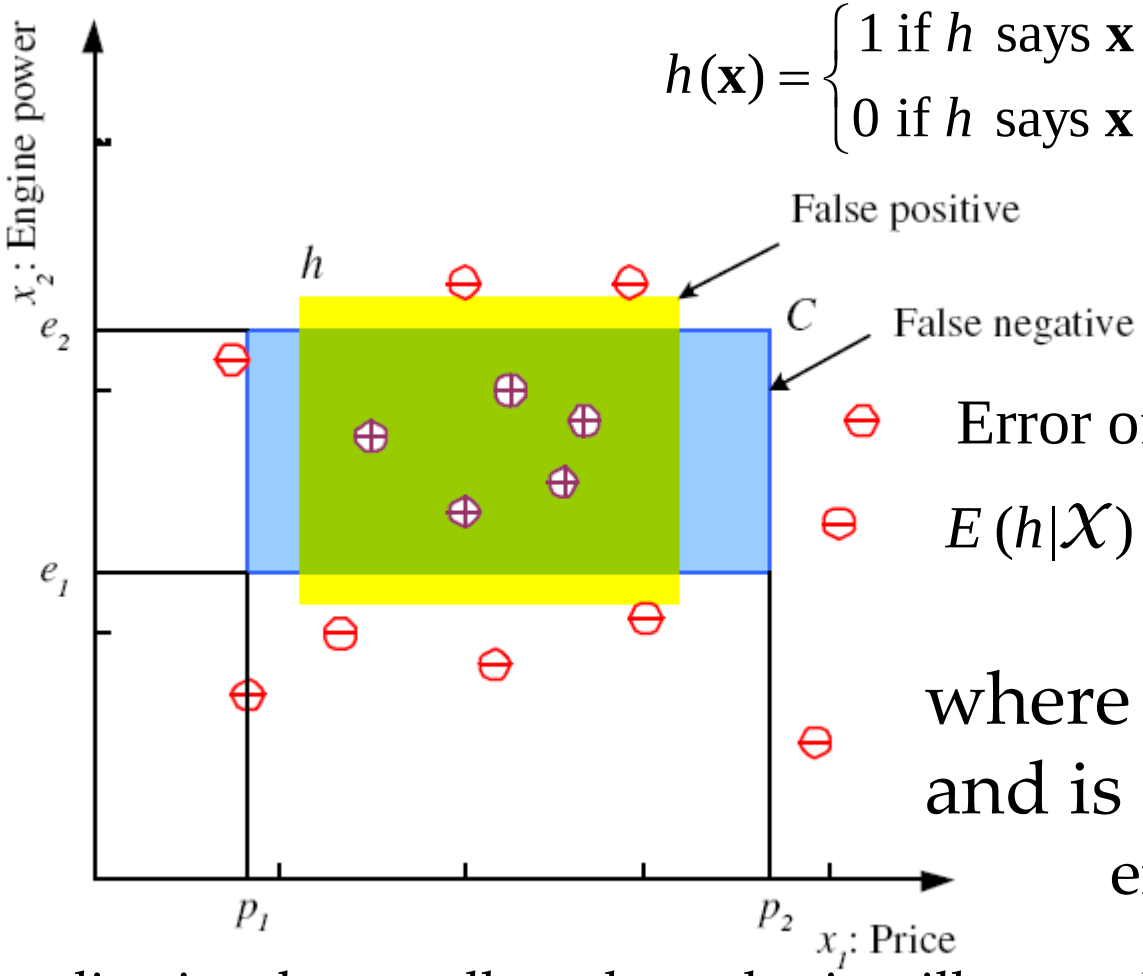
Family Car Decision Tree



Hypothesis class H

H, the hypothesis class (the set of rectangles) from which we believe *C* is drawn

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



Error of h on H

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

where $1(a \neq b)$ is 1 if $a \neq b$ and is 0 if $a = b$

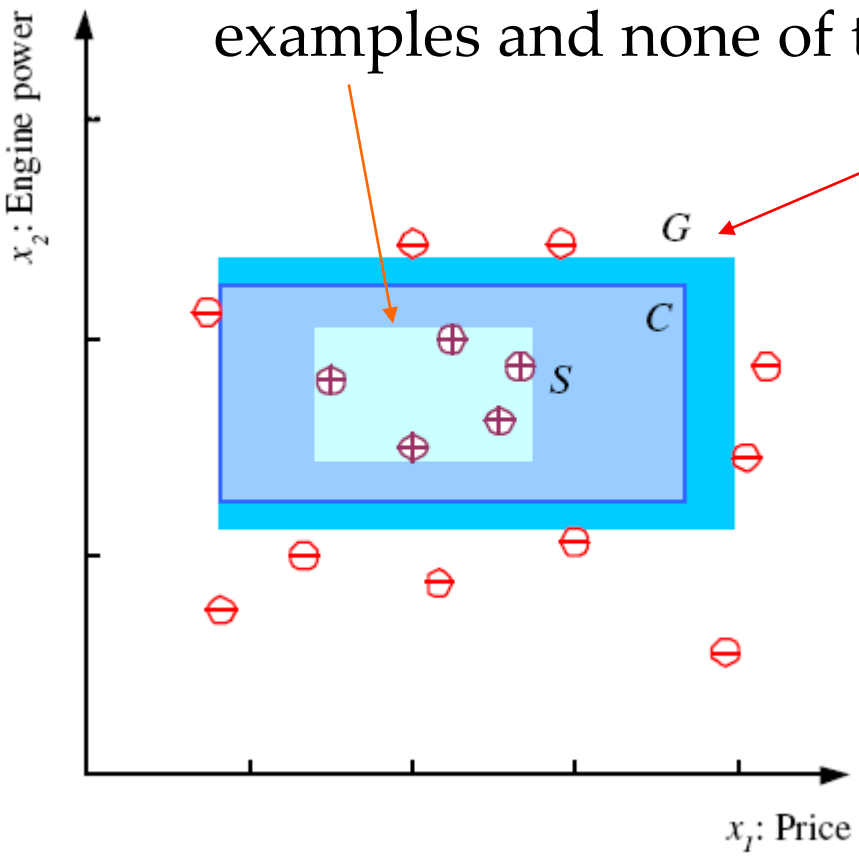
empirical error

Generalization: how well our hypothesis will correctly classify future examples that are not part of the training set.

S, G, and the Version Space

The most specific hypothesis, *S, the* tightest rectangle that includes all the positive examples and none of the negative examples

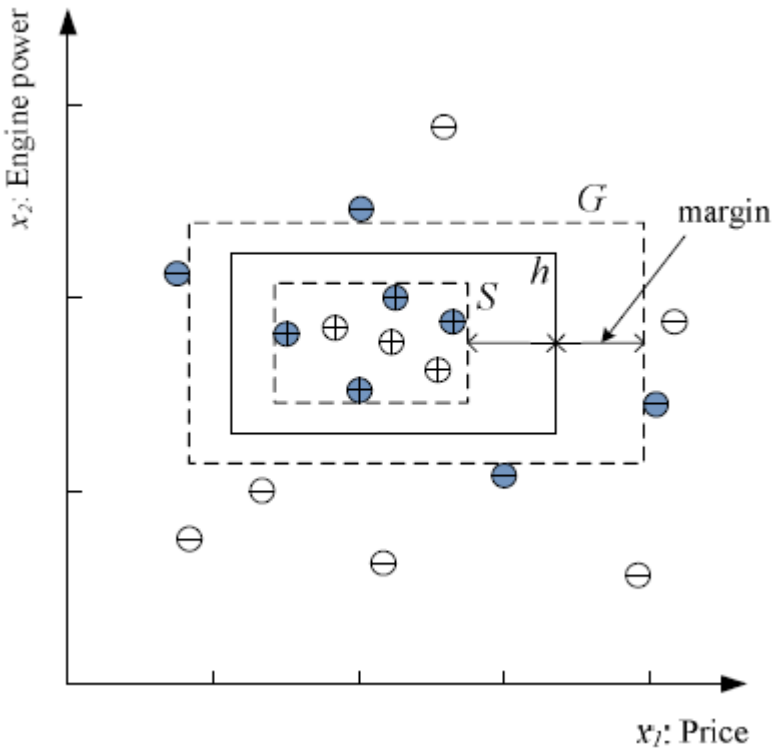
The most general hypothesis, G



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

Margin

- Choose *h* with largest margin. It seems intuitive to choose *h* halfway between S and G; this is to increase the margin, which is the distance between the margin boundary and the instances closest to it.

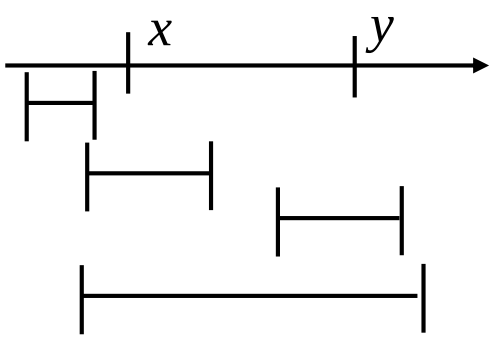


Doubt

- In some applications, a wrong decision may be very costly and in such a case, we can say that any instance that falls in between S and G is a case of doubt, which we cannot label with certainty due to lack of data. In such a case, the system rejects the instance and defers the decision to a human expert.

Shattering Instances

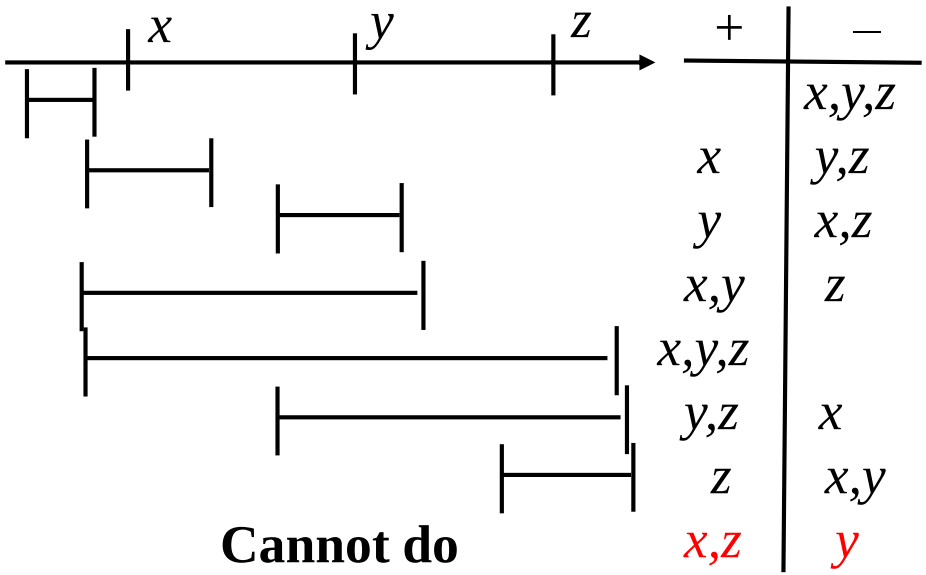
- A hypothesis space is said to shatter a set of instances iff for every partition of the instances into positive and negative, there is a hypothesis that produces that partition.
- For example, consider 2 instances described using a single real-valued feature being shattered by intervals.



+	–
	x,y
x	y
y	x
x,y	

Shattering Instances (cont)

- But 3 instances cannot be shattered by a single interval.



- Since there are 2^m partitions of m instances, in order for H to shatter instances: $|H| \geq 2^m$.

VC Dimension

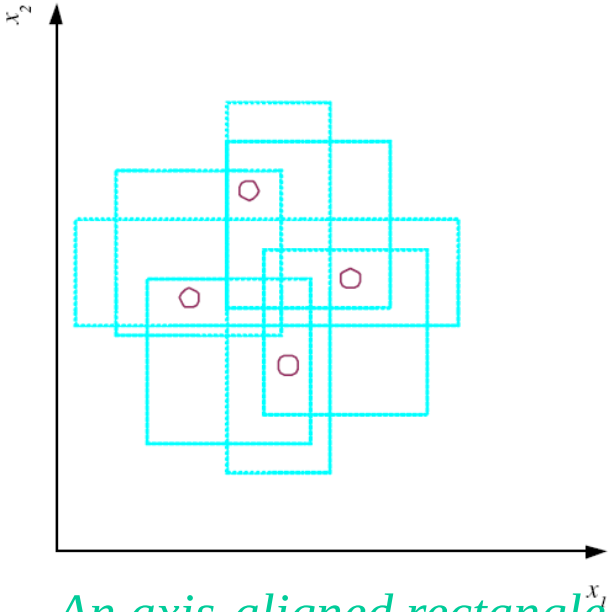
- An unbiased hypothesis space shatters the entire instance space.
- The larger the subset of X that can be shattered, the more expressive the hypothesis space is, i.e. the less biased.
- The Vapnik-Chervonenkis dimension, $VC(H)$ of hypothesis space H defined over instance space X is the size of the largest finite subset of X shattered by H. If arbitrarily large finite subsets of X can be shattered then $VC(H) = \infty$
- If there exists at least one subset of X of size d that can be shattered then $VC(H) \geq d$. If no subset of size d can be shattered, then $VC(H) < d$.
- For a single intervals on the real line, all sets of 2 instances can be shattered, but no set of 3 instances can, so $VC(H) = 2$.
- Since $|H| \geq 2^m$, to shatter m instances, $VC(H) \leq \log_2|H|$

VC Dimension

- N points can be labeled in 2^N ways as +/–
- H shatters N if there exists h ε H consistent for any of these.

That is, any learning problem definable by N examples can be learned with no error by a hypothesis drawn from H.

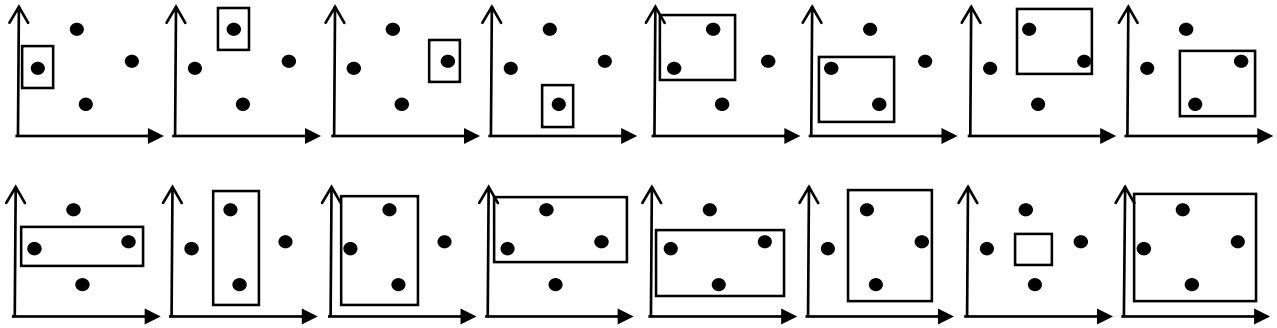
The maximum number of points that can be shattered by H is called the Vapnik-Chervonenkis (*VC*) dimension.



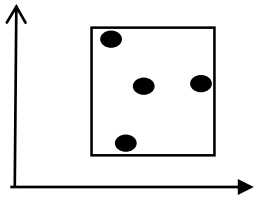
An axis-aligned rectangle shatters 4 points only !

VC Dimension Example

- Consider axis-parallel rectangles in the real-plane, i.e. conjunctions of intervals on two real-valued features. Some 4 instances can be shattered.

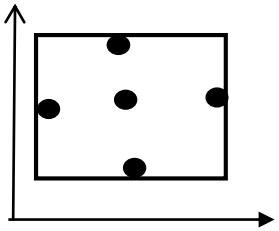


Some 4 instances cannot be shattered:



VC Dimension Example (cont)

- No five instances can be shattered since there can be at most 4 distinct extreme points (min and max on each of the 2 dimensions) and these 4 cannot be included without including any possible 5th point.



- Therefore $VC(H) = 4$
- Generalizes to axis-parallel hyper-rectangles (conjunctions of intervals in n dimensions): $VC(H)=2n$.

Probably Approximately Correct (PAC) Learning

- The only reasonable expectation of a learner is that with ***high probability*** it learns a ***close approximation*** to the target concept.
- In the PAC model, we specify two small parameters, ε and δ, and require that with probability at least $(1 - \delta)$ a system learn a concept with error at most ε.

Formal Definition of PAC-Learnable

- Consider a concept class C defined over an instance space X containing instances of length n, and a learner, L, using a hypothesis space, H. C is said to be ***PAC-learnable*** by L using H iff for all $c \in C$, distributions D over X, $0<\varepsilon<0.5$, $0<\delta<0.5$; learner L by sampling random examples from distribution D, will with probability at least $1-\delta$ output a hypothesis $h \in H$ such that $\text{error}_D(h) \leq \varepsilon$, in time polynomial in $1/\varepsilon$, $1/\delta$, n and size(c).

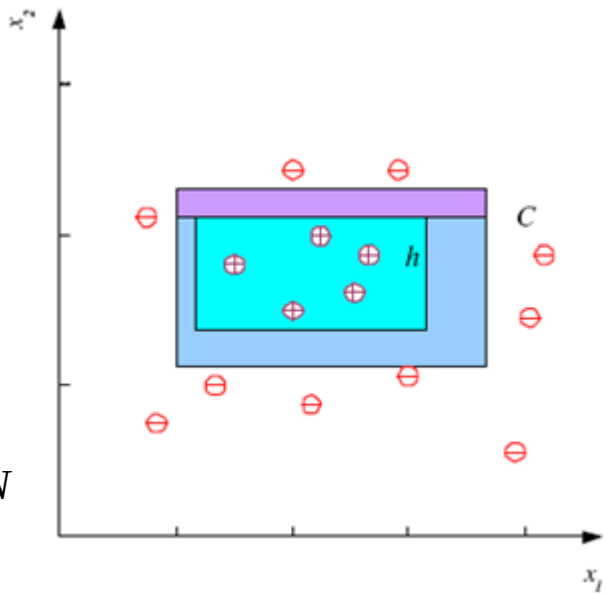
Issues of PAC Learnability

- The computational limitation also imposes a polynomial constraint on the training set size, since a learner can process at most polynomial data in polynomial time.
- How to prove PAC learnability:
  - First prove sample complexity of learning C using H is polynomial.
  - Second prove that the learner can train on a polynomial-sized data set in polynomial time.
- To be PAC-learnable, there must be a hypothesis in H with arbitrarily small error for every concept in C, generally $C \subseteq H$.

Probably Approximately Correct (PAC) Learning (2)

- 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)



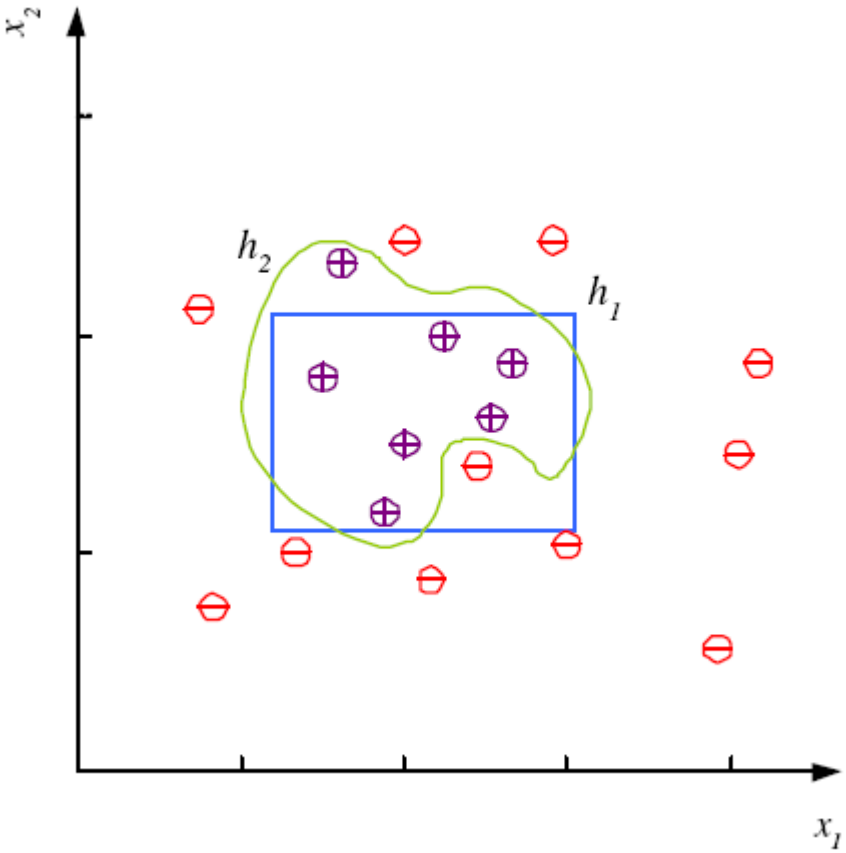
- Each strip is at most $\varepsilon/4$
- Pr that we miss a strip $1 - \varepsilon/4$
- Pr that N instances miss a strip $(1 - \varepsilon/4)^N$
- Pr that N instances miss 4 strips $4(1 - \varepsilon/4)^N$
- $(1 - x) \leq \exp(-x) \rightarrow 4(1 - \varepsilon/4)^N \leq \delta$
- $4\exp(-\varepsilon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

$N \geq (4/\varepsilon)\log(4/\delta)$

- Therefore, provided that we take at least $(4/\varepsilon)\log(4/\delta)$ independent examples from C and use the tightest rectangle as our hypothesis h, with confidence probability at least $1 - \delta$, a given point will be misclassified with error probability at most ε.
- We can have arbitrary large confidence by decreasing δ and arbitrary small error by decreasing ε, and we see in above equation that the number of examples is a slowly growing function of $1/\varepsilon$ and $1/\delta$, linear and logarithmic, respectively.

Noise and Model Complexity

- Imprecision in recording the input attributes
- Errors in labeling the data points
- May be additional attributes, which we have not taken into account,



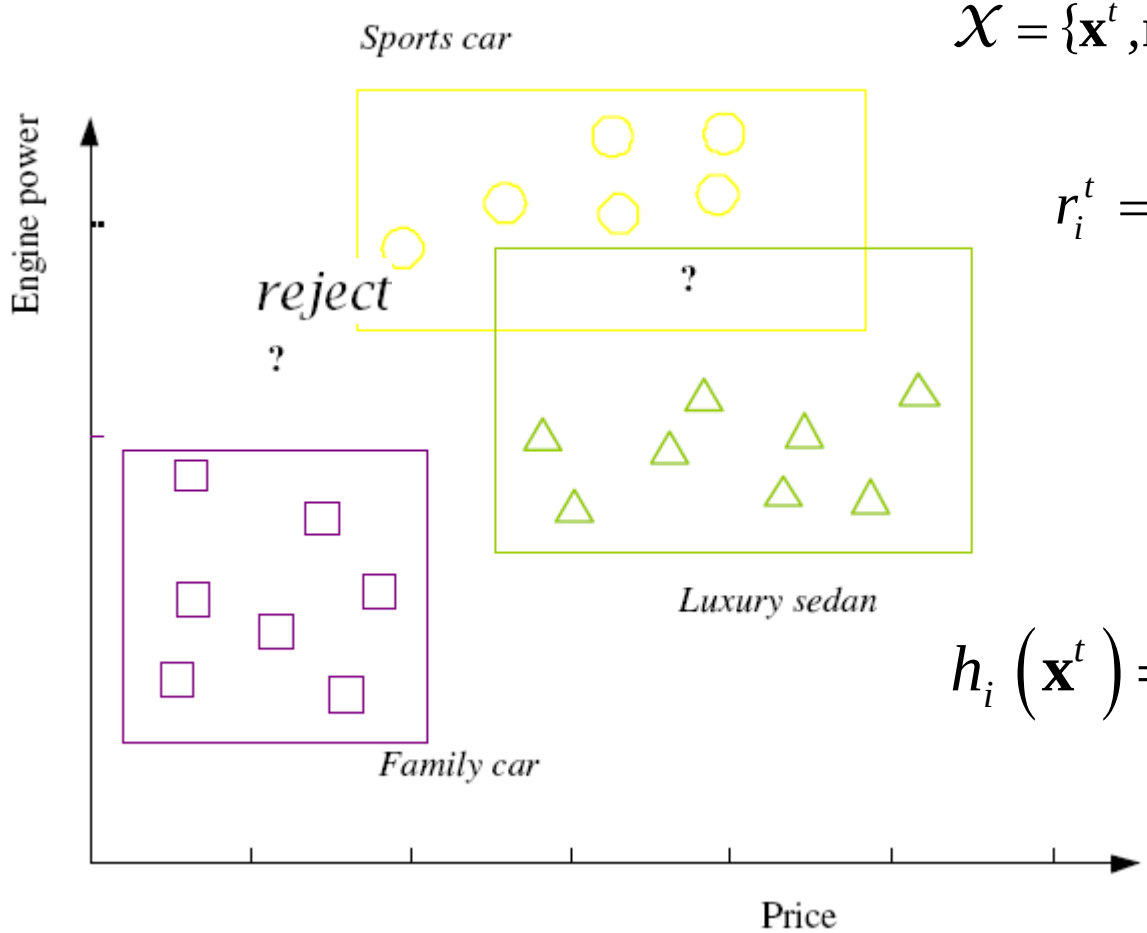
Noise and Model Complexity

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)
  - Note: A simpler model has more *bias*. Finding the optimal model corresponds to minimizing both the bias and the variance.

- Occam's razor: Simpler explanations are more plausible and any unnecessary complexity should be shaved off.

Multiple Classes, C_i ,i=1,...,K



$$\mathcal{X} = \{\mathbf{x}^t, \mathbf{r}^t\}_{t=1}^N \text{ , } \mathbf{r} \text{ is } K\text{-dim}$$

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

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

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

The total empirical error $E(\{h_i\}_{i=1}^K | \mathcal{X}) = \sum_{t=1}^N \sum_{i=1}^K 1(h_i(\boldsymbol{x}^t) \neq r_i^t)$

Regression

- In classification, given an input, the output that is generated is Boolean; it is a yes/no answer.
- If the output is continuous and there is no noise the task is *interpolation*. $\mathrm{X} = \left\{ \mathbf{x}^t, r^t \right\}_{t=1}^{N}, \; r^t \in \Re, \;\; r^t = f\left(\mathbf{x}^t\right)$
- In *regression*, there is noise added to the output of the unknown function $r^t = f\left(\mathbf{x}^t\right) + \varepsilon$
- The explanation for noise is that there are extra *hidden* variables that we cannot observe

$$r^t = f^*\left(\mathbf{x}^t, \mathbf{z}^t\right)$$

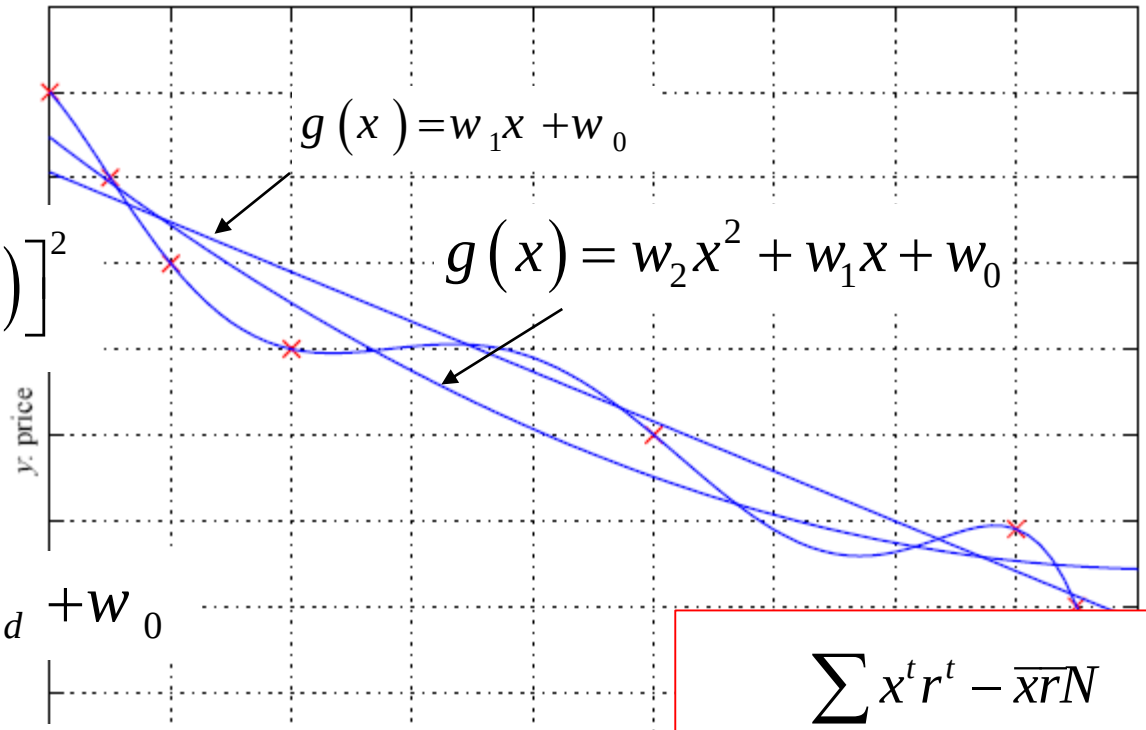
$\mathbf{z}^t$ denote those hidden variables.

Regression

We would like to approximate the output by our model $g(\mathbf{x})$.

The empirical error is:

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



If $g(\mathbf{x})$ is linear:

$$g(\mathbf{x}) = w_1 x_1 + \ldots + w_d x_d + w_0$$

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

Error minimization

$$w_1 = \frac{\sum_t x^t r^t - \overline{xr}N}{\sum_t \left(x^t\right)^2 - N\overline{x}^2},$$

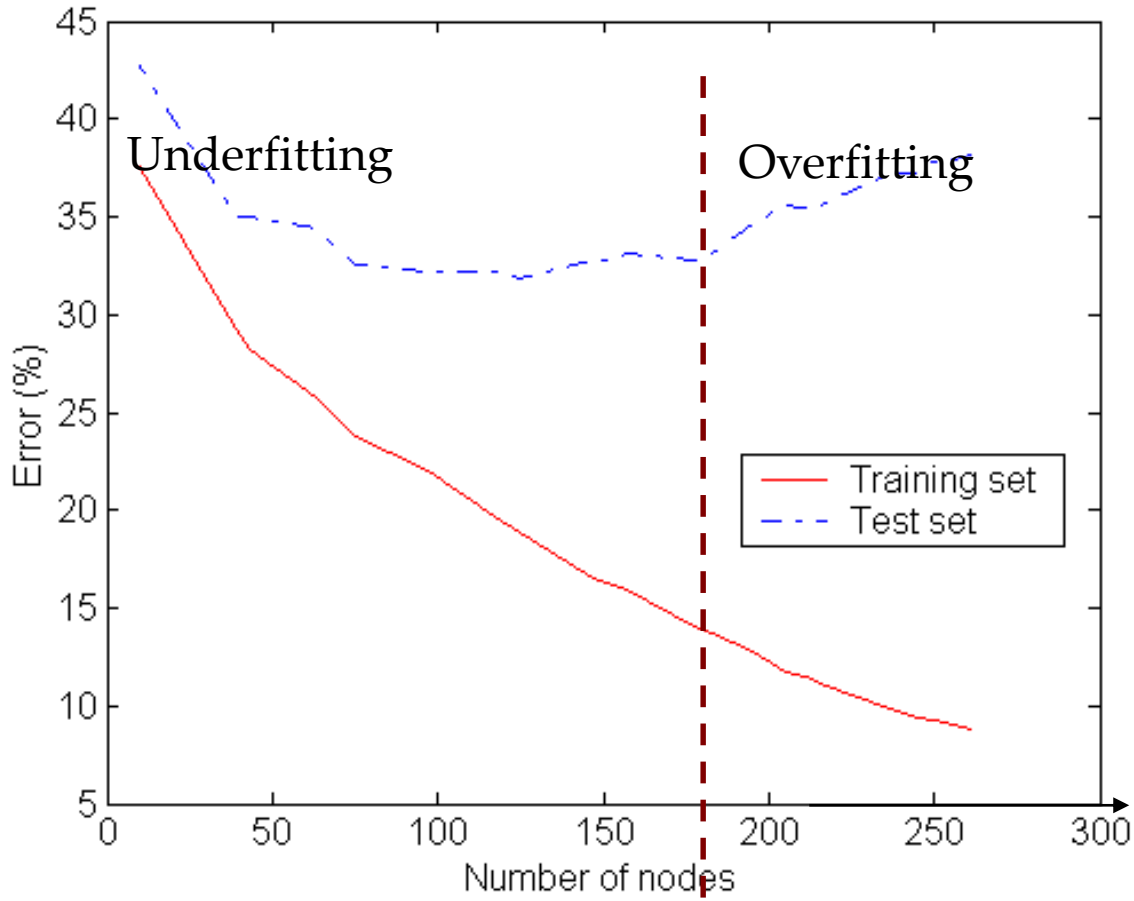
$$w_0 = \overline{r} - w_1\overline{x}$$

Model Selection & Generalization

- Learning is an ill-posed problem when, data is not sufficient to find a unique solution.
- Exp: There are 2^d possible ways to write d binary values and therefore, with d inputs, the training set has at most 2^d examples.
- After seeing N example cases, there remain $2^{2^d - N}$ possible functions.
- The need for inductive bias, assumptions about H; make some extra assumptions to have a unique solution with the data we have.

- Assuming the shape of a rectangle is one inductive bias, and then the rectangle with the largest margin for example, is another inductive bias.
- Generalization:
- How well a model performs on new data
- Overfitting: H more complex than C or f
- Underfitting: H less complex than C or f

Underfitting and Overfitting



Complexity of a Decision Tree = number of nodes it uses

Complexity of the classification function

Underfitting: when model is too simple, both training and test errors are large

Overfitting: when model is too complex and test errors are large although training errors are small.

Triple Trade-Off

- There is a trade-off between three factors (Dietterich, 2003):
  1. Complexity of H, $c(H)$,
  2. Training set size, N,
  3. Generalization error, E, on new data
- As $N \uparrow$, $E \downarrow$
- As $c(H) \uparrow$, first $E \downarrow$ and then $E \uparrow$

Cross-Validation

Error on new examples; actually the testing error is used as an estimation of the generalization error!

- Two errors: training error, and testing error usually *called generalization error*. Typically, the training error is smaller than the generalization error.
- To estimate generalization error, we need data unseen during training. We could split the data as
  - Training set (50%)
  - Validation set (25%)→optional, for selecting ML algorithm parameters (e.g. model complexity)
  - Test (publication) set (25%)
- Resampling when there is few data

Dimensions of a Supervised Learner

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

The sample is independent and identically distributed (*iid*); the ordering is not important and all instances are drawn from the same joint distribution p(**x**,**r**).

The aim is to build a good and useful approximation to $\mathbf{r}^t$ using the model $g(\mathbf{x}^t \mid \boldsymbol{\theta})$.

Three following decisions we must make:

1. Model: $g\left(\mathbf{x} \mid \boldsymbol{\theta}\right)$

where g(·) is the model, **x** is the input, and **θ** are the parameters.

The model (inductive bias), or H, is fixed by the machine learning system designer based on his or her knowledge of the application and the hypothesis h is chosen (parameters are tuned) by a learning algorithm using the training set, sampled from $p(\mathbf{x},\mathbf{r})$.

2. Loss function: The approximation error, or loss, is the sum of losses over the individual instances

$$E\left(\theta|\mathrm{X}\right)=\sum_{t}L\left(r^{t},g\left(\mathbf{x}^{t}|\theta\right)\right)$$

3. Optimization procedure: To find θ^* that minimizes the total error

$$\theta^{*}=\arg\min_{\theta}E\left(\theta|\mathrm{X}\right)$$

Remark: This procedure is typical for Parametric approaches to supervised learning; Non-parametric approaches work differently!

Conditions

- For this setting to work well, the following conditions should be satisfied:
  - The hypothesis class of $g(\cdot)$ should be large enough, that is, have enough capacity, to include the unknown function that generated the data that is represented in X in a noisy form.
  - There should be enough training data to allow us to pinpoint the correct (or a good enough) hypothesis from the hypothesis class.
  - We should have a good optimization method that finds the correct hypothesis given the training data.