Supervised Learning

Berlin Chen

Graduate Institute of Computer Science & Information Engineering

National Taiwan Normal University

References:

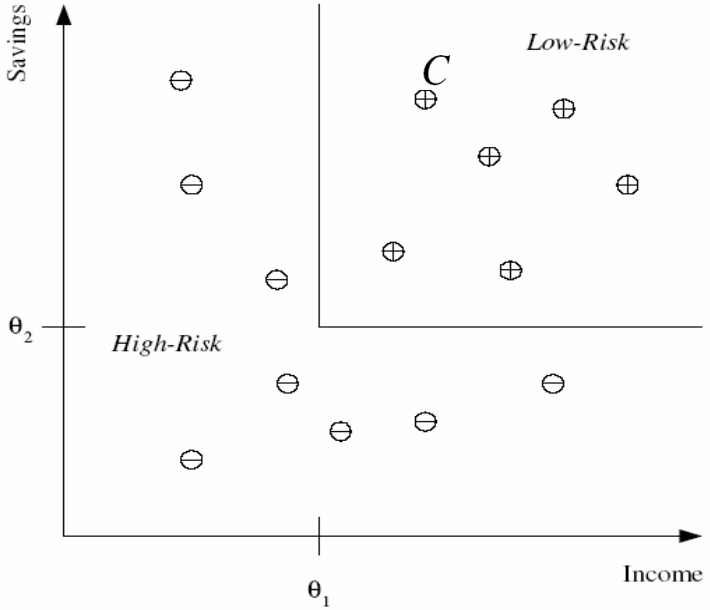
1. **E. Alpaydin, *Introduction to Machine Learning* , Chapter 2**
2. Tom M. Mitchell, *Machine Learning*, Chapter 7
3. T. Hastie et al., *The Elements of Statistical Learning*, Chapter 2

Supervised Learning

- Learning with a Teacher
  - The training examples (instances) are equipped with (class) label information
- Simplest Case
  - Two-class problem (dichotomy): the training set is composed of positive examples (Class 1) and negative examples (Class 0)
    - Output is discrete (symbolic)
  - Use the learned machine to make predictions of the examples not seen before (*Predictive Machine Learning*)
- More Complicated Cases
  - Multi-class problem
    - Output is discrete (symbolic)
  - Regression or function approximation
    - Output is numeric (continuous)

Learning a Class from Examples (1/2)

- Case 1: Credit Scoring
  - Examples (dataset) - customer data of the bank
  - Features (or attributes) of the customers
    - **Income, savings**, collaterals, profession, etc.



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

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

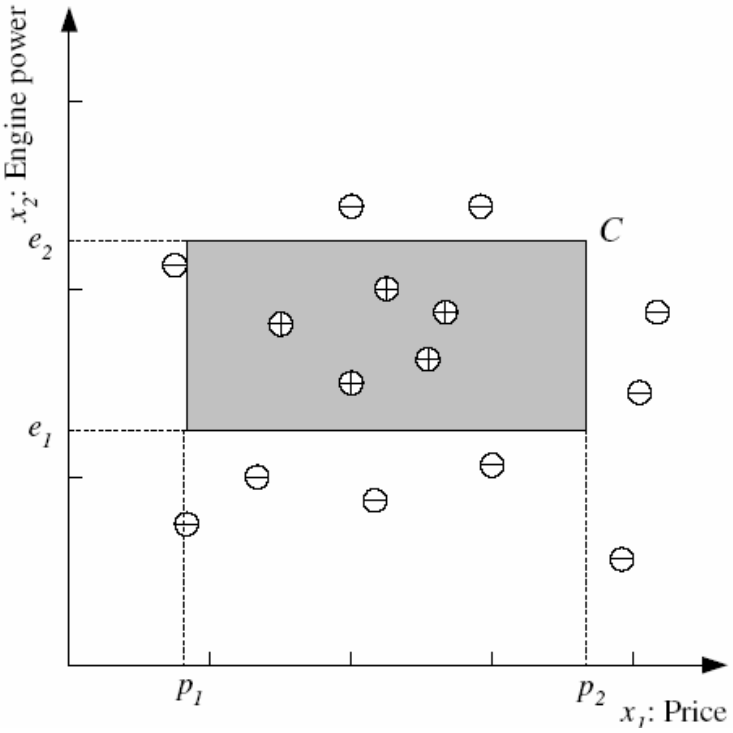
$$\boldsymbol{X} = \left\{ \boldsymbol{x}^t, r^t \right\}_{t=1}^{N}$$

A learned classification rule:

If $income > \theta_1$ AND $savings > \theta_2$ Then *low-risk* ELSE *high-risk*

Learning a Class from Examples (2/2)

- Case 2: Car Class ("Family Car")
  - Examples (dataset) - a set of cars
  - Features (or attributes) of the cars
    - **Price, engine power**, seat capacity, color, etc.



$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \text{Price} \\ \text{Engine Power} \end{bmatrix}$$

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

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

A learned classification rule :

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

THEN family - car

Hypothesis Class (1/3)

- The learning algorithm should find a hypothesis class $h \in H$ to approximate the ideal class (concept) C as closely as possible
  - h can make a prediction for any instance $\boldsymbol{x}$

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

  - In "family car" case, a possible hypothesis class is defined by

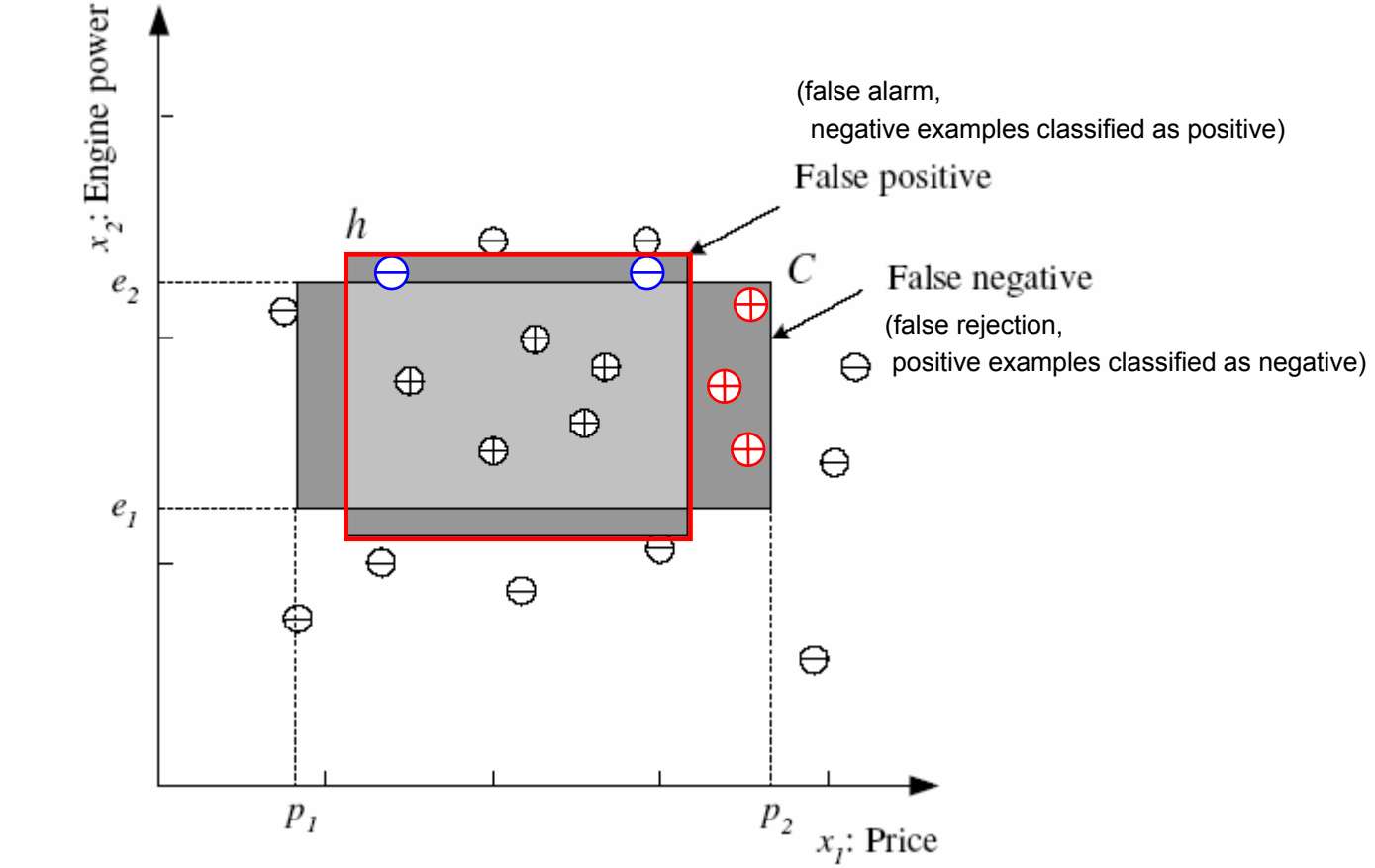
$$\left(p_1^h, p_2^h, e_1^h, e_2^h\right)$$ an axis-aligned rectangle hypothesis

- Empirical Error
  - The proportion of training instances $\boldsymbol{X}$ where predictions of h do not match the true class labels

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

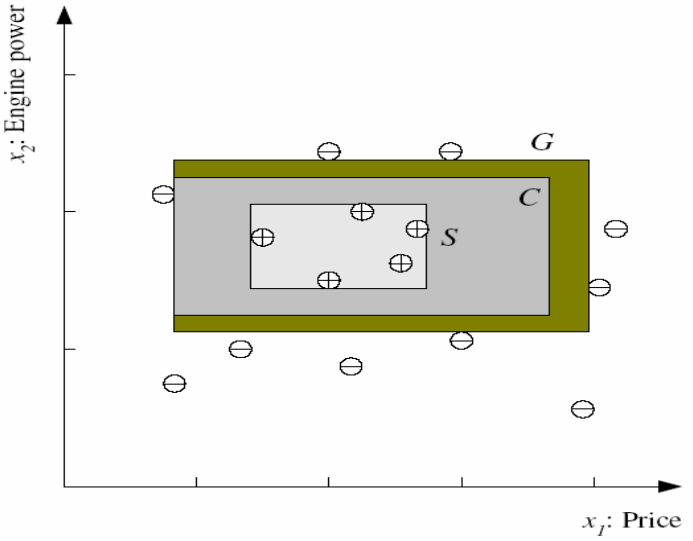
Hypothesis Class (2/3)

- Generalization
  - The issue about how well the hypothesis will correctly classify future (new) examples that are not part of the training set



Hypothesis Class (3/3)

- Given the hypothesis in the form of rectangles
  - Most Specific Hypothesis S
    - The tightest rectangle that includes all the positive examples and none of the negative examples
  - Most General Hypothesis G
    - The largest rectangle that includes all the positive examples and none of the negative examples
  - Version Space Mitchell, 1997
    - Any $h \in H$ between S and G
    - Consisting of valid hypotheses with no error (consistent with the training set)
    - *To be discussed in more detail later on !*



Some Assessments

- The difficulty of the machine learning problems
  - E.g., inherent computational complexity
- The capability of various types of machine learning algorithms
  - E.g., numbers of mistakes will be made
- Frameworks for analyzing, e.g.
  - Vapnik-Chervonenkis (VC) Dimension
  - Probably Approximately Correct (PAC) Learning

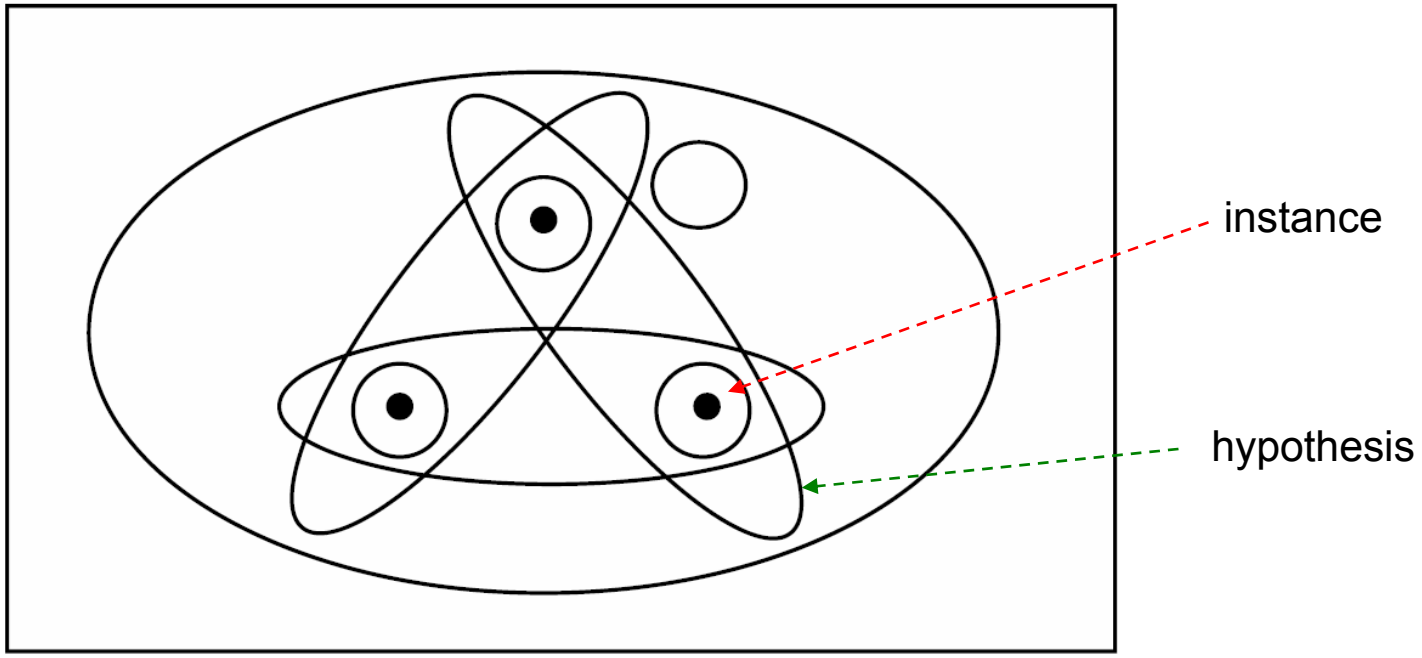
Shattering a Set of Instances (1/2)

- A dichotomy (two-class problem) with a dataset of N points
  - 2^N ways to be labeled as positive and negative
    - Namely, 2^N labeling problems

- Shattering for dichotomy
  - A hypothesis $h \in H$ that can separates the positive examples form the negative
    - H shatters N points

Shattering a Set of Instances (2/2)

- A set of instances *S* is shattered by hypothesis space *H* iff for every dichotomy of *S* there exists some hypothesis in *H* consistent with this dichotomy



  - Eight dichotomies of three instances using ellipses

Vapnik-Chervonenkis (VC) Dimension (1/7)

1970s

- The complexity of the hypothesis space H is not measured by the number of distinct hypotheses $|H|$, but instead by the number of distinct instances S ($|S|=N$) from the instance space X that can be completely discriminated using H
  - Each hypothesis imposes some dichotomy on S ($S \subseteq X$)
  - $2^{|S|}$ possible dichotomies
    - Some of them cannot be shattered
  - The large the subset S that can be shattered, the more expressive H

$$H_a : a_1 x + a_0$$
$$H_b : b_2 x^2 + b_1 x + b_0$$

- A tight bound on sample complexity

VC Dimension (2/7)

- **Def:** The VC 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 sets of X can be shattered by H
    - $VC(H) \equiv \infty$
  - For any finite H
    - $VC(H) \leqq \log_2 |H|$

Given $VC(H) = d$, H will reqire 2^d hypotheses to shatter d instances (for dichotomy).

$\Rightarrow |H| \geq 2^d$

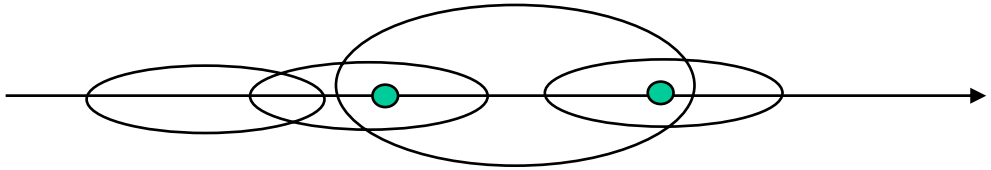
$\Rightarrow \log_2 |H| \geq d = VC(H)$

If we find any set of instances of size d can be shattered, then $VC(H) \geqq d$

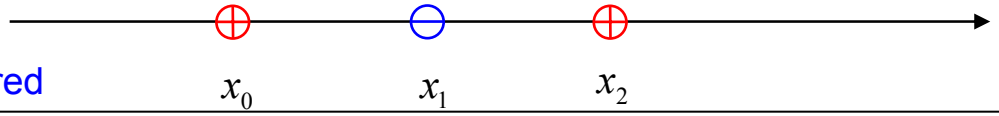
To show that $VC(H) < d$, we must show that no set of size d can be shattered

VC Dimension (3/7)

- Case 1
  - *S*: a set of real values
  - *H*: a set of possible intervals on the real number lines ($|H| \equiv \infty$)
    $H : a < x < b$
  - Suppose that for a subset of *S* consisting of two instances
    - All can be shattered
    - VC(*H*)$\geqq$2 (possibly)



  - Suppose that for a subset of *S* consisting of three instances
    - Cannot be shattered in all cases
    - ∴ VC(*H*)=2 ***?***



No subset of size three can be shattered

VC Dimension (4/7)

- Case 2
  - S: a set of points on the two-dimensional (x, y) plane
  - H: a set of all linear decision surface ($|H| \equiv \infty$)
  - Suppose that for a subset of S consisting of two instances
    - All can be shattered
    - VC(H)$\geqq$2 (possibly)
  - Suppose that for a subset of S consisting of three instances
    - Can be shattered for points not colinear
    - $\therefore$ VC(H)=3

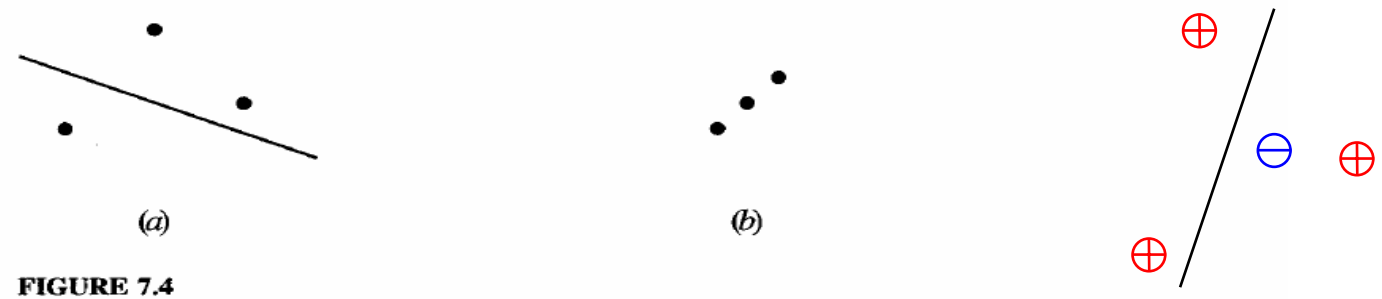
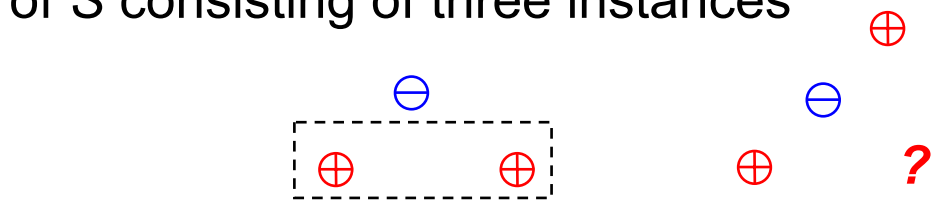
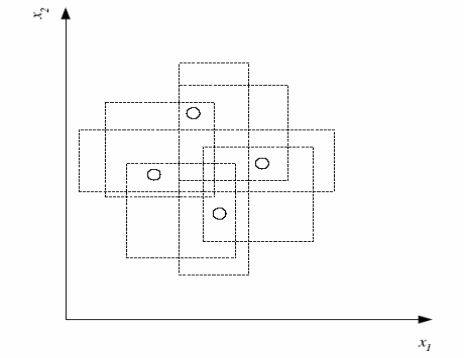


FIGURE 7.4
The VC dimension for linear decision surfaces in the x, y plane is 3. (a) A set of three points that can be shattered using linear decision surfaces. (b) A set of three that cannot be shattered.

VC Dimension (5/7)

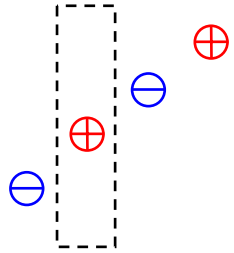
- Case 3
  - *S*: a set of points on the two-dimensional (*x*, *y*) plane
  - *H*: a set of all axis-aligned rectangle ($|H| \equiv \infty$)
  - Suppose that for a subset of *S* consisting of three instances
    - All can be shattered
    - VC(*H*)$\geqq$3 (possibly)
  - Suppose that for a subset of *S* consisting of four instances
    - Can be shattered for points not colinear
    - $\therefore$ VC(*H*)=4



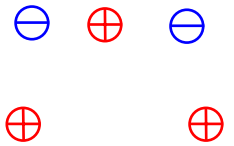


S: four points on a line

H: intervals



We can't dichotomize 5 points for all possible labelings



VC Dimension (6/7)

- Case 4
  - *S*: a set of instances described by the conjunction of exactly three literals
  - *H*: a set of hypothesis each of which is described by the conjunction of up to three literals
  - VC(*H*)=*3*

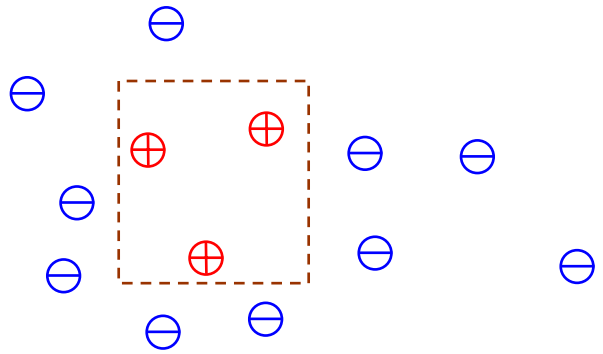
$instance_1$: 100
$instance_2$: 010
$instance_3$: 001

To exclude $instance_1$ and $instance_3$
$\neg l_1 \wedge \neg l_3$

  - The VC dimension for conjunctions of (up to) *n* Boolean literals is at least *n*

VC Dimension (7/7)

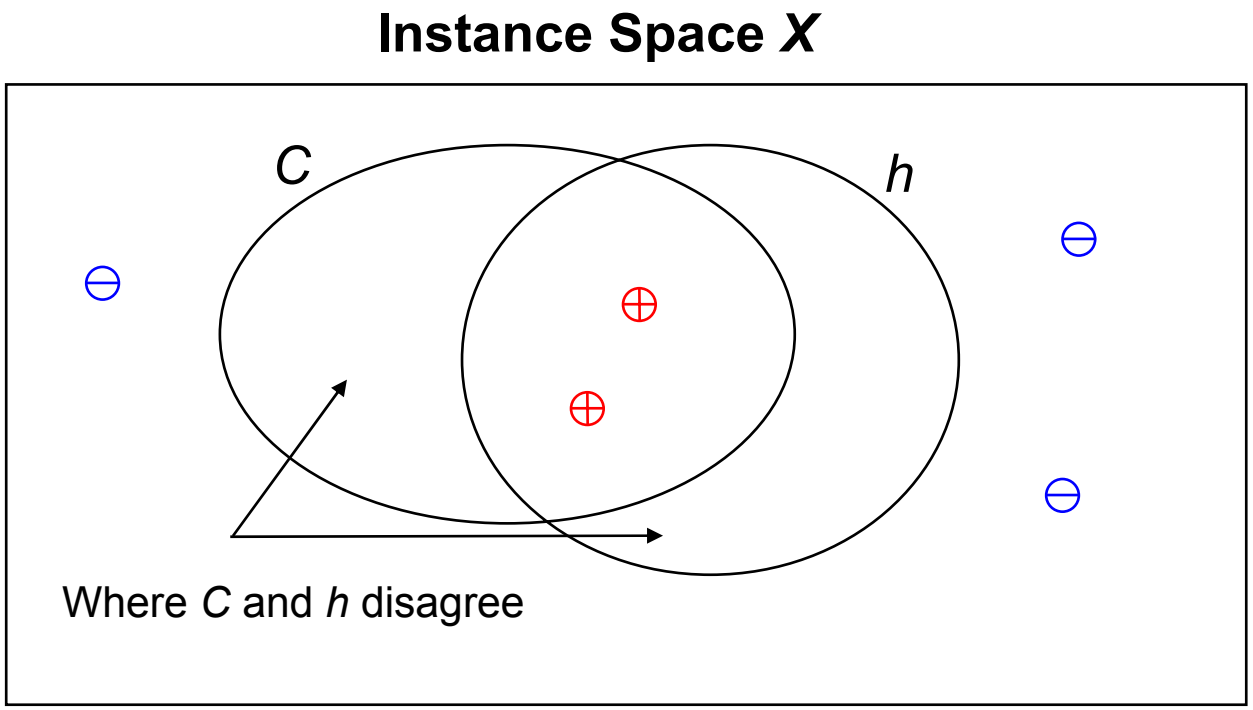
- VC Dimension seems pessimistic
  - E.g., rectangle hypotheses can lean only datasets containing four 2-dimensional points
- VC Dimension does not take the probability distribution of data samples into consideration



positions & labelings

  - Instances close by most of the time have the same color
  - Dataset containing much more than four points might be learned

Probably Approximately Correct (PAC) Learning (1/6)



– True Error ?

$$error_D(h) = \underset{x \in D}{P}[c(\boldsymbol{x}) \neq h(\boldsymbol{x})]$$

$$(\text{where } D \text{ is the distribution of instances})$$

– Training Error ?

PAC Learning (2/6)

- Weakened Demands
  - The error made by the hypothesis is bounded by some constant ε
  - The probability that the hypothesis fails to shattering the instances is bounded δ

PAC Learning (3/6)

(Valiant, 1984)

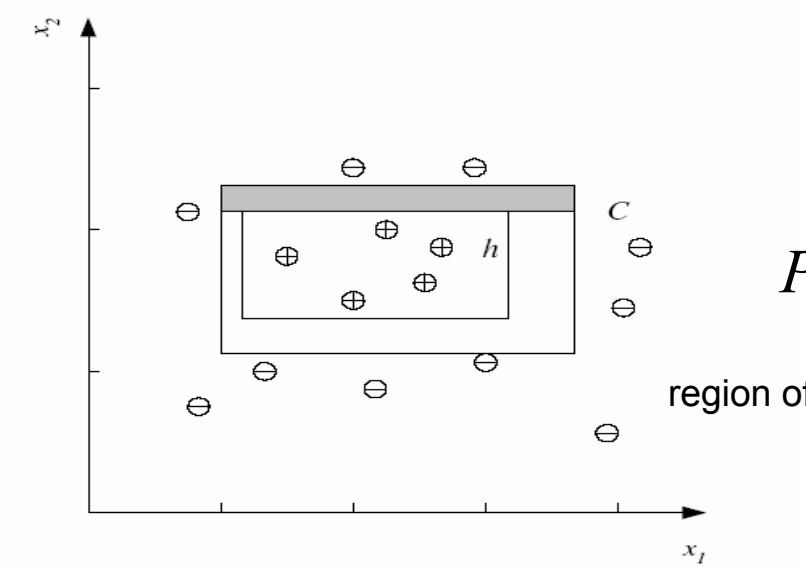
- Def: Given a class, *C*, and examples drawn from some unknown but fixed probability distribution, $p(\mathbf{x})$, we want to find the number of examples, N, such that with probability at least $1-\delta$, the hypothesis h has error at most ε, for arbitrary $\delta \le 1/2$ and $\varepsilon > 0$

$$P\{C \Delta h \le \varepsilon\} \ge 1-\delta$$

  - Measure how closely the hypothesis h approximates the actual target class (concept) *C*

PAC Learning (4/6)

- An Illustrative case (Blumer et al., 1984)
  - *S*: a set of points on the two-dimensional (*x*, *y*) plane
  - *H*: a set of all axis-aligned rectangle
    - *h*: the tightest rectangle hypothesis



$$P\{C \Delta h \le \varepsilon\} \ge 1-\delta$$

region of difference between *C* and *h*

  - We want positive examples falling in $C \Delta h$ is at most ε
    - Falling in one of the four strips is less than $\varepsilon / 4$

PAC Learning (5/6)

- An Illustrative case (cont.)
  - The probability that *N* independent draws missing (not falling in) any of the four strips is

$$4\left(1-\varepsilon/4\right)^N$$

    - Which we like to be at most δ , namely

$$4\left(1-\varepsilon/4\right)^N \le \delta$$

  - We also have that $\log x \le x-1$

$$4\left(1-\varepsilon/4\right)^N \le \delta$$
$$N \log\left(1-\varepsilon/4\right) \le \log\left(\delta/4\right)$$
$$N\left(-\varepsilon/4\right) \le \log\left(\delta/4\right)$$
$$\therefore N \ge \left(4/\varepsilon\right)\log\left(4/\delta\right)$$

let $\left(1-\varepsilon/4\right) = x$

$\therefore x-1 = \left(-\varepsilon/4\right)$

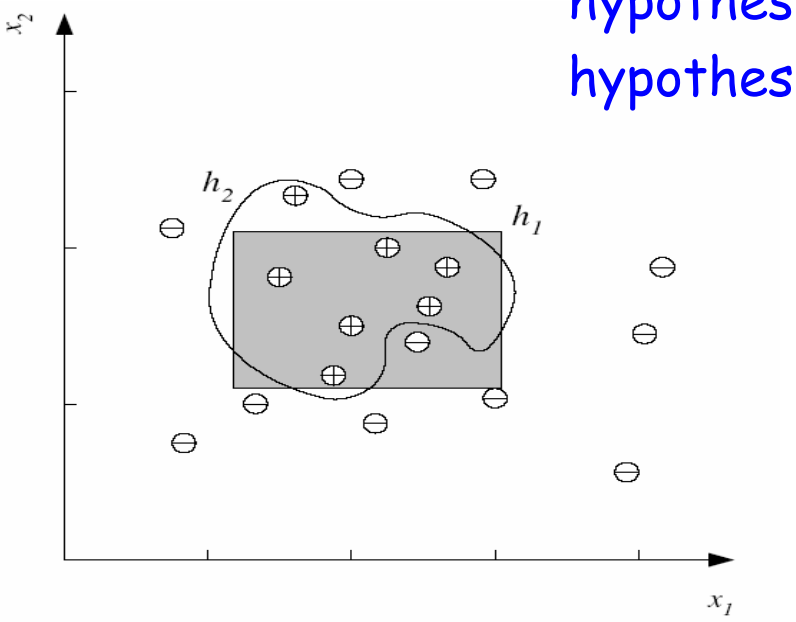
(A loose bound)

PAC Learning (6/6)

- An Illustrative case (cont.)
  - The number of examples *N* is a slowly growing function of $1/\varepsilon$ and $1/\delta$, linear and logarithmic respectively
  - Provided that we take at least $(4/\varepsilon)\log(4/\delta)$ independent examples 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

Noise (1/2)

- Unwanted anomaly in the data due to:
  - Imprecision in recording the input attributes
  - Errors in labeling the data points (teacher noise)
  - Misuse of features, or there are hidden or latent features which are unobservable



hypotheses with simple shapes vs.
hypotheses with wiggly shapes

Noise (2/2)

- But simple models (like axis-aligned rectangles) are
  - Easy to use
  - With few parameters $\rightarrow$ a small training set needed $\left(p_1^h, p_2^h, e_1^h, e_2^h\right)$
  - Easy to explain (interpret)
  - Less affected by single instances

Occam's Razor

- Simple explanations are more plausible and unnecessary complexity should be shaved off
  - A simple model has more bias
  - A complex model has more variance
  - Tradeoff between bias and variance
    - *To be discussed in more detail later on*

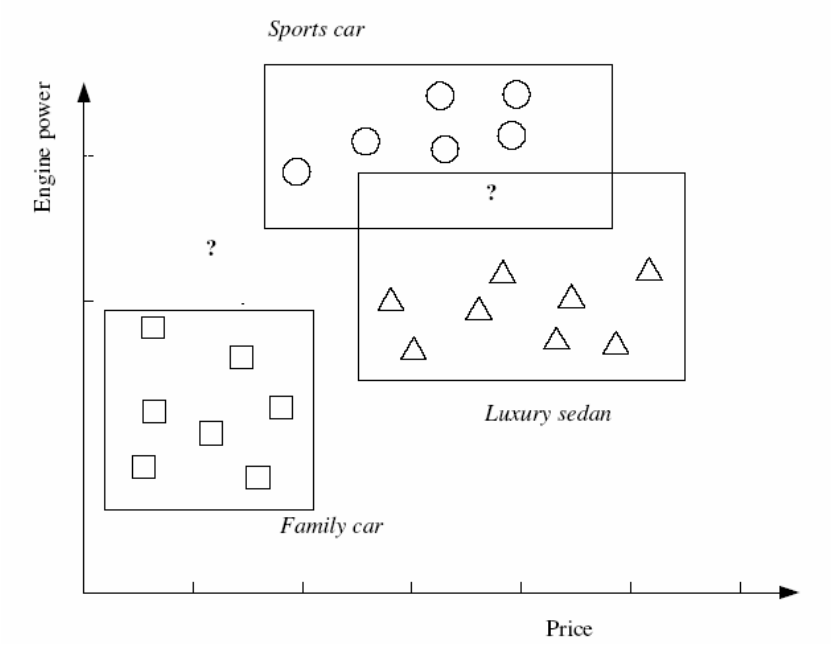
Simple is Better!

Learning Multiple Classes (1/2)

- Extend a two-class problem to a *K*-class problem

$$\mathbf{X} = \left\{\mathbf{x}^t, \mathbf{r}^t\right\}_{t=1}^{N} \qquad (\mathbf{r}^t \text{ has } K \text{ dimensions})$$

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



Learning Multiple Classes (2/2)

- A *K*-class classification problem can be viewed as *K* two-class problems
  - For each class C_i, training examples can be either positive (belonging to C_i) and negative (belonging to all other classes)
  - Build a hypothesis h_i for each class C_i

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

  - What if no, or two or more $h_i\left(\boldsymbol{x}\right)$ is 1
    - The case of *doubt* : reject such cases or defer the decision to human experts

Interpolation/Regression (1/2)

- To learn an unknown function $f(\cdot)$ whose output is numeric from a training set of examples

$$\boldsymbol{X} = \left\{\boldsymbol{x}^t, r^t\right\}_{t=1}^{N} \qquad (r^t \text{ is numeric, i, e. } r^t \in R)$$

- Interpolation: when no error is presented, we want to find a function $f(\boldsymbol{x})$ that pass through those training points
  - Extrapolation: to predict the output for any $\boldsymbol{x}$ unseen in the training set

N points
$\Rightarrow (N-1)$st degree polynomial

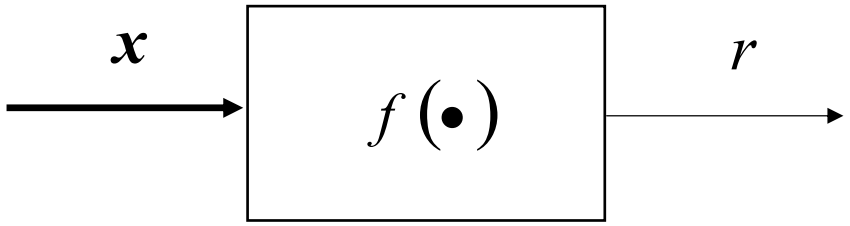
- Regression
  - There is noise added to the output of the unknown function

$$r^t = f(\boldsymbol{x}) + \varepsilon \quad \Rightarrow \quad r^t = f^*(\mathbf{x}, \mathbf{z})$$

- ε : random noise (thought of as extra hidden variables $\boldsymbol{z}$)

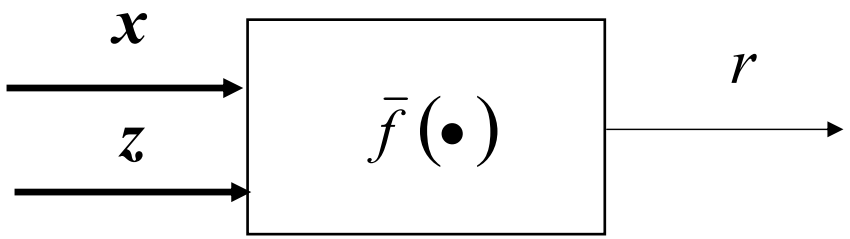
Interpolation/Regression (2/2)

- Interpolation



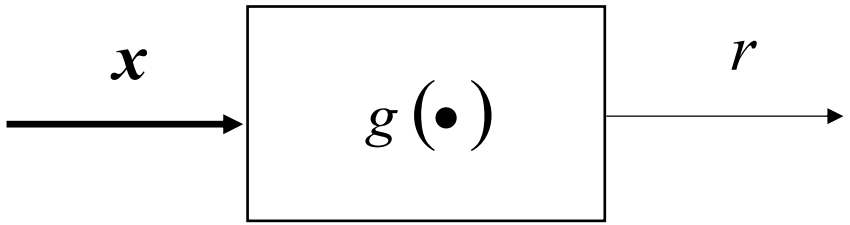
$$r = f(\boldsymbol{x})$$

- Regression



$$r = \bar{f}(\boldsymbol{x}, \boldsymbol{z})$$
$$= \hat{f}(\boldsymbol{x}) + e$$

($\boldsymbol{z}$: hidden var iable)



$$r \approx g(\boldsymbol{x})$$

Regression (1/4)

- Given the noise as the hidden variables z, try to approximate the function output by our model $g(x)$
- Empirical error on the training set X defined as

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

- E.g., If g is linear and x is one-dimensional examples

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

- Take partial derivatives of $E(\cdot)$ with respect to parameters w_1 and w_0, and setting them equal to zero

$$w_1 = \frac{\sum_t x^t r^t - \bar{x}\bar{r}N}{\sum_t (x^t)^2 - \bar{x}^2 N} \qquad where \quad \bar{x} = \sum_t x^t / N, \quad \bar{r} = \sum_t r^t / N$$

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

Regression (2/4)

- Appendix A:

$$E\left(g\middle|\mathbf{X}\right)=\frac{1}{N}\sum_{t=1}^{N}\left[r^t-\left(w_1x^t+w_0\right)\right]^2$$

(i) $$\frac{\partial E\left(g\middle|\mathbf{X}\right)}{\partial w_1}=\frac{2}{N}\sum_{t=1}^{N}x^t\left[r^t-\left(w_1x^t+w_0\right)\right]=0$$

$$\Rightarrow \sum_{t=1}^{N}x^tr^t-w_1\sum_{t=1}^{N}\left(x^t\right)^2-w_0N\bar{x}=0$$

(ii) $$\frac{\partial E\left(g\middle|\mathbf{X}\right)}{\partial w_0}=\frac{2}{N}\sum_{t=1}^{N}\left[r^t-\left(w_1x^t+w_0\right)\right]=0$$

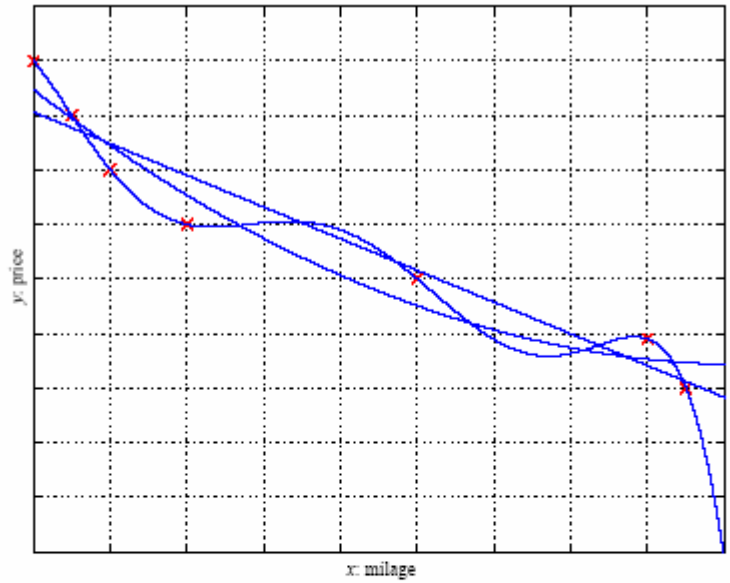
$$\Rightarrow N\bar{r}-w_1N\bar{x}-w_0N=0$$

$$\Rightarrow w_0=\bar{r}-w_1\bar{x}$$

(iii) $$\sum_{t=1}^{N}x^tr^t-w_1\sum_{t=1}^{N}\left(x^t\right)^2-\left(\bar{r}-w_1\bar{x}\right)N\bar{x}=0$$

$$\Rightarrow w_1=\frac{\sum_{t=1}^{N}x^tr^t-N\bar{r}\bar{x}}{\sum_{t=1}^{N}\left(x^t\right)^2-N\bar{x}^2}$$

Regression (3/4)



Second - order polynomial (quadratic)

$$g(x) = w_2 x^2 + w_1 x + w_0$$

Figure 2.9: Linear, second-order, and sixth-order polynomials are fitted to the same set of points. The highest order gives a perfect fit but given this much data, it is very unlikely that the real curve is so shaped. The second order seems better than the linear fit in capturing the trend in the training data.

Regression (4/4)

- Two-dimensional Inputs

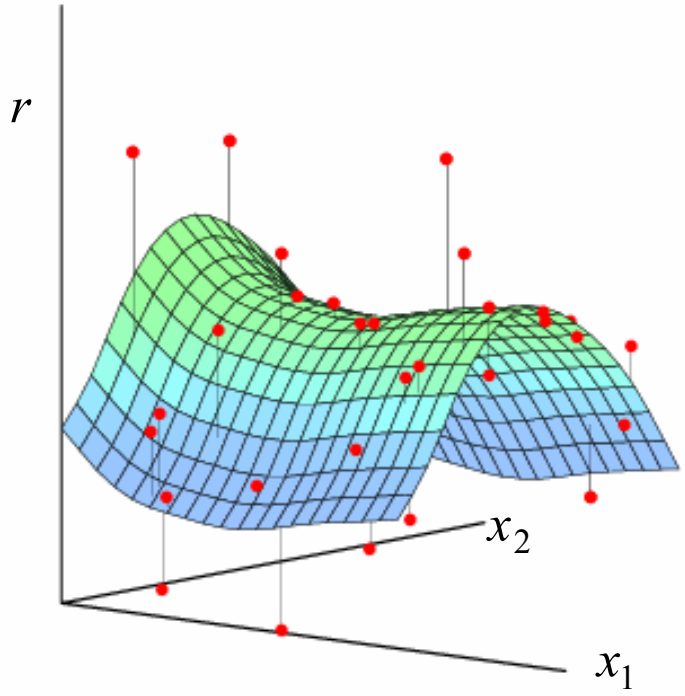


Figure 2.10: *Least squares fitting of a function of two inputs. The parameters of $f_\theta(x)$ are chosen so as to minimize the sum-of-squared vertical errors.*

Model Selection (1/3)

- An Illustrative case
  - Binary *d*-dimensional examples (examples with *d* inputs/features)
  - Binary output values (0 and 1) $\{\mathbf{x}^t, r^t\}$
  - If *d*=2 2^{2^d} possible Boolean functions

Table 2.1 With two inputs, there are four possible cases and sixteen possible Boolean functions.

x_1	x_2	h_1	h_2	h_3	h_4	h_5	h_6	h_7	h_8	h_9	h_{10}	h_{11}	h_{12}	h_{13}	h_{14}	h_{15}	h_{16}
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

inputs

output

  - When more training samples observed, more hypotheses inconsistent with them are removed
  - Each distinct training example removes half of the remaining hypotheses

Model Selection (2/3)

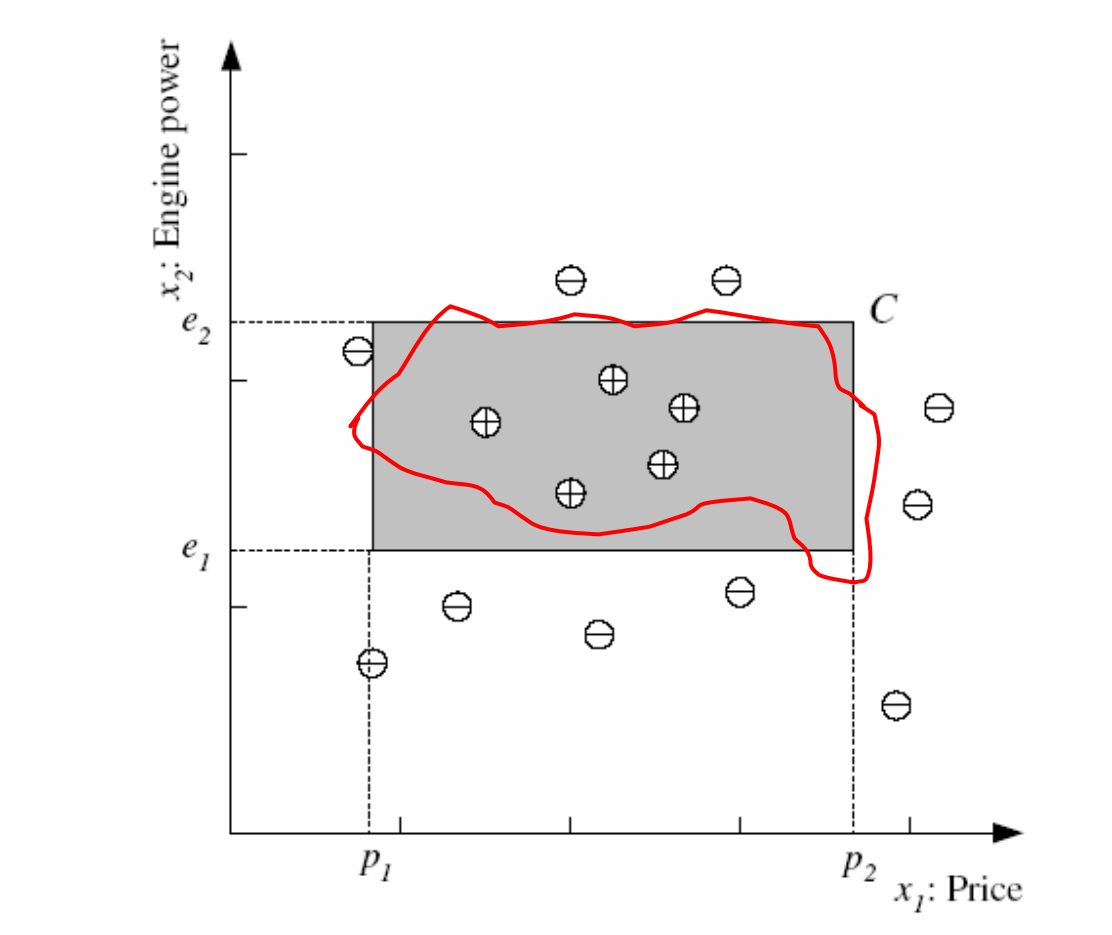
- An Illustrative case (cont.)
  - Given *N* *d*-dimensional training examples observed
    - There will be $2^{2^d - N}$ possible functions remained

- Ill-Posted Problem
  - The training data by itself is not sufficient to find a unique solution

model selection: how to choose the right (inductive) bias

- Inductive Bias
  - The assumptions made to have learning possible (e.g. to find a unique solution given the insufficient training data)
  - Car classification: hypothesis class in rectangles instead of wiggly shapes
  - Regression: linear hypothesis class instead of polynomial ones

Model Selection (3/3)



Model Complexity (1/3)

- We should match the complexity of the hypothesis with the complexity of the function underlying the training/test data
  - Underfitting: the hypothesis is less complex than the function
    - E.g., try to fit a line to data sampled from a third-order polynomial
  - Overfitting: the hypothesis is more complex than the function
    - E.g., try to fit a six-order polynomial to data sampled from a third-order polynomial

Model Complexity (2/3)

- Trade-off between three factors (Triple Trade-off)
  - The complexity (capacity) of the hypothesis
  - The amount of training data
  - The generalization error on new samples

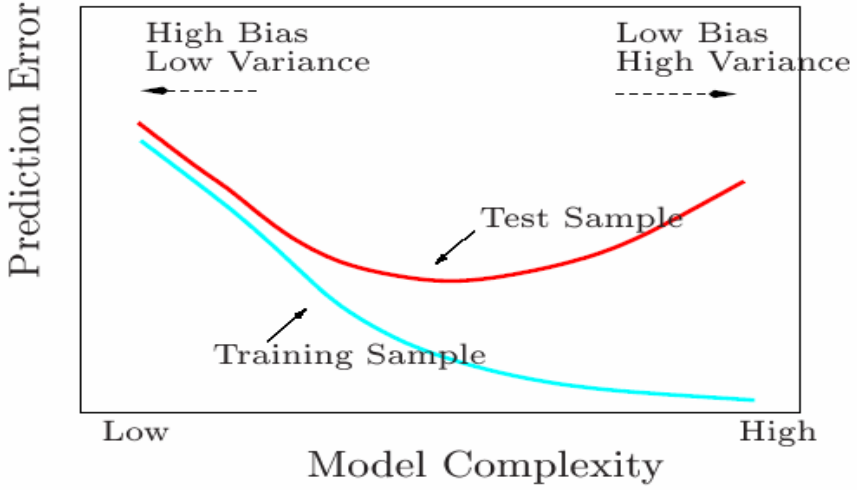


Figure 2.11: *Test and training error as a function of model complexity.*

Model Complexity (3/3)

- Data sets (for measuring generalization ability)
  - Training Set
  - Validation Set (Development)
    - Test the generalization ability
    - Select the best model (complexity)
  - Test Set (Evaluation)

50% 25% 25%

Summary

- Given a training set of labeled examples $\mathbf{X} = \left\{\mathbf{x}^t, r^t\right\}$, supervised learning aims to build a good and useful approximate to r^t with three issues taken into account

1. Model $g\left(\mathbf{x}|\theta\right)$ to be employed
   - Car classification: rectangles with four coordinates making up θ
   - Linear regression: linear function with slope and intercept
2. Loss function computing difference between desired output and the approximation

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

3. Optimization procedure fining θ^* minimizing the approximation error

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