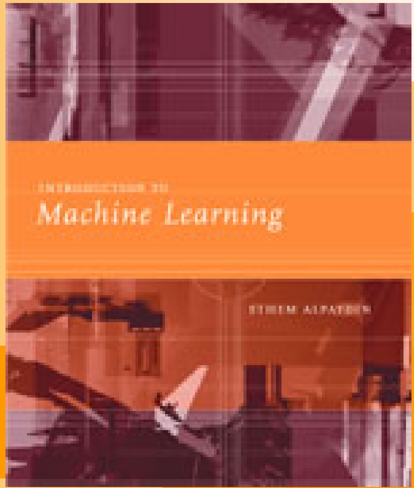



*Lecture Slides for*

# INTRODUCTION TO *Machine Learning*

ETHEM ALPAYDIN


*alpaydin@boun.edu.tr*
*http://www.cmpe.boun.edu.tr/~ethem/i2ml*

CHAPTER 2:

# *Supervised Learning*

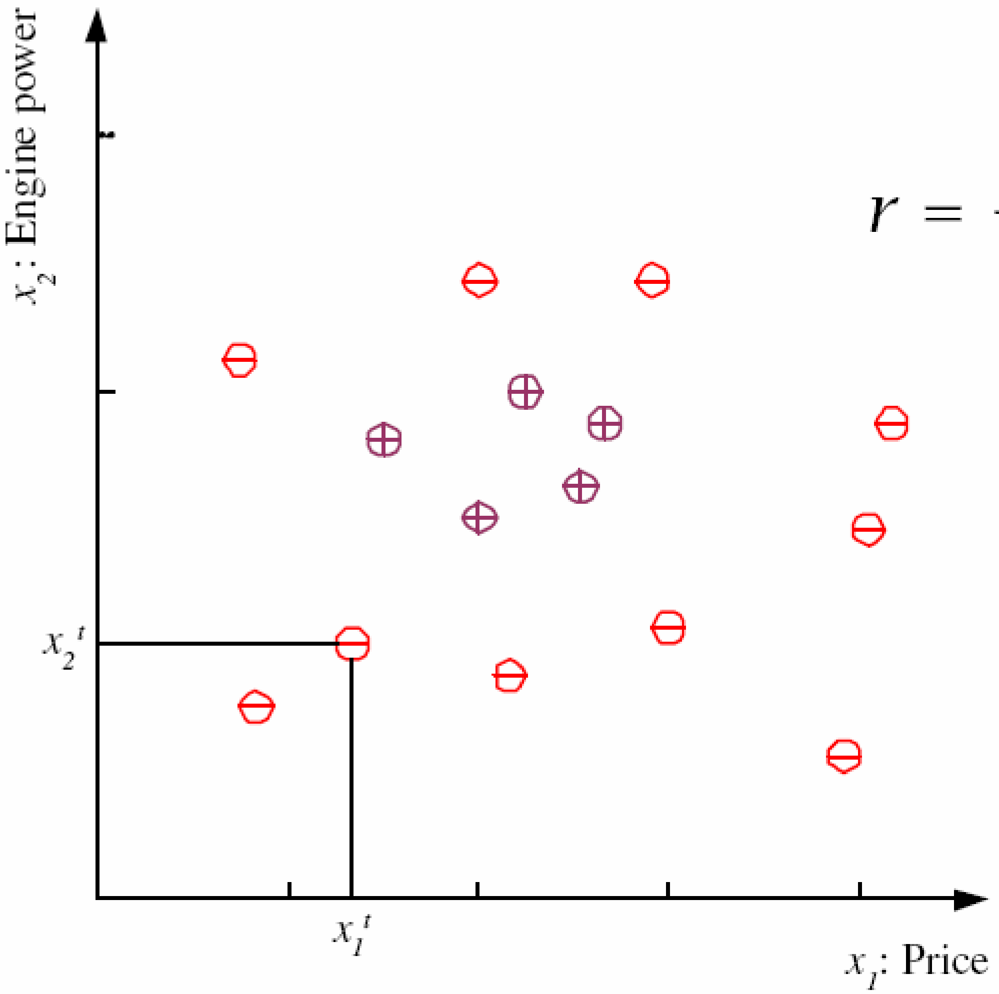
# *Learning a Class from Examples*

- 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 $\mathcal{X}$*

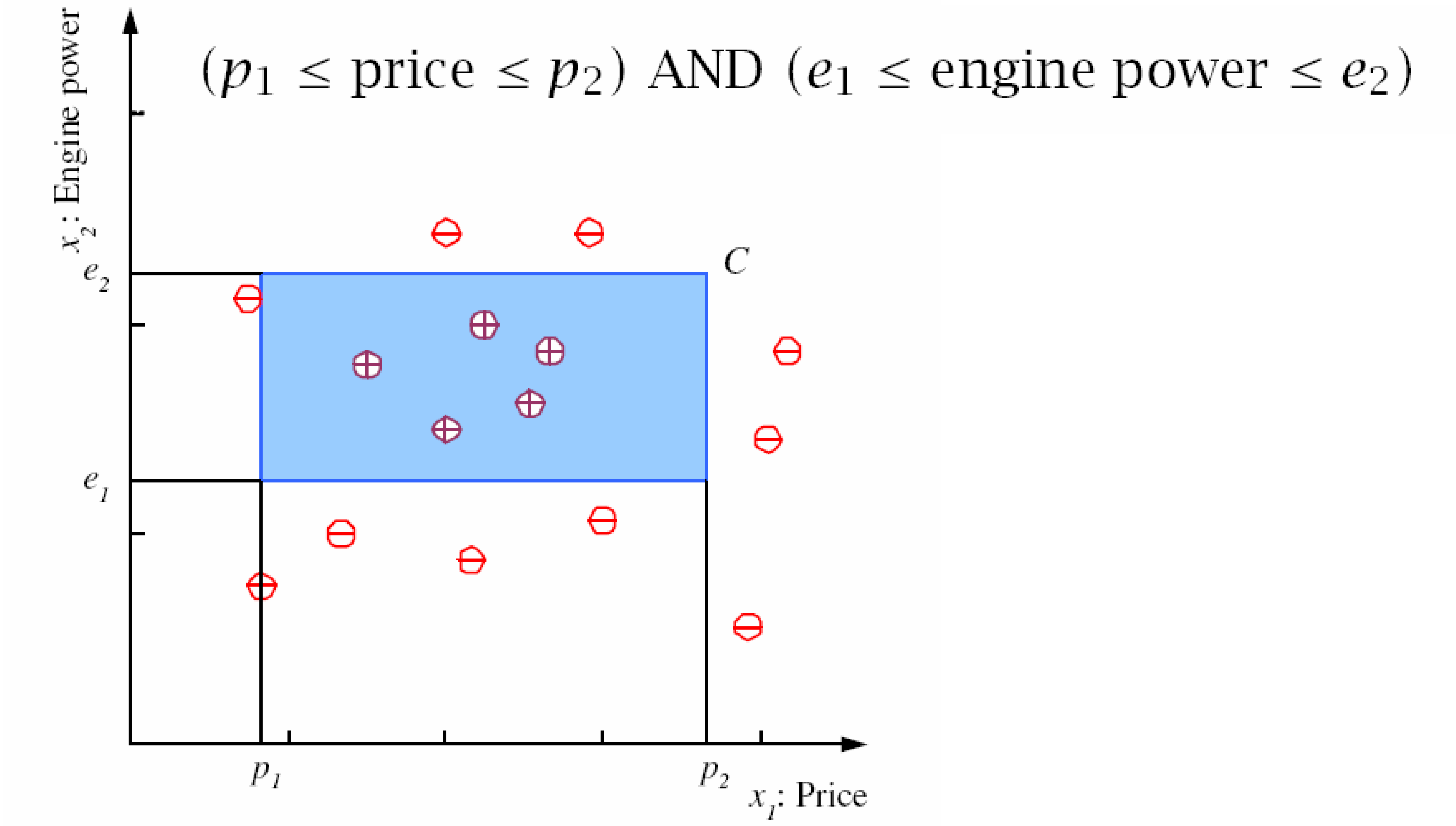



$$\mathcal{X} = \{\mathbf{x}^t, r^t\}_{t=1}^N$$

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

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

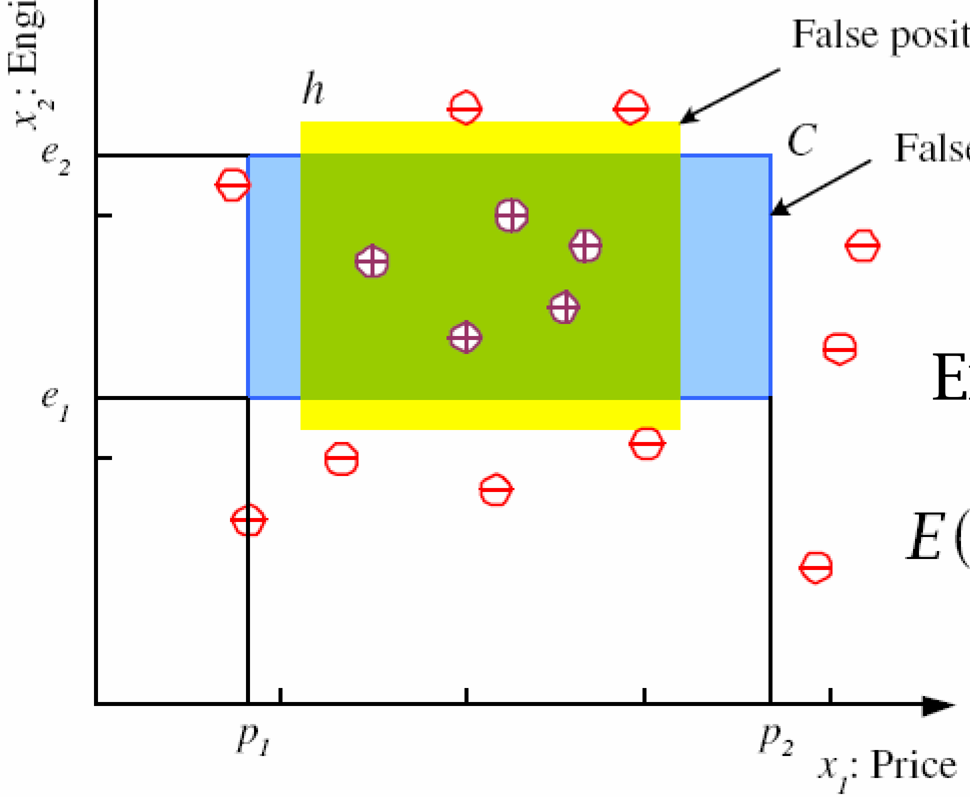
# Class C



$(p_1 \leq \text{price} \leq p_2)$ AND $(e_1 \leq \text{engine power} \leq e_2)$
$x_2$ : Engine power
$e_2$
$e_1$
$C$
$p_1$
$p_2$
$x_1$: Price

# Hypothesis class $\mathcal{H}$

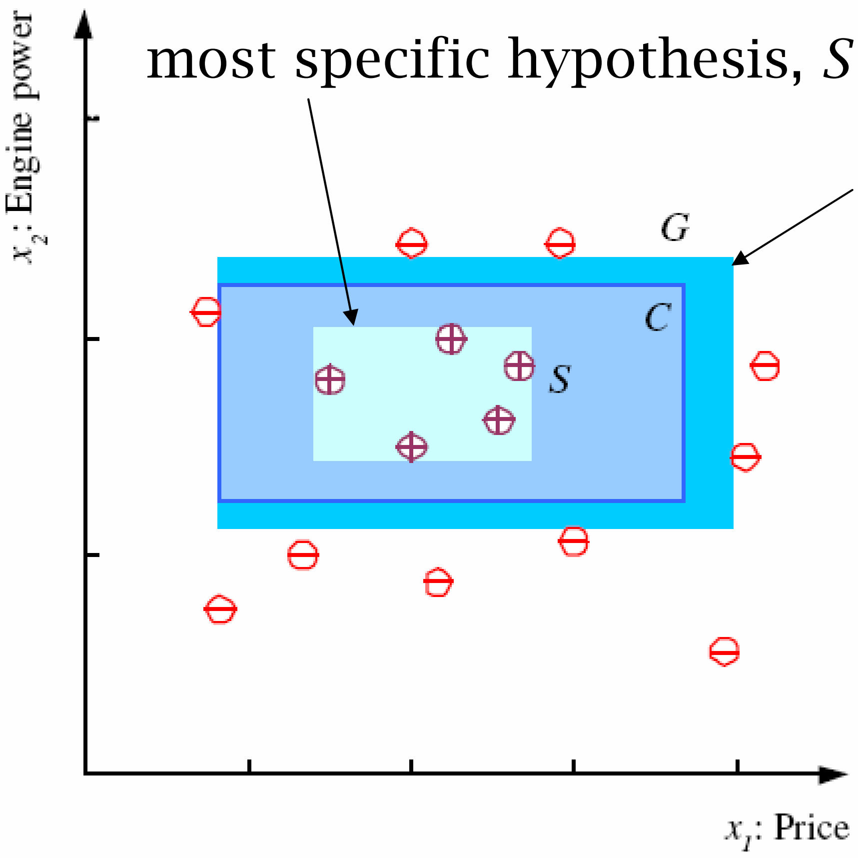
$$h(\mathbf{x}) = \begin{cases} 1 & \text{if } h \text{ classifies } \mathbf{x} \text{ as a positive example} \\ 0 & \text{if } h \text{ classifies } \mathbf{x} \text{ as a negative example} \end{cases}$$




Error of $h$ on $\mathcal{H}$

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

# *S, G, and the Version Space*



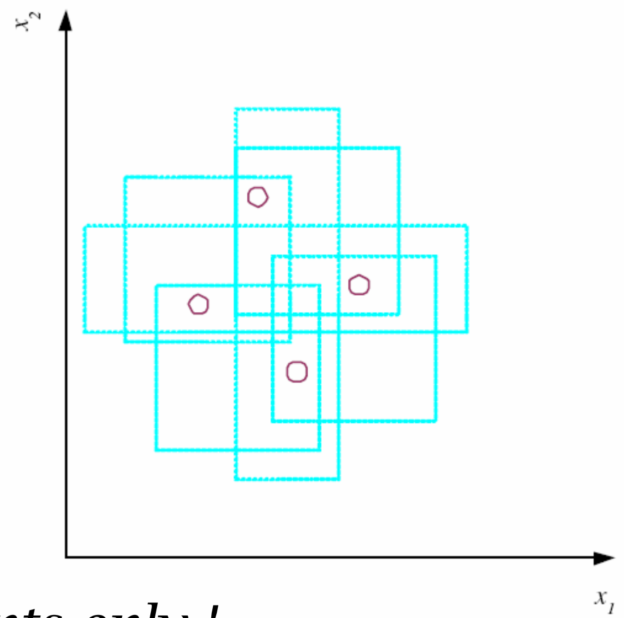

$h \in \mathcal{H}$, between $S$ and $G$ is consistent

and make up the version space

(Mitchell, 1997)

# VC Dimension

- $N$ points can be labeled in $2^N$ ways as +/–
- $\mathcal{H}$ shatters $N$ if there exists $h \in \mathcal{H}$ consistent for any of these:
  VC($\mathcal{H}$) = $N$

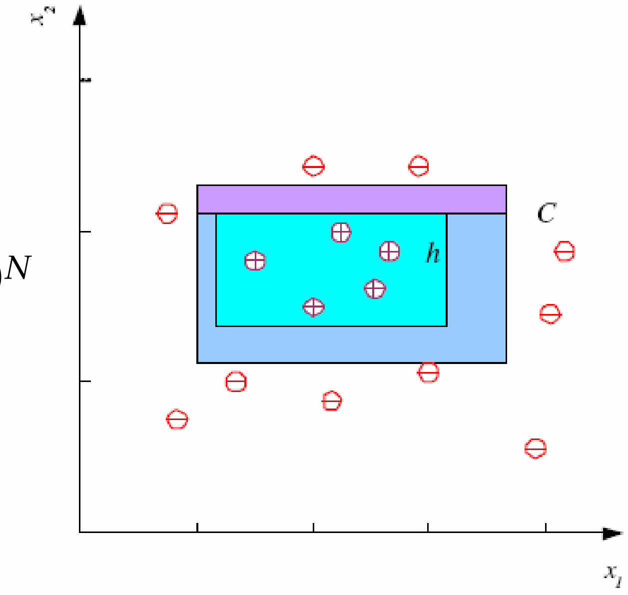



*An axis-aligned rectangle shatters 4 points only !*

# *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 $\varepsilon$ ?
  (Blumer et al., 1989)

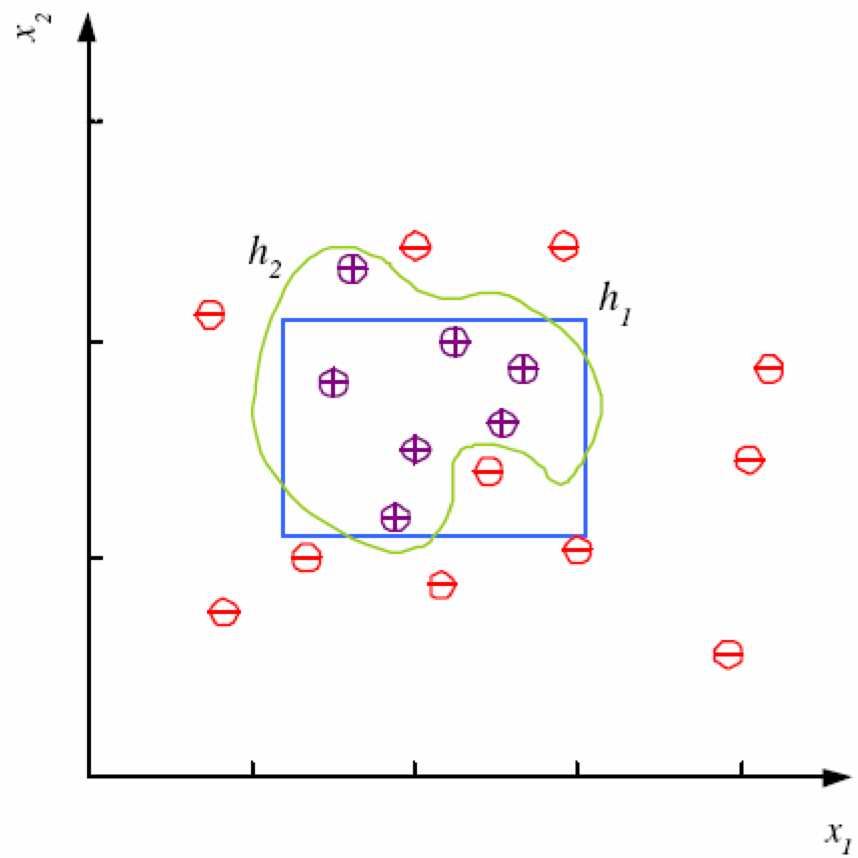
- Each strip is at most $\varepsilon/4$
- Pr that we miss a strip $1 - \varepsilon/4$
- Pr that $N$ instances miss a strip $(1 - \varepsilon/4)^N$
- Pr that $N$ instances miss 4 strips $4(1 - \varepsilon/4)^N$
- $4(1 - \varepsilon/4)^N \leq \delta$ and $(1 - x) \leq \exp(-x)$
- $4\exp(-\varepsilon N/4)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



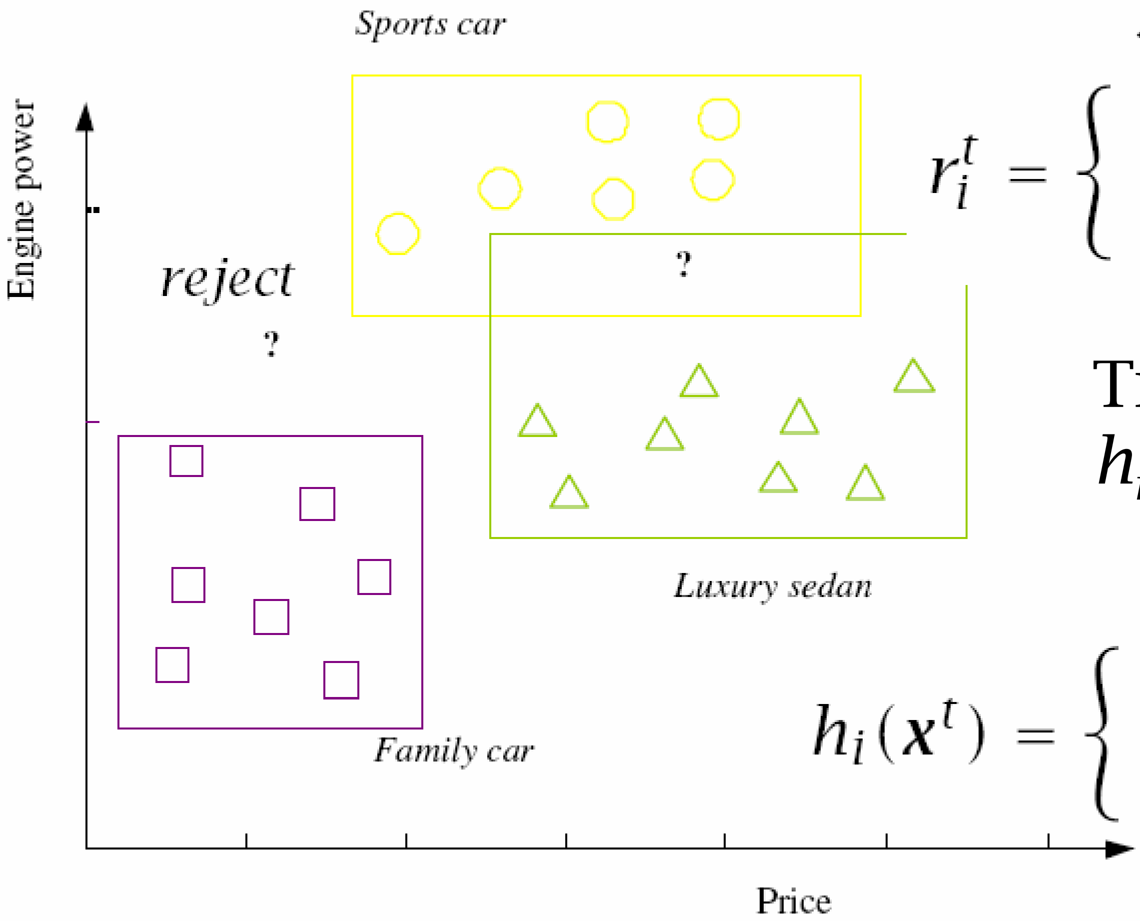
# *Noise and Model Complexity*

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, $C_i$ $i=1,...,K$*




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

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

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

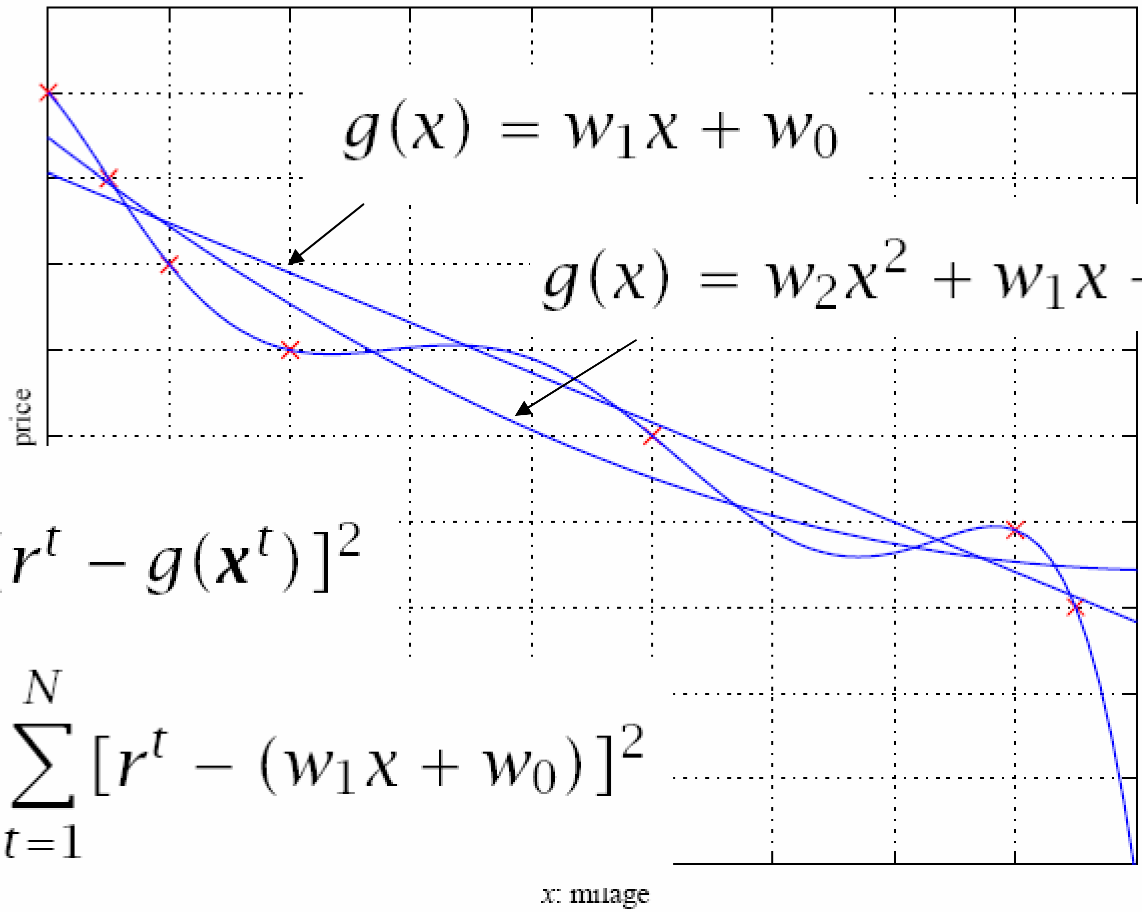
$$h_i(\boldsymbol{x}^t) = \begin{cases} 1 & \text{if } \boldsymbol{x}^t \in C_i \\ 0 & \text{if } \boldsymbol{x}^t \in C_j, j \neq i \end{cases}$$

# Regression

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

$$r^t \in \Re$$

$$r^t = f(\mathbf{x}^t) + \epsilon$$




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

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

# *Model Selection & Generalization*

- 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
- 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 $E\uparrow$

# *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 when there is few data

# *Dimensions of a Supervised Learner*

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

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

3. Optimization procedure:

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