Ch.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:

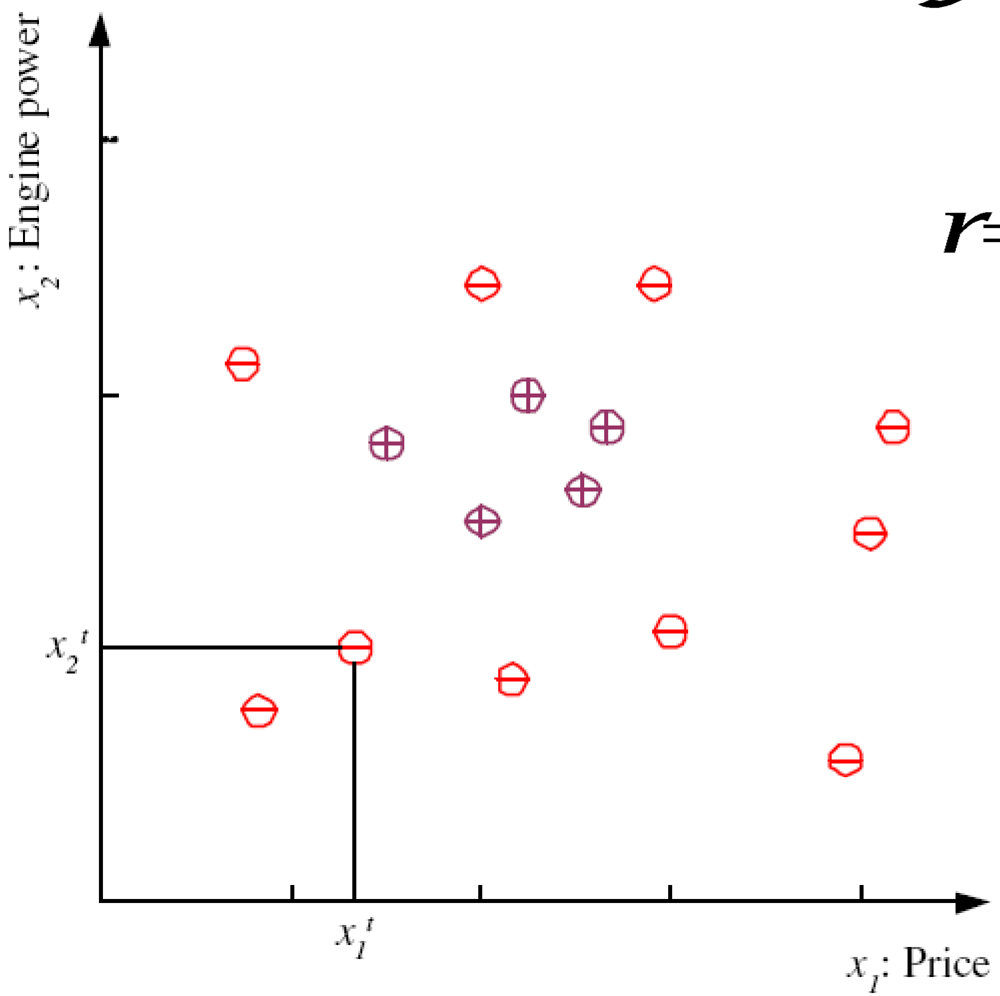
  x_1: price, x_2 : engine power

Training set $\mathcal{X}$

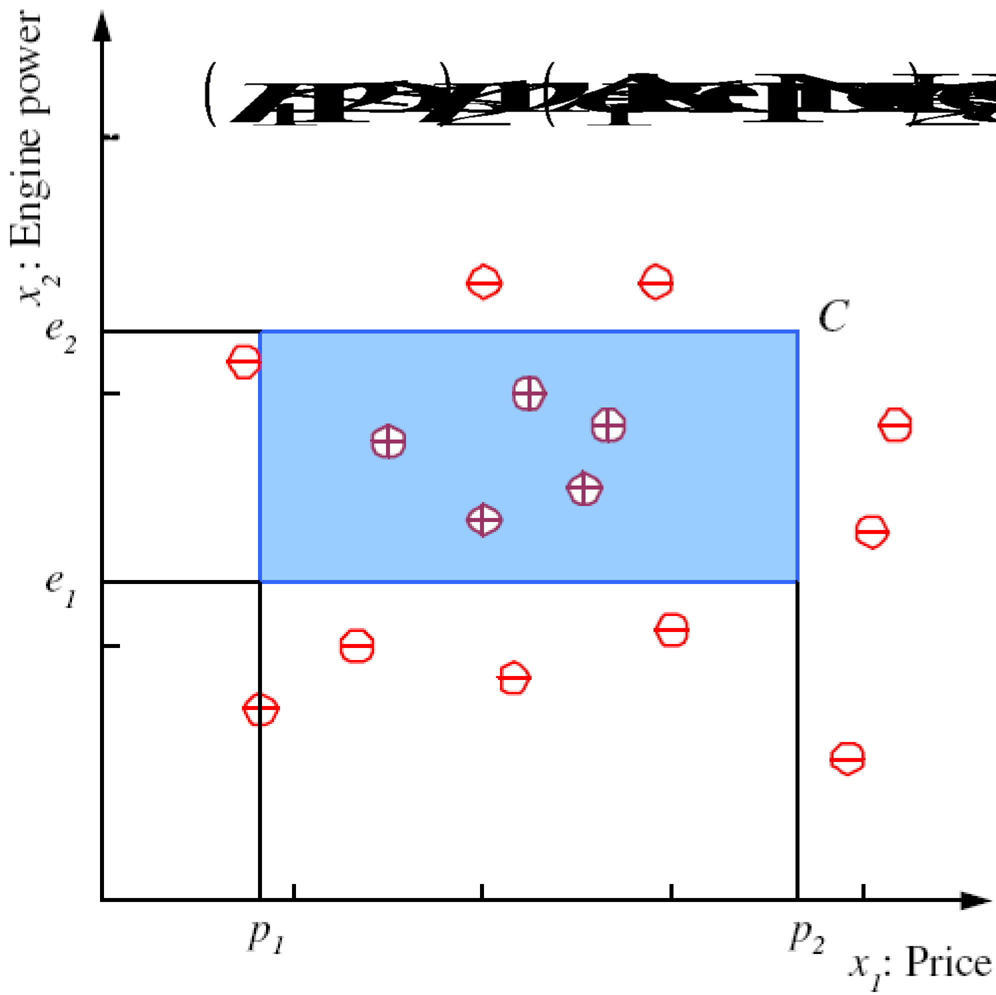
$$\mathcal{X} = \{\mathbf{x}^t, r^t\}_{t=1}^N$$

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

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

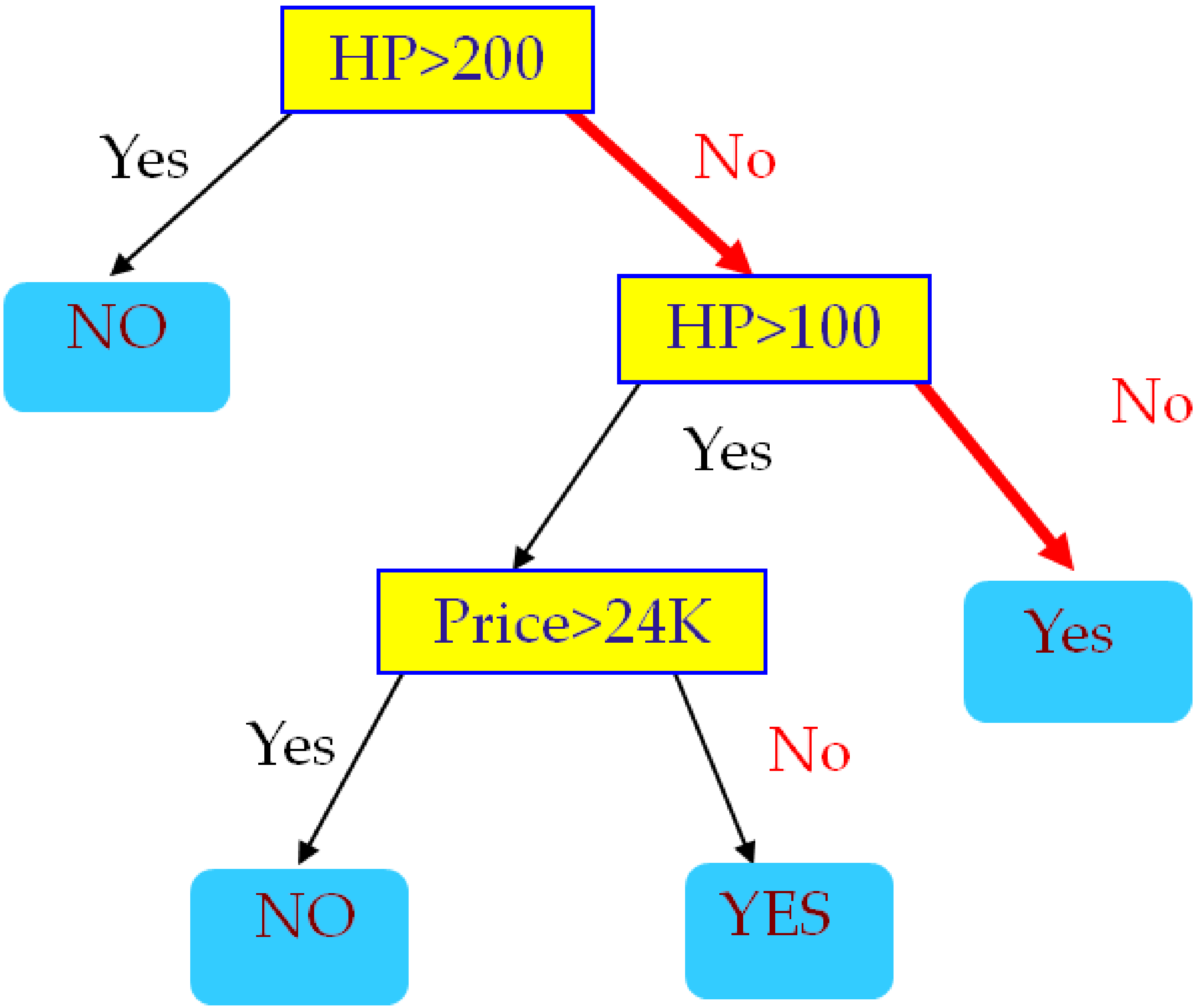


Class C



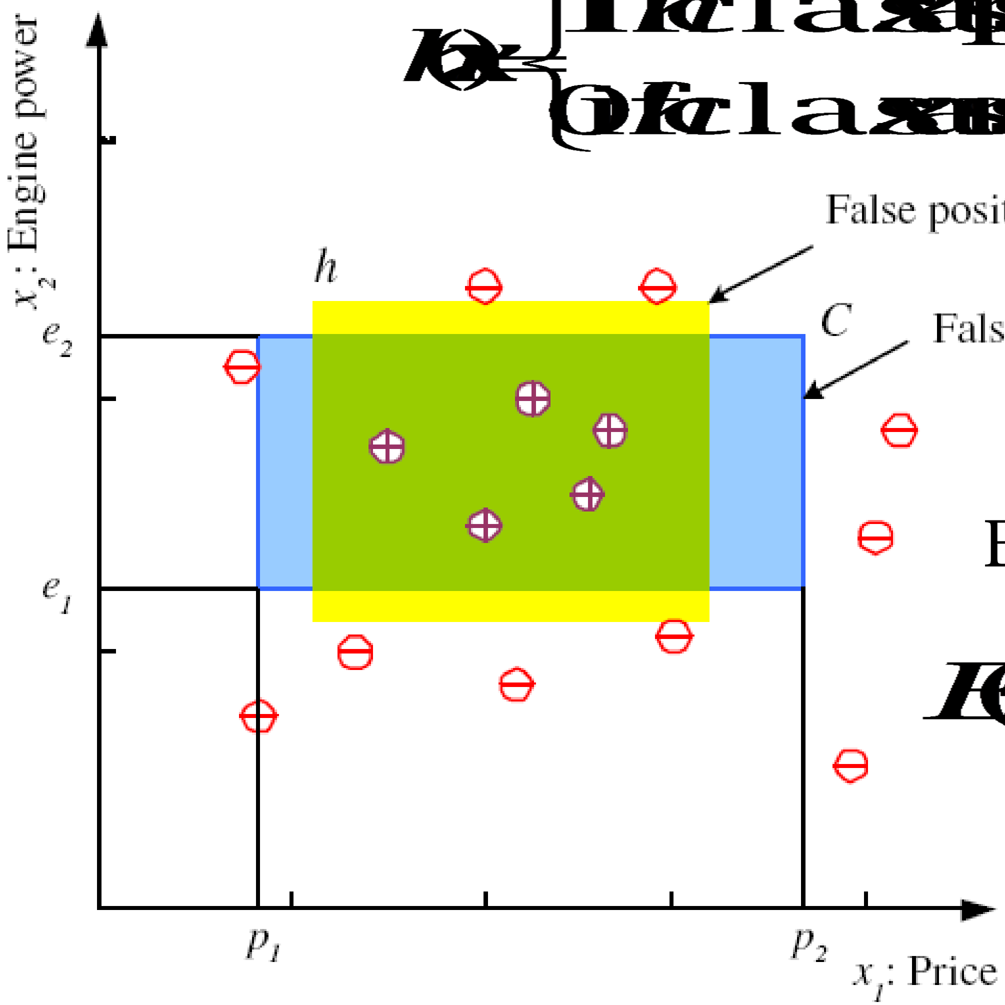
*

Family Car Decision Tree



Hypothesis class $\mathcal{H}$

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

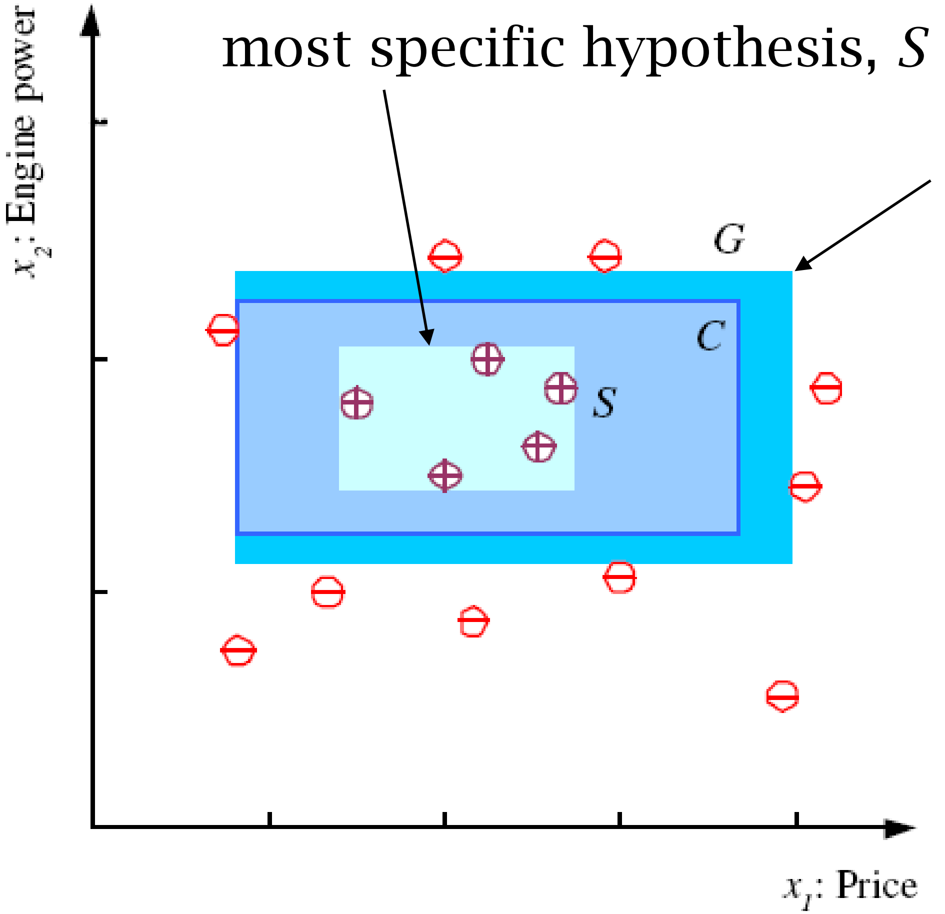


Error of h on $\mathcal{H}$

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

*

S, G, and the Version Space



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

and make up the version space

(Mitchell, 1997)

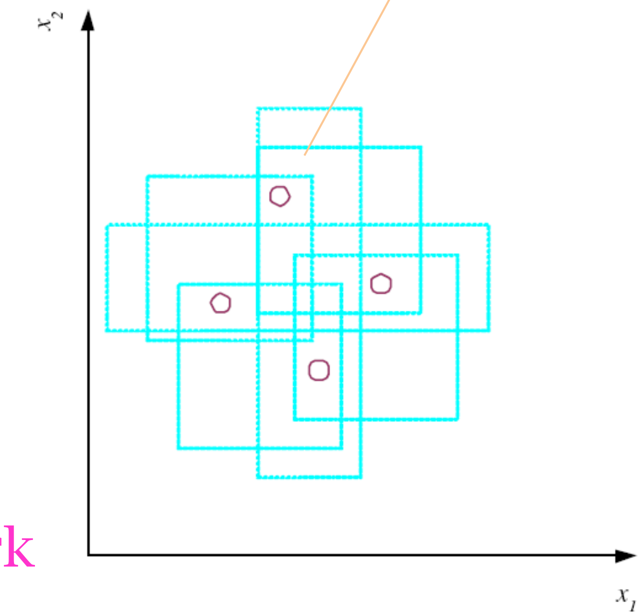
VC Dimension

Vapnik-Chervonenkis

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

rectangles here

Only 2 +/2- depicted



- *Does not work for 5 points!*

x x x x x

try it: Homework

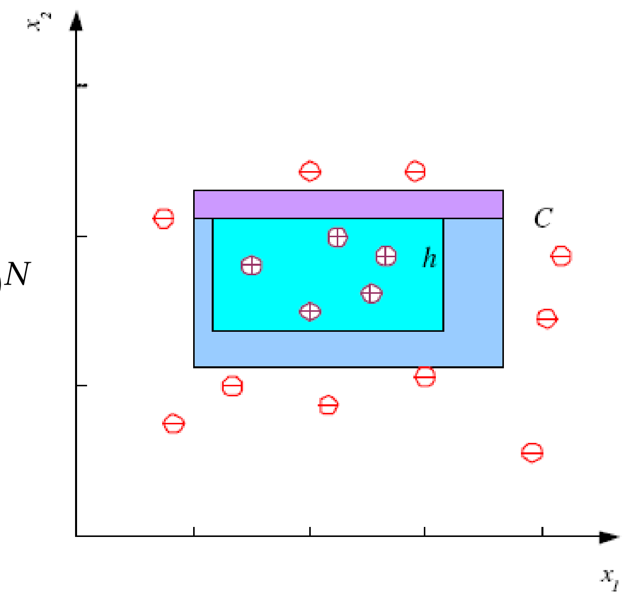
An axis-aligned rectangle shatters 4 points only !

Skipped!

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)

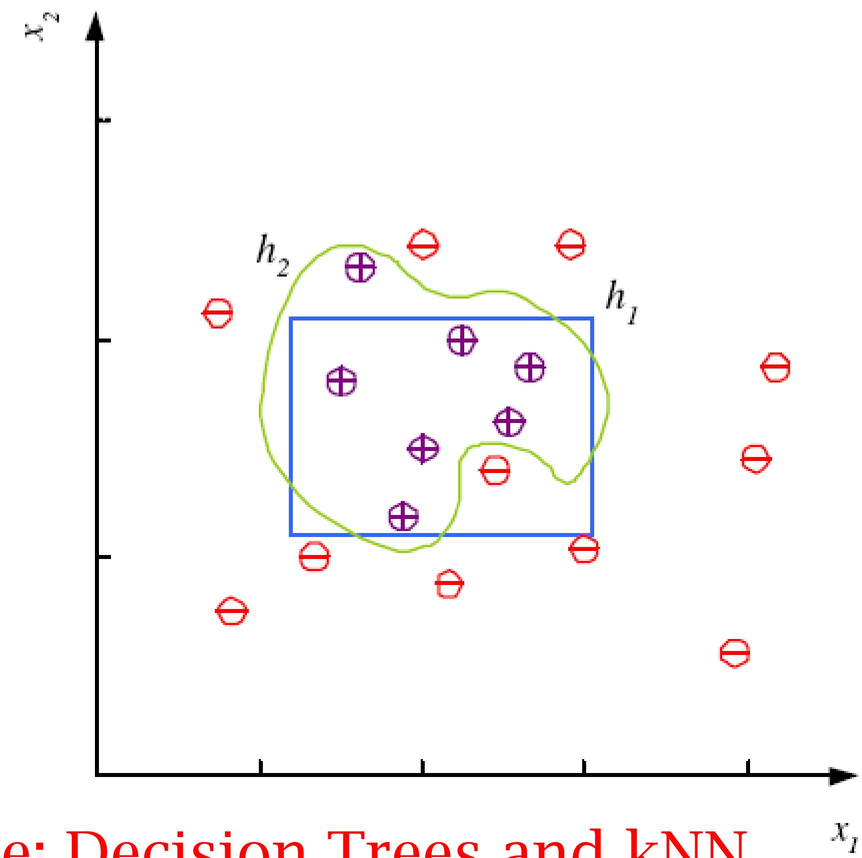
- Each strip is at most $\varepsilon/4$
- Pr that we miss a strip $1 - \varepsilon/4$
- Pr that N instances miss a strip $(1 - \varepsilon/4)^N$
- Pr that N instances miss 4 strips $4(1 - \varepsilon/4)^N$
- $4(1 - \varepsilon/4)^N \leq \delta$ and $(1 - x) \leq \exp(-x)$
- $4\exp(-\varepsilon N/4) \leq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



Noise and Model Complexity

Use the simpler one because

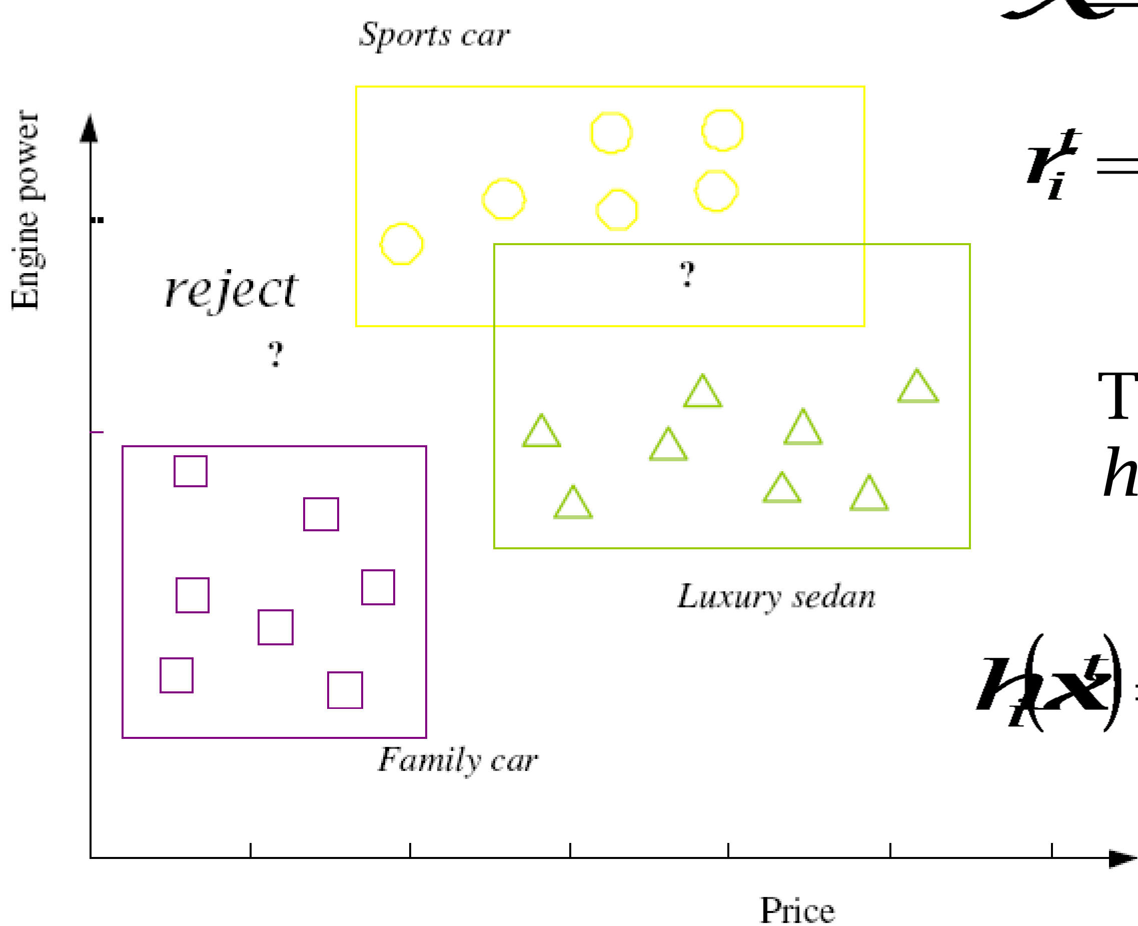
- Simpler to use (lower computational complexity)
- Easier to train (lower space complexity)
- Easier to explain (more interpretable)
- Generalizes better (lower variance - Occam's razor)



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

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

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



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

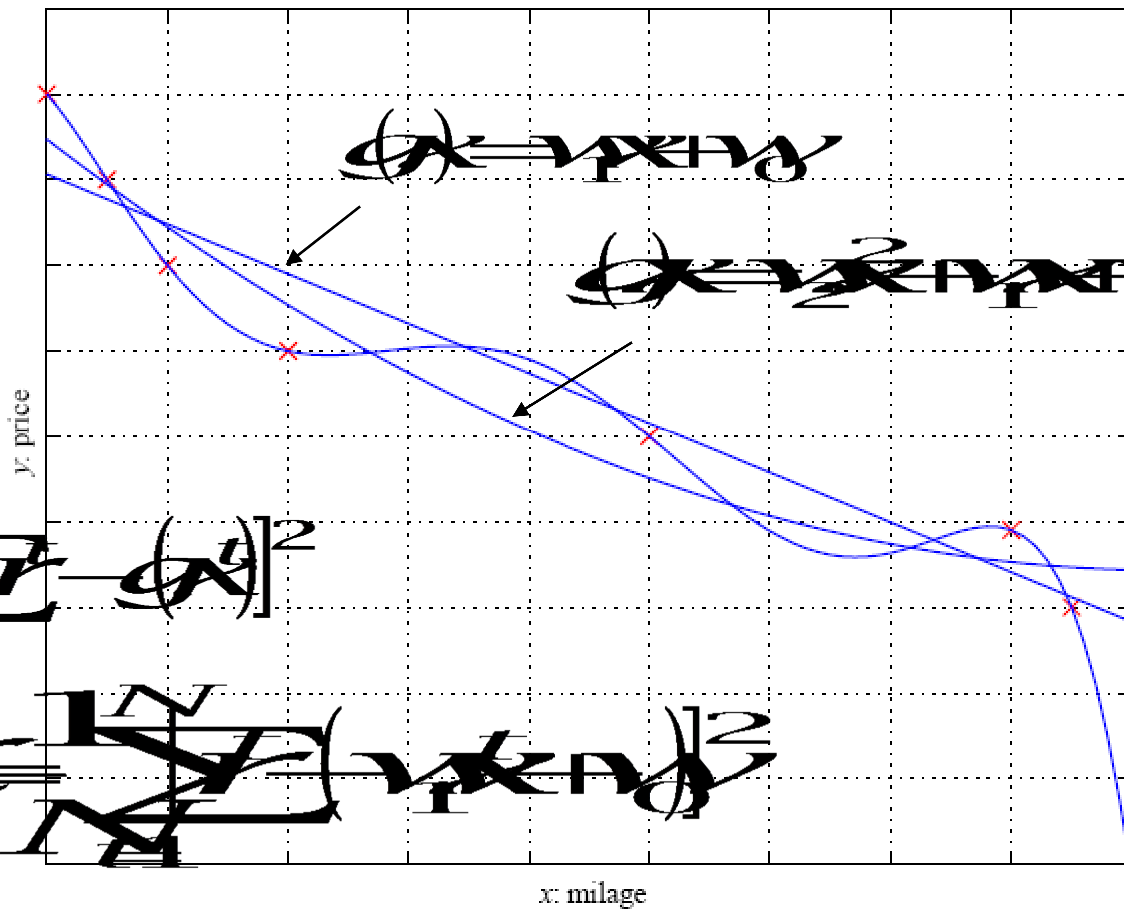
$$h_i(\mathbf{x}^t) = \begin{cases} 1 & \text{if } \mathbf{x}^t \in C_i \\ 0 & \text{if } \mathbf{x}^t \in C_j, j \neq i \end{cases}$$

Regression

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

$$r^t \in \Re$$

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



$$g(x) = w_1 x + w_0$$

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

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

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

Finding Regression Coefficients

$$g(x) = w_1 x + w_0$$

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

$$r^t \in \Re$$

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

How to find w_1 and w_0?

Solve: $dE/dw_1=0$ and $dE/dw_0=0$

And solve the two obtained equation

Ungraded Homework!

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

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

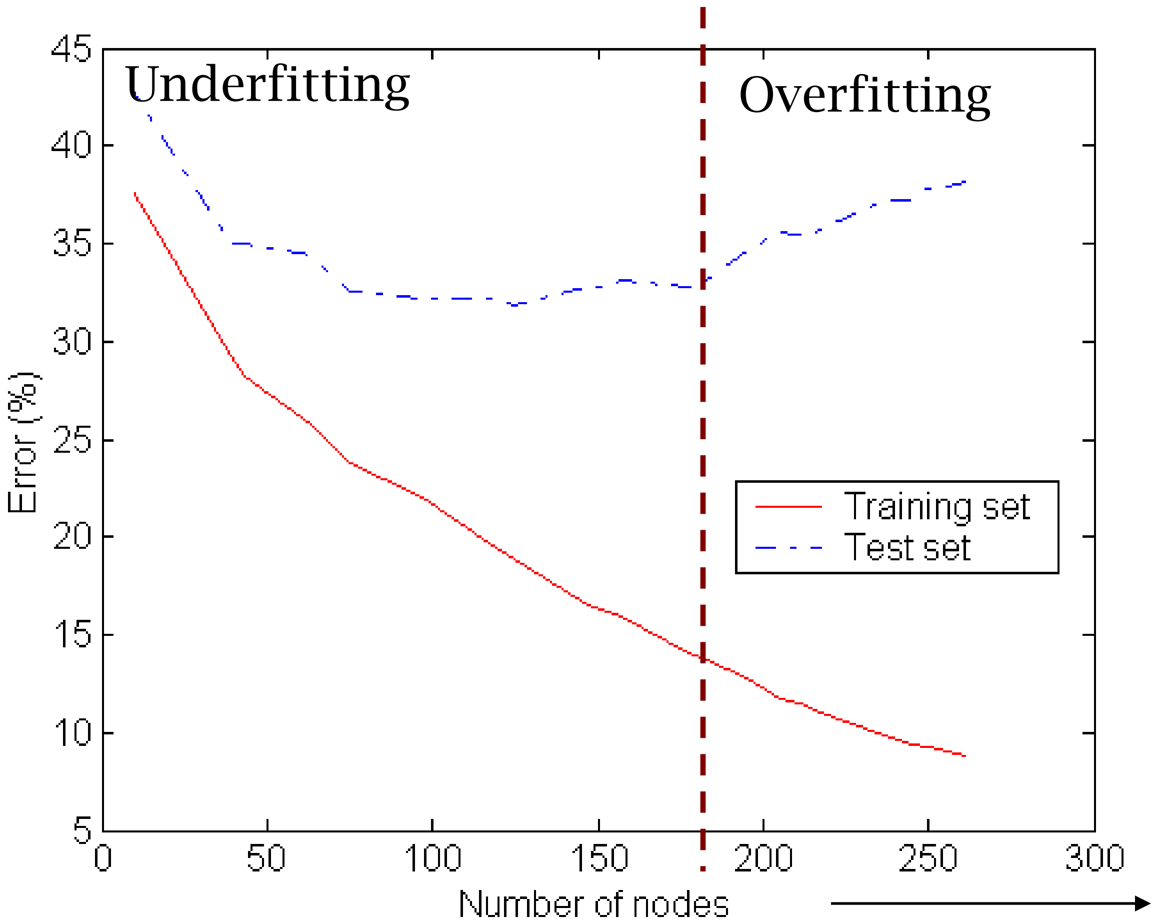
http://en.wikipedia.org/wiki/Simple_linear_regression

*

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

Underfitting and Overfitting



Complexity of the classification function

Complexity of a Decision Tree := number of nodes It uses

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.

Cross-Validation

Error on new examples; actually the testing error is used as an estimatio 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

*

Triple Trade-Off

overfitting

- 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 $E\uparrow$
- As $c(\mathcal{H})\uparrow$ the training error decreases for some time and then stays constant (frequently at 0)

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

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