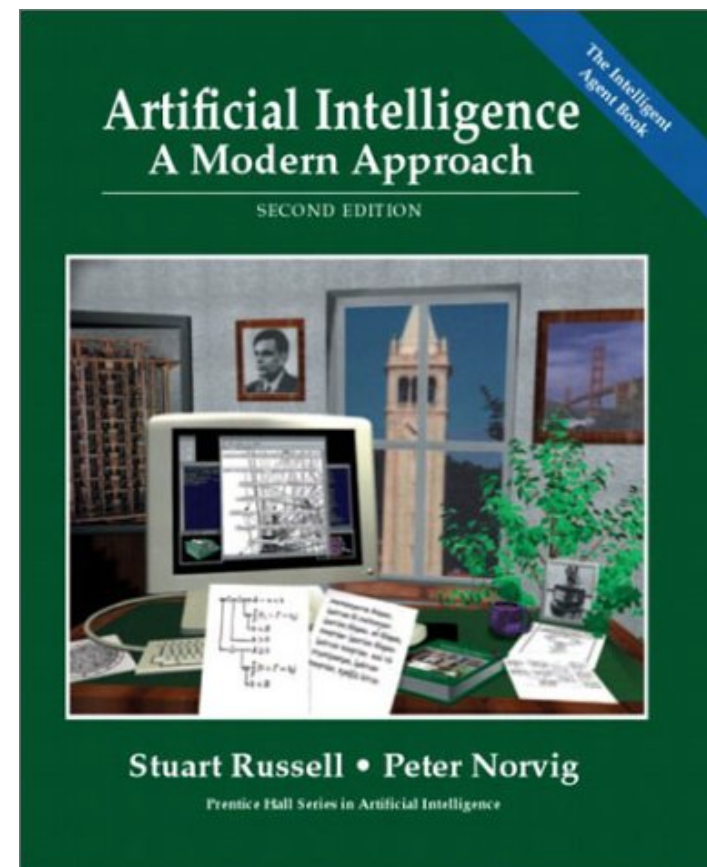
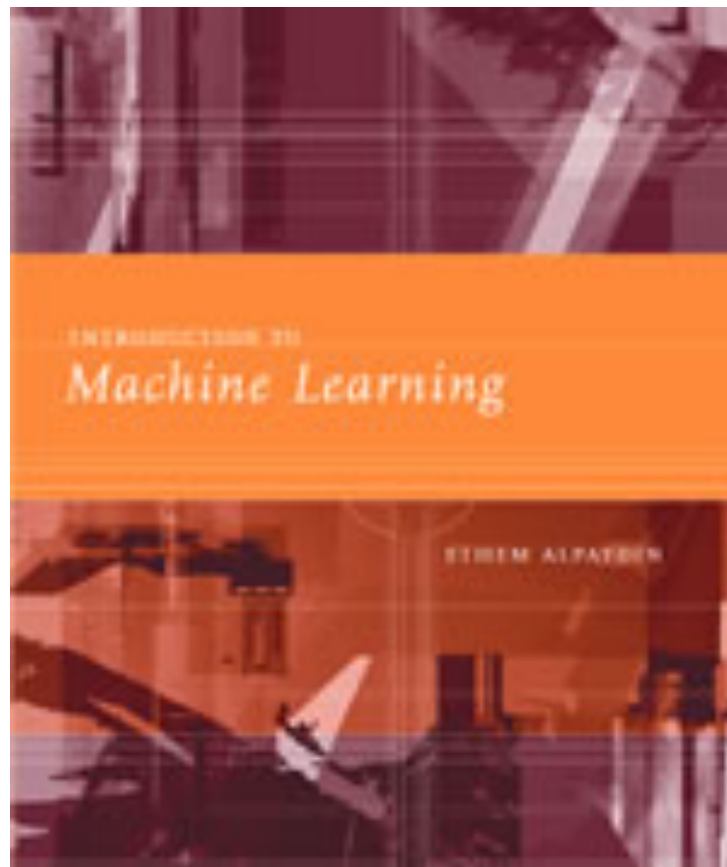


Foundations of Machine Learning and Data Mining

Introduction

Ralf Moeller
Hamburg Univ. of Technology

Literature



Why “Learn” ?

- Machine learning is programming computers to optimize a performance criterion using example data or past experience.
- There is no need to “learn” to calculate payroll
- Learning is used when:
 - ◆ Human expertise does not exist (navigating on Mars),
 - ◆ Humans are unable to explain their expertise (speech recognition)
 - ◆ Solution changes in time (routing on a computer network)
 - ◆ Solution needs to be adapted to particular cases (user biometrics)

What We Talk About When We Talk About “Learning”

- Learning general models from a data of particular examples
- Data is cheap and abundant (data warehouses, data marts); knowledge is expensive and scarce.
- Example in retail: Customer transactions to consumer behavior:
 - People who bought “Da Vinci Code” also bought “The Five People You Meet in Heaven” (www.amazon.com)*
- Build a model that is *a good and useful approximation* to the data.

Data Mining

- Retail: Market basket analysis, Customer relationship management (CRM)
- Finance: Credit scoring, fraud detection
- Manufacturing: Optimization, troubleshooting
- Medicine: Medical diagnosis
- Telecommunications: Quality of service optimization
- Bioinformatics: Motifs, alignment
- Web mining: Search engines
- ...

What is Machine Learning?

- Optimize a performance criterion using example data or past experience.
- Role of Statistics: Inference from a sample
- Role of Computer science: Efficient algorithms to
 - ◆ Solve the optimization problem
 - ◆ Representing and evaluating the model for inference

Applications

- Association
- Supervised Learning
 - ◆ Classification
 - ◆ Regression
- Unsupervised Learning
- Reinforcement Learning

Learning Associations

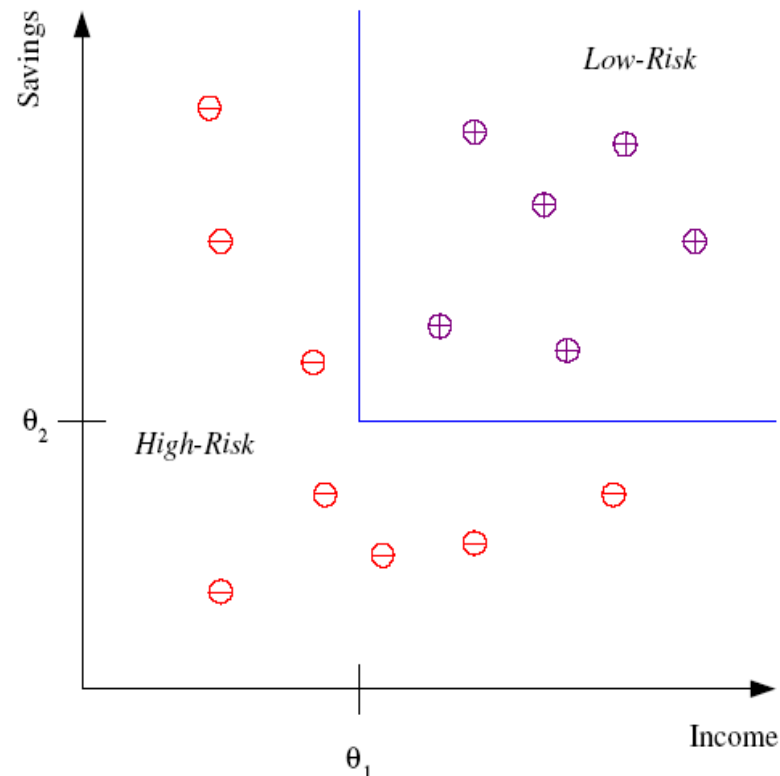
- Basket analysis:

$P(Y | X)$ probability that somebody who buys X also buys Y where X and Y are products/services.

Example: $P(\text{chips} | \text{beer}) = 0.7$

Classification

- Example: Credit scoring
- Differentiating between **low-risk** and **high-risk** customers from their *income* and *savings*



*Discriminant: IF income $> \theta_1$ AND savings $> \theta_2$
THEN **low-risk** ELSE **high-risk***

Classification: Applications

- Aka Pattern recognition
- Face recognition: Pose, lighting, occlusion (glasses, beard), make-up, hair style
- Character recognition: Different handwriting styles.
- Speech recognition: Temporal dependency.
 - ◆ Use of a dictionary or the syntax of the language.
 - ◆ Sensor fusion: Combine multiple modalities; eg, visual (lip image) and acoustic for speech
- Medical diagnosis: From symptoms to illnesses
- ...

Face Recognition

Training examples of a person

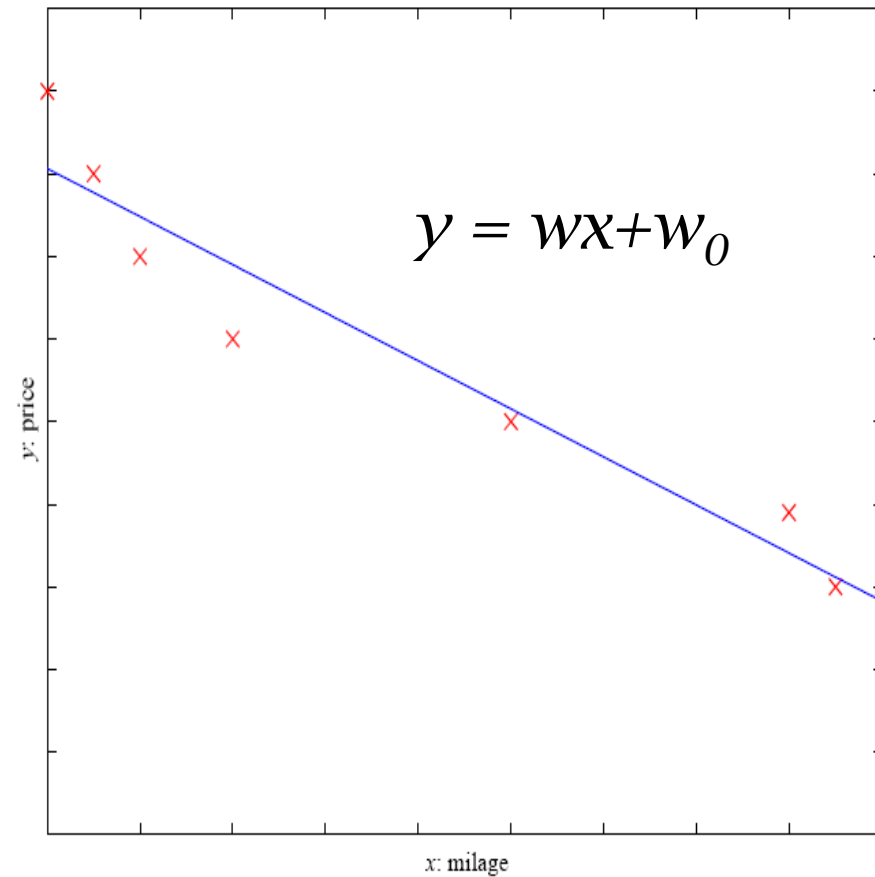


Test images



Regression

- Example: Price of a used car
- x : car attributes
- y : price
- $y = g(x | \theta)$
- $g()$ model,
- θ parameters



Supervised Learning: Uses

- Prediction of future cases: Use the rule to predict the output for future inputs
- Knowledge extraction: The rule is easy to understand
- Compression: The rule is simpler than the data it explains
- Outlier detection: Exceptions that are not covered by the rule, e.g., fraud

Unsupervised Learning

- Learning “what normally happens”
- No output
- Clustering: Grouping similar instances
- Example applications
 - ◆ Customer segmentation in CRM
 - ◆ Image compression: Color quantization
 - ◆ Bioinformatics: Learning motifs

Reinforcement Learning

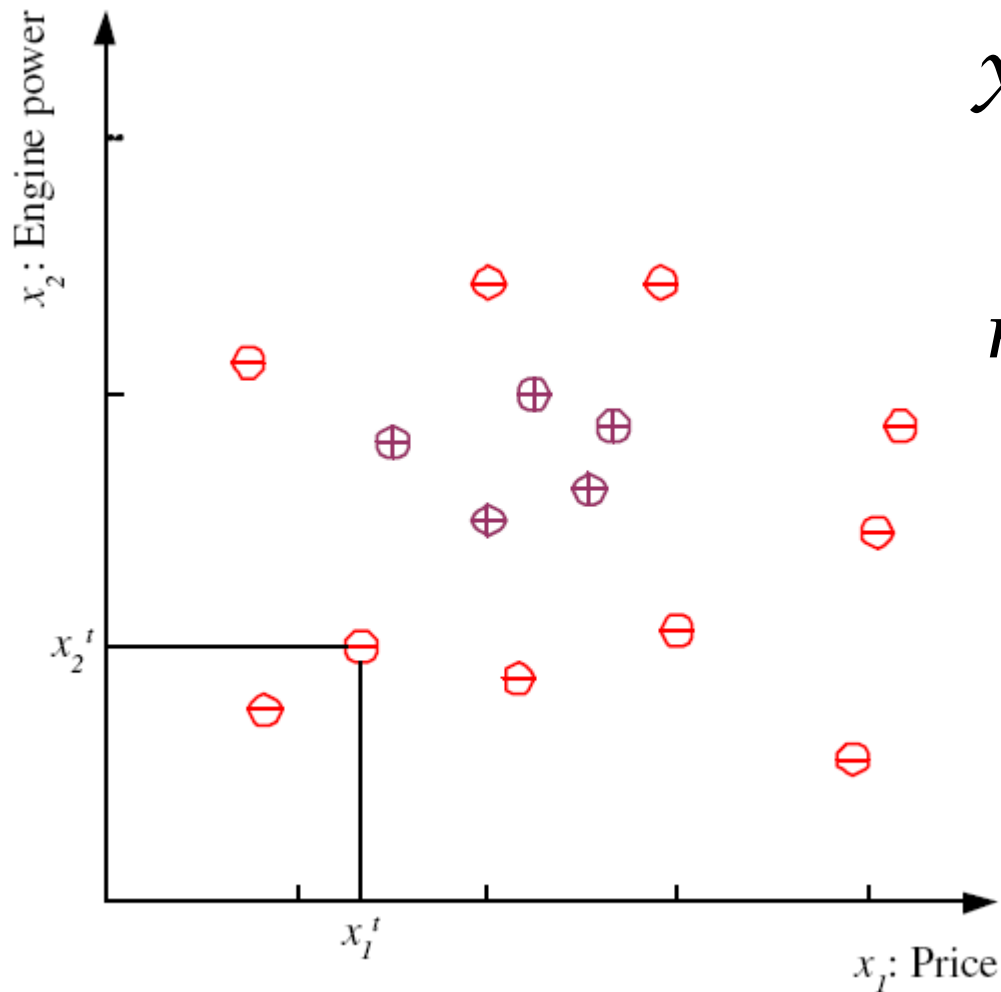
- Learning a policy: A sequence of outputs
- No supervised output but delayed reward
- Credit assignment problem
- Game playing
- Robot in a maze
- Multiple agents, partial observability, ...

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}



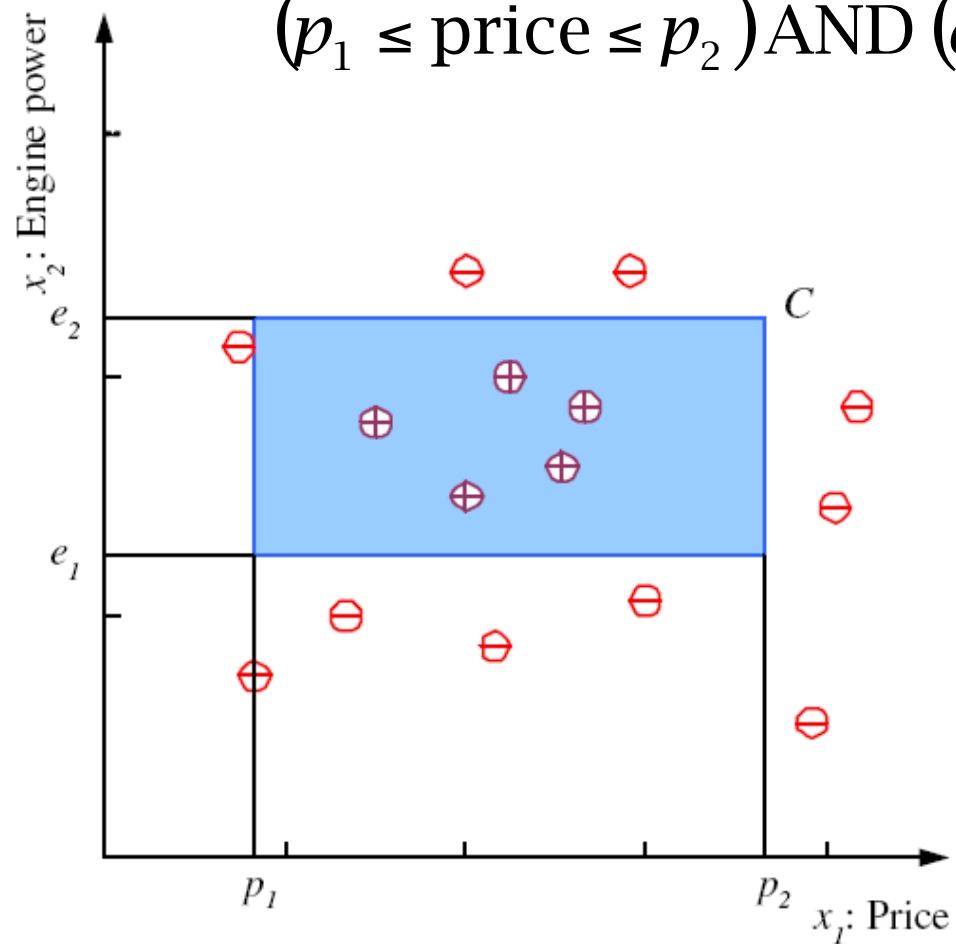
$$\mathcal{X} = \{\mathbf{x}^t, r^t\}_{t=1}^N$$

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

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

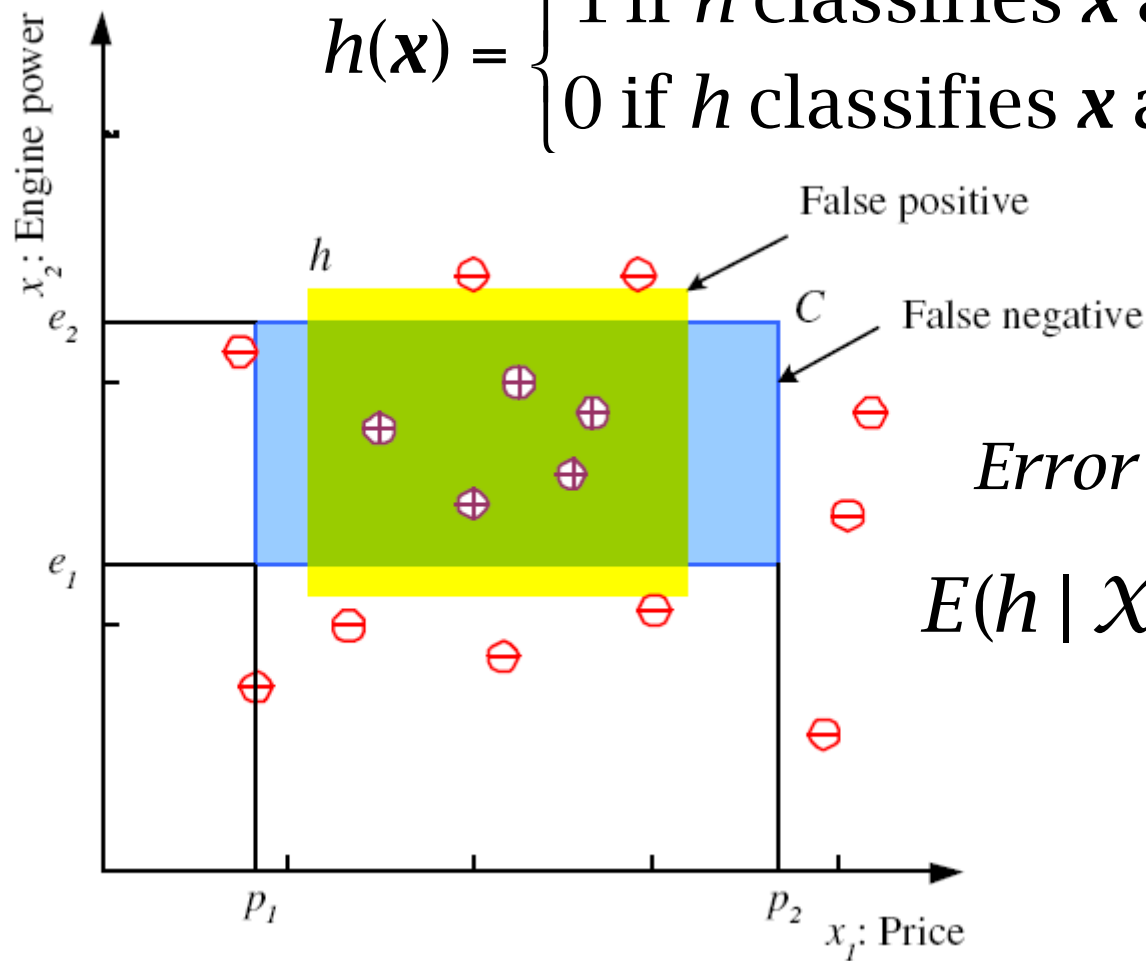
Class C

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



Hypothesis class \mathcal{H}

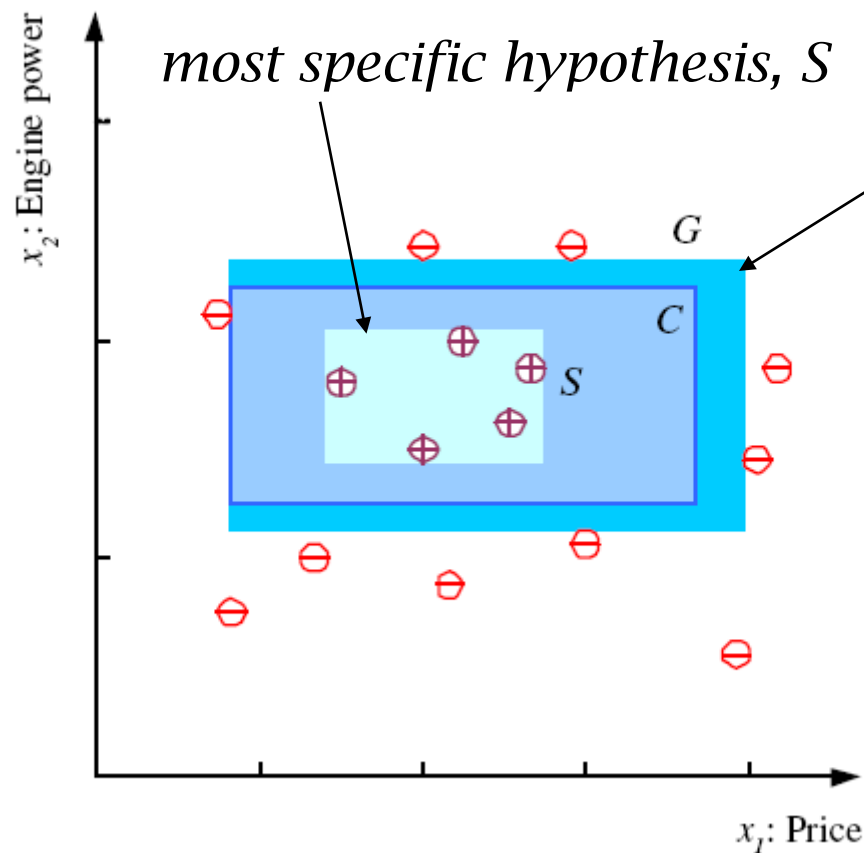
$$h(\mathbf{x}) = \begin{cases} 1 & \text{if } h \text{ classifies } \mathbf{x} \text{ as positive} \\ 0 & \text{if } h \text{ classifies } \mathbf{x} \text{ as negative} \end{cases}$$



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

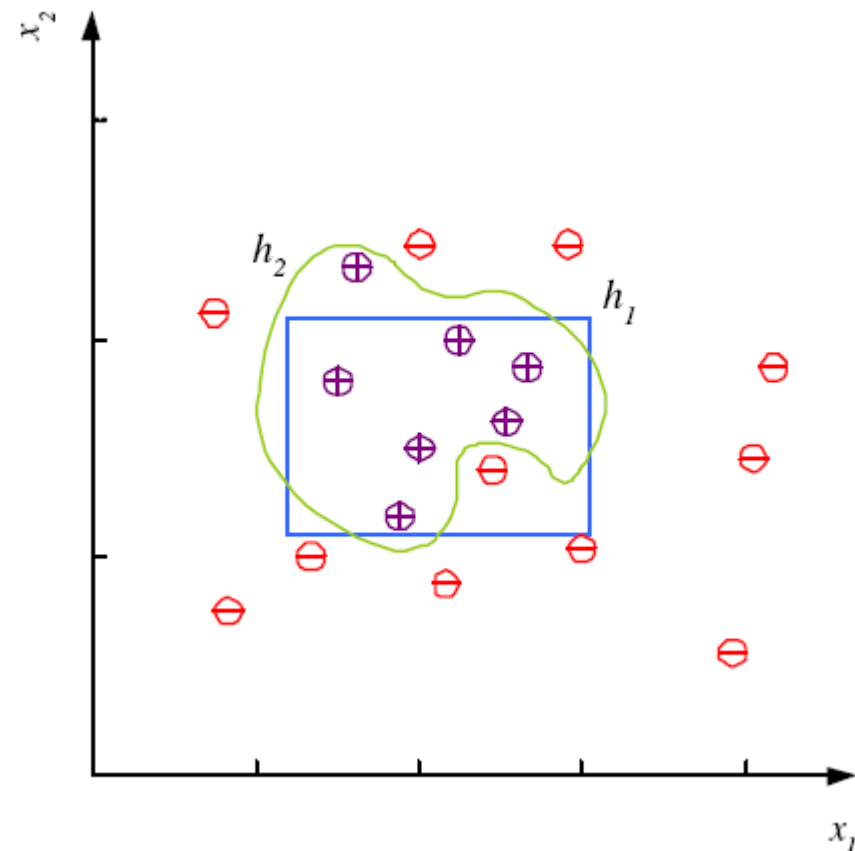
and make up the version space

(Mitchell, 1997)

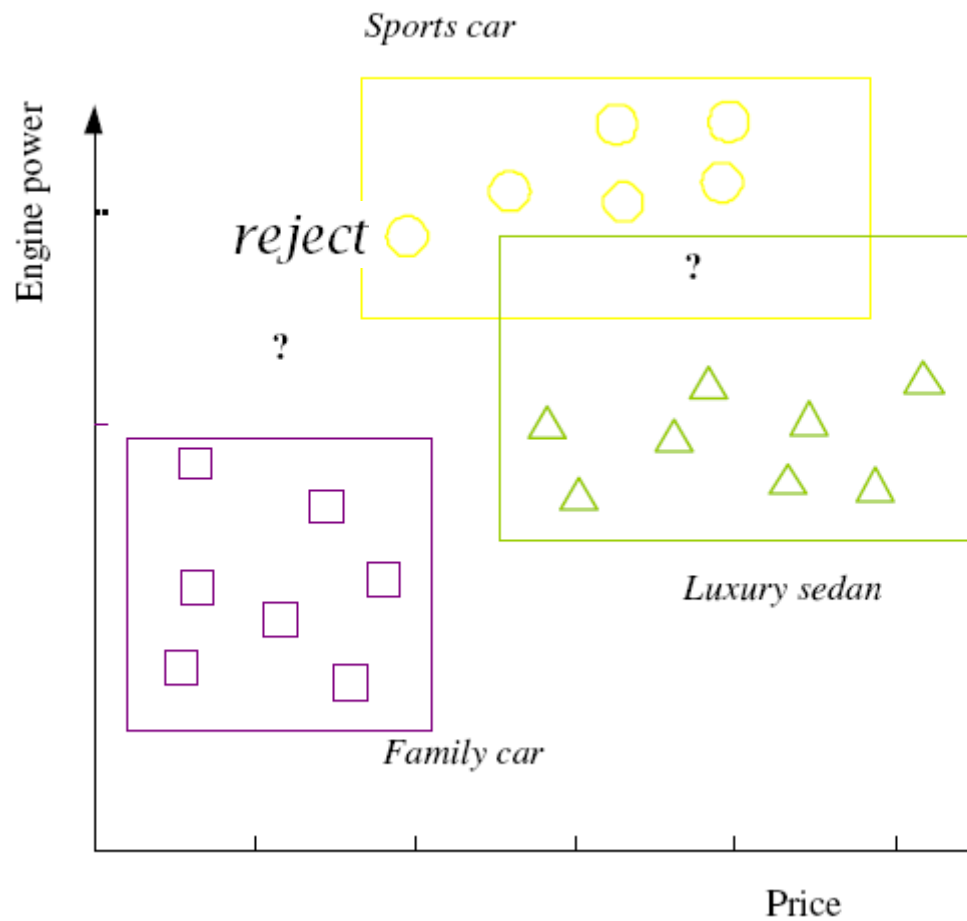
Noise and Model Complexity

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 $i=1,\dots,K$



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

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

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

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

Regression

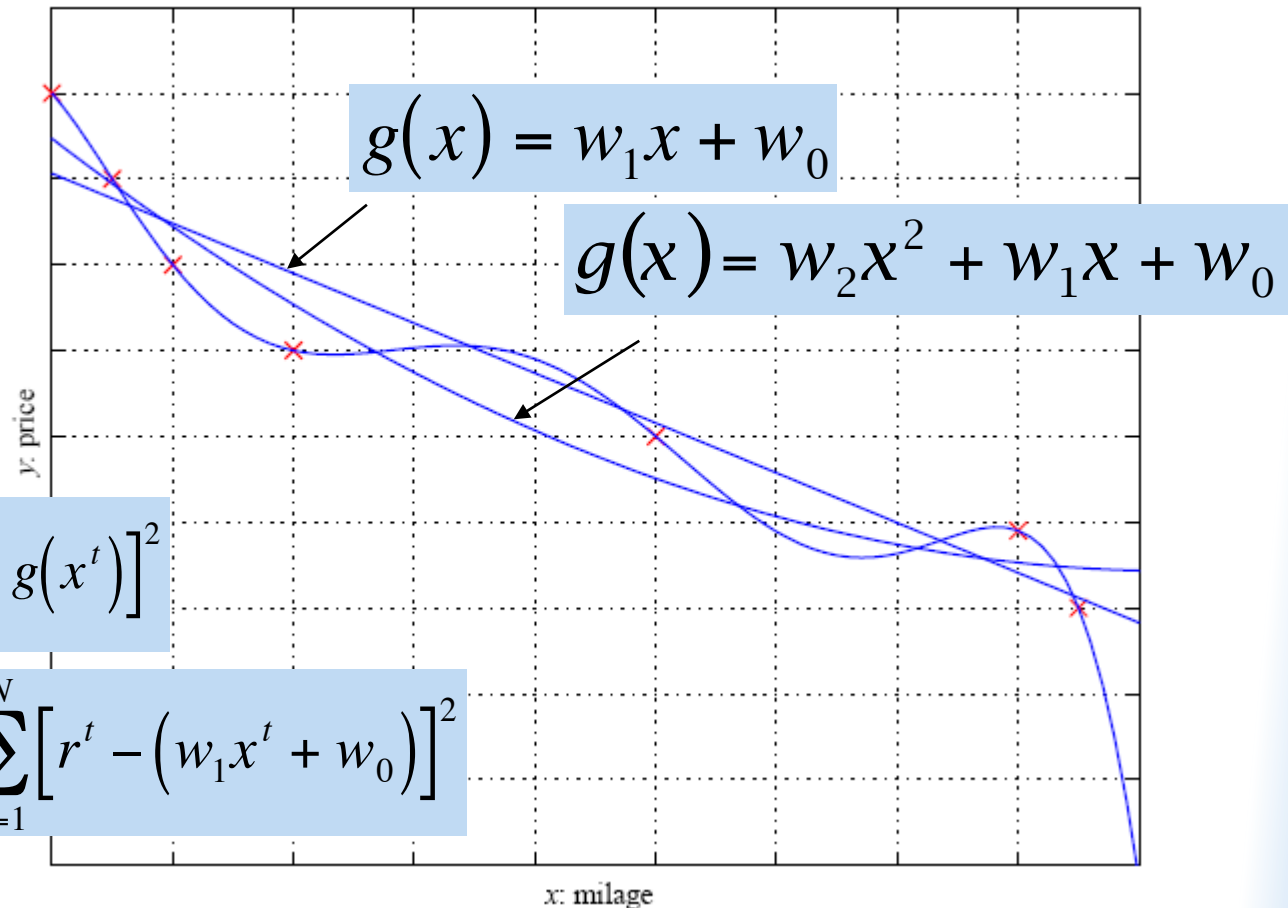
$$\mathcal{X} = \left\{ x^t, r^t \right\}_{t=1}^N$$

$$r^t \in \mathfrak{R}$$

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

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

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



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):
 - ♦ Complexity of \mathcal{H} , $c(\mathcal{H})$,
 - ♦ Training set size, N ,
 - ♦ 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