CHAPTER 2:

SUPERVİSED LEARNİNG

Learning a Class from Examples

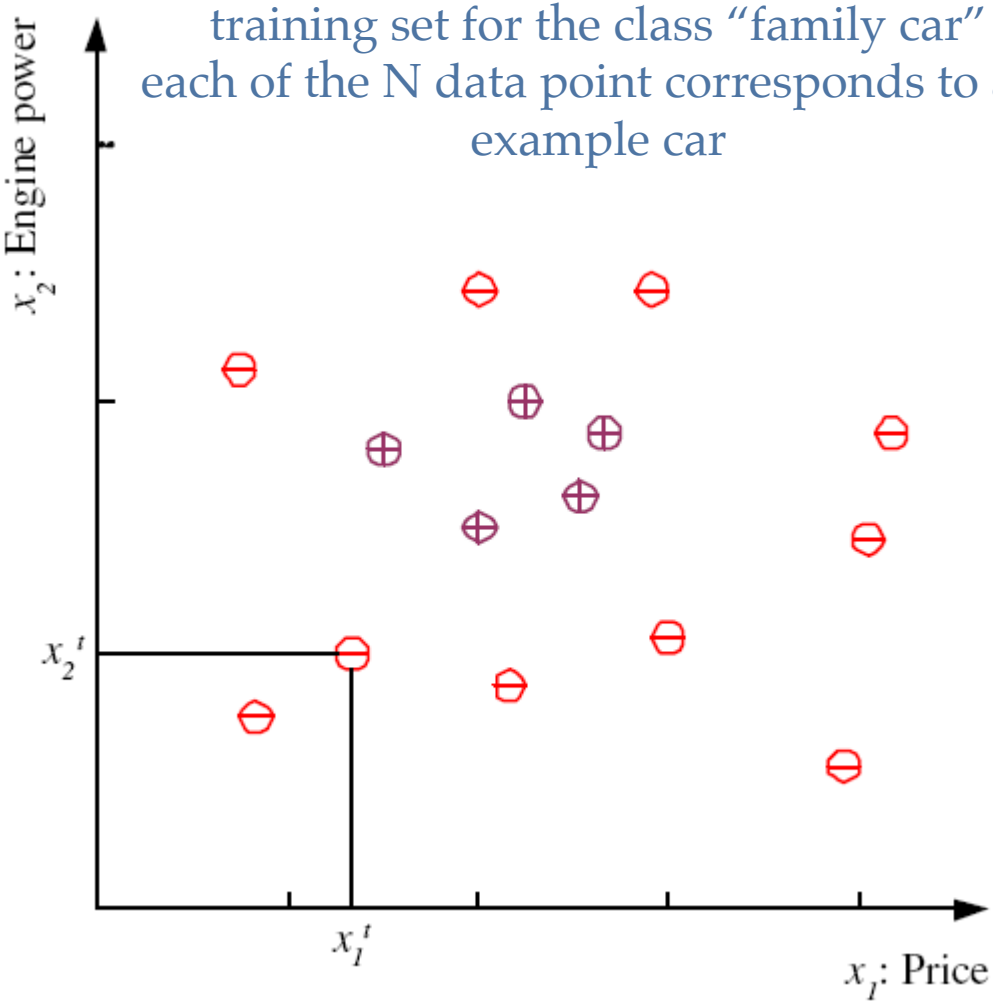
- Class C of a "family car"
  - Prediction: Is car x a family car?
  - Knowledge extraction: What do people expect from a family car?
- Output:

  Positive (+) "yes, it's a family car" and

  negative (–) "no, it's not a family car" examples
- Input representation: information we have about each car

  x_1: price, x_2 : engine power

Training set X



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

each car is represented as:

$$\{\mathbf{x}^t, r^t\}$$

where:

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

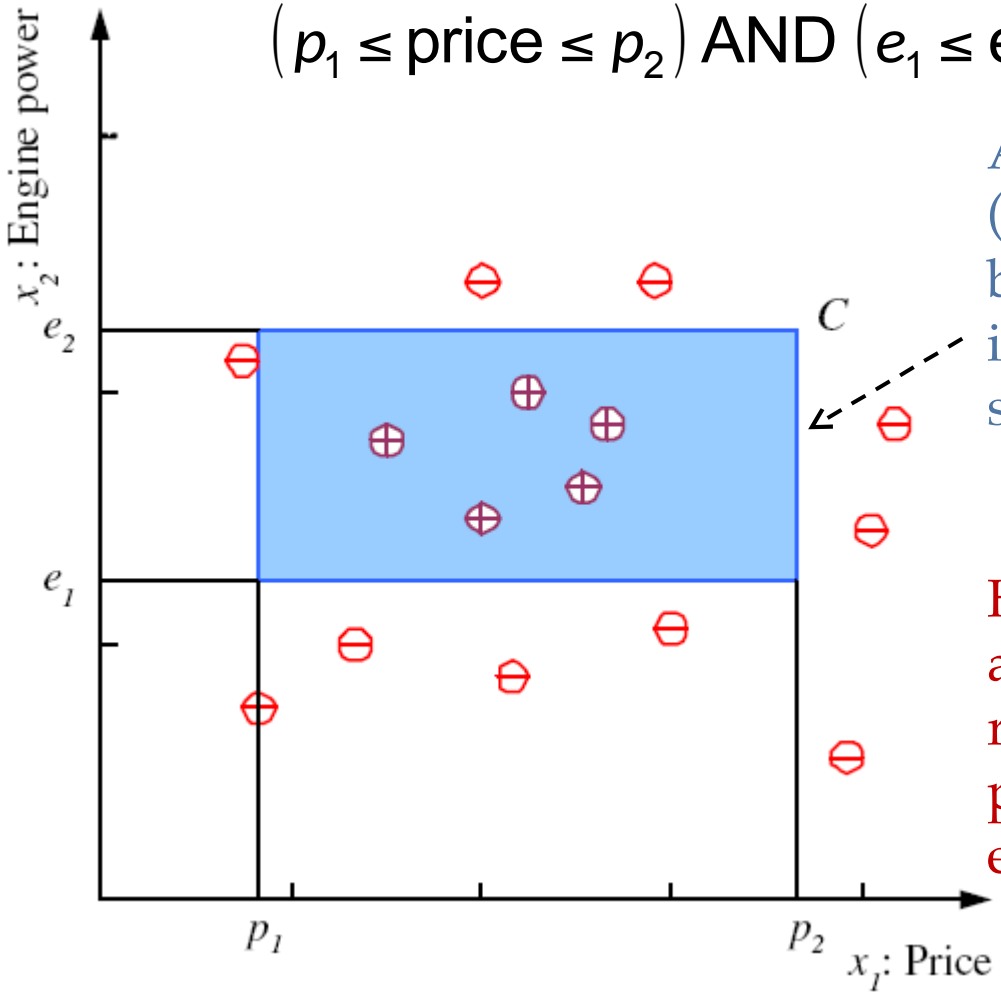
r is the example's "class"

also called "label",
"desired class" or "actual class"

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

Class C:

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



Assume that the actual Class ("family car") is characterized by this blue rectangle, defined in the price and engine power space

But that rectangle is not explicitly given to us!

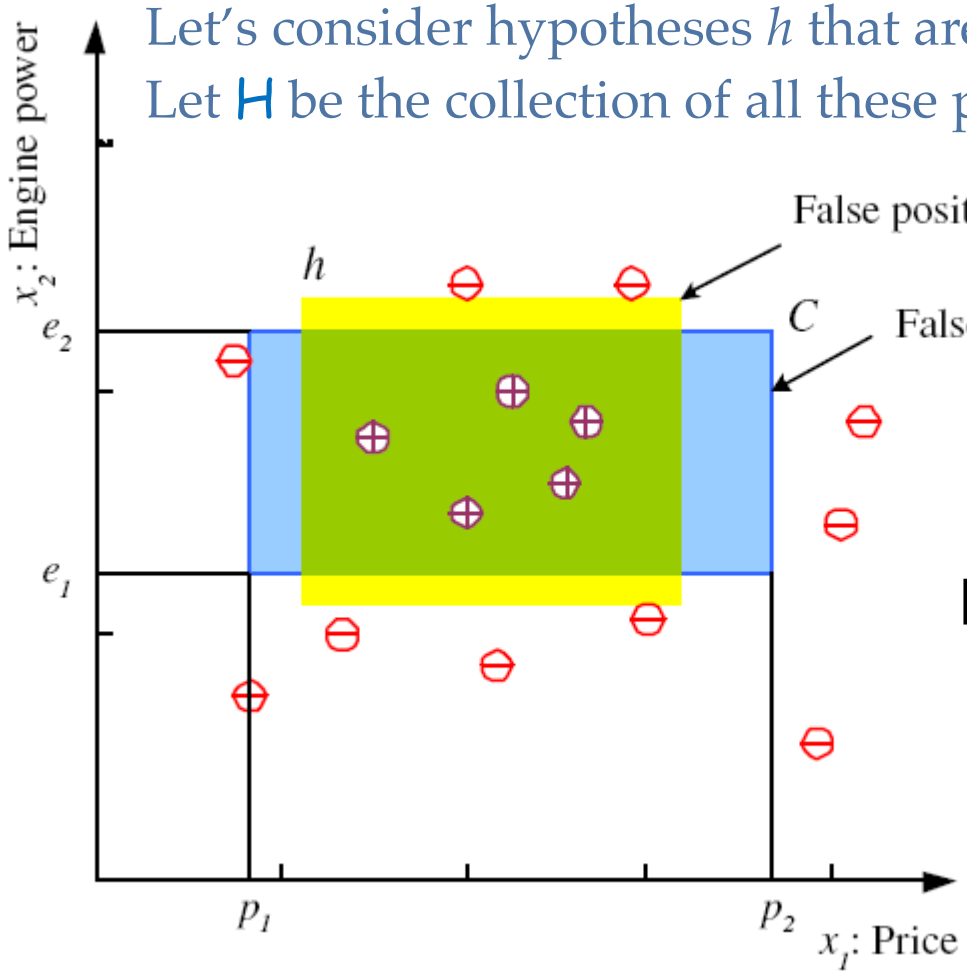
How to learn a good approximation to this rectangle based just on the positive and negative examples given in data set X?

We create hypotheses!

Hypothesis ~~class~~ space H

Let's consider hypotheses h that are rectangles in the x_1 , x_2 space

Let H be the collection of all these possible rectangles



Take for example h to be the yellow rectangle

$$h(\mathbf{x}) = \begin{cases} 1, \text{if } \mathbf{x} \text{ is inside of } h \\ 0, \text{if } \mathbf{x} \text{ is outside of } h \end{cases}$$

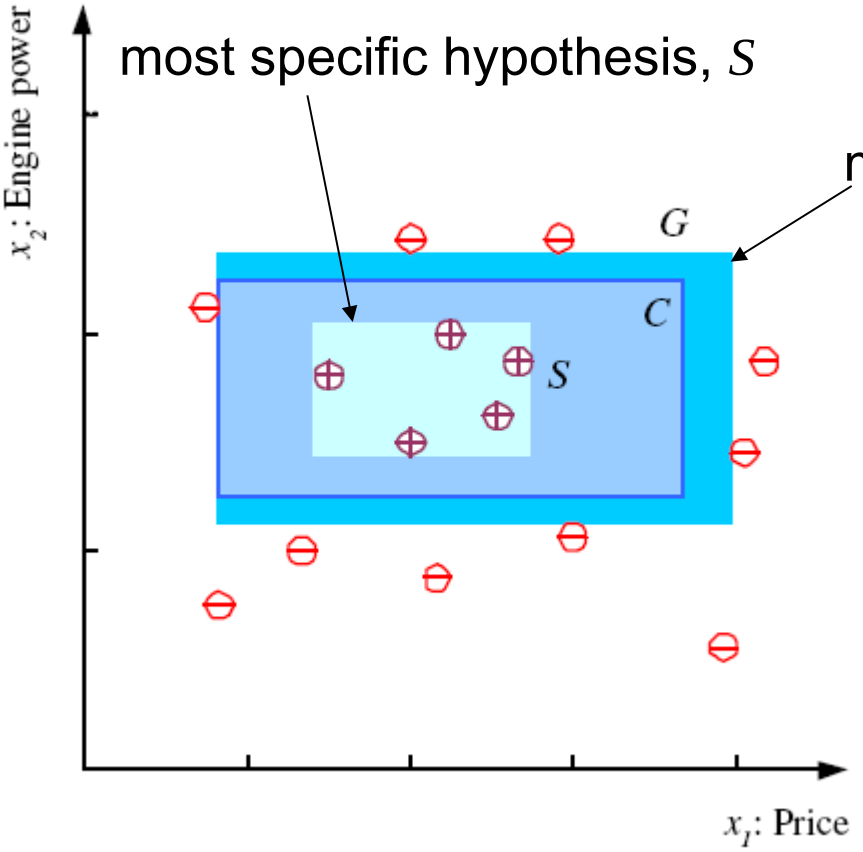
Error of h on H

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

where

$$1(u) = \begin{cases} 1, \text{if } u = \text{true} \\ 0, \text{if } u = \text{false} \end{cases}$$

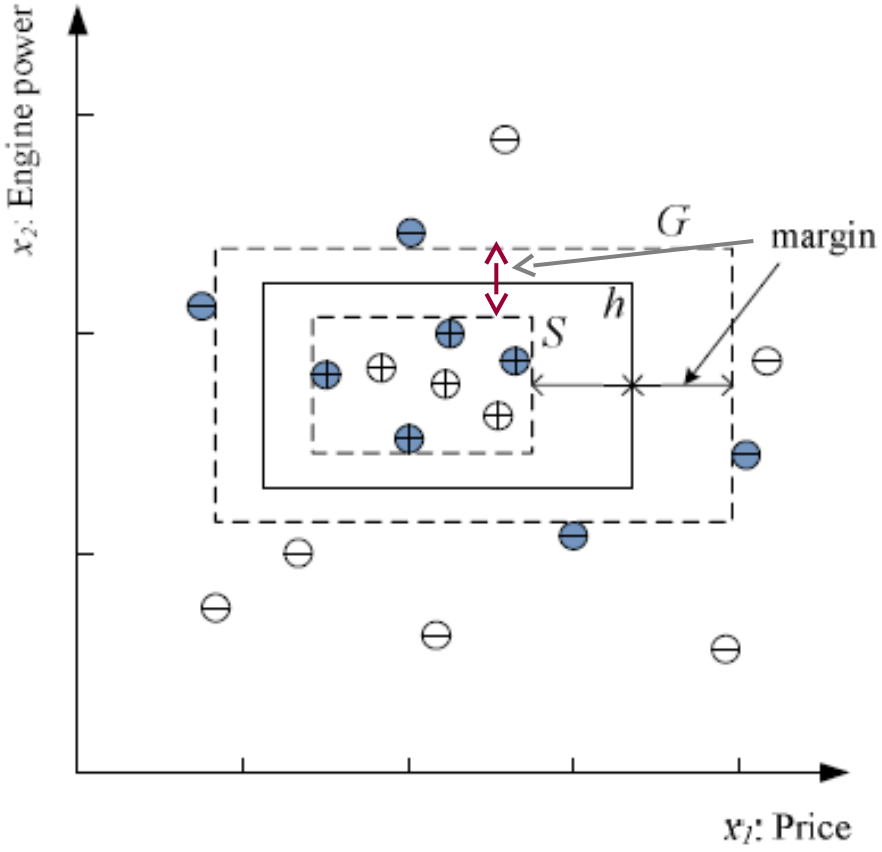
S, G, and the Version Space



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

Margin

- Choose h with largest margin



Why the largest margin?

because we want to minimize the generalization error.

What's "generalization"?

Generalization: How well a model (i.e., hypothesis) performs on new data

Vapnik-Chernonenkis (VC) Dimension

- N points can be labeled in 2^N ways as +/–
- $\mathcal{H}$ shatters N if and only if:
  - there is a set of N points in 2D such that for each of the 2^N possible ways of labelling these N points, there exists $h \in \mathcal{H}$ that is consistent with this labelling (i.e., h correctly separates the + from the – points)
- VC($\mathcal{H}$) = maximum N that can be shattered by $\mathcal{H}$

measures the "capacity" of $\mathcal{H}$

see example on the next slide

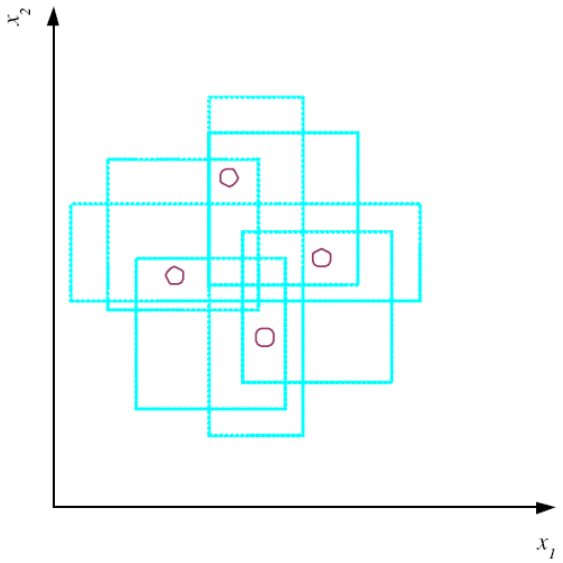
VC Dimension: Example

- Let H be the set of 2D axis-aligned rectangles
- VC(H) = ? (value)
- Remember:
  - VC(H) = maximum N that can be shattered by H
  - H shatters N if and only if:
    - there is a set of N points in 2D such that for each of the 2^N possible ways of labelling these N points, there exists $h \in H$ that is consistent with this labelling (i.e., correctly separates the + from the – points)

(cont.)

VC Dimension: Example (cont.)

- Let H be the set of 2D axis-aligned rectangles
- $VC(H) = 4$



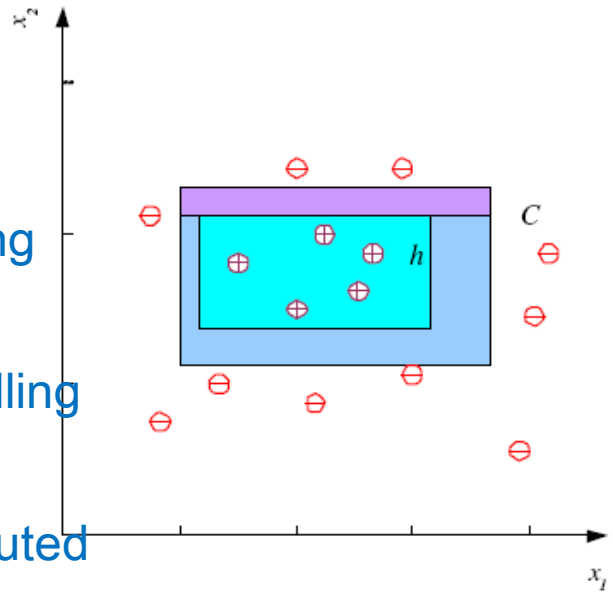
The family of axis-aligned rectangles shatters 4 points only !

Probably Approximately Correct (PAC) Learning

- If we are to use the tightest rectangle h as our chosen hypothesis, how many training examples do we need so that we can guarantee that our hypothesis is approximately correct? In other words:
- Given δ (1-δ is the desired minimum confidence) and ε (desired maximum error): How many training examples, N, should we have such that with probability at least 1–δ, h has error at most ε ?

  that is, $\Pr(E(h|X) \leq \varepsilon) \geq 1 - \delta$

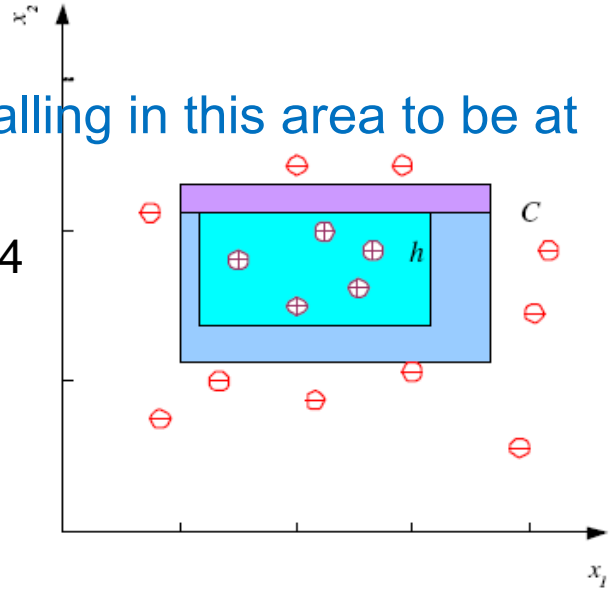
(Blumer et al., 1989)



- E(h|X) would come from a positive example falling in the area between C and h
- We need the probability of a positive example falling in this area to be at most ε
- Assume that data instances are uniformly distributed in space and are independent from each other

Probably Approximately Correct (PAC) Learning (cont.)

- Given δ (1-δ is the desired minimum confidence) and ε (desired maximum error): How many training examples, N, should we have such that with probability at least 1–δ, the tightest rectangle h has error at most ε ? Pr(E(h|X) ≤ ε) ≥ 1 - δ
- E(h|X) would come from a positive example falling in the area between C and h
- We need the Probab. of a positive example falling in this area to be at most ε
- Pr of an instance falling in a strip should at most ε/4
- Pr that the data instance misses a strip is1– ε/4
- Pr that N instances miss a strip $(1 – ε/4)^N$
- Pr that N instances miss 4 strips $4(1 – ε/4)^N$
- Hence we need: $4(1 – ε/4)^N ≤ δ$



(cont.)

Probably Approximately Correct (PAC) Learning (cont.)

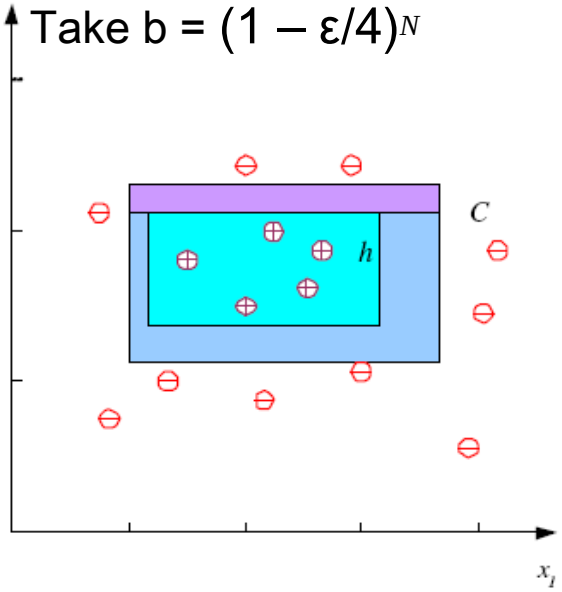
- Given δ (1-δ is the desired minimum confidence) and ε (desired maximum error): How many training examples, N, should we have such that with probability at least 1–δ, the tightest rectangle h has error at most ε ? $\Pr(E(h|X) \le \varepsilon) \ge 1 - \delta$
- …. we need: $4(1 - \varepsilon/4)^N \le \delta$

Note that $b = e^{\ln(b)} = \exp(\ln(b))$ and that $(1 - x) \le e^{-x}$

Take $b = (1 - \varepsilon/4)^N$

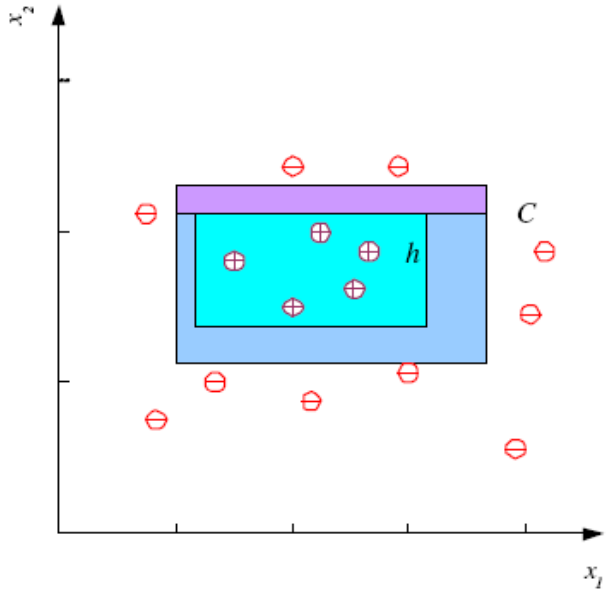
- $4(1 - \varepsilon/4)^N \le \delta$

  $4 \exp(\ln[(1 - \varepsilon/4)^N]) \le \delta$
- $4 \exp(N \ln(1 - \varepsilon/4)) \le \delta$, now using $(1 - x) \le \exp(-x)$:
- $4 \exp(N \ln(1 - \varepsilon/4)) \le 4 \exp(N \ln(\exp(-\varepsilon/4)))$
- Let's make $4 \exp(N \ln(\exp(-\varepsilon/4))) = 4 \exp(N (-\varepsilon/4)) \le \delta$
- $4/\delta \le \exp(\varepsilon N/4)$
- $\ln(4/\delta) \le \ln(\exp(\varepsilon N/4)) = \varepsilon N/4$
- and so $N \ge (4/\varepsilon)\ln(4/\delta)$



Probably Approximately Correct (PAC) Learning (cont.)

- Given δ (1-δ is the desired minimum confidence) and ε (desired maximum error): How many training examples N should we have, such that with probability at least 1–δ, the tightest rectangle h has error at most ε ? Pr(E(h|X) ≤ ε) ≥ 1 - δ
- Answer: $N \geq (4/\varepsilon)\ln(4/\delta)$
- Example: How many training examples, N, do we need so that Pr(E(h|X) ≤ 0.1) ≥ 95%?
  - Here ε = 0.1 and 1 – δ = 0.95 and so δ = 0.05
  - $N \geq (4/\varepsilon)\ln(4/\delta) = (4/0.1)\ln(4/0.05)$
  - $N \geq 40 \ln(80) = 175.28$
  - Hence our training set X should contain at least N = 176 data instances.



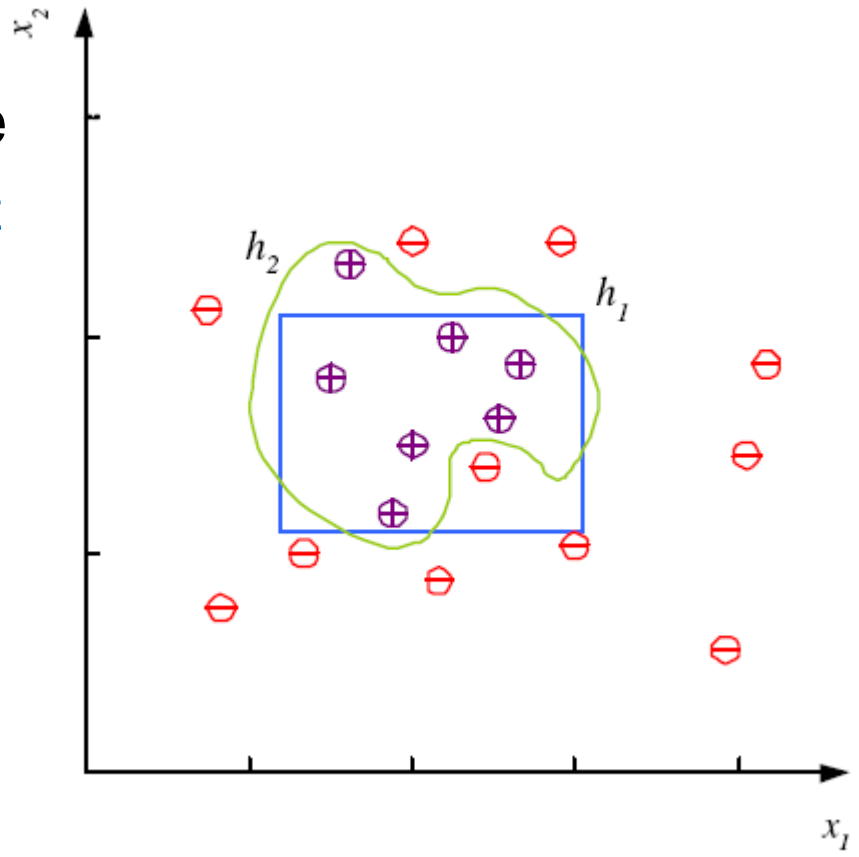
Noise and Model Complexity

Noise:

- Unwanted anomaly in the data

Potential Sources of Noise

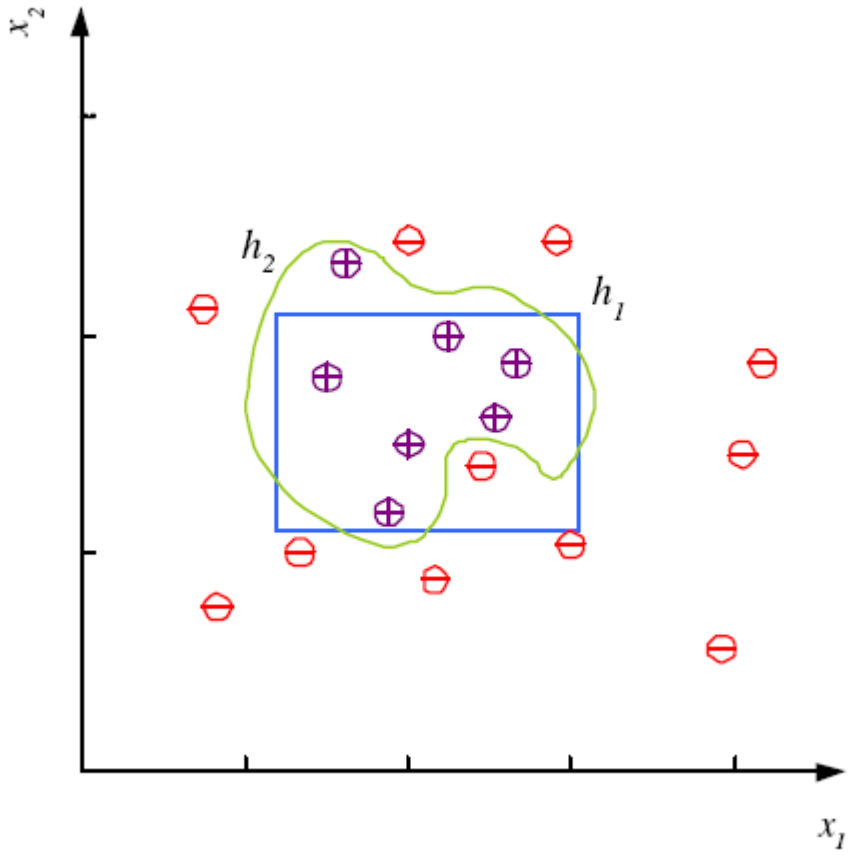
- Imprecision in recording input attributes
- Errors in labeling the data points (“teacher noise”)
- “Hidden” or “latent” attributes that affect the labels of the instances, but which are not considered in the data



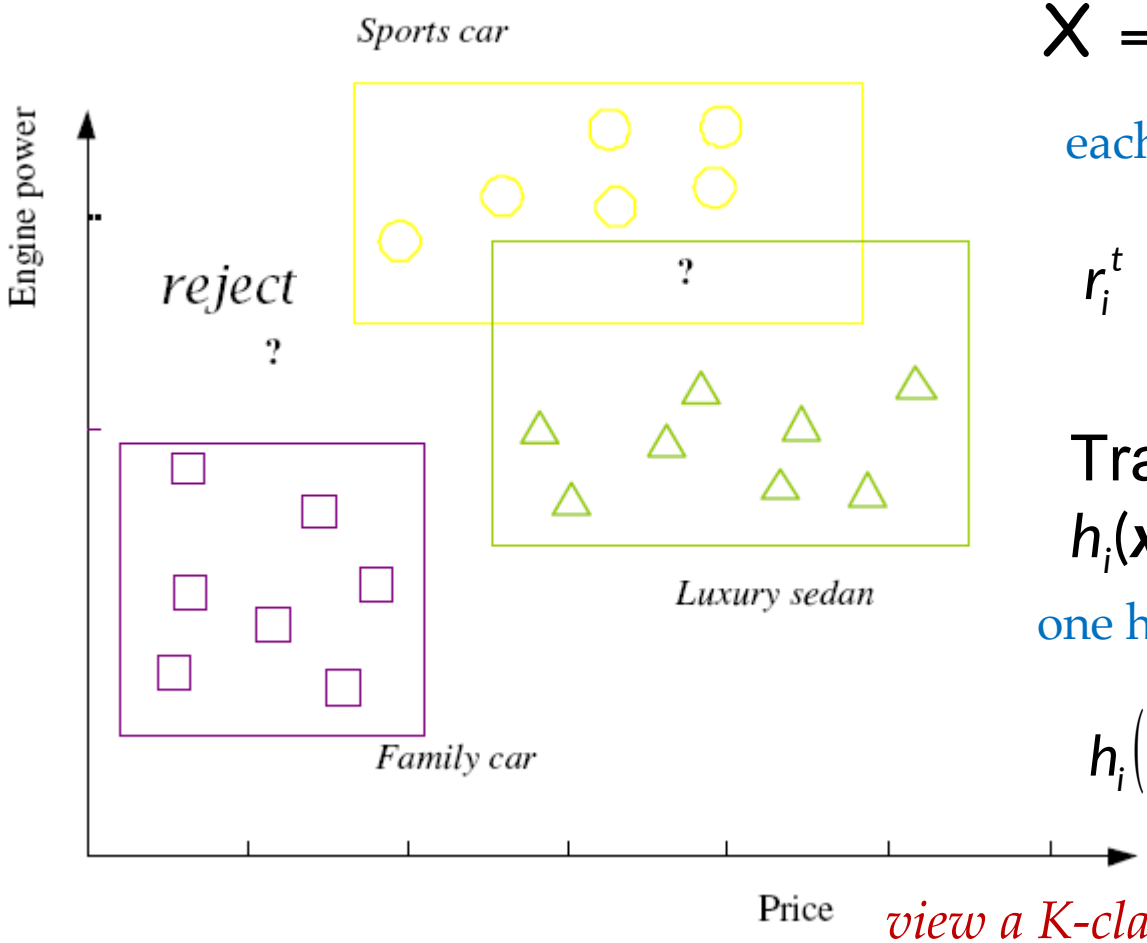
Noise and Model Complexity (cont.)

Use simplest model 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



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

each r^t is now a vector

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

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

one hypothesis for each class

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

view a K-class classification problem as K two-class classification problems

Regression

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

$$r^t \in \Re$$

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

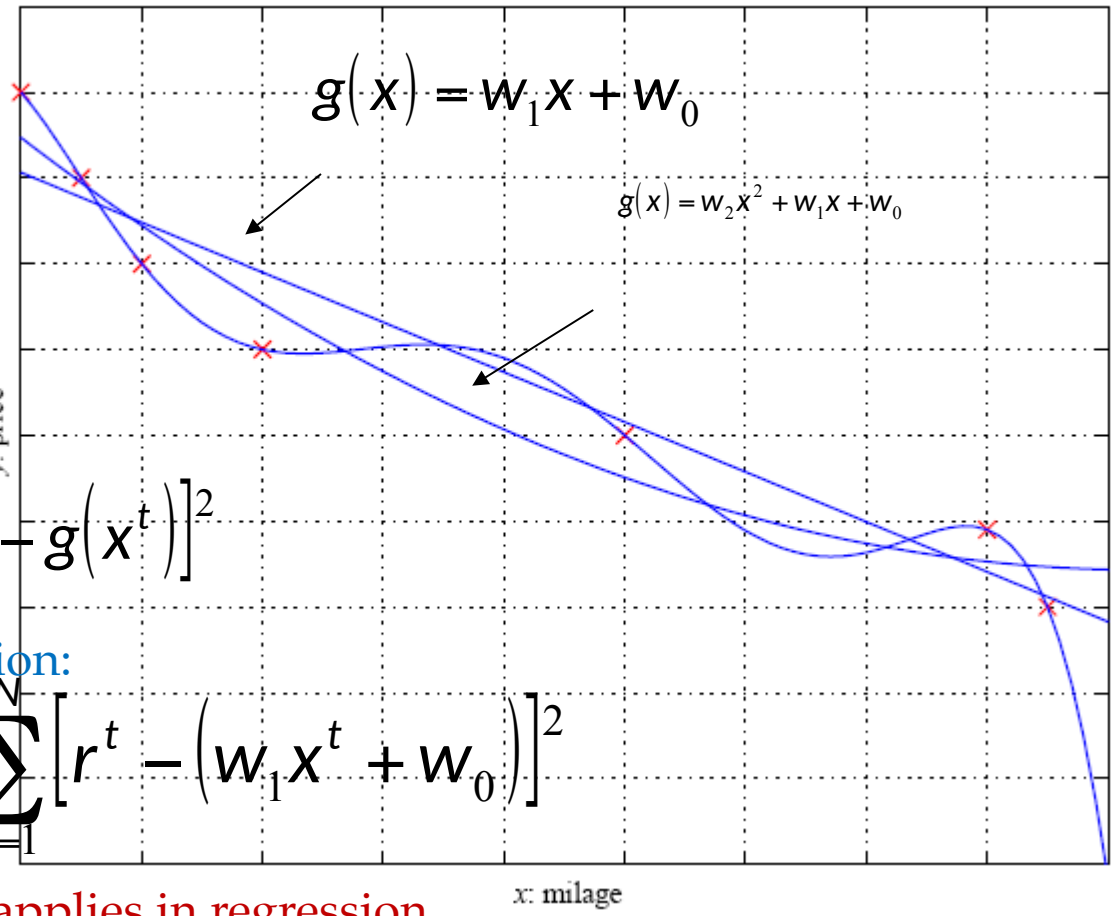
f: underlying function that we want to learn from data

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

Error of g in linear regression:

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

linear, second-order, and sixth-order polynomials



Occam's razor also applies in 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 $\mathcal{H}$

(e.g., our inductive bias in these slides' classification examples was to select rectangles as hypotheses - they could have been circles, squares or something else)

- Generalization: How well a model performs on new data
- Overfitting: $\mathcal{H}$ more complex than C or f

(e.g., fitting a 6th-order polynomial to noisy data sampled from a 2nd – order polynomial)

- Underfitting: $\mathcal{H}$ less complex than C or f

(e.g., fitting a 2nd-order polynomial to noisy data sampled from a 6th – order polynomial)

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 (e.g., 50%)

    data instances used to construct the model
  - Validation set (e.g., 25%)

    data instances used to test preliminary versions of the model and/or to refine the model
  - Test (publication) set (e.g., 25%)

    data instances used to test the final model after it has been fully constructed
- Resampling when there is few data

Dimensions of a Supervised Learner

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

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

   to compute the difference between the actual and predicted values

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