CSCI 5521: Machine Learning Fundamentals (Spring 2023)

Supervised Learning (Chpt 2)

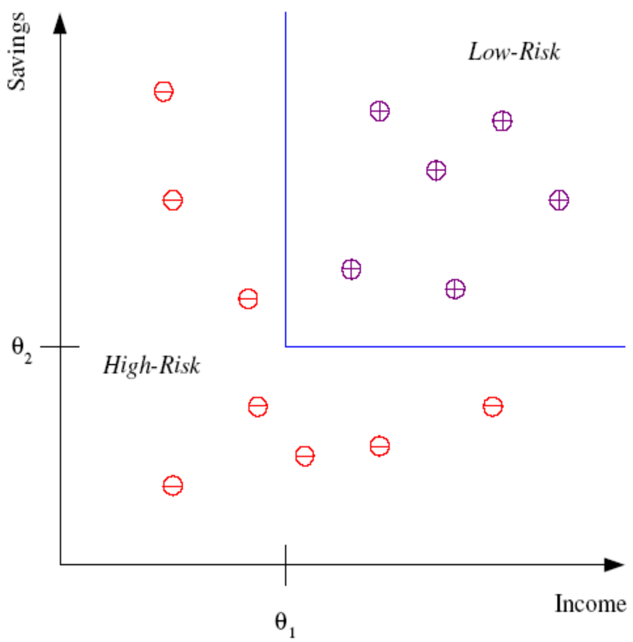
Rui Kuang

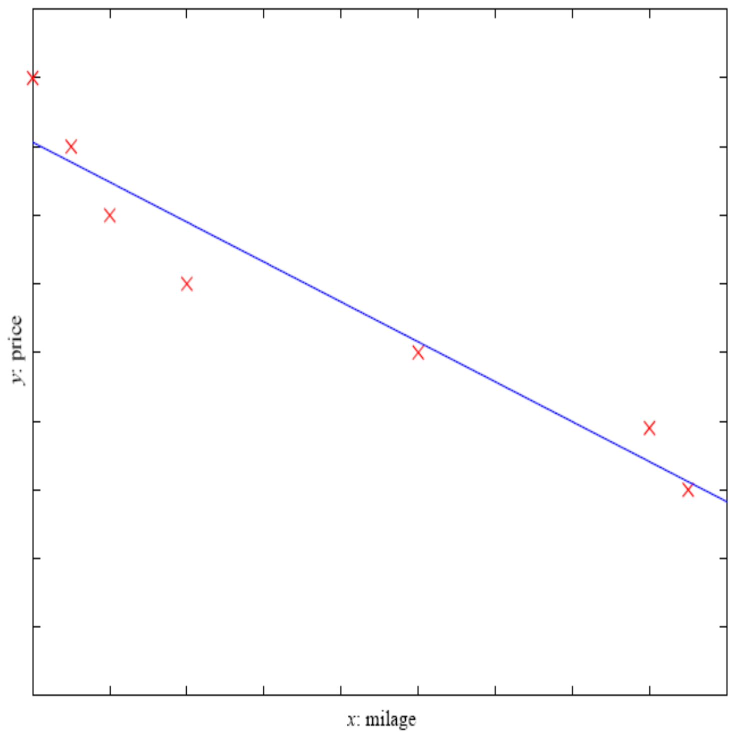
Department of Computer Science and Engineering

University of Minnesota

Supervised Learning

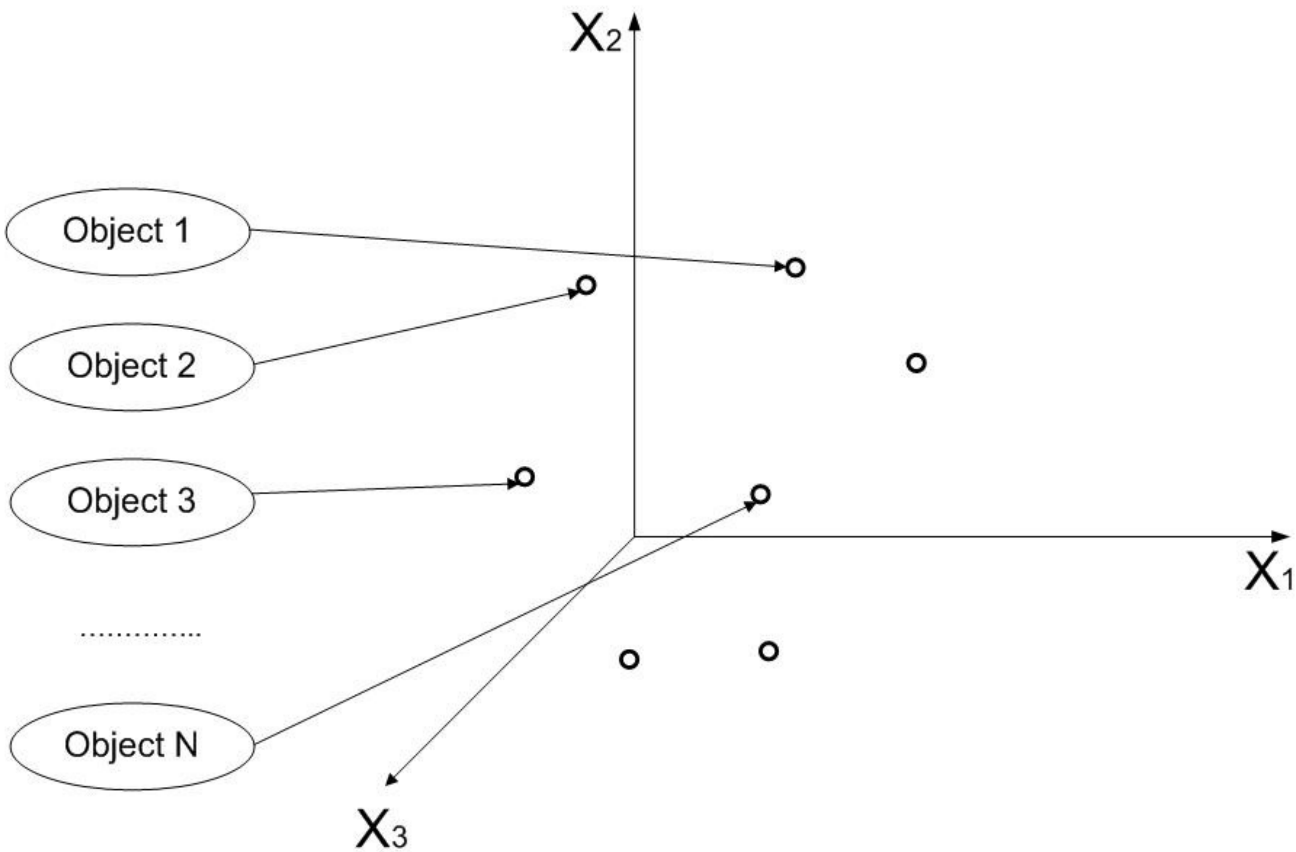
- Classification
- Regression





Input Feature Space

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



Supervised Learning

- Classification
- Regression

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

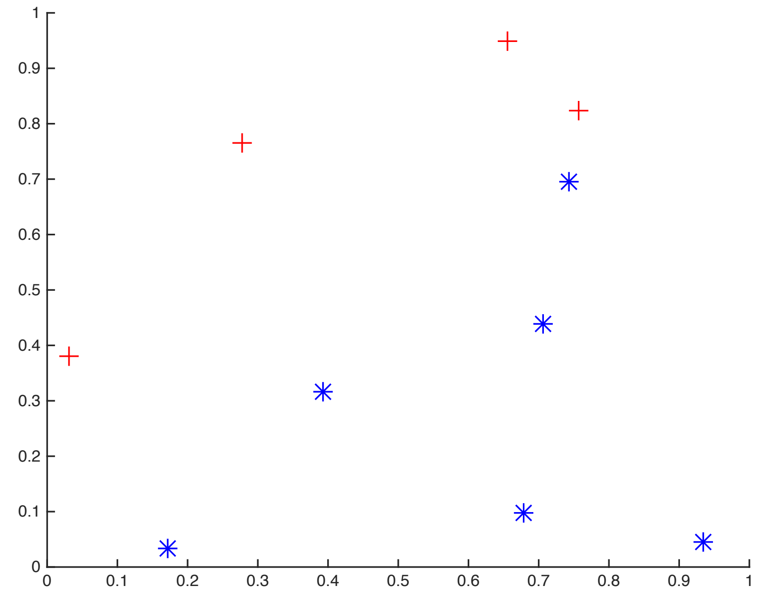
Output: $r^t = \begin{cases} 1 \text{ if } \mathbf{x} \text{ is positive} \\ 0\,/-1 \text{ if } \mathbf{x} \text{ is negative} \end{cases}$ $\qquad$ $r^t \in \Re$

(Class label) $\qquad$ (Response)

Classification

Data: $\mathcal{X} = \{\mathbf{x}^t, r^t\}_{t=1}^N$ Output: $r = \begin{cases} 1 \text{ if } \mathbf{x} \text{ is positive} \\ 0/-1 \text{ if } \mathbf{x} \text{ is negative} \end{cases}$

X_1	X_2	r
0.934	0.046	-1
0.679	0.097	-1
0.758	0.823	1
0.743	0.695	-1
0.392	0.317	-1
0.655	0.950	1
0.171	0.034	-1
0.706	0.439	-1
0.032	0.382	1
0.277	0.766	1



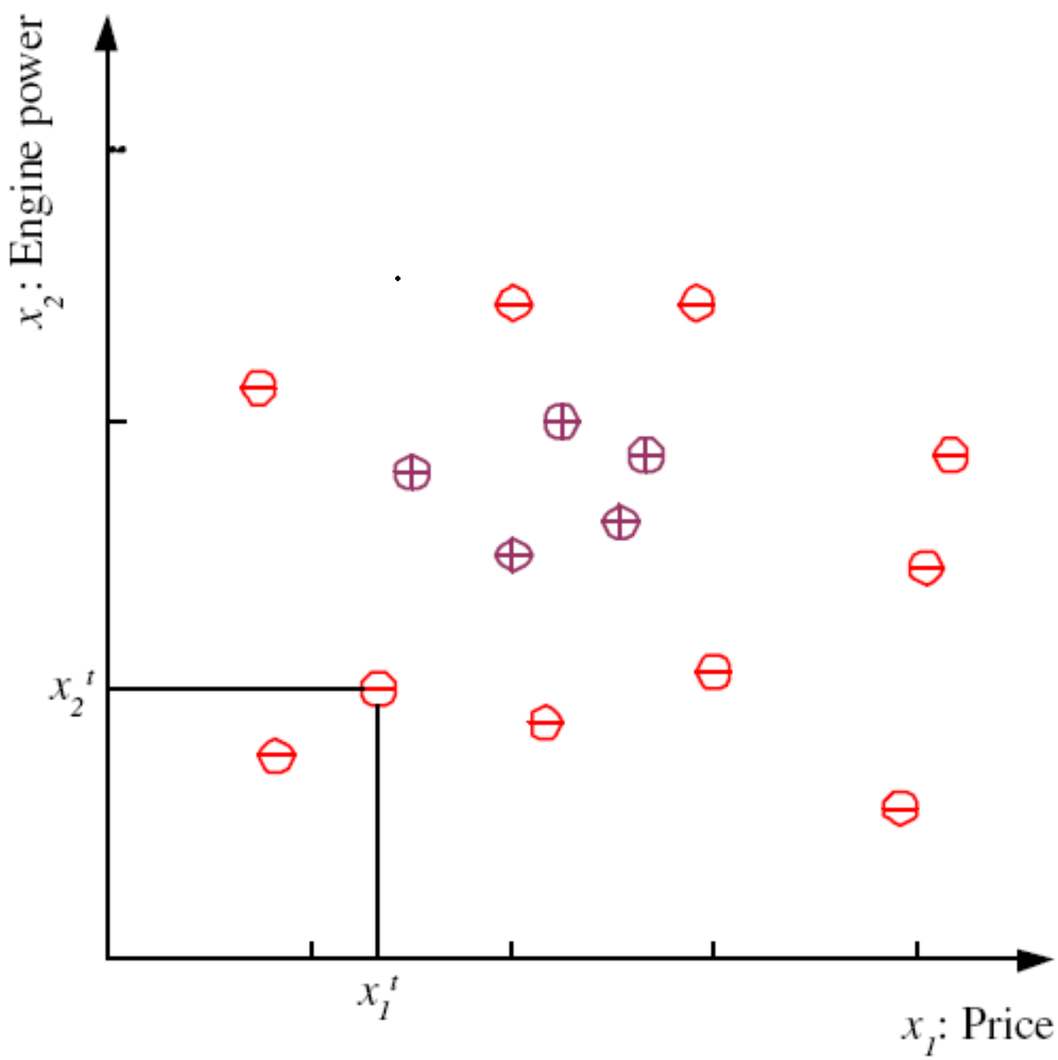
Learning a Class from Examples

- 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 $\mathcal{X}$

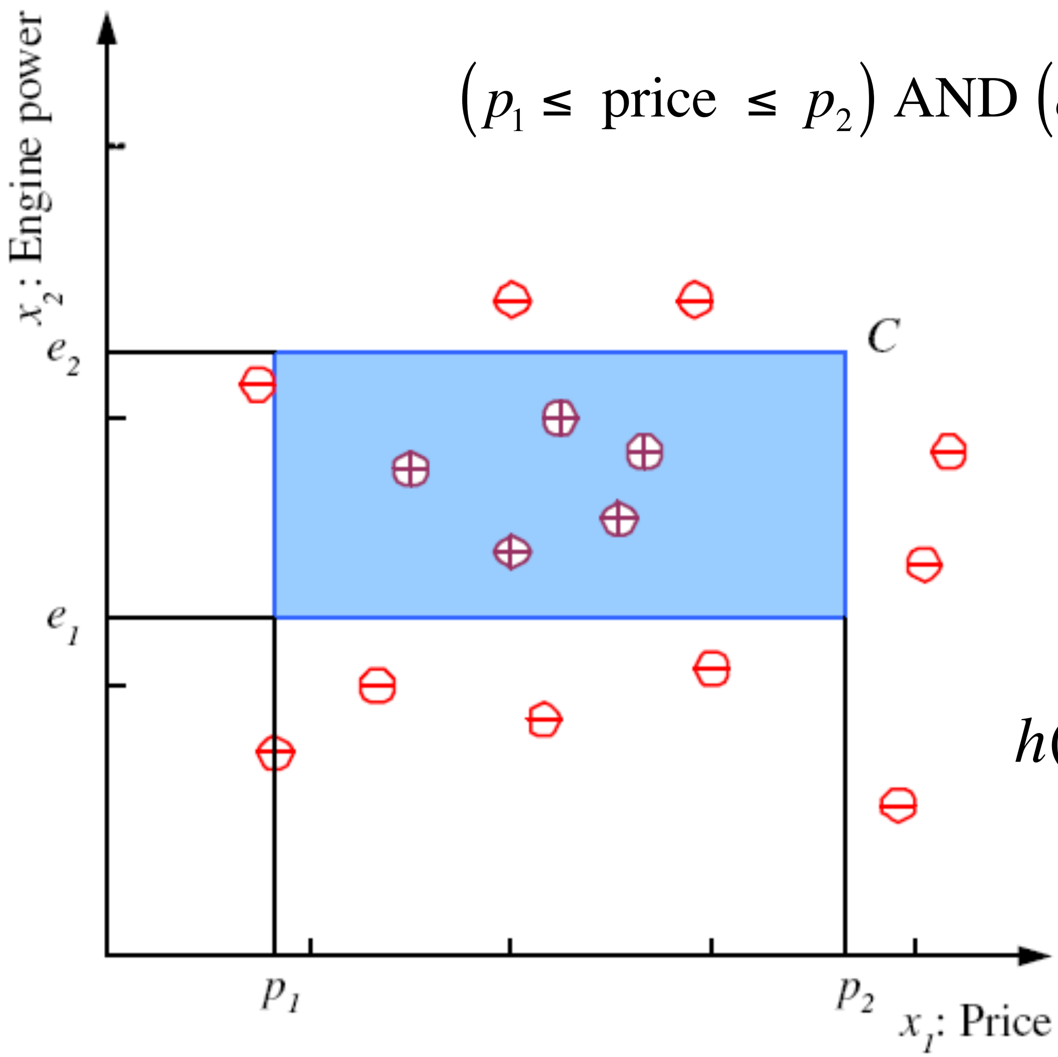


$$\mathcal{X} = \{\mathbf{x}^t, r^t\}_{t=1}^{N}$$

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

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

Class in a Rectangle

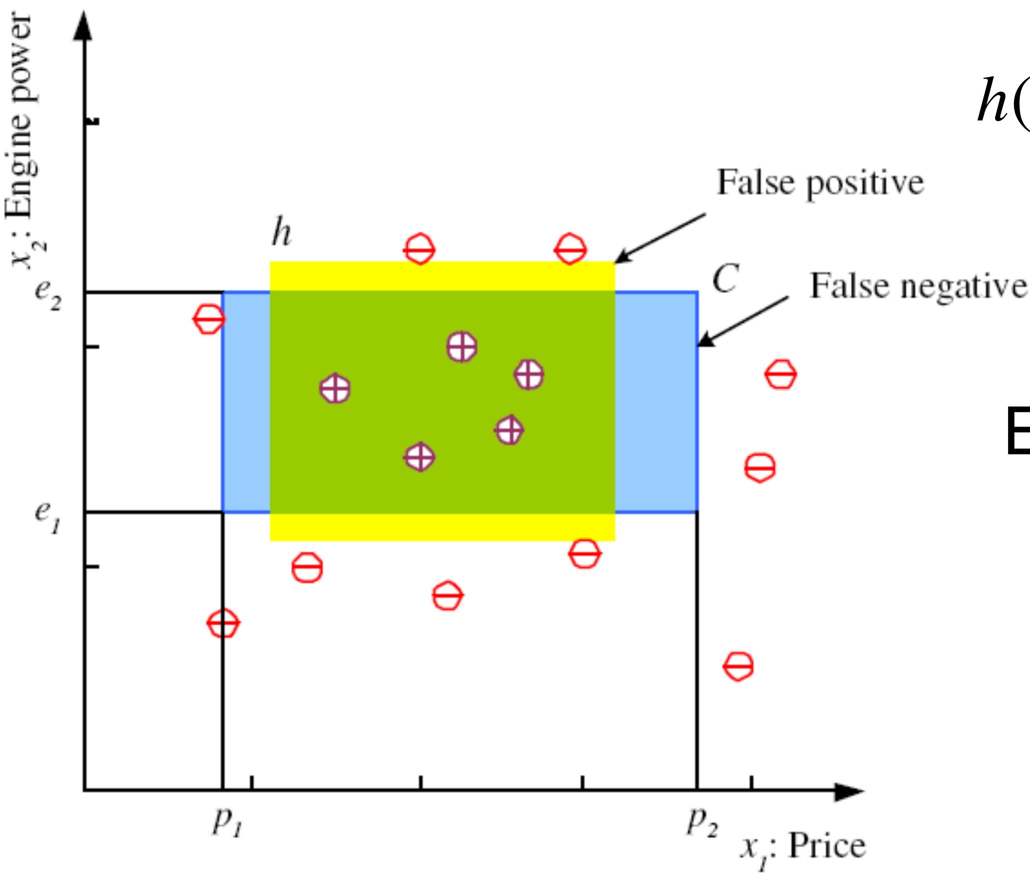


$$\left(p_1 \le \text{ price } \le p_2\right) \text{ AND } \left(e_1 \le \text{ engine power } \le e_2\right)$$

$$h(\mathbf{x}) = \begin{cases} 1 \text{ if } h \text{ says } \mathbf{x} \text{ is positive} \\ 0 \text{ if } h \text{ says } \mathbf{x} \text{ is negative} \end{cases}$$

Hypothesis class $\mathcal{H}$

Consider $\mathcal{H}$: the set of all rectangles

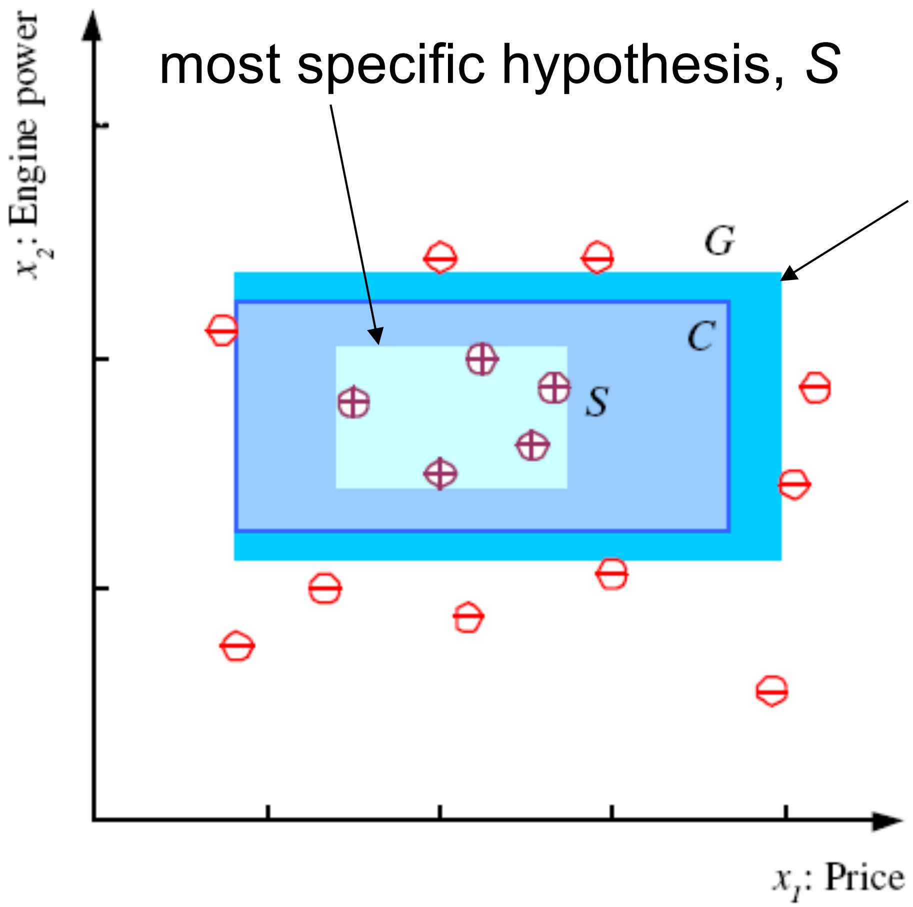


$$h(\mathbf{x}) = \begin{cases} 1 \text{ if } h \text{ says } \mathbf{x} \text{ is positive} \\ 0 \text{ if } h \text{ says } \mathbf{x} \text{ is negative} \end{cases}$$

Error of *h* on $\mathcal{X}$

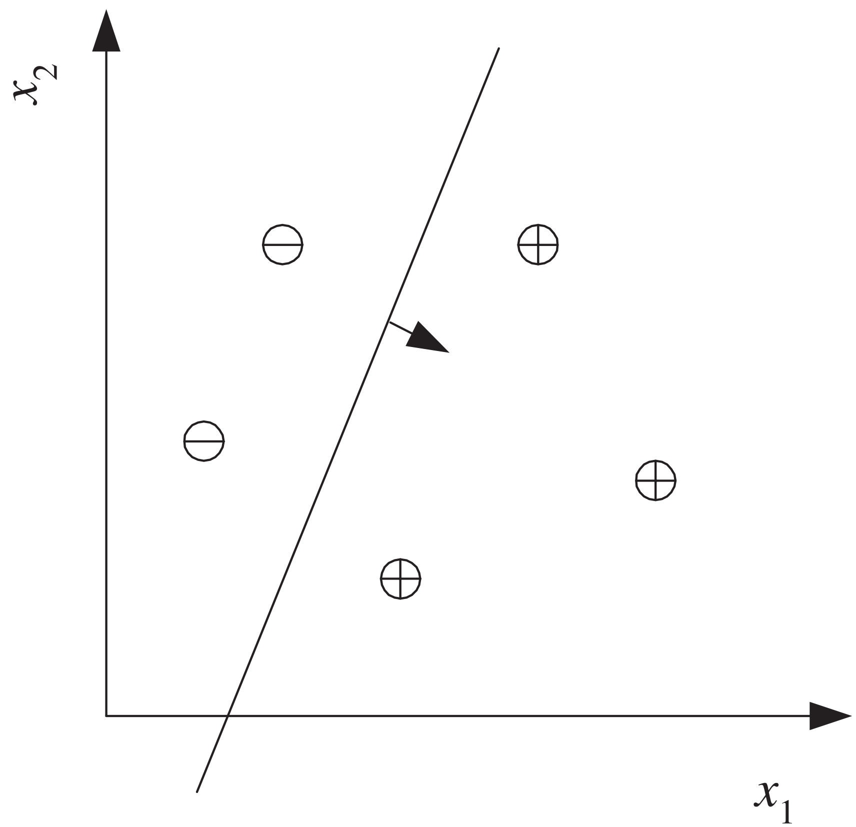
$$E(h \mid \mathcal{X}) = \sum_{t=1}^{N} 1\left(h\left(\mathbf{x}^t\right) \neq r^t\right)$$

Version Space



h ∈ H, between *S* and *G* is consistent and make up the version space (Mitchell, 1997)

Linear Classifier



h(x)=<*w*,*x*>+b is a linear classifier

h(x)>*0* positive
h(x)<*0* negative

h $\in$ *H*, *H*?

Perceptron Learning

- Perceptron algorithm, Rosenblatt, 1957.
- Initialization:

$$w = 0$$

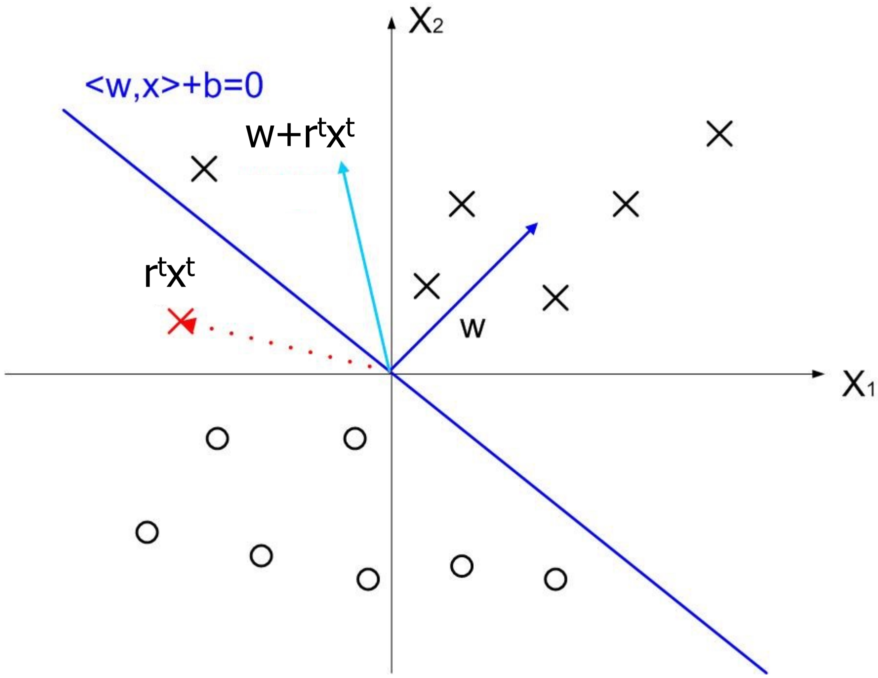
- Iterate until converge (no mistake)

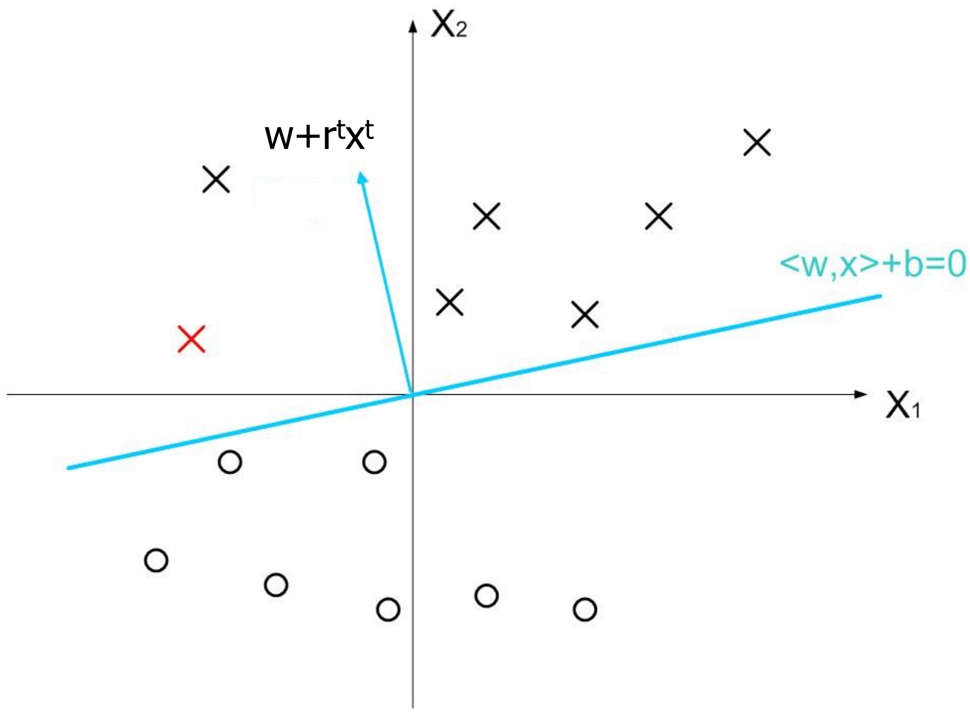
$$\text{for each example } (\mathbf{x}^t, r^t):$$

$$if\,(<\mathbf{w}, \mathbf{x}^t> * r^t \leq 0)$$

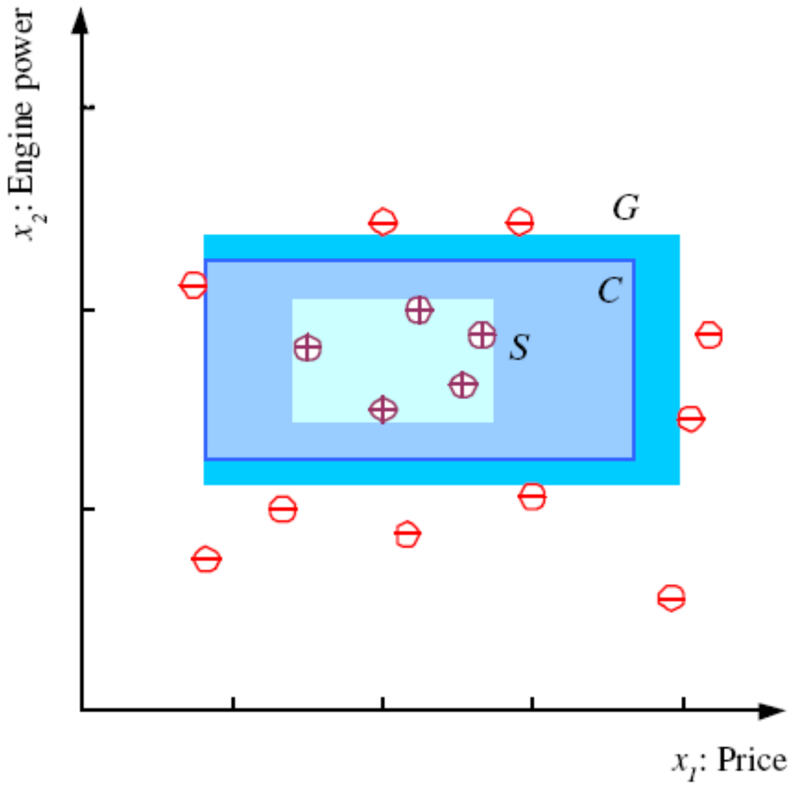
$$\mathbf{w} = \mathbf{w} + r^t \mathbf{x}^t$$

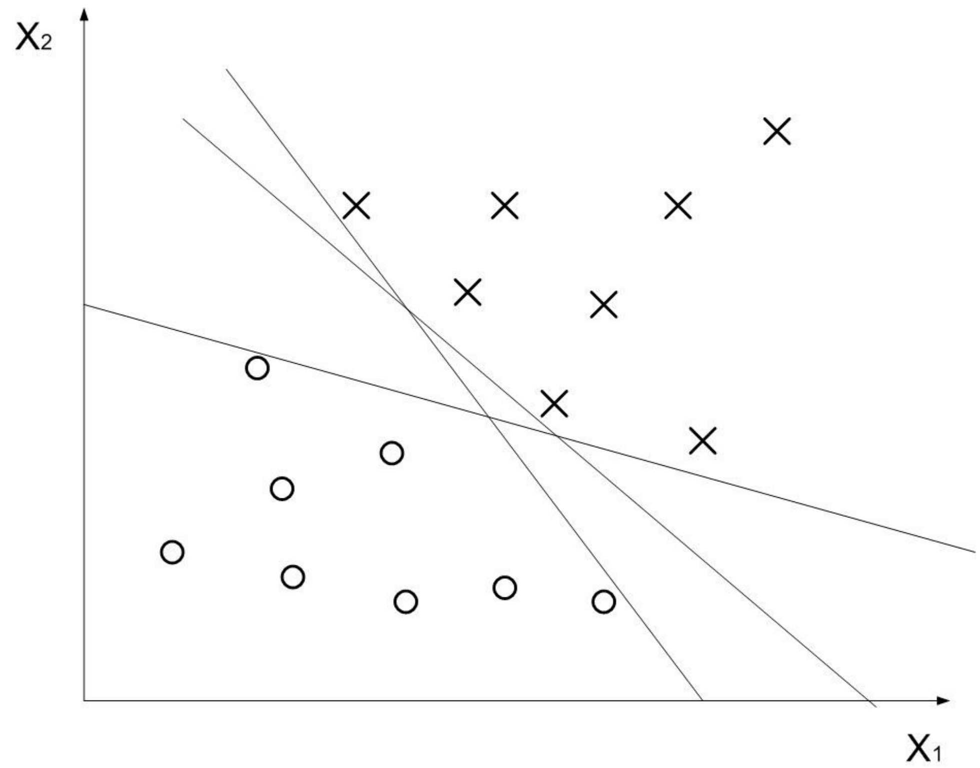
Perceptron Learning





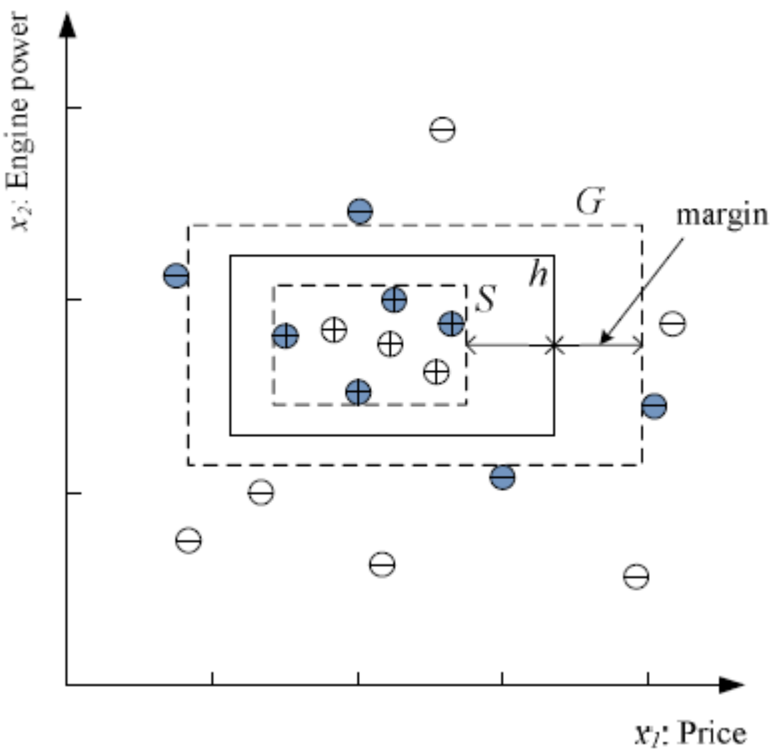
Best in the Version Space

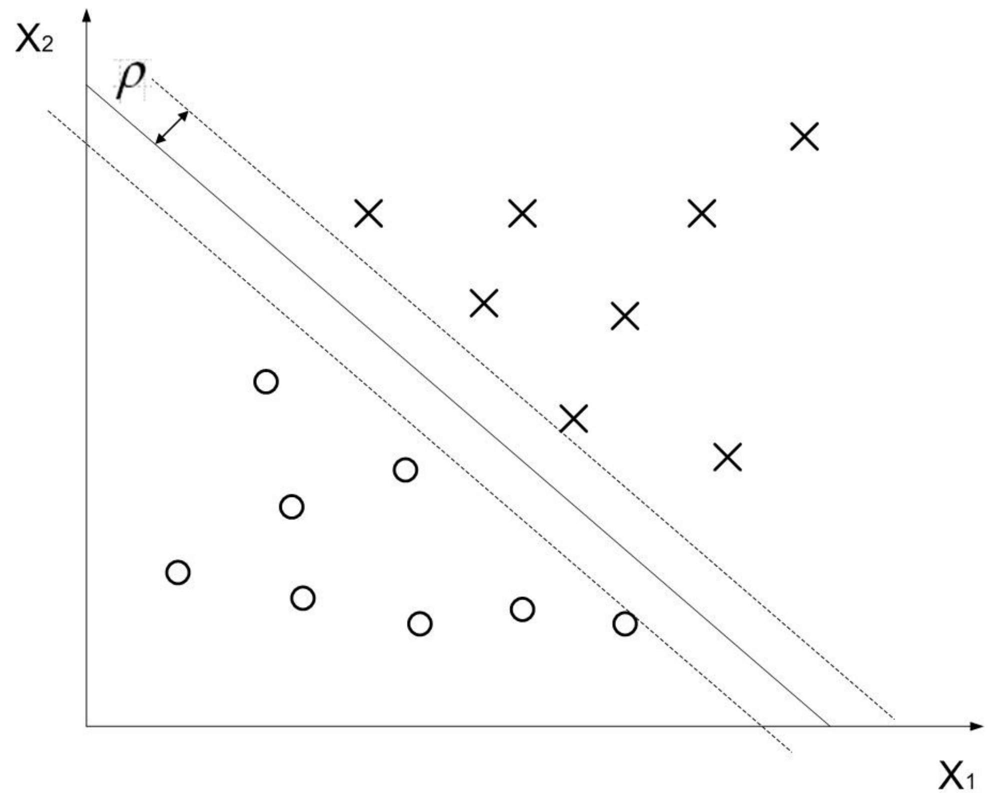




Margin

- Choose *h* with largest margin
- Why?





Model Capacity

- Different models have different capacity meaning the ability to handle more complex data.

- How to measure model capacity?
- The maximum number of data points that can be classified perfectly in any labeling.

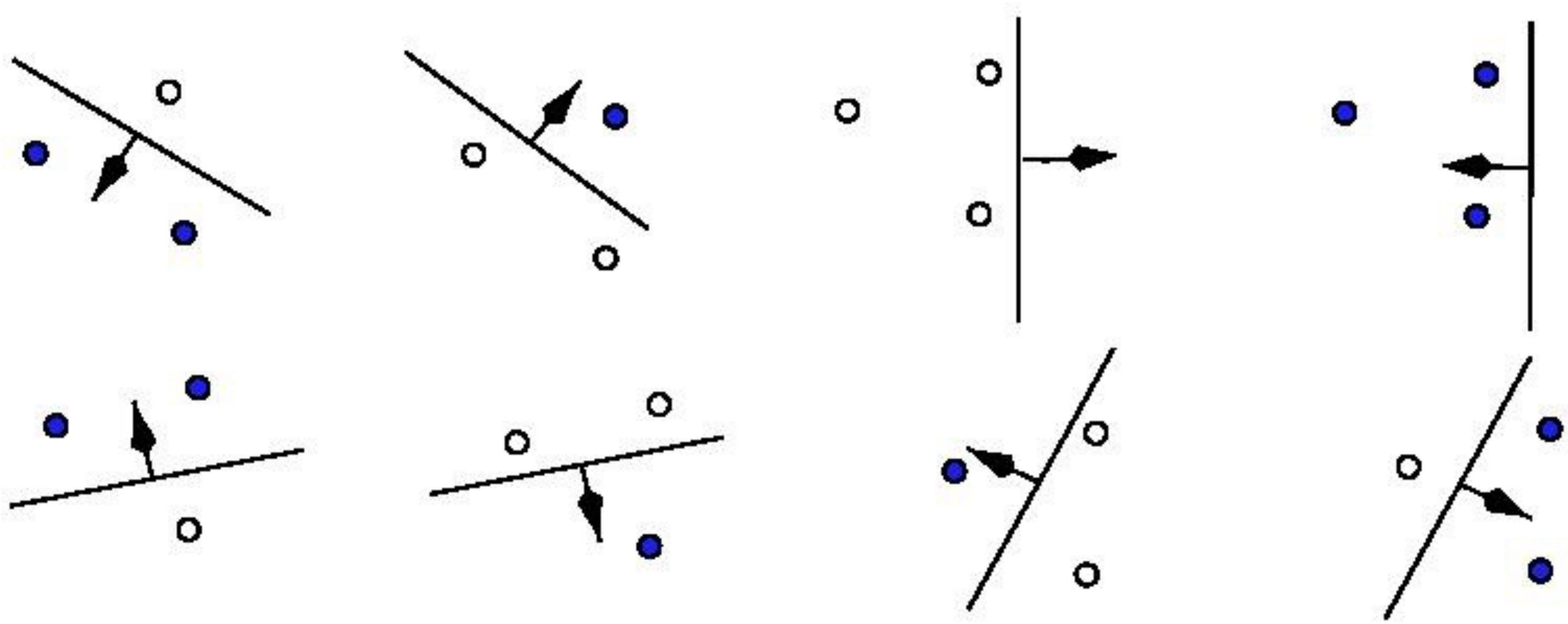
VC (Vapnik Chervonenkis) Dimension

- N points can be labeled in 2^N ways as +/–
- In a particular arrangement, $\mathcal{H}$ shatters N if there exists $h \in \mathcal{H}$ consistent for any of the 2^N ways:

$$\mathrm{VC}(\mathcal{H}) = N$$

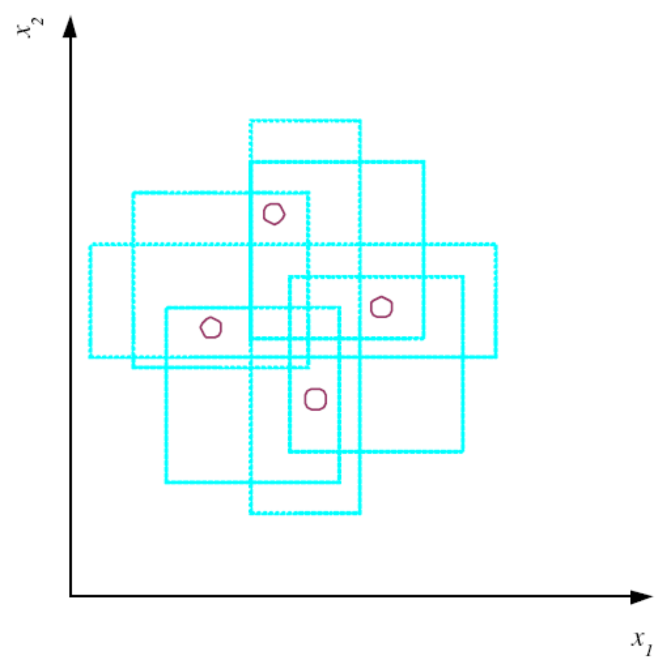
VC Dimension

How many points can be shattered by a line?



VC (Vapnik Chervonenkis) Dimension

- How about axis-aligned rectangles?



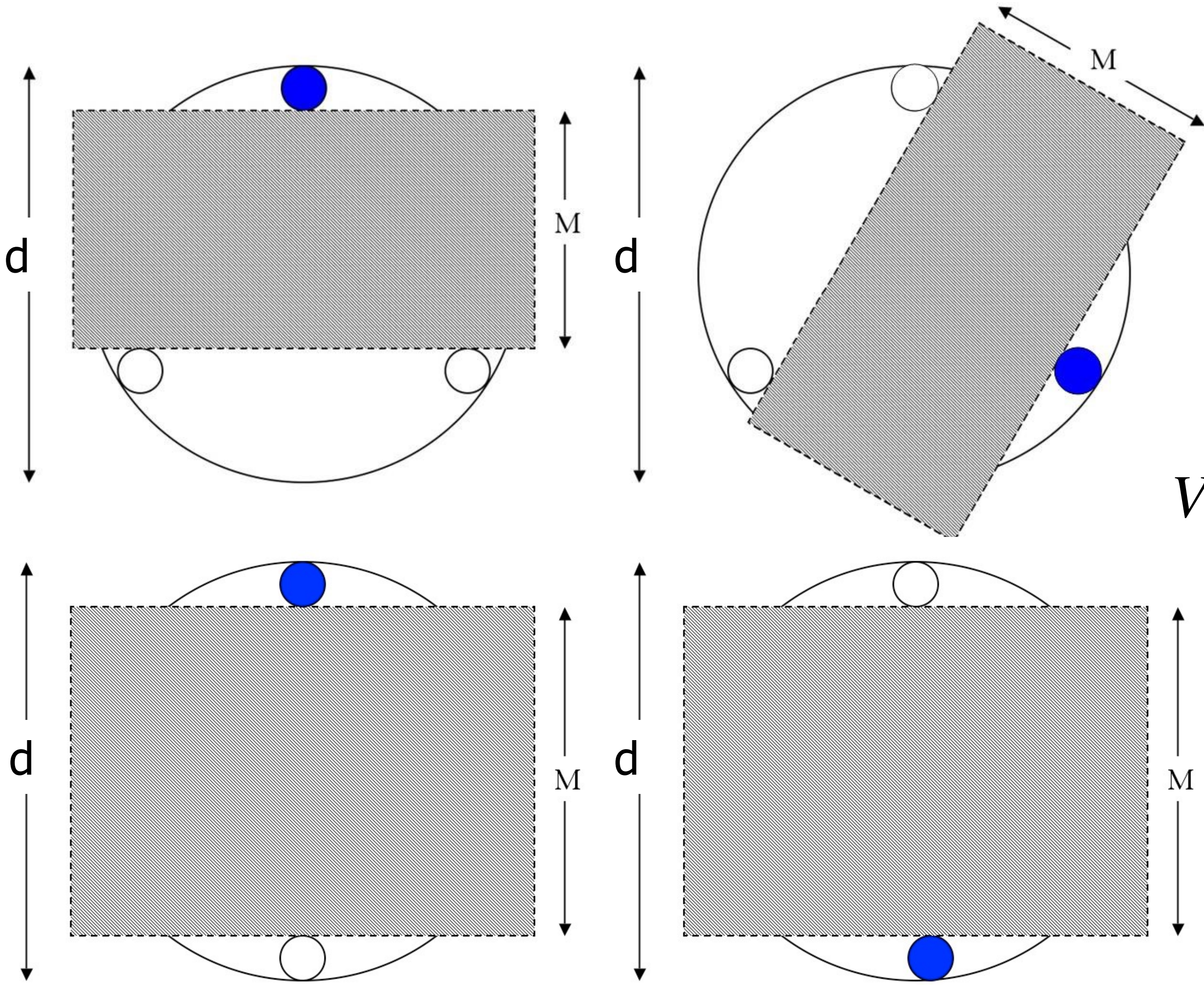
VC Summary

- The capacity of function is measured by the number of data points that can be shattered by the function.
- VC dimension can be motived by the proof of No-Free-Lunch theorem for PAC learning theory (section 2.3 EA book).
- Rectangle classifier in 2-D space: 4.
- A line : 3.
- More ...

VC Dimension

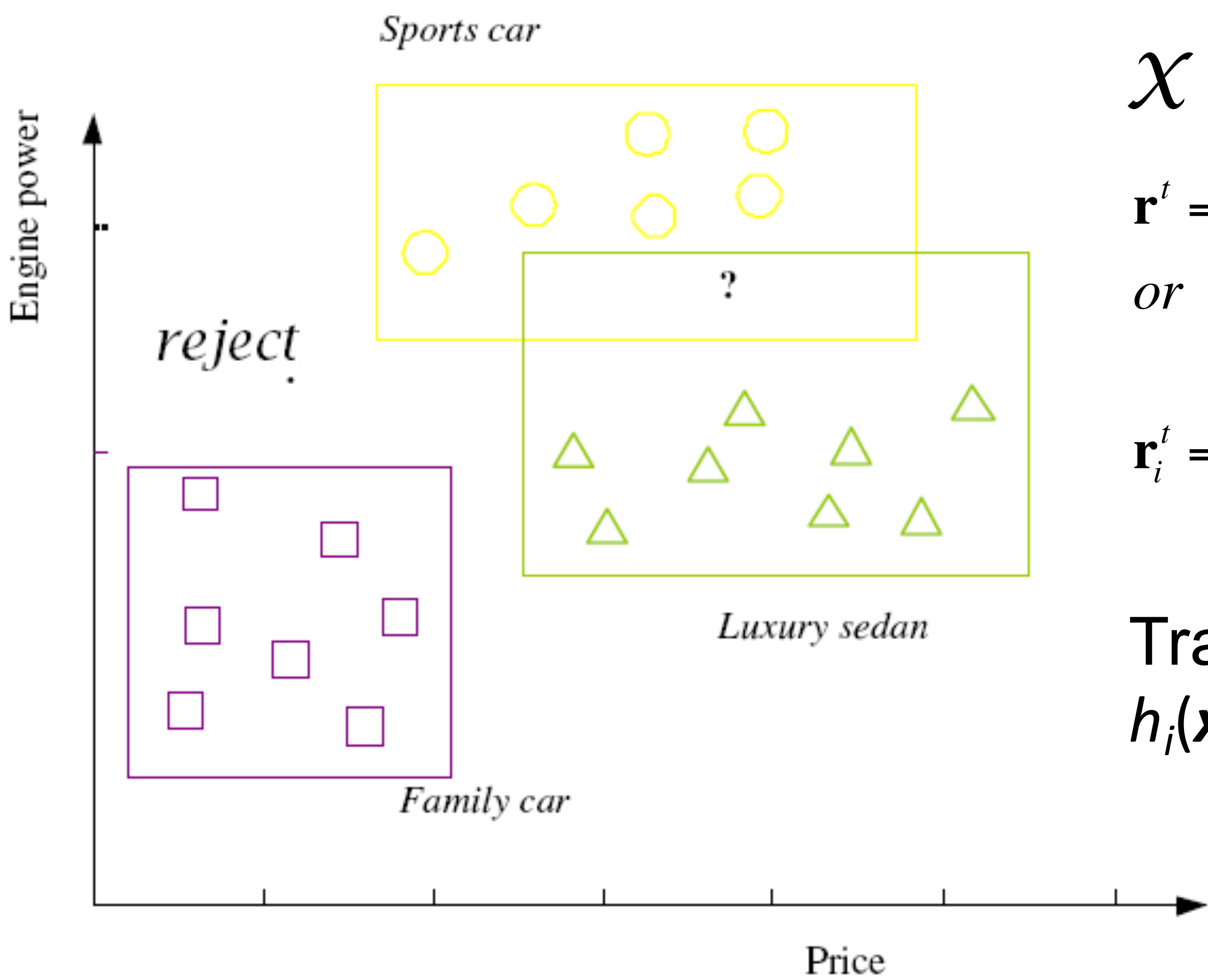
- More generally, in R^D space, what is the VC of a hyperplane?
- What is the VC of a triangle classifier?
- Is an algorithm that can shatter only 4 or 3 data points useful?
- How easy it is to determine the VC dimension for the hypothesis class?

VC Dimension: Why Large Margin



$$VC \leq \min(ceil[\frac{d^2}{M^2}], D) + 1$$

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



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

$$\mathbf{r}^t = C_i \text{ if } \mathbf{x}^t \in C_i$$

or

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

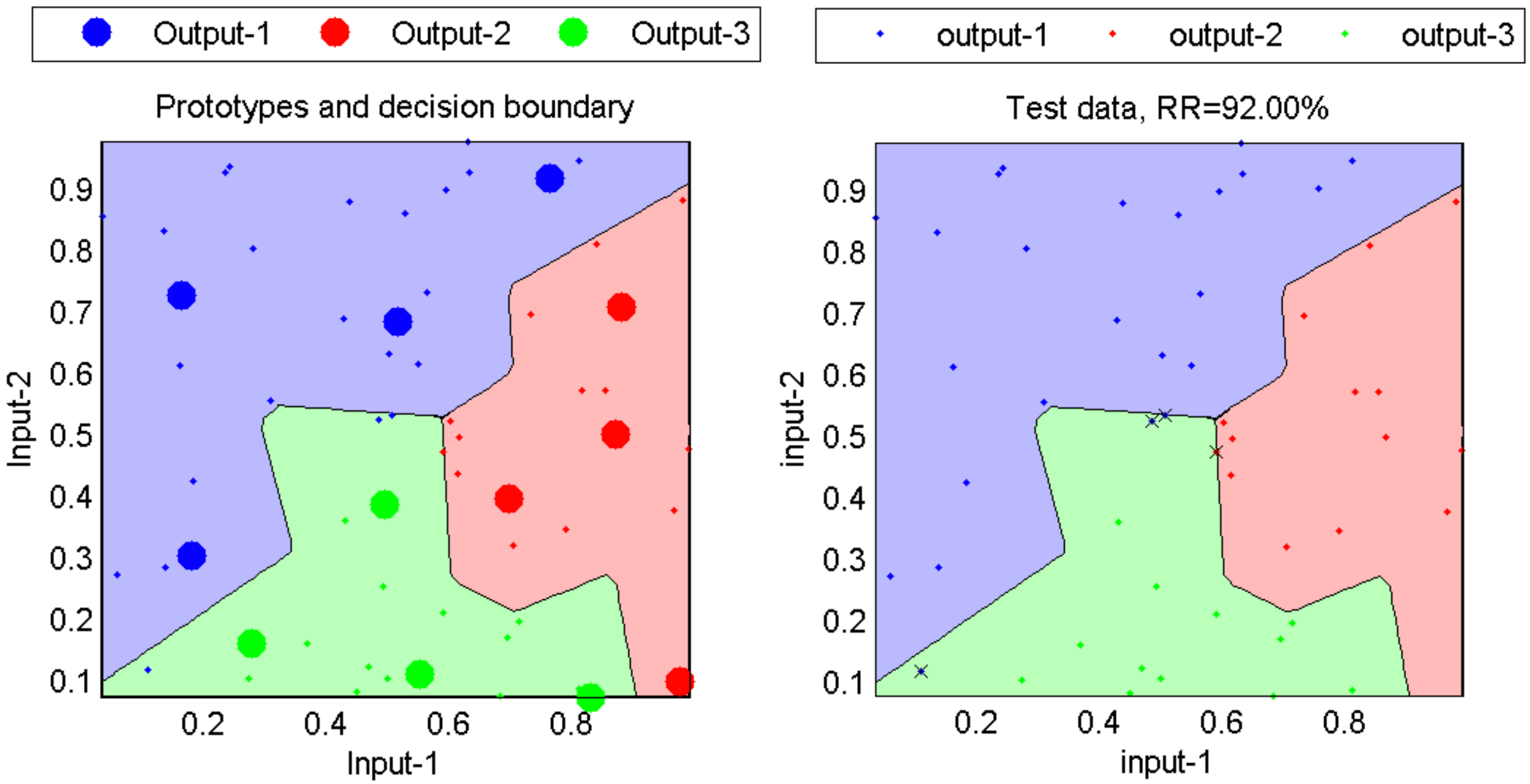
Train hypotheses
$h_i(\boldsymbol{x})$, i =1,...,K:

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

KNN Classification

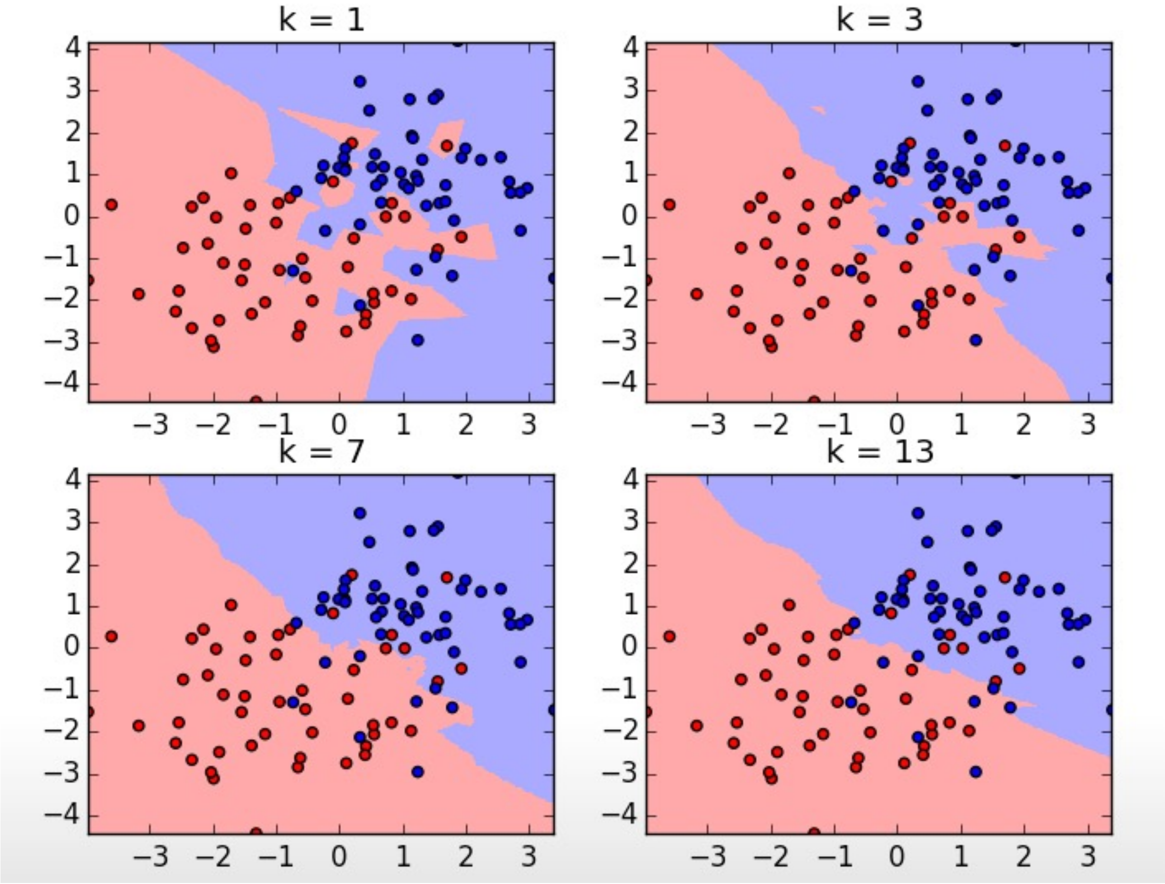
- K nearest neighbor

$$h_i(x) = |\{(x^t, r^t) \mid r^t = C_i \,\&\, x^t \in N_x^{(k)}\}|$$



http://mirlab.org/jang/books/dcpr/prKnnc.asp?title=5-2%20K-nearest-neighbor%20Classifiers&language=english

How to Choose K for KNN?



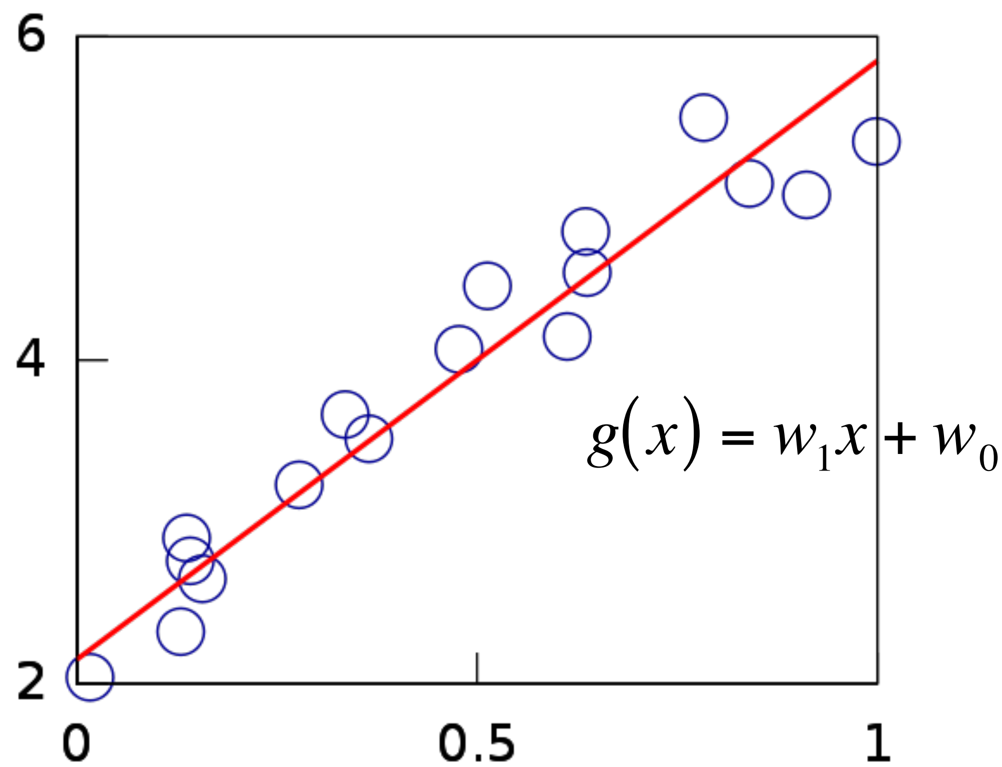
- What is the VC dimension of KNN?
- Is VC proportional to the # of parameters (appeared complexity)?

http://ljdursi.github.io/ML-for-scientists/

Regression

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

$$r^t = g\left(x^t\right) + \varepsilon, \ (\varepsilon\text{: random noise})$$



Training Error:

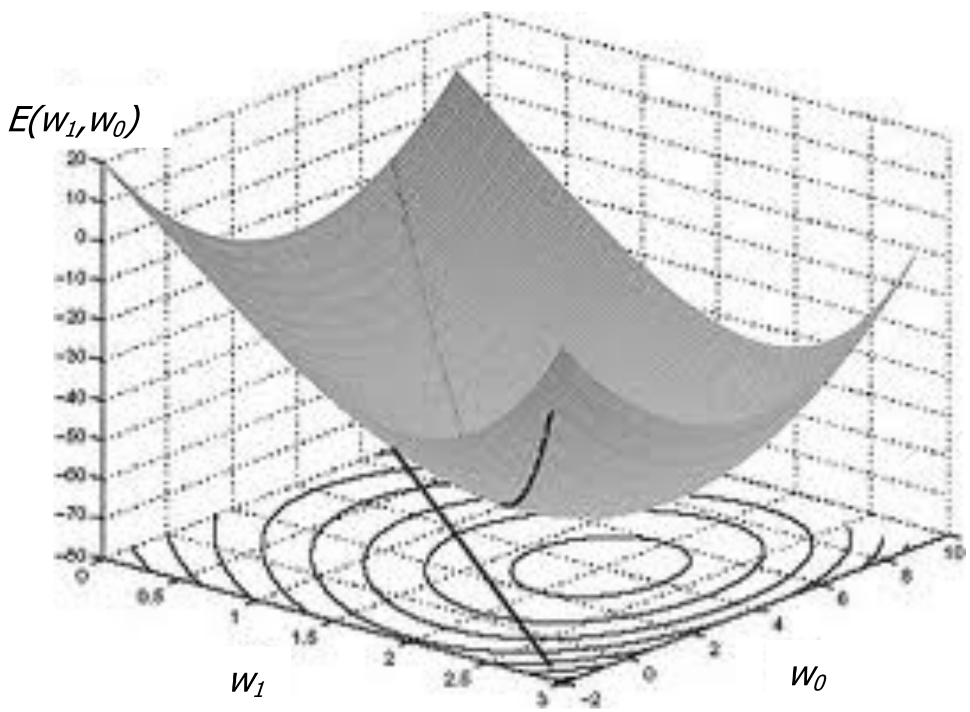
$$E(g \mid \mathcal{X}) = \frac{1}{N} \sum_{t=1}^{N} \left[r^t - g\left(x^t\right)\right]^2$$

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

Regression

- How does the error function look like?

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



Regression

- Find the *g* to minimize training error

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

$$\frac{\partial E\left(w_1, w_0 \mid \mathcal{X}\right)}{\partial w_0} = \frac{1}{N}\sum_{t=1}^{N}\left[(r^t - w_1 x^t - w_0)(-1)\right] = 0$$

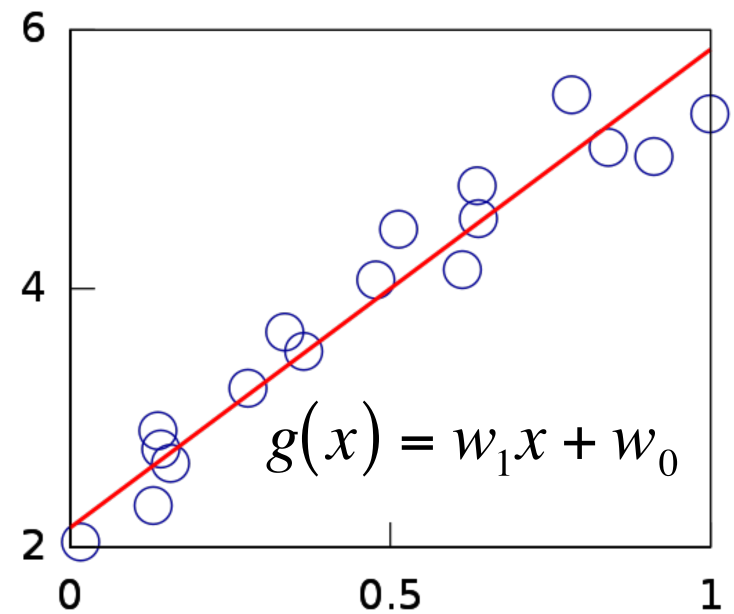
$$\frac{\partial E\left(w_1, w_0 \mid \mathcal{X}\right)}{\partial w_1} = \frac{1}{N}\sum_{t=1}^{N}\left[(r^t - w_1 x^t - w_0)(-x^t)\right] = 0$$

$$w_1 = \frac{\sum_t x^t r^t - N\overline{x}\overline{r}}{\sum_t (x^t)^2 - N\overline{x}^2}, w_0 = \overline{r} - w_1 \overline{x}$$

Regression: Understand Solution

$$r^t = g\left(x^t\right) + \varepsilon, \ (\varepsilon\text{: random noise})$$

$$\Rightarrow \varepsilon^t = r^t - g\left(x^t\right), \ (\varepsilon^t\text{: error on sample } t)$$



- Property 1:

$$\frac{1}{N}\sum_{t=1}^{N}\left[(r^t - w_1 x^t - w_0)\right] = \sum_{t=1}^{N}\varepsilon^t = 0$$

Average error is 0.

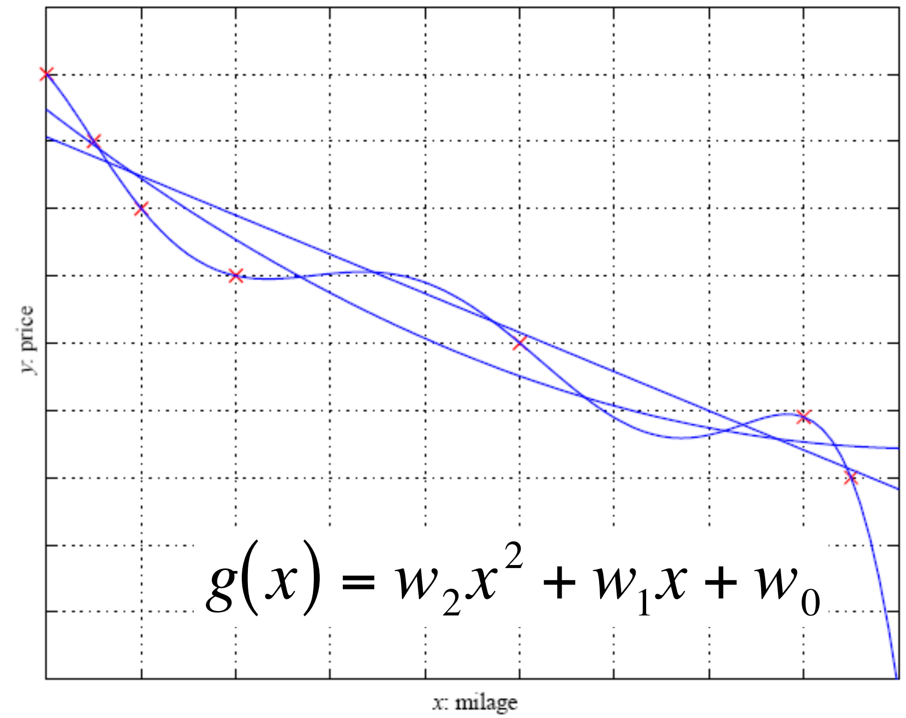
- Property 2: $$\frac{1}{N}\sum_{t=1}^{N}\left[(r^t - w_1 x^t - w_0)(-x^t)\right] = \sum_{t=1}^{N}\varepsilon^t x^t = 0$$

Error is uncorrelated with data

Polynomial Regression

- Is polynomial fitting very different?

$$g(x) = \sum_{i=1}^{P} w_p (x)^p + w_0$$



- It is the same as linear regression with a polynomial mapping.

$$g(x) = w^T x$$

$$w = [w_P, \ldots, w_1, w_0]$$

$$x = [x^P, \ldots, x^1, x^0]$$

Summary of Supervised Learning

1. Model: $g(\mathbf{x} \mid \theta)$ $\qquad g(x) = w_1 x + w_0$

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

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

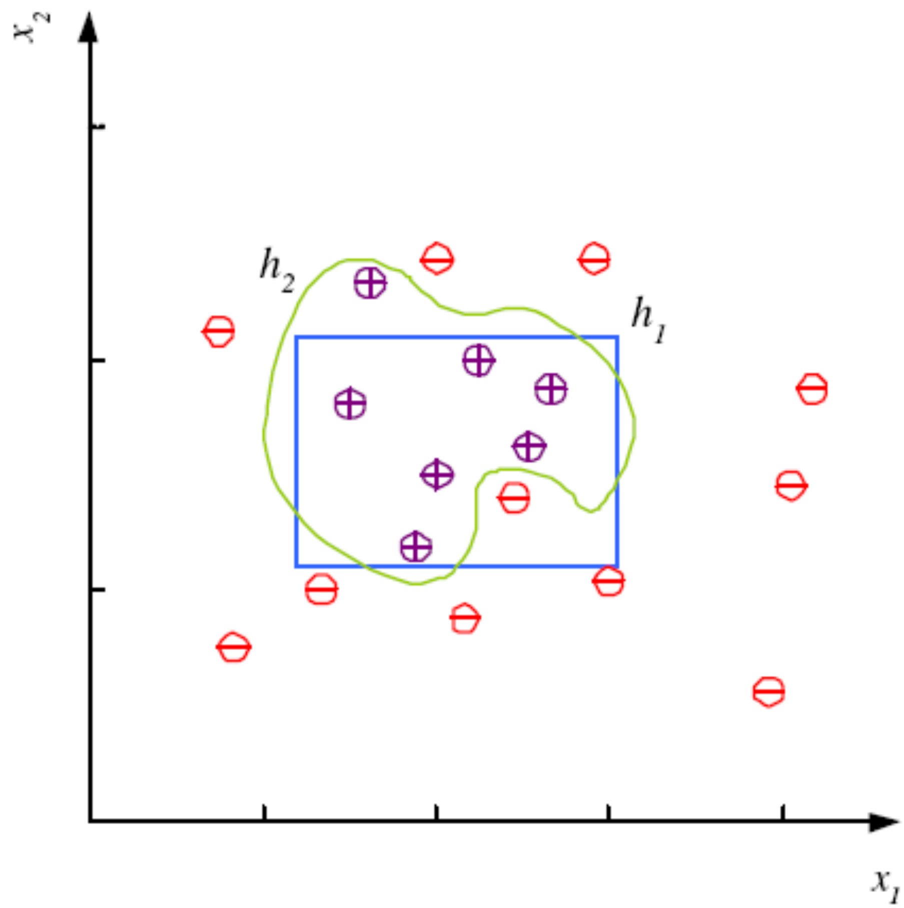
3. Optimization procedure: $\theta^* = \arg \min_{\theta} E(\theta \mid \mathcal{X})$

Algorithms: KNN, percepton, linear regression

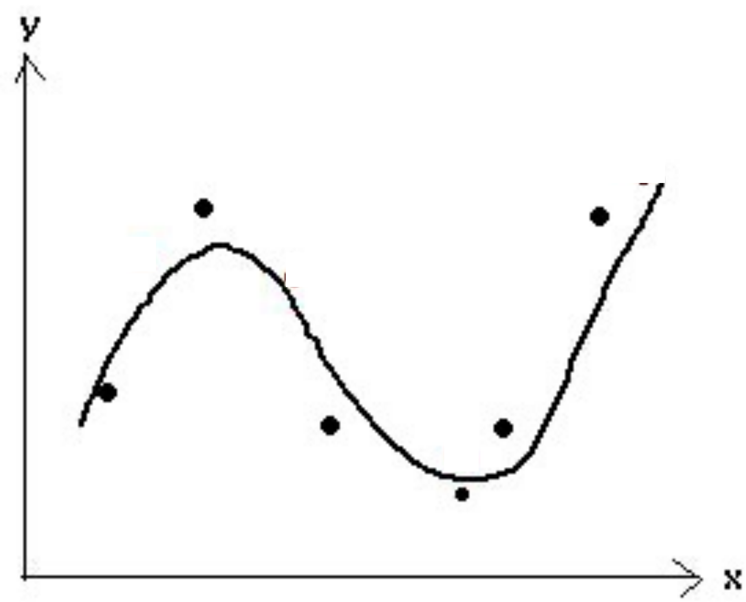
Noise and Model Complexity

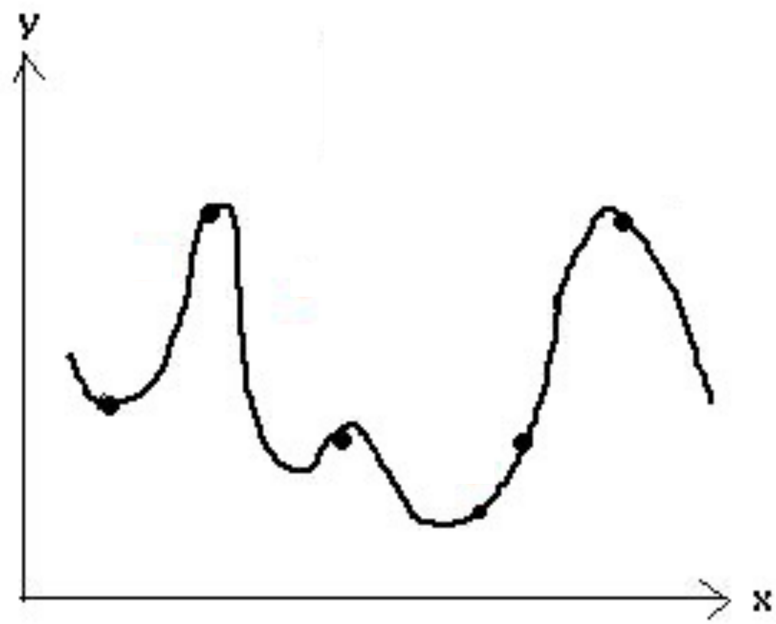
Data is not perfect

- Data recording might not be perfect (shifted data points)
- Wrong labeling of the data
- There might be additional unobervable hidden variables.



Noise and Model Complexity





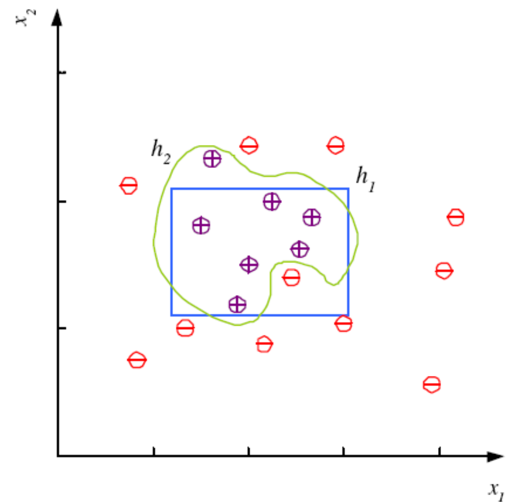
Options:

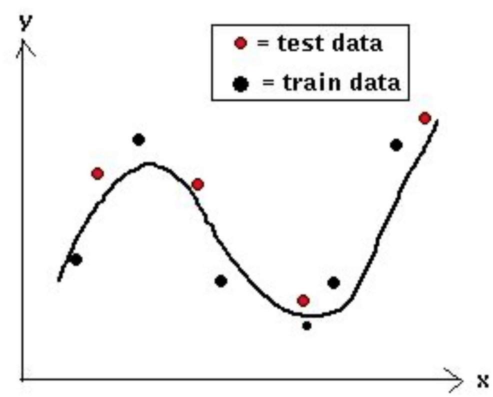
- Simple model with training errors
- Complex comdel with no training error

Noise and Model Complexity

Given similar training error use the simpler one

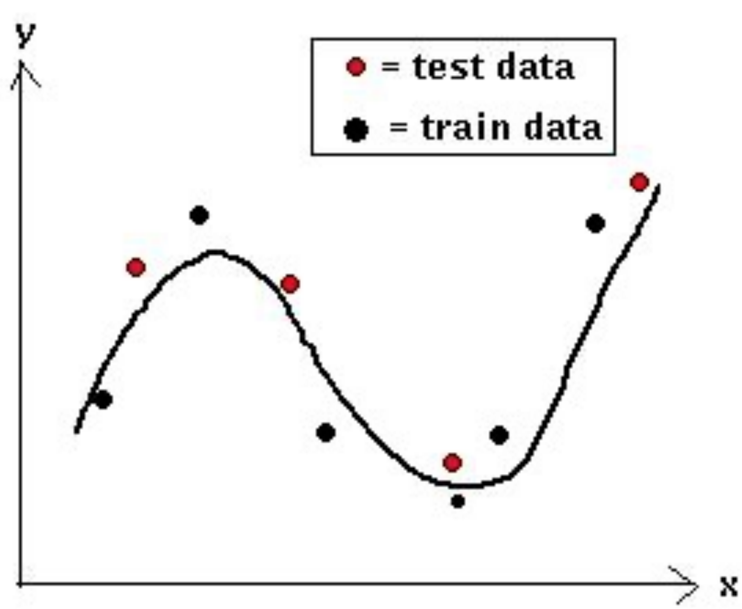
- Simpler to use (lower computational complexity)
- Easier to train (lower space complexity)
- Easier to explain (more interpretable)
- Generalizes better (lower variance - Occam's razor)

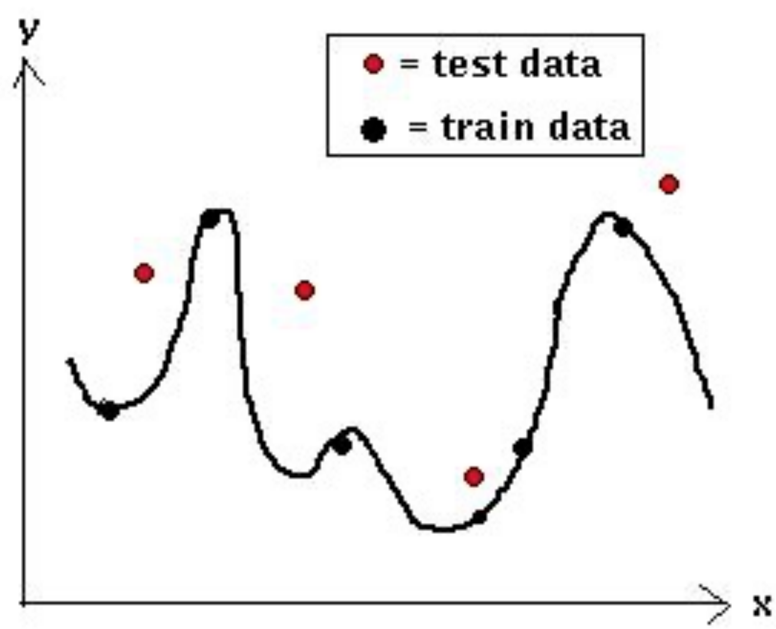




Generatlization and Overfitting

- 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





Model Selection & Generalization

- Learning is an ill-posed problem; data is not sufficient to find a unique solution
- Given d binary inputs, there are at most 2^D samples, and 2^{2^D} binary functions
- Each sample eliminates half of the functions;
- Thus, N samples leaves 2^{2^D-N} viable functions

- Not possible to check all functions. Need for inductive bias, assumptions about $\mathcal{H}$

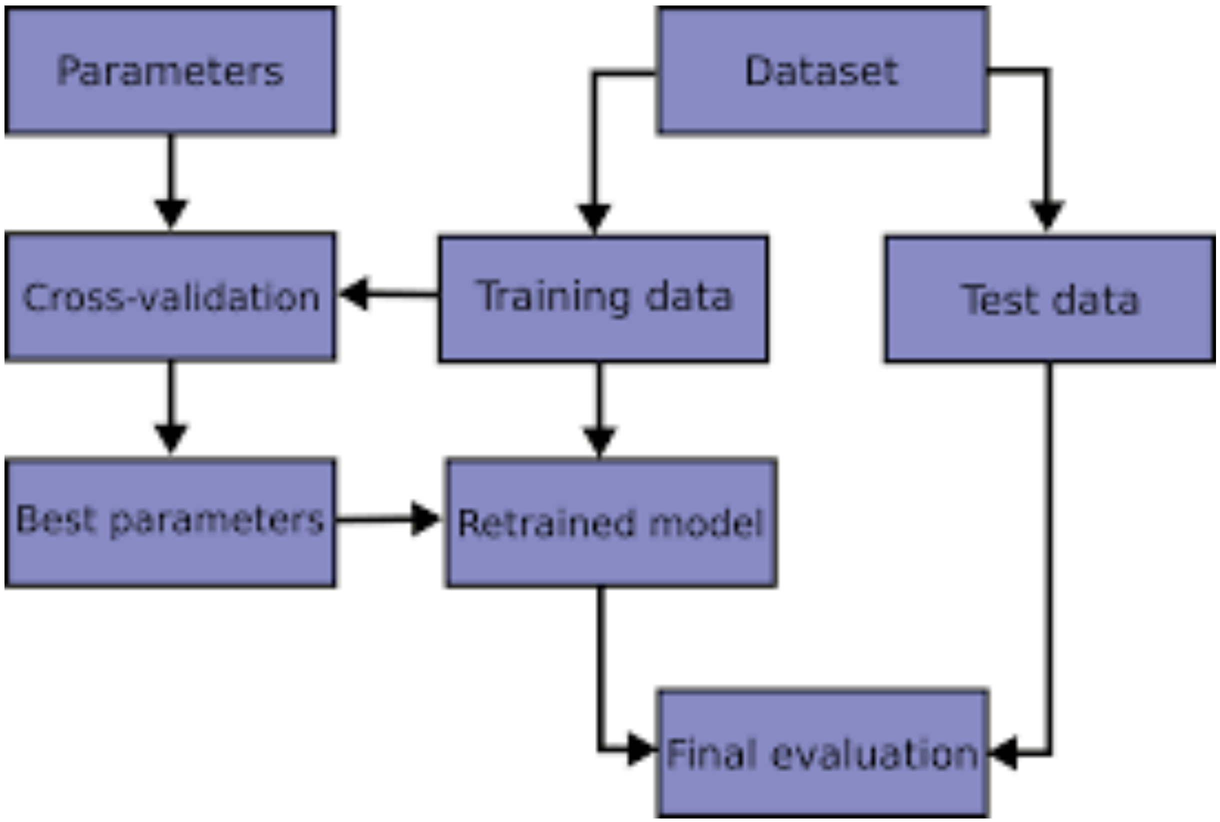
Cross-Validation

- To better estimate generalization error, we need data unseen during training. We split the data as
  - Training set (50%)
  - Validation set (25%)
  - Test set (25%)
- Resampling when there is few data

Cross-Validation

https://www.mathworks.com/discovery/cross-validation.html

Cross-Validation (good practice)



https://scikit-learn.org/stable/modules/cross_validation.html