Lecture 3: Linear models

Presentation · March 2019

CITATIONS
0

READS
187

1 author:

Alaa Tharwat
Fachhochschule Bielefeld
120 PUBLICATIONS **6,195** CITATIONS

SEE PROFILE

Lecture 3: Linear models

Alaa Tharwat

Lecture 3: Linear models

- Review of Lecture 2
- Input representation
- Linear classification
- Linear Regression
- Nonlinear transformation

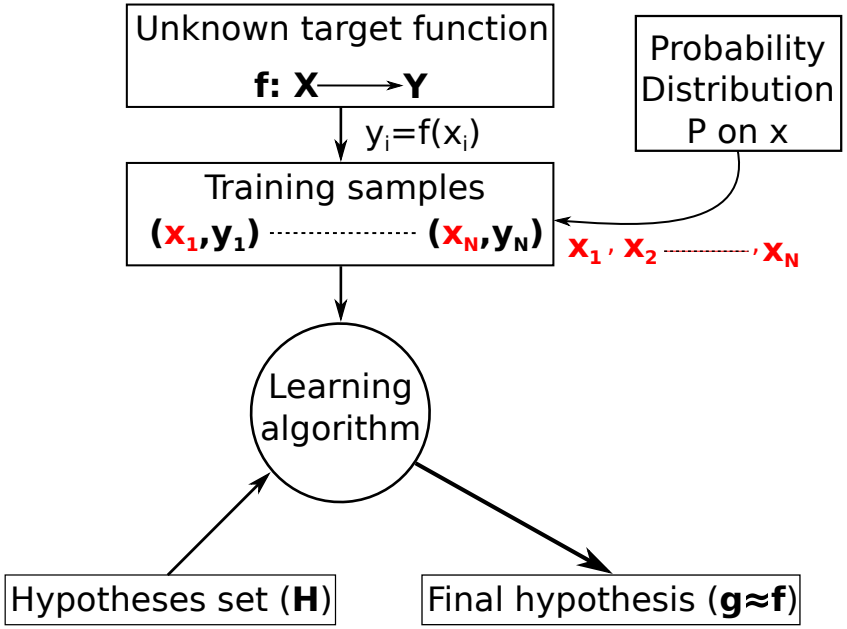
Lecture 3: Linear models

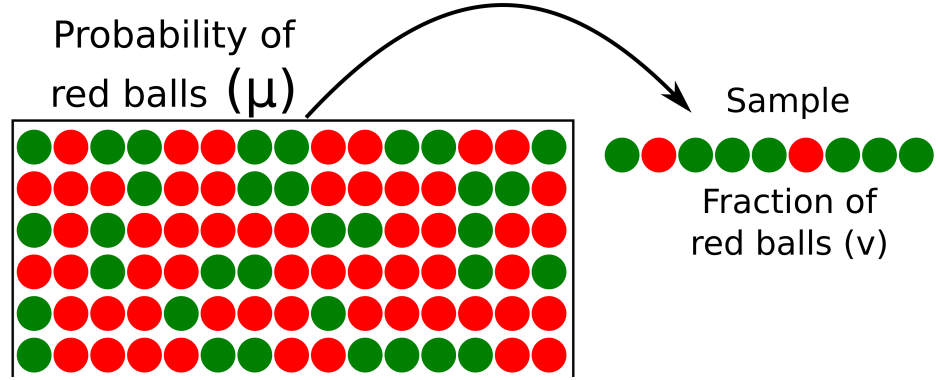
- Review of Lecture 2
- Input representation
- Linear classification
- Linear Regression
- Nonlinear transformation

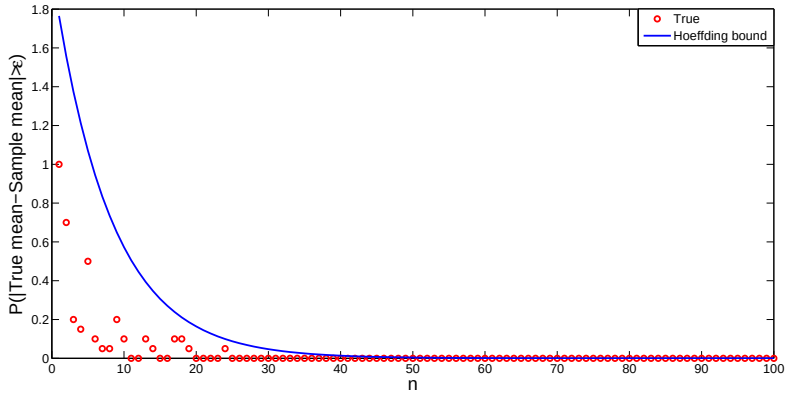
- Hoeffding's inequality,

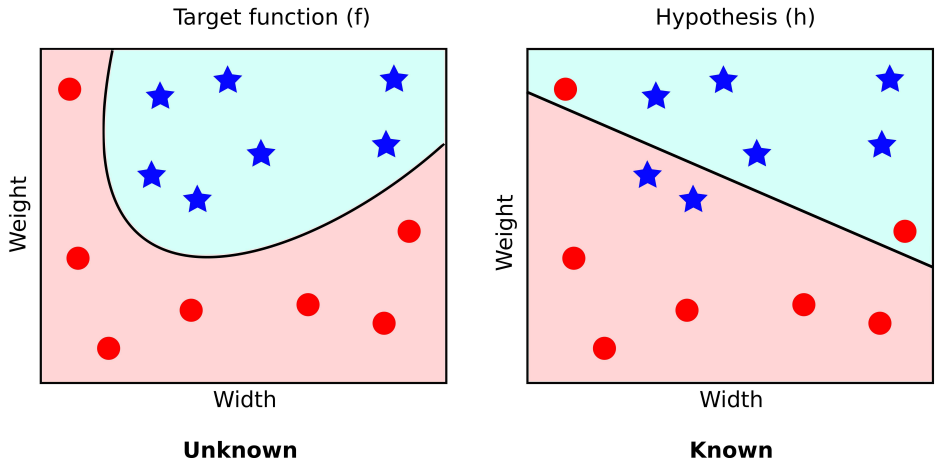
$$P[|v - \mu| > \epsilon] \leq 2e^{-2\epsilon^2 N}$$

- Tradeoff N, ϵ (large sample size N or looser gab ϵ)



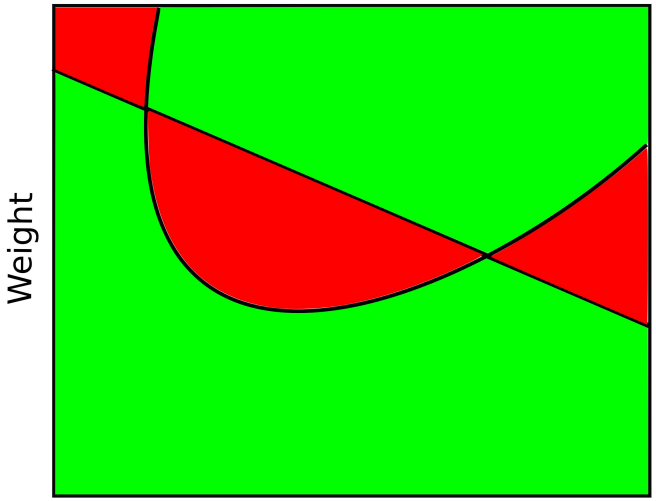


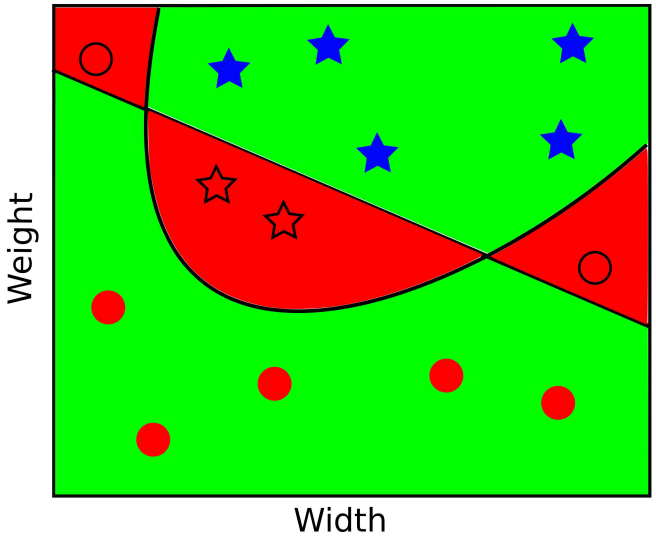




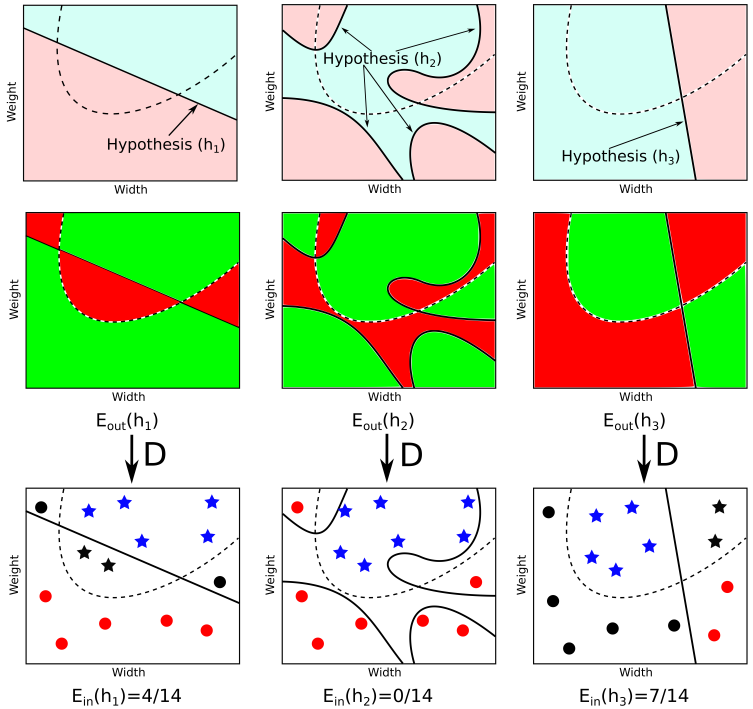
- In-sample Error ($E_{in}(h)$): how many samples in the *dataset* are misclassified, $E_{in}(h) = \frac{1}{N}\sum_{i=1}^{N}[h(\mathbf{x}_i) \neq f(\mathbf{x}_i)]$, N is the number of samples in the dataset (here is $E_{in}(h) = \frac{4}{14}$)
- Out-of-sample Error ($E_{out} = (h)$): red regions, $E_{out}(h) = P_x[h(x) \neq f(x)]$ (we cannot calculate it)

$$P[|E_{in}(h) - E_{out}(h)| > \epsilon] \leq 2e^{-2\epsilon^2 N}$$





Learning Model	Box Model
Input space X	Box of balls
x, where $h(x) = f(x)$ (h is right)	Green ball
x, where $h(x) \neq f(x)$ (h is wrong)	Red ball
$P(x)$	Randomly pick balls
dataset $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ with i.i.d. x_n	Sample of N balls of i.i.d. balls
Out-of-sample (E_{out})	$\mu \rightarrow$ probability of picking red ball in the box
In-sample (E_{in})	$v \rightarrow$ probability of picking red ball in a sample



$g \approx f$ inside D, but, not outside D

$$P[|E_{in}(g) - E_{out}(g)| > \epsilon]$$

$$\leq \sum_{i=1}^{M} P[|E_{in}(h_i) - E_{out}(h_i)|]$$

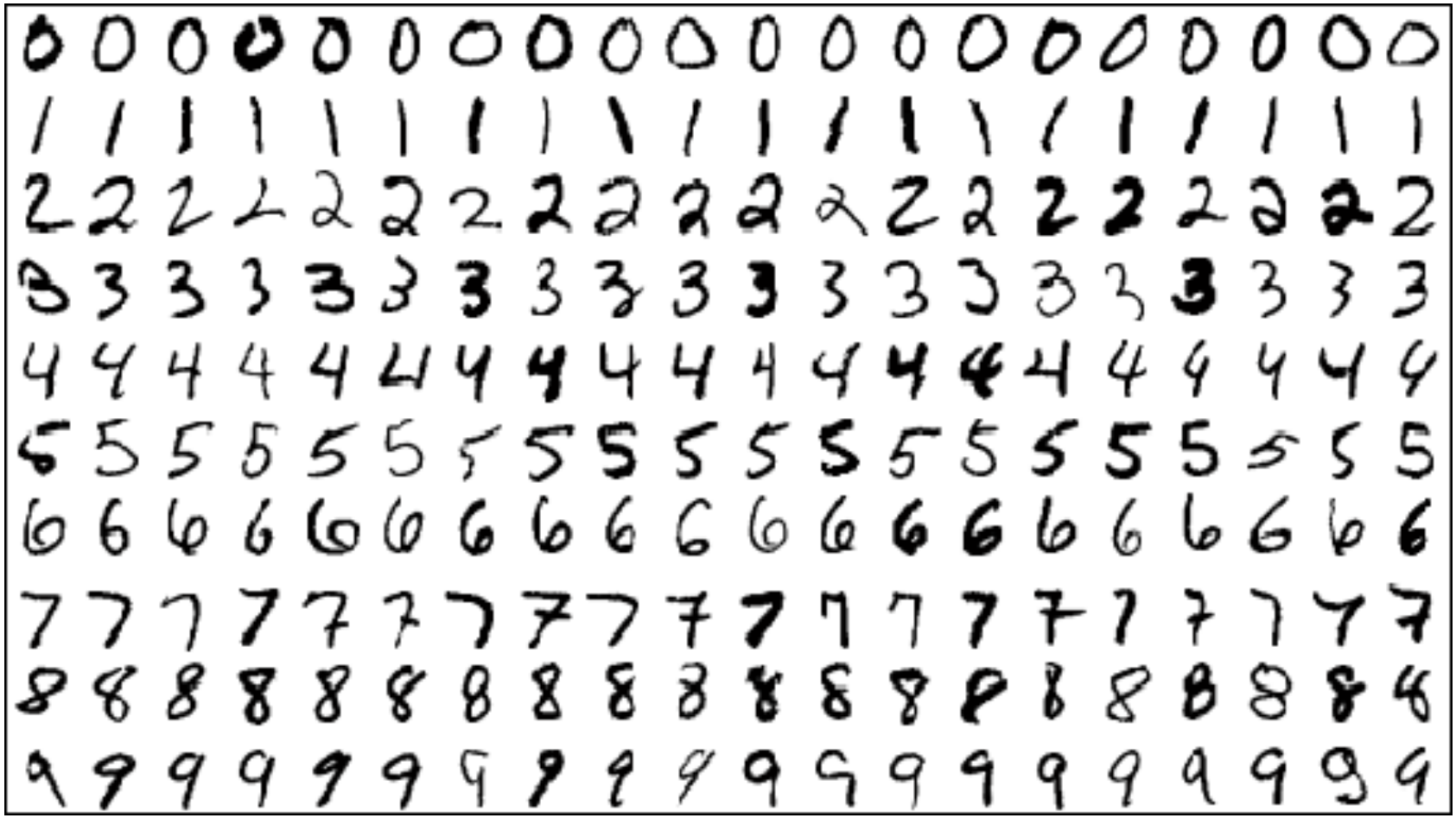
$$\leq \sum_{i=1}^{M} 2e^{-2\epsilon^2 N} \leq M 2e^{-2\epsilon^2 N}$$

x_n	y_n	g	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8
0 0 0	○	○	○	○	○	○	○	○	○	○
0 0 1	X	X	X	X	X	X	X	X	X	X
0 1 0	X	X	X	X	X	X	X	X	X	X
0 1 1	○	○	○	○	○	○	○	○	○	○
1 0 0	X	X	X	X	X	X	X	X	X	X
1 0 1		?	○	○	○	○	X	X	X	X
1 1 0		?	○	○	X	X	○	○	X	X
1 1 1		?	○	X	○	X	○	X	○	X

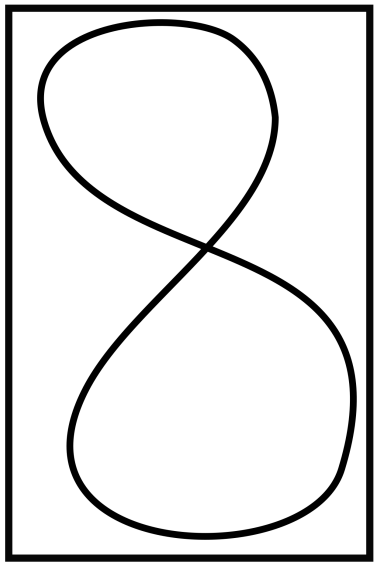
Lecture 3: Linear models

- Review of Lecture 2
- Input representation
- Linear classification
- Linear Regression
- Nonlinear transformation

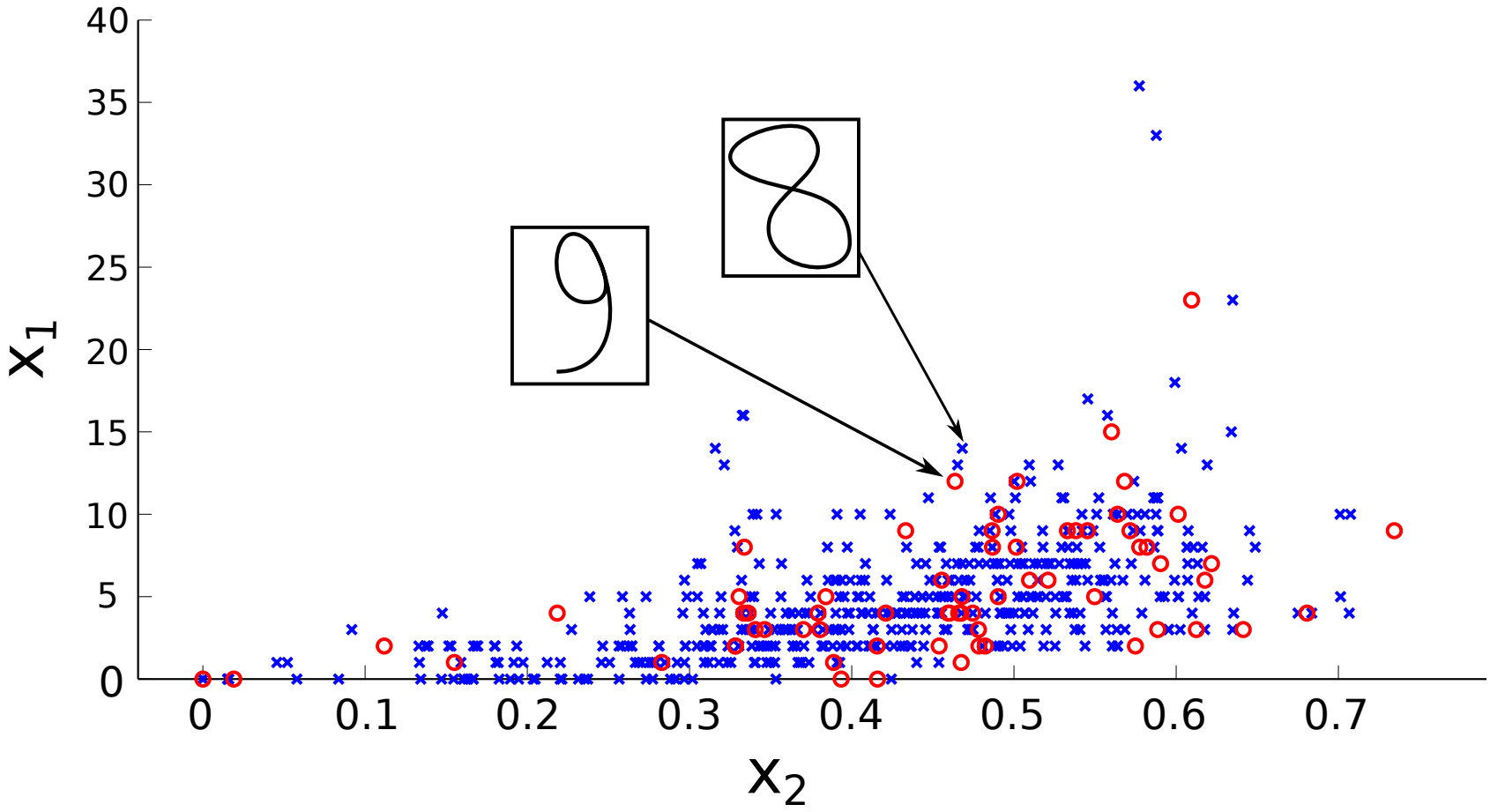
Handwriting digit dataset



- The image has many pixels (16×16), $(\mathbf{x} = \{x_0, x_1, \ldots, x_{256}\})$.
- Hence, the linear model has many parameters (weights) $(\mathbf{w} = \{w_0, w_1, \ldots, w_{256}\})$
- Extracting useful features such as symmetry (x_1) and ratio of foreground to background (x_2) reduces $\mathbf{x}$ as follows, $\mathbf{x} = \{x_0, x_1, x_2\}$
- The linear model has three parameters w_0, w_1, w_2



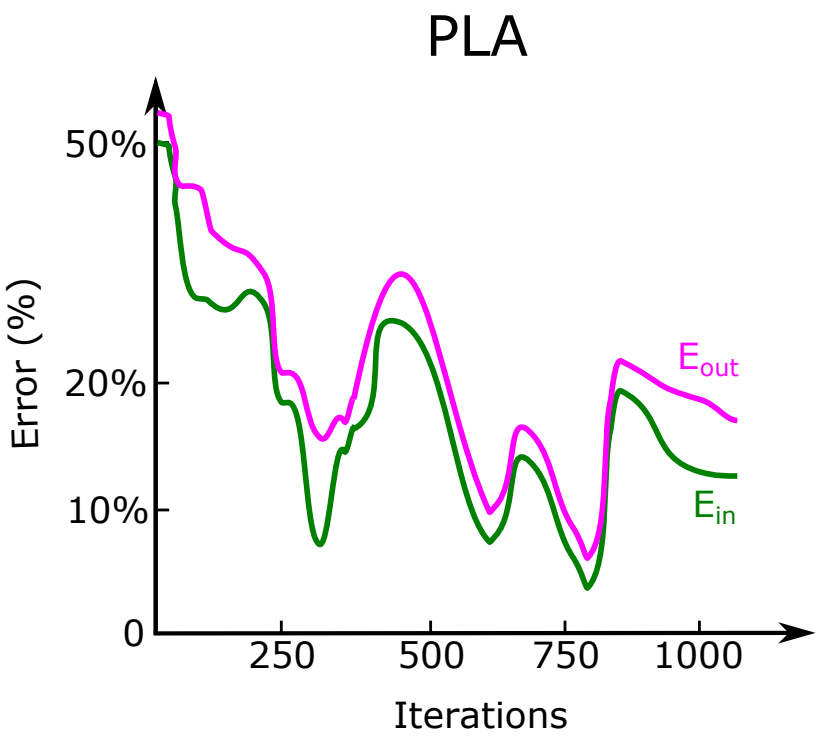
- Features illustrations (scatter plot)

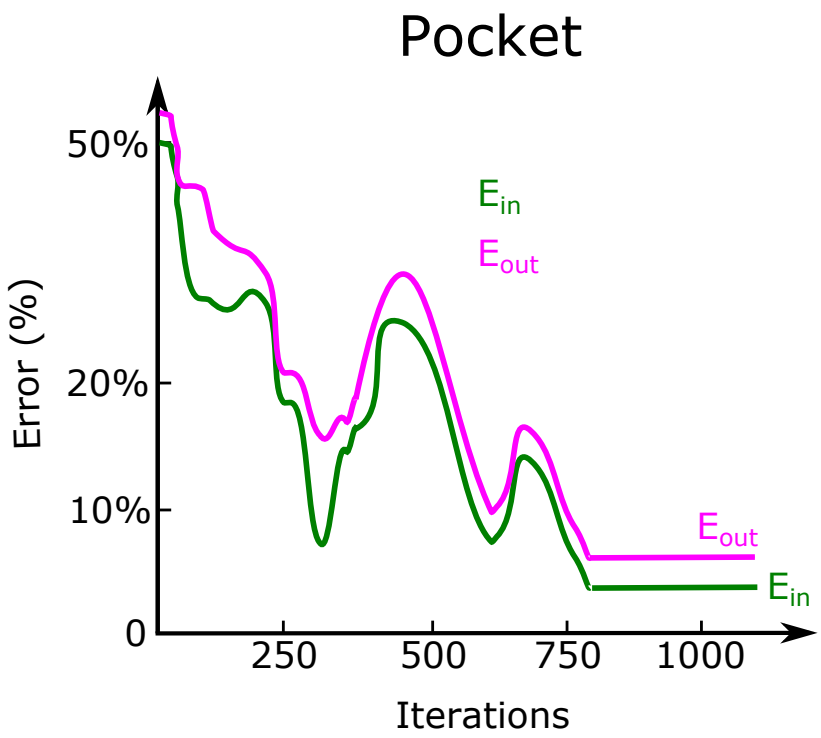


Lecture 3: Linear models

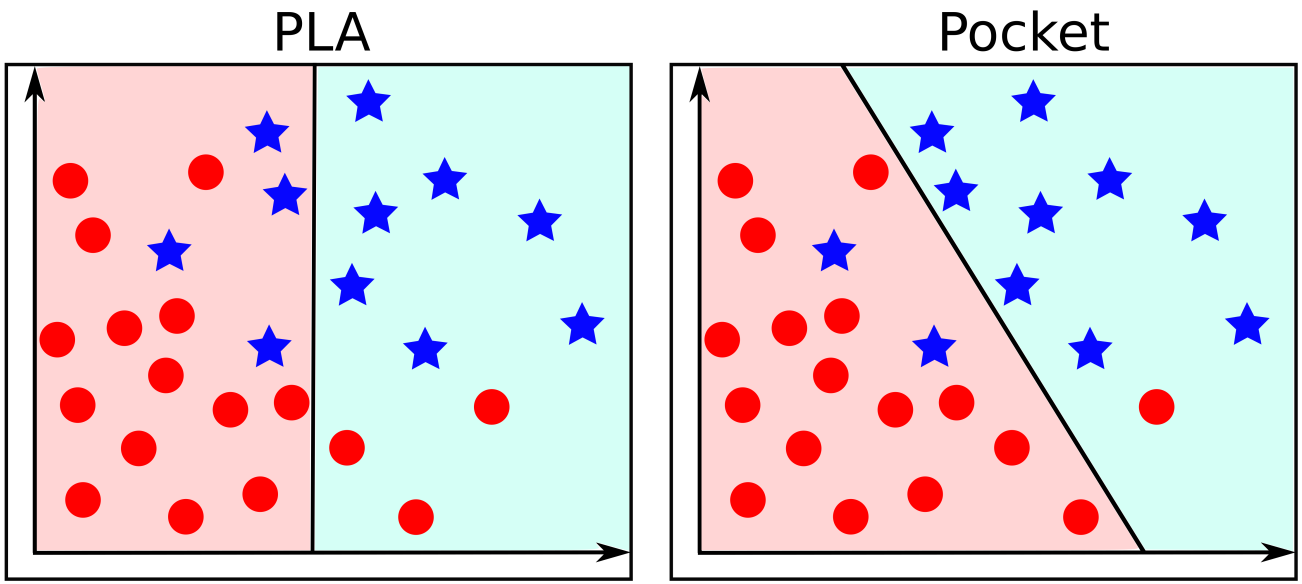
- Review of Lecture 2
- Input representation
- Linear classification
- Linear Regression
- Nonlinear transformation

- Given a dataset and we apply PLA on it, and the data is nonlinearly separable
- There is a difference between E_{in} and E_{out}; but, E_{out} tracks E_{in} very well, so the model generalizes well
- Pocket algorithm is a modified version from the PLA, it saves the hypothesis/solution with the minimum E_{in} (best solution) in your pocket





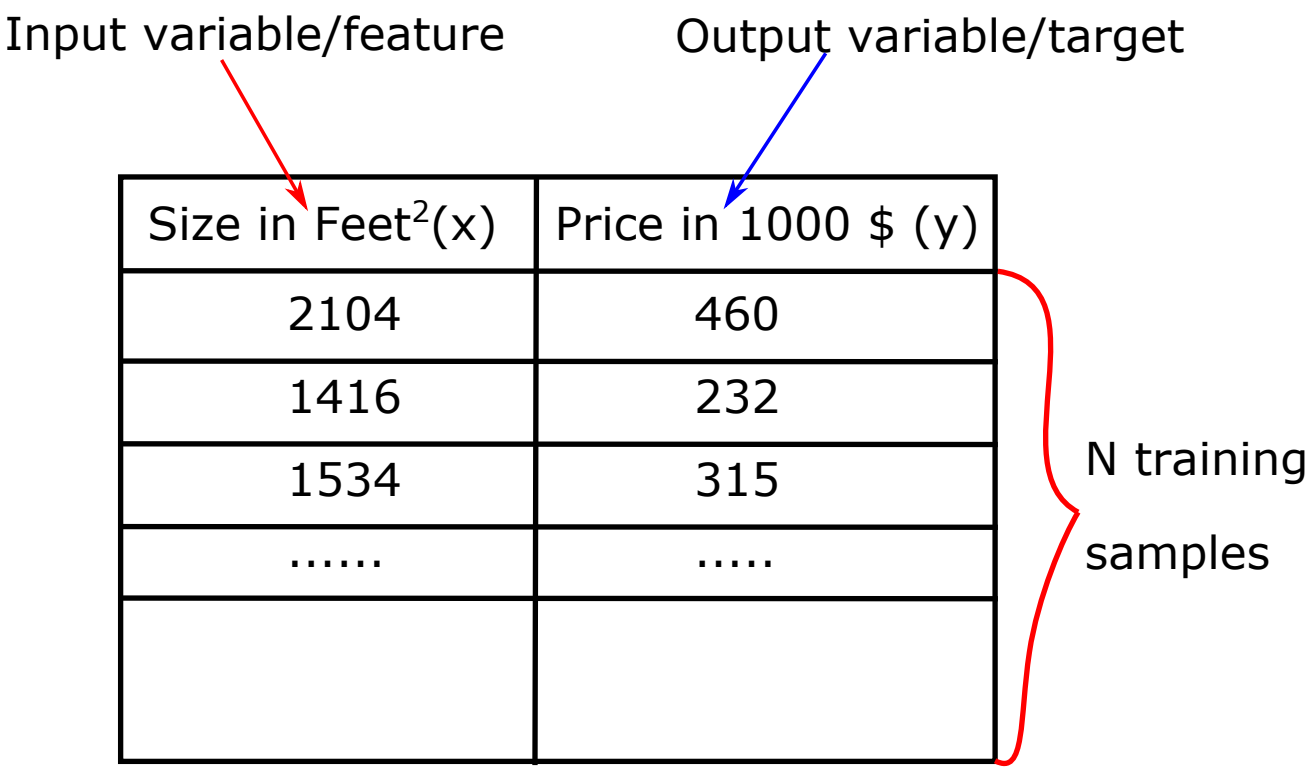
- Classification or decision boundary: PLA vs. Pocket



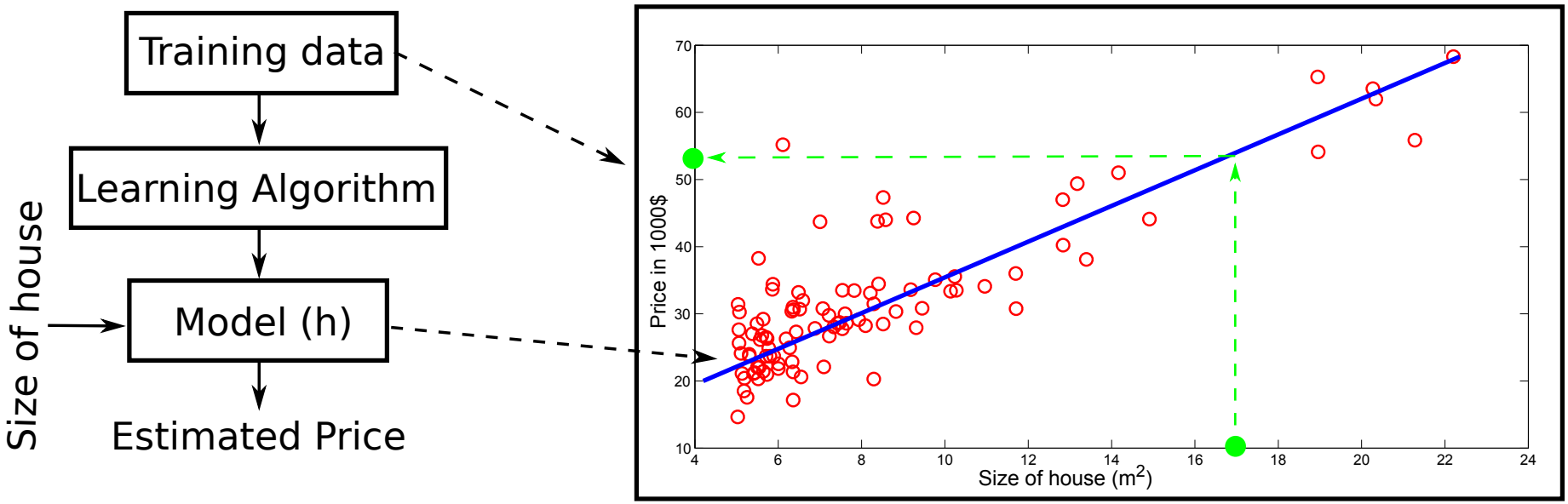
Lecture 3: Linear models

- Review of Lecture 2
- Input representation
- Linear classification
- Linear Regression
- Nonlinear transformation

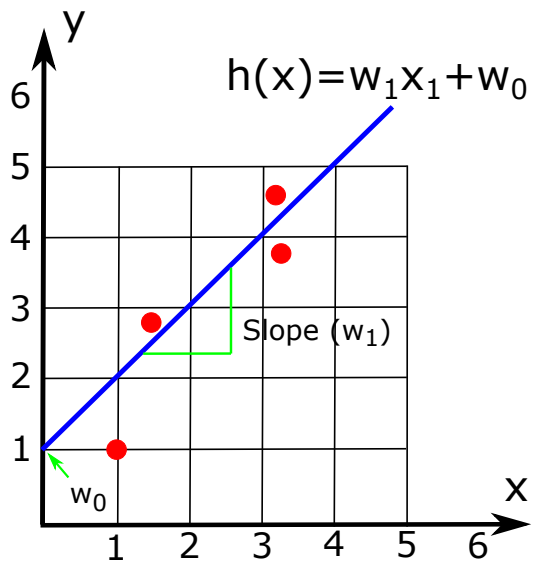
- In the classification, the outputs are class labels (e.g. in credit approval example, the output is Yes/No)
- Regression $\equiv$ real-valued output (e.g. in credit approval example, the output is the dolar amount ($y_n \in R$))
- Linear regression model: $h(\mathbf{x}) = \sum_{i=0}^{d} w_i x_i = \mathbf{w}^T \mathbf{x}$



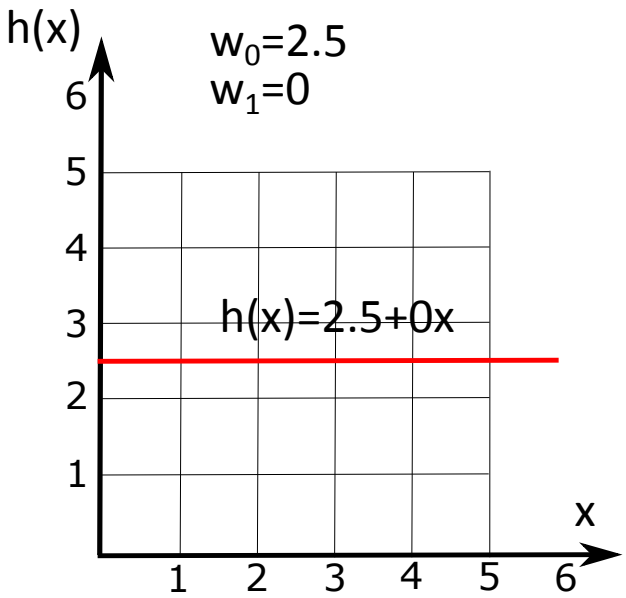
- The training data are used for training the model h.

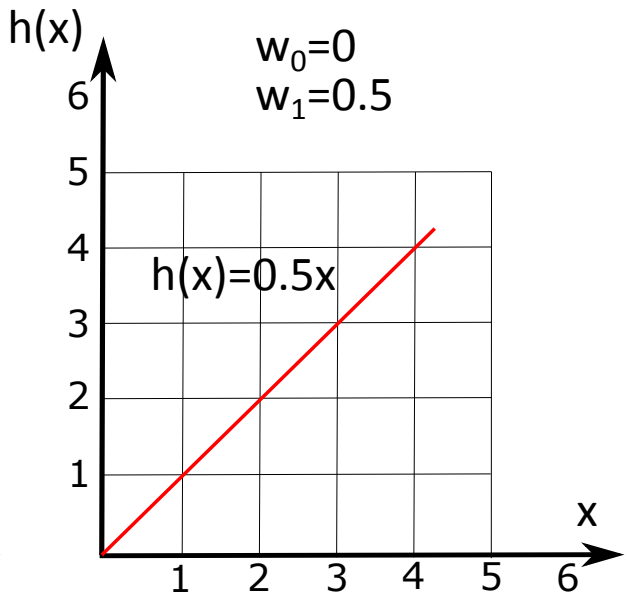


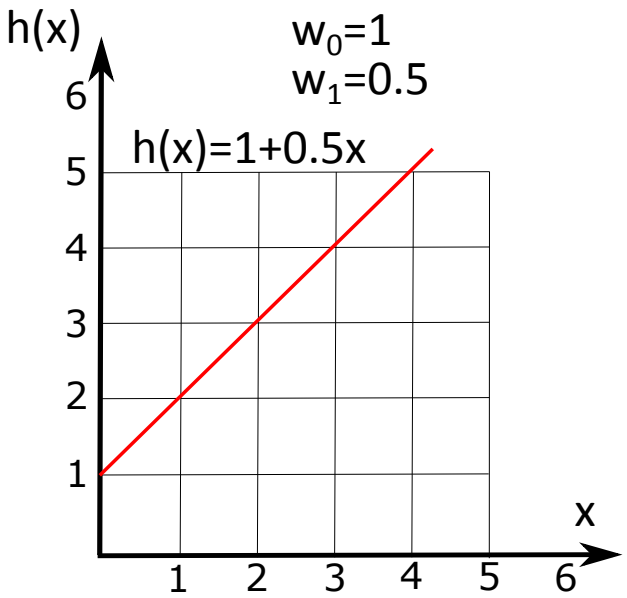
- In linear regression model, given one feature (as in our example); hence, the model will be $h(x) = w_1 x_1 + w_0$
  - w_1 is the slope of the regression line
  - w_0 is the shift of the line
  - w_0 and w_1 are called regression coefficients
- How to measure the error?
- the **cost function** is used to measure the error, it is defined as follows, $(h(\mathbf{x}) - f(\mathbf{x}))^2$ (this is called *squared error function*)
- In-sample error: $E_{in} = \frac{1}{N} \sum_i^N (h(\mathbf{x}_i) - y_i)^2$ or $E_{in}(\mathbf{w}) = (\mathbf{w}^T \mathbf{x} - y)^2$
- $\mathbf{w}_{lin} = \underset{\mathbf{w} \in R^{d+1}}{\arg\min}\ E_{in}(\mathbf{w})$



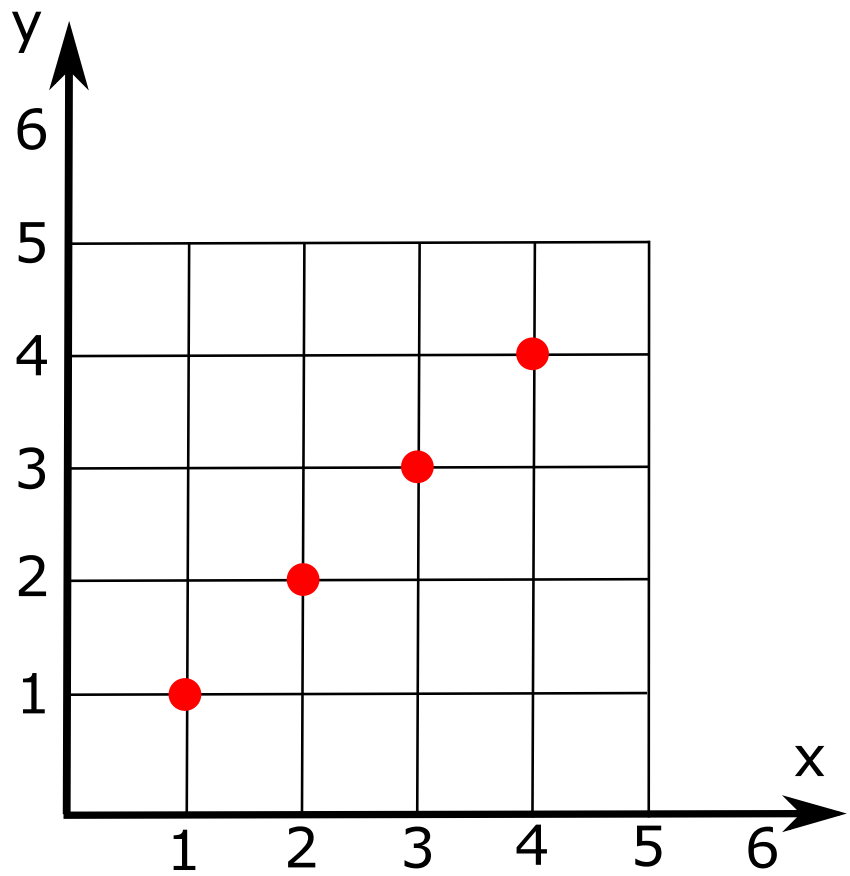
- $h(\mathbf{x}) = w_1 x + w_0$
- The idea of linear regression is to search for w_0 and w_1 so that the line (hypothesis) $h(\mathbf{x})$ is close to y for our training samples. In other words, linear regression finds the line which **minimizes** the vertical distances, i.e., the distance between y values and their corresponding estimated values on the line/hypothesis, these vertical distances are called **residuals**.



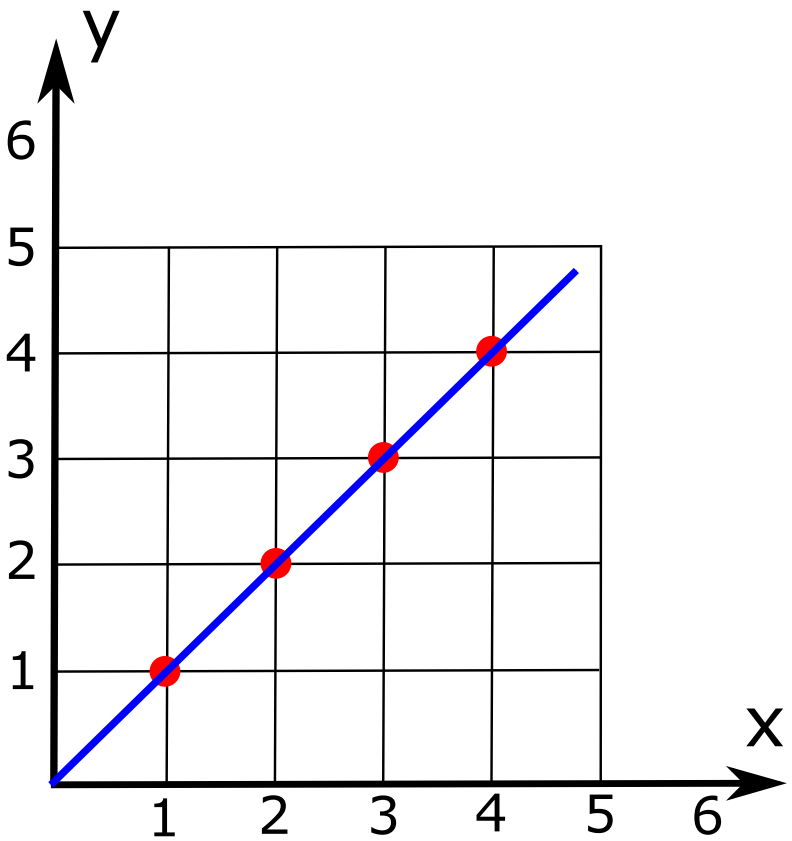


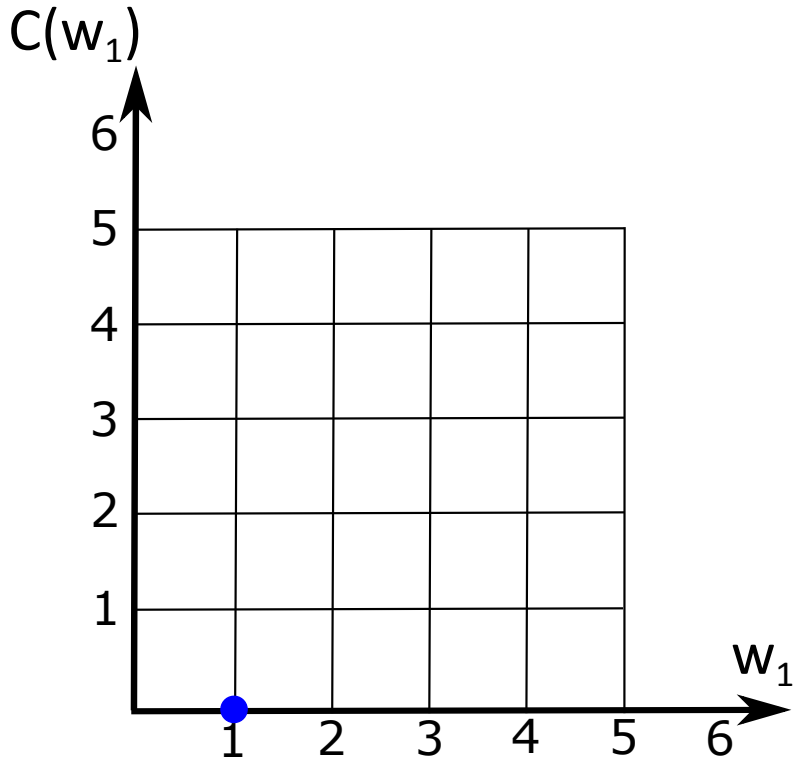


- Given four training data samples as in the figure,
- For simplicity, assume $w_0 = 0$, hence, changing w_1 changes only the slope of the line.

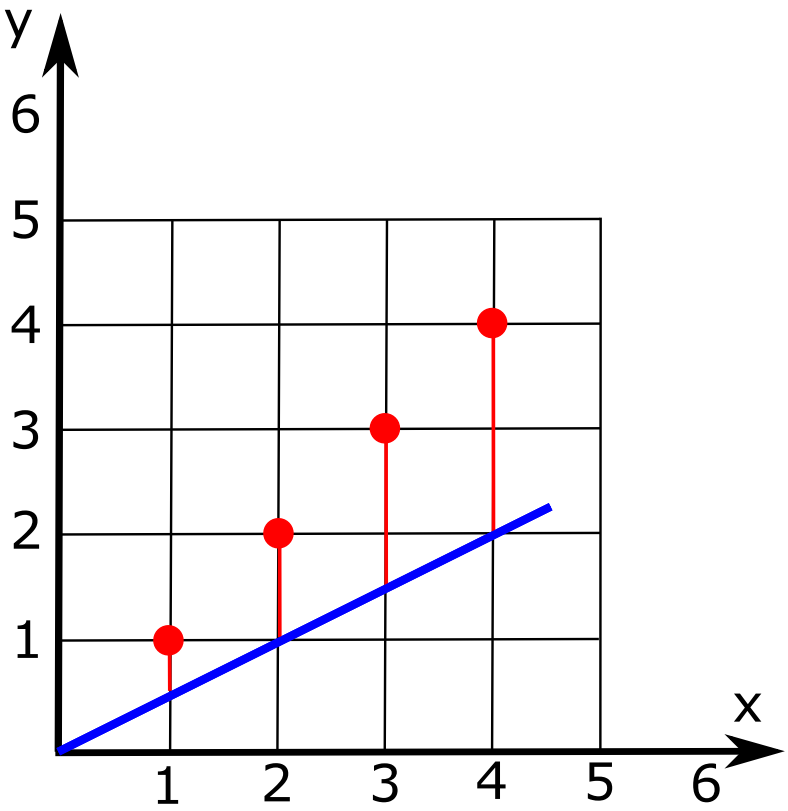


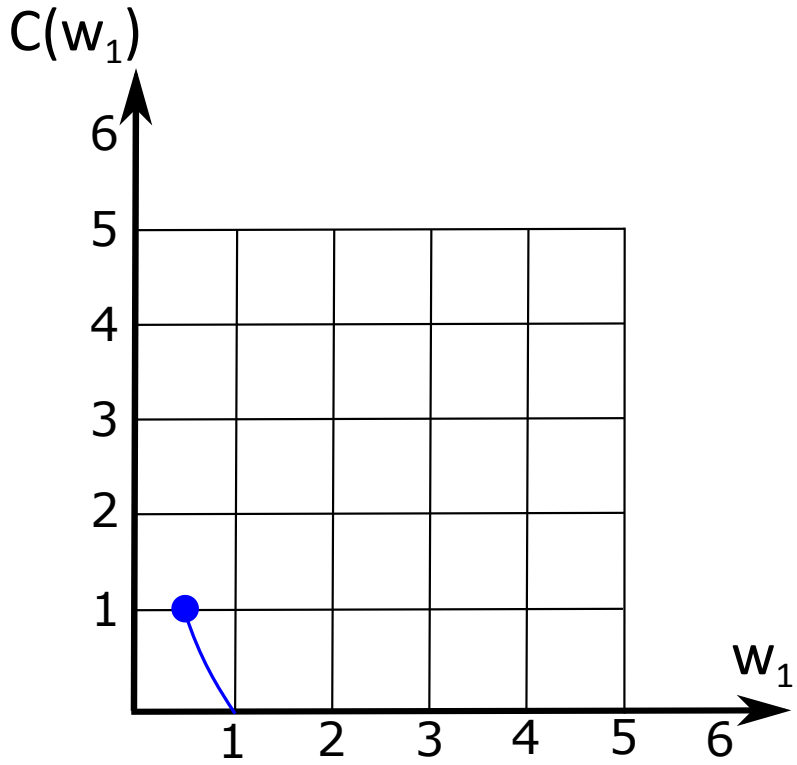
- Assume the line $h(\mathbf{x}) = \mathbf{x}$, i.e. $w_1 = 1$,
- $C = \frac{1}{2N} \sum_i^N (h(\mathbf{x}_i) - y_i)^2 = \frac{1}{2N} \sum_i^{N=4} (w_1 \mathbf{x}_i - y_i)^2$,
- $C = \frac{1}{2 \times 4} (0^2 + 0^2 + 0^2 + 0^2) = 0$.



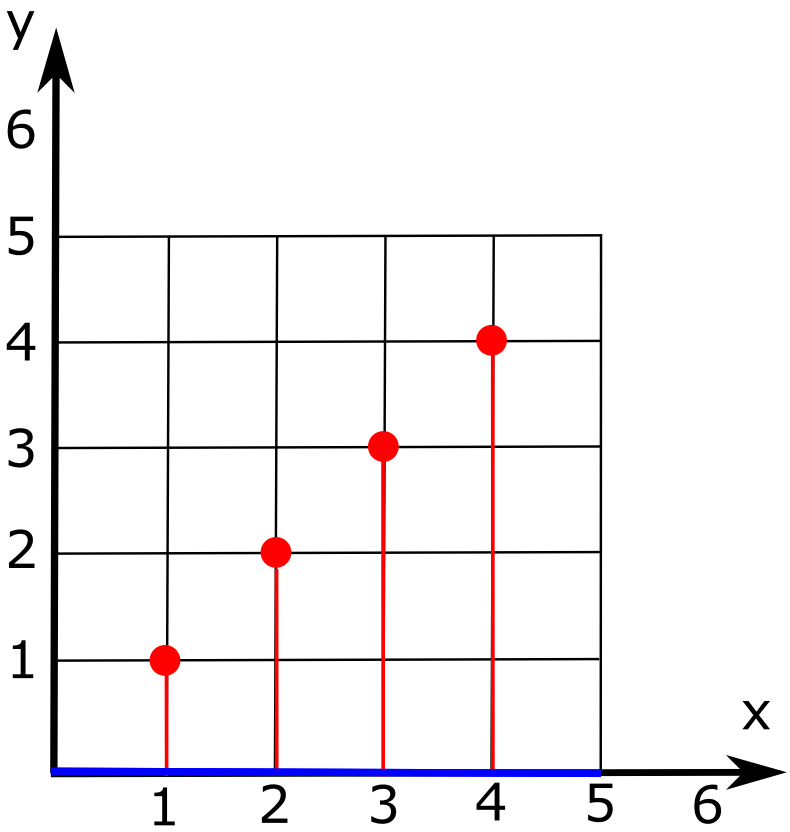


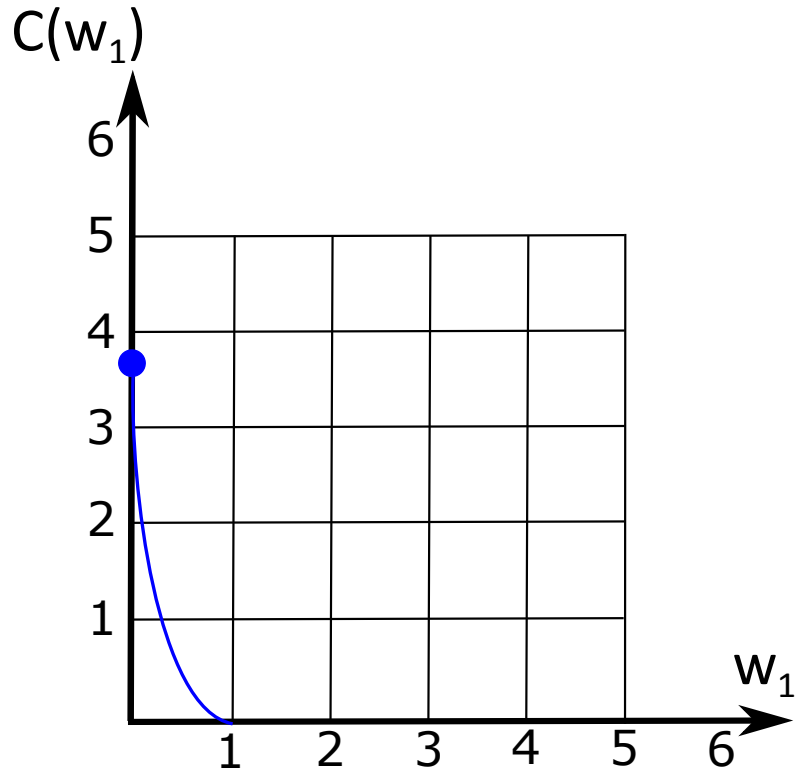
- Assume the line $h(\mathbf{x}) = \frac{1}{2}\mathbf{x}$, i.e. $w_1 = \frac{1}{2}$,
- $C(w_1) = \frac{1}{2\times 4}(0.5^2 + 1^2 + 1.5^2 + 2^2) = \frac{7.75}{8} = 0.96875.$



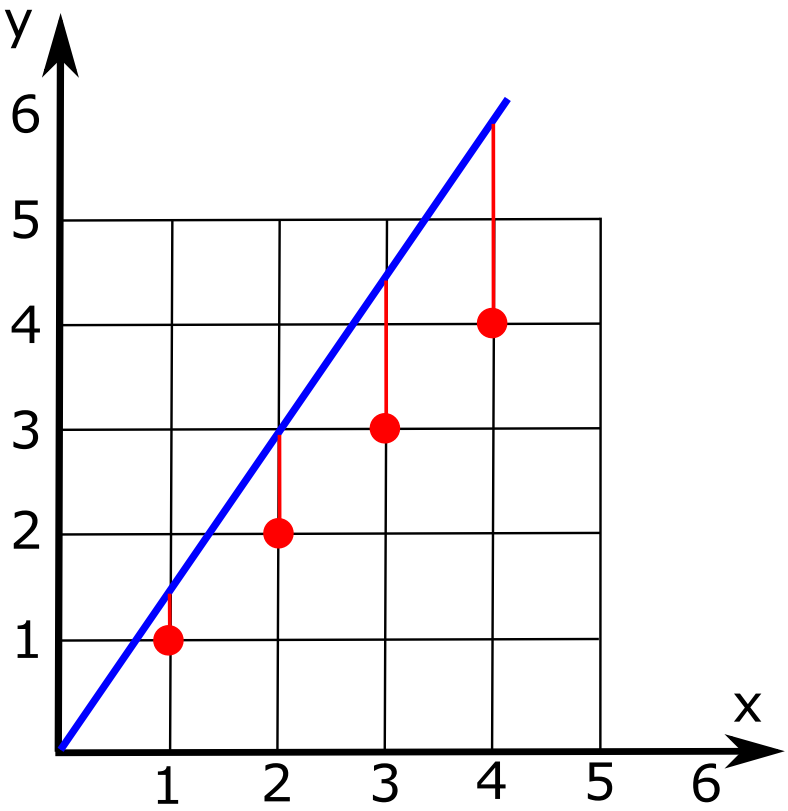


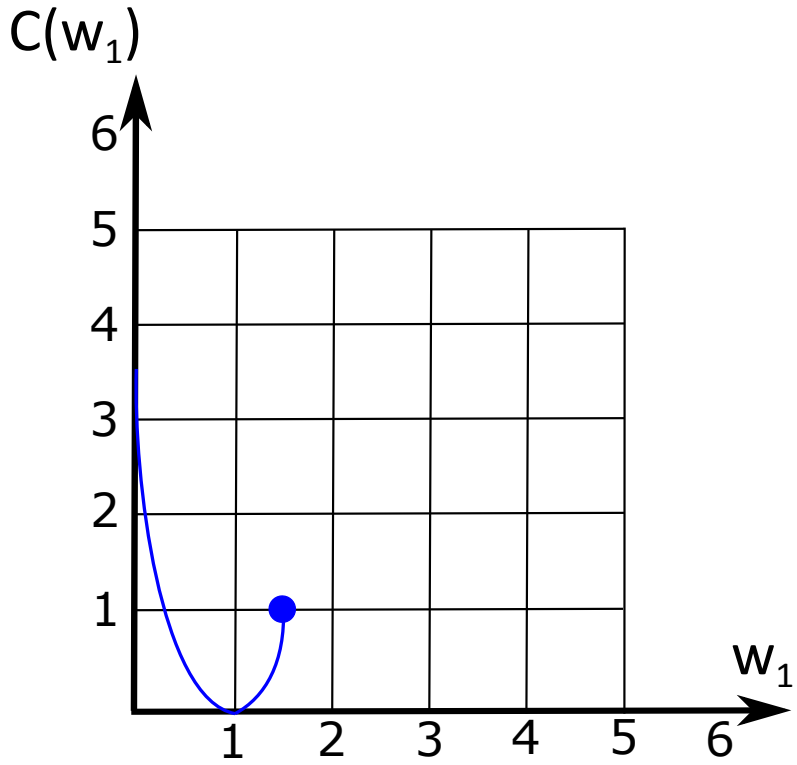
- Assume the line $h(\mathbf{x}) = 0$, i.e. $w_1 = 0$,
- $C(w_1) = \frac{1}{2\times 4}(1^2 + 2^2 + 3^2 + 4^2) = \frac{30}{8} = 3.75.$



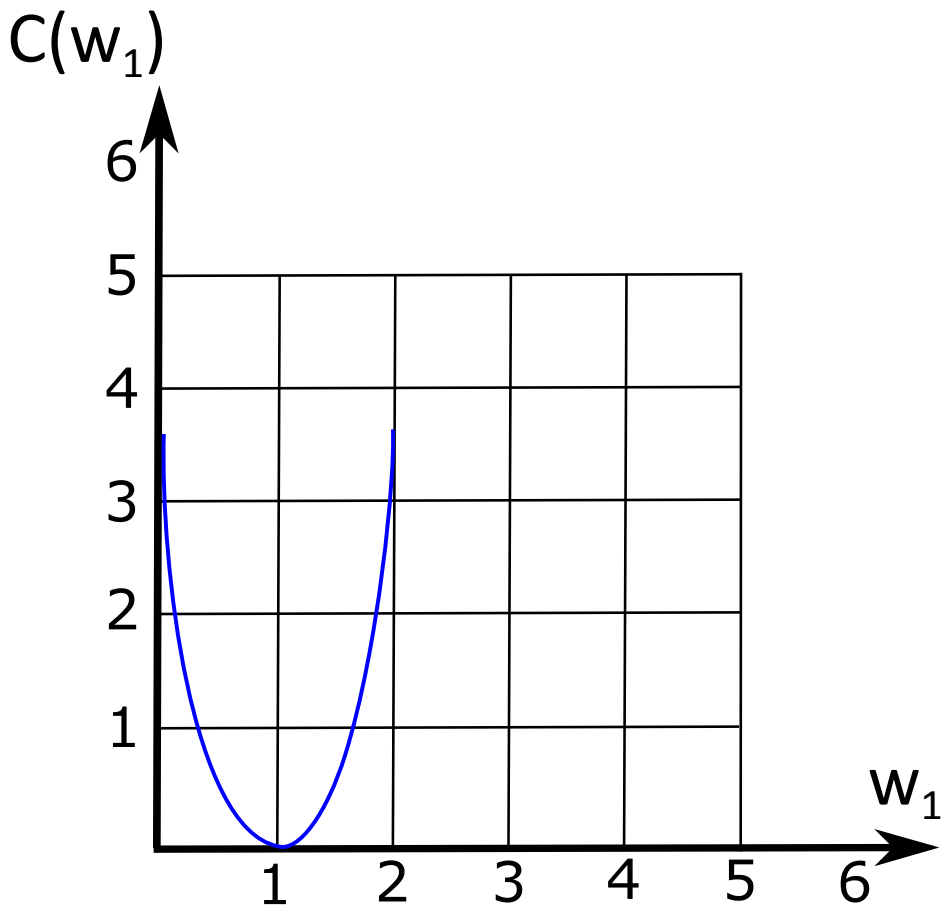


- Assume the line $h(\mathbf{x}) = 1.5x$, i.e. $w_1 = 1.5$,
- $C(w_1) = \frac{1}{2\times 4}(0.5^2 + 1^2 + 1.5^2 + 2^2) = \frac{7.75}{8} = 0.96875.$

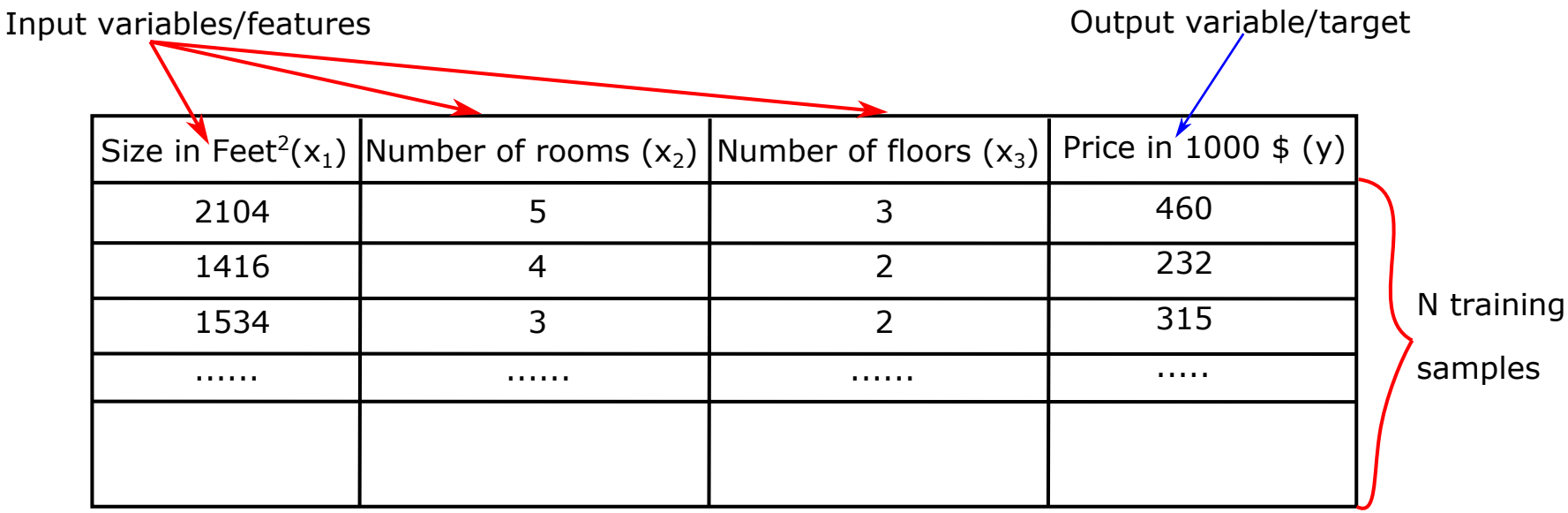




- The goal in linear regression is to find the minimum cost
- The cost function is convex,



- Instead of using only one feature, we can use d features $(x_1, x_2, \ldots, x_d)$. Hence, the dependent variable Y is related to d variables,
- the regression model still linear but with multiple features $(h(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_d x_d)$.



- For convenience of notation, define $x_0 = 1$, hence $\mathbf{X} \in \mathcal{R}^{d+1}$, $\mathbf{w} \in \mathcal{R}^{d+1}$

$$\mathbf{X} = \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix}, \ \mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_n \end{bmatrix} \tag{1}$$

$$h(x) = w_0 \cancel{x_0} \overset{1}{+} w_1 x_1 + w_2 x_2 + \cdots + w_d x_d = \mathbf{w}^T \mathbf{X} \tag{2}$$

- Normal equation can be used for calculating the parameters of linear regression.

$$\mathbf{X} = \begin{bmatrix} x_0 & x_1 & \dots & x_d \end{bmatrix} \tag{3}$$

$$\mathbf{X} = \begin{bmatrix} 1 & x_1^1 & \dots & x_1^d \\ 1 & x_2^1 & \dots & x_2^d \\ \vdots & \vdots & \dots & \vdots \\ 1 & x_N^1 & \dots & x_N^d \end{bmatrix}, \ \mathbf{y} = \begin{bmatrix} y^1 \\ y^2 \\ \vdots \\ y^N \end{bmatrix} \tag{4}$$

where

- $\mathbf{X} \in (N \times (d+1))$ and $\mathbf{y} \in (N \times 1)$

- The goal of any model is reduce the error, i.e. minimize J,

$$C(\mathbf{w}) = \frac{1}{2N}\sum_{i=1}^{N}(h(\mathbf{x}_i) - y(\mathbf{x}_i))^2 = \frac{1}{2N}\left\|\mathbf{w}^T\mathbf{X} - \mathbf{y}\right\|^2$$

- $C = E_{in}$ can be minimized as follows:

$$\begin{aligned}\bigtriangledown C(\mathbf{w}) = 0 \Rightarrow &= 2\frac{1}{2N}\mathbf{X}\left\|\mathbf{w}^T\mathbf{X} - \mathbf{y}\right\| = 0 \\ &= \mathbf{X}^T\mathbf{X}\mathbf{w} = \mathbf{X}^T\mathbf{y} \Rightarrow \mathbf{w} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \rightarrow \\ \mathbf{w} &= \mathbf{X}\dagger\,\mathbf{y}\end{aligned}$$

- $\mathbf{w}$ does not produce $\mathbf{y}$, but $\hat{\mathbf{y}}$ as follows: $\hat{\mathbf{y}} = \mathbf{X}\mathbf{w}$
- $\mathbf{X}\dagger \in \mathcal{R}^{(d+1)\times N}$ is called "pseudo-inverse" of $\mathbf{X}$.
- $\mathbf{X}\dagger\,\mathbf{X} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{X} = \mathbf{I}$; thus, we can say $\mathbf{X}\dagger$ is also inverse of $\mathbf{X}$

- The values of **w** parameters are calculated as follows,

$$\mathbf{w} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \tag{5}$$

- this can be interpreted as follows:
  - As shown before, $\mathbf{Xw} = \mathbf{y}$, but $\mathbf{X}$ is not a square matrix, so we need to use $\mathbf{X}^T$ to have a square matrix as follows, $\mathbf{X}^T\mathbf{Xw} = \mathbf{X}^T\mathbf{y} \rightarrow (\mathbf{X}^T\mathbf{X})\mathbf{w} = \mathbf{X}^TY$, and then the value of **w** can be calculated as follows, $\mathbf{w} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$.

- Given six ($N = 6$) points as shown in the table, try to find the line which has the least square errors

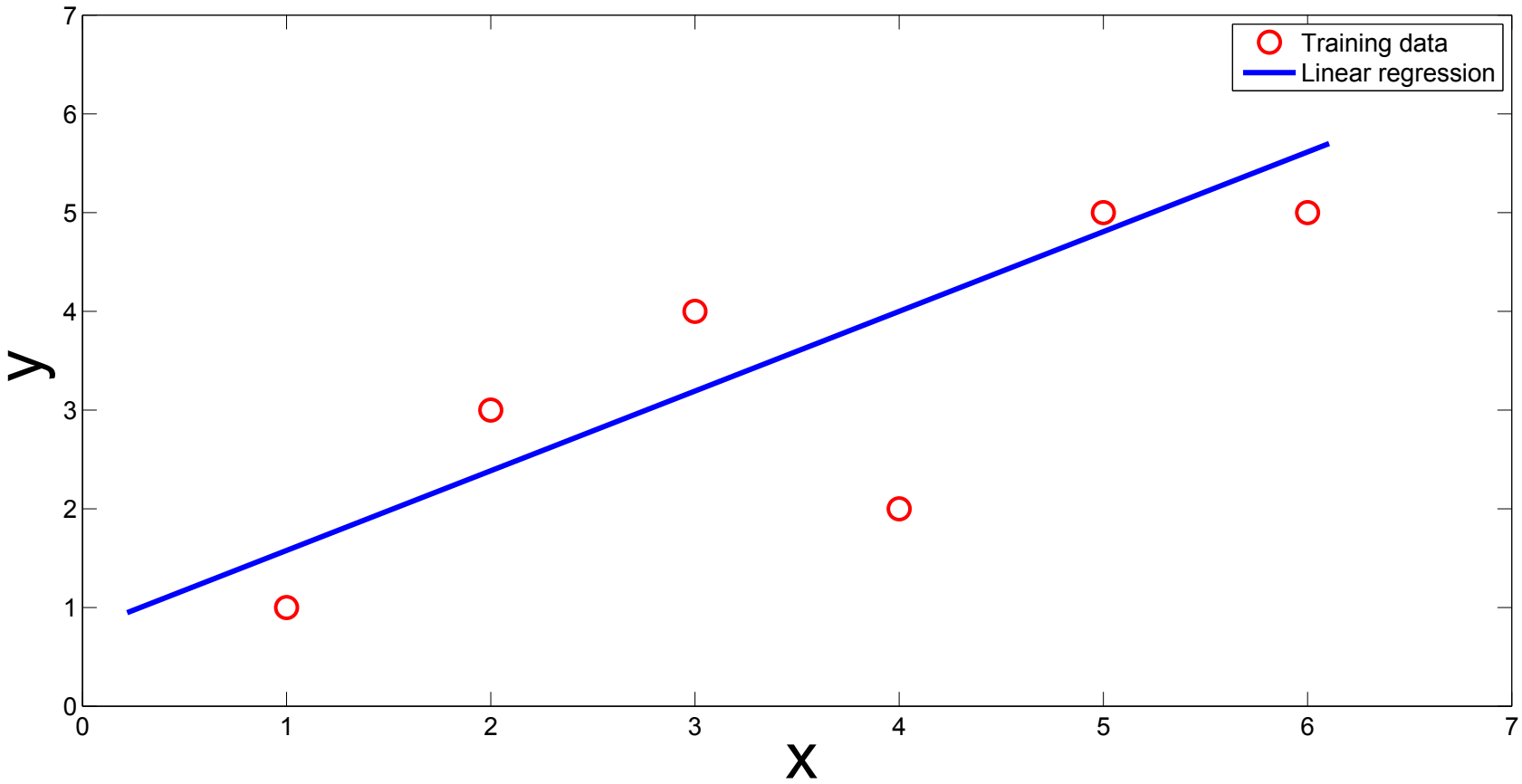
i	x	y
1	1	1
2	2	3
3	3	4
4	4	2
5	5	5
6	6	5

- Normal equation can be used for solving our example as follows:

$$\mathbf{X}^T\mathbf{X} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 & 6 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 1 & 4 \\ 1 & 5 \\ 1 & 6 \end{bmatrix} = \begin{bmatrix} 6 & 21 \\ 21 & 91 \end{bmatrix} \tag{6}$$

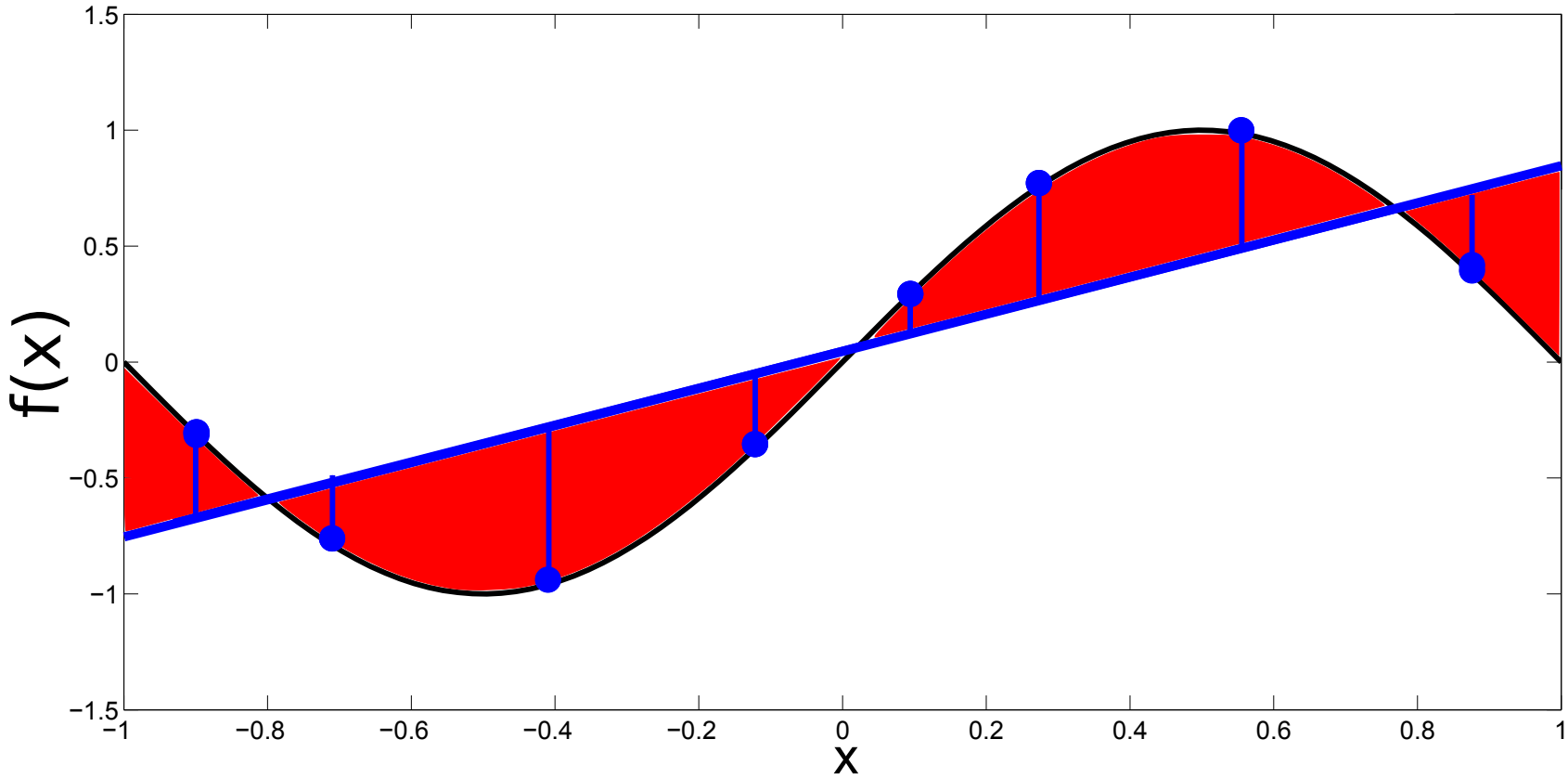
$$\mathbf{w} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} = (\begin{bmatrix} 6 & 21 \\ 21 & 91 \end{bmatrix})^{-1} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 & 6 \end{bmatrix} \begin{bmatrix} 1 \\ 3 \\ 4 \\ 2 \\ 5 \\ 5 \end{bmatrix} = \begin{bmatrix} 0.9333 \\ 0.6875 \end{bmatrix} \tag{7}$$

- $w_0 = 0.9333$ and $w_1 = 0.6875$.


Training data
Linear regression
y
x
0
1
2
3
4
5
6
7

What is the E_{in} and E_{out} in regression

- E_{in}: is the cost function, $E_{in} = \frac{1}{N}\sum_{i=1}^{N}(h(\mathbf{x}_i) - y(\mathbf{x}_i))^2$
- E_{out}: we cannot measure it!!

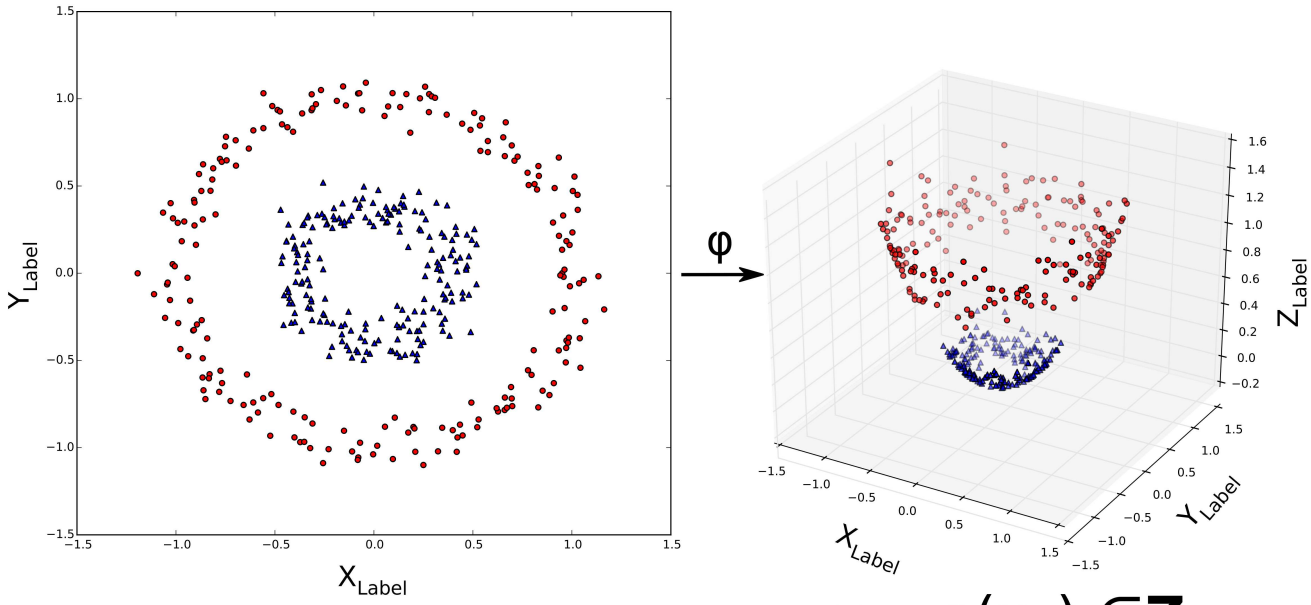


- Binary classification has two classes (± 1), can we use regression instead of classification?
  - Answer: practically, yes. In classification, sign($\mathbf{w}^T$)x_n is likely to agree with $y_n = \pm 1$. In regression, $\mathbf{w}^T x_n \approx y_n = \pm 1$
  - But, -3 is classified as a correct sample, but regression considers it wrong and tries to reduce it to -1

Lecture 3: Linear models

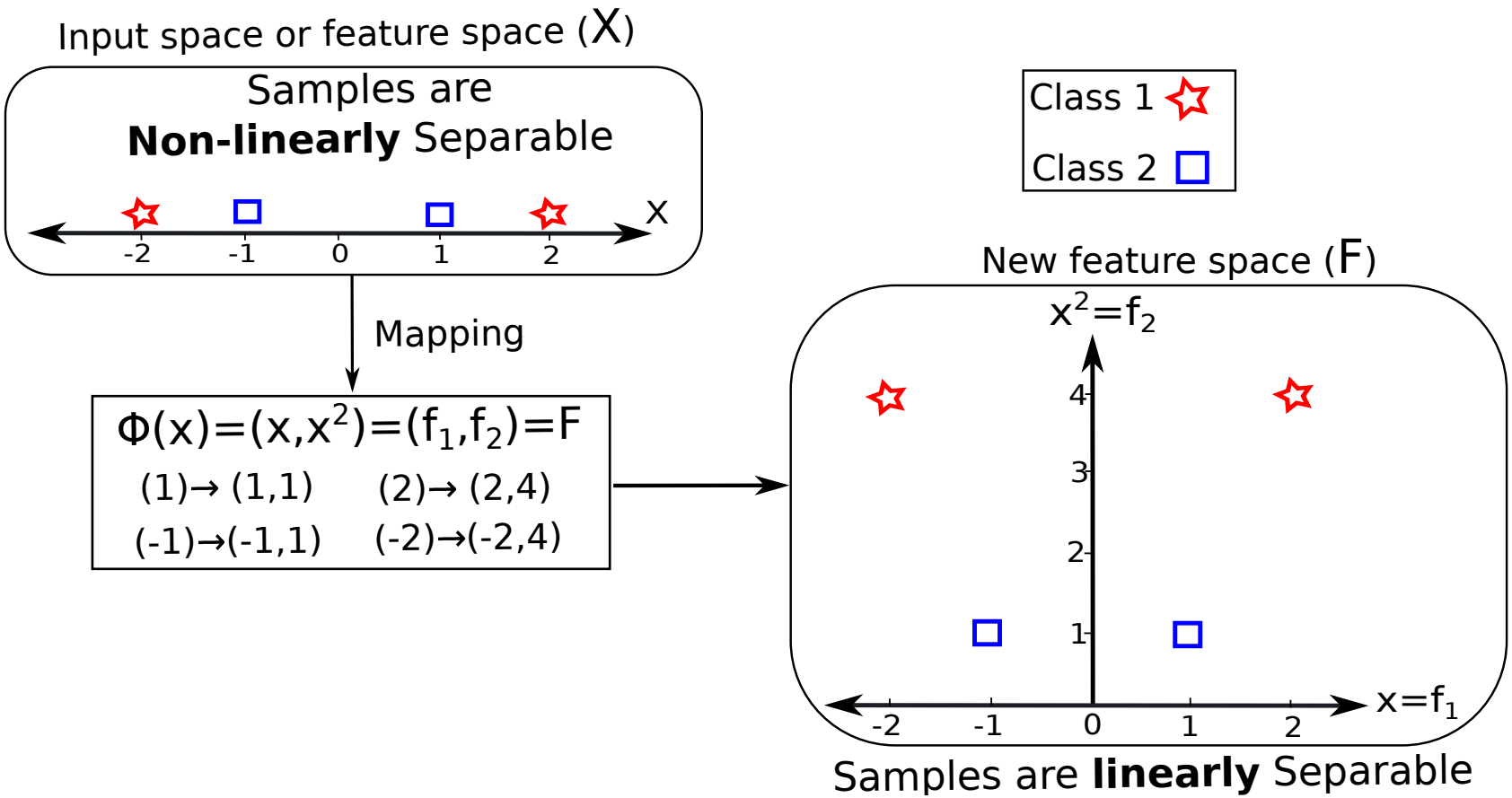
- Review of Lecture 2
- Input representation
- Linear classification
- Linear Regression
- Nonlinear transformation

- We cannot classify nonlinear separable data using linear classifiers
- Transforming data into higher spaces help for finding linear classifier
- $\mathbf{w}^T\mathbf{x}$ is linear in $\mathbf{w}$, and the transformation preserves this linearity
- In the figure below, the data are transformed as follows, $(x, y, x^2 + y^2)$
- In the figure below:
  - The point $(-1, -0.5)$ transformed to $(-1, -0.5, 1.25)$ (red class)
  - The point $(-0.5, 0)$ transformed to $(-0.5, 0, 0.25)$ (blue class)



$x_n \in \mathbf{X}$

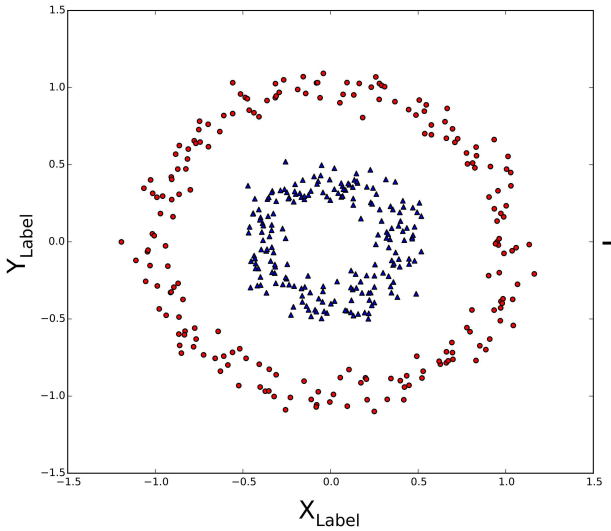
$z_n = \varphi(x_n) \in \mathbf{Z}$
Transformed data


Input space or feature space (X)
Samples are
Non-linearly Separable
x
-2
-1
0
1
2
Class 1
Class 2
Mapping
Φ(x)=(x,x²)=(f₁,f₂)=F
(1)→ (1,1)
(2)→ (2,4)
(-1)→(-1,1)
(-2)→(-2,4)
New feature space (F)
x²=f₂
4
3
2
1
x=f₁
-2
-1
0
1
2
Samples are **linearly** Separable

- One sample transformation:

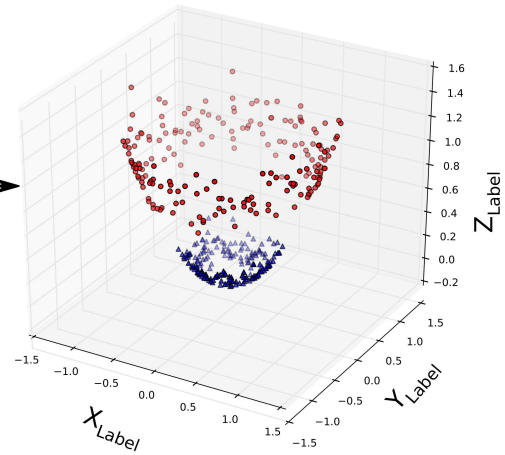
  $\mathbf{x} = (x_0, x_1, \ldots, x_d) \xrightarrow{\phi} \mathbf{z} = (z_0, z_1, \ldots, z_{\tilde{d}})$

- Data transformation: $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N \xrightarrow{\phi} \mathbf{z}_1, \ldots, \mathbf{z}_N$
- Labels transformation: $y_1, \ldots, y_N \xrightarrow{\phi} y_1, \ldots, y_N$
- Where are the weights, in $\mathbf{X}$ or in $\mathbf{Z}$?
  - Answer: the weights are in $\mathbf{Z}$, $\tilde{\mathbf{w}} = \{w_0, w_1, \ldots, w_{\tilde{d}}\}$
  - $g(\mathbf{x}) = \tilde{g}(\phi(\mathbf{x})) = \text{sign}(\tilde{\mathbf{w}}\phi(\mathbf{x}))$

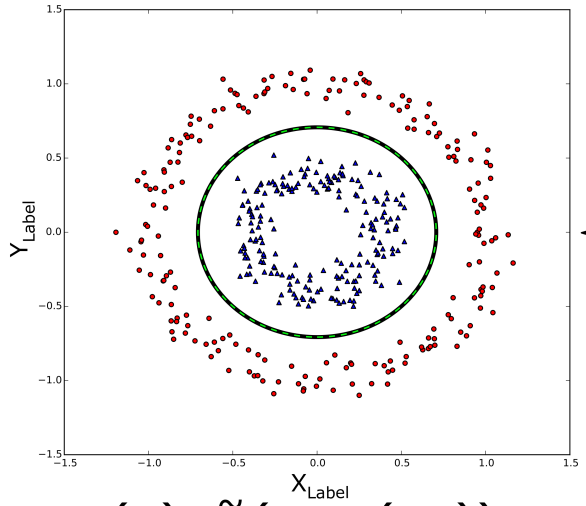


$x_n \in \mathbf{X}$

φ

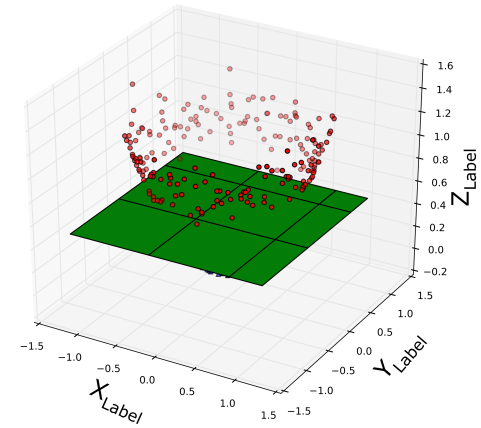


$z_n = \varphi(x_n) \in \mathbf{Z}$
Transformed data



$g(x) = \tilde{g}(z = \varphi(xn))$
$= \text{sign}(\tilde{w}^T \varphi(xn))$

φ^{-1}



Classification
$\tilde{g}(z) = \text{sign}(\tilde{w}^T z)$