Machine Learning

Lecture 6
Linear Regression & Regularization

Chen-Kuo Chiang (江 振 國)
ckchiang@cs.ccu.edu.tw

中正大學 資訊工程學系

The Storyline

How Can Machines Learn?

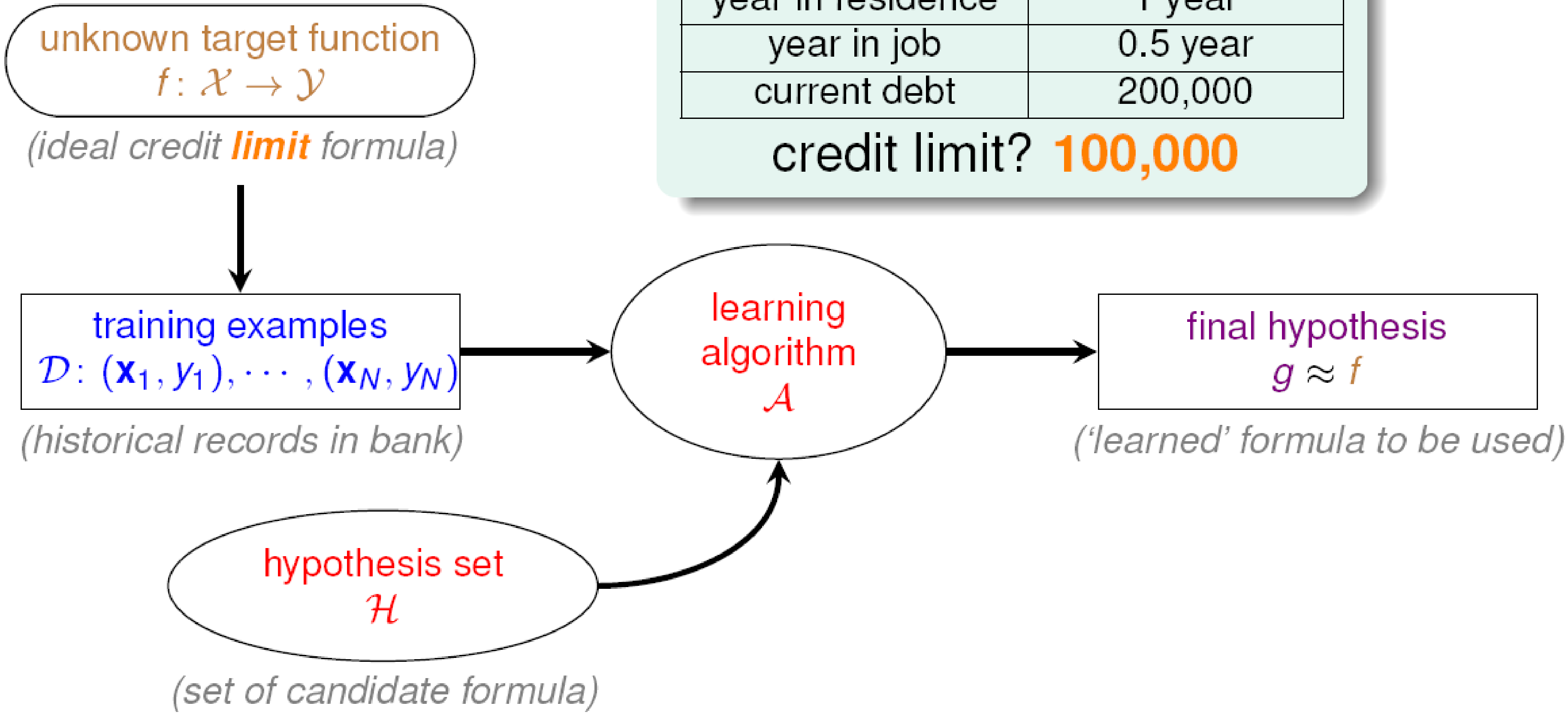
Linear Regression

- Linear Regression Problem
- Linear Regression Algorithm
- Linear Regression for Binary Classification

Credit **Limit** Problem

age	23 years
gender	female
annual salary	NTD 1,000,000
year in residence	1 year
year in job	0.5 year
current debt	200,000

credit limit? **100,000**



$\mathcal{Y} = \mathbb{R}$: **regression**

High-dimensional Data as Input

age	23 years
annual salary	NTD 1,000,000
year in job	0.5 year
current debt	200,000

- For $\mathbf{x} = (x_1, x_2, \cdots, x_d)$ 'features of customer', compute a weighted 'score' and

$$\text{approve credit if} \quad \sum_{i=1}^{d} w_i x_i > \text{threshold}$$

$$\text{deny credit if} \quad \sum_{i=1}^{d} w_i x_i < \text{threshold}$$

- $\mathcal{Y}$: $\{+1(\textbf{good}), -1(\textbf{bad})\}$, 0 ignored—linear formula $h \in \mathcal{H}$ are

$$h(\mathbf{x}) = \text{sign}\left(\left(\sum_{i=1}^{d} w_i x_i\right) - \text{threshold}\right)$$

Vector Form of Perceptron Hypothesis

$$
\begin{aligned}
h(\mathbf{x}) &= \operatorname{sign}\left(\left(\sum_{i=1}^{d} w_i x_i\right) - \text{threshold}\right) \\
&= \operatorname{sign}\left(\left(\sum_{i=1}^{d} w_i x_i\right) + \underbrace{(-\text{threshold})}_{w_0} \cdot \underbrace{(+1)}_{x_0}\right) \\
&= \operatorname{sign}\left(\sum_{i=0}^{d} w_i x_i\right) \\
&= \operatorname{sign}\left(\mathbf{w}^T \mathbf{x}\right)
\end{aligned}
$$

- each ‘tall’ **w** represents a hypothesis *h* & is multiplied with ‘tall’ **x** —**will use tall versions to simplify notation**

Linear Regression Hypothesis

age	23 years
annual salary	NTD 1,000,000
year in job	0.5 year
current debt	200,000

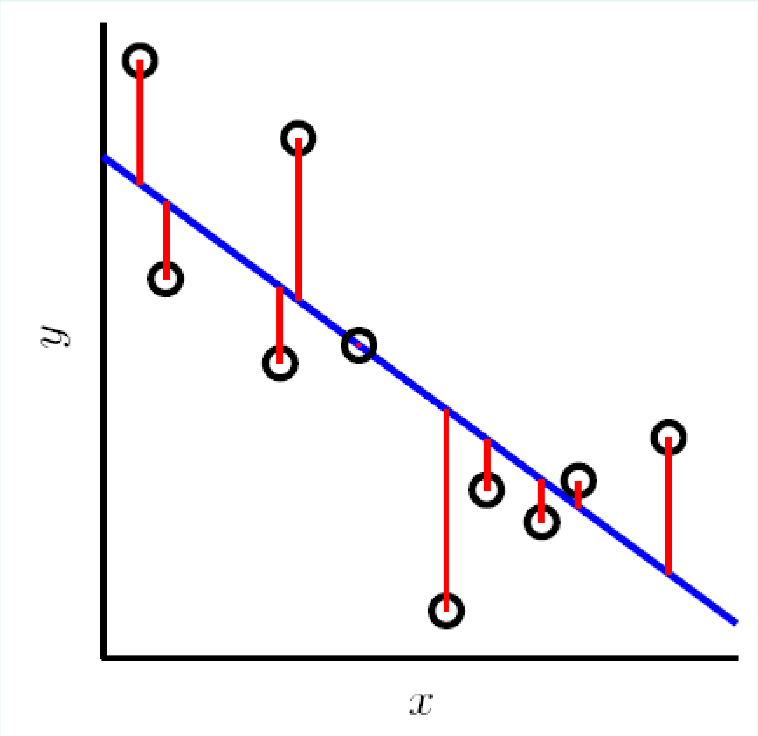
- For $\mathbf{x} = (x_0, x_1, x_2, \cdots, x_d)$ 'features of customer', approximate the desired credit limit with a weighted sum:

$$y \approx \sum_{i=0}^{d} w_i x_i$$

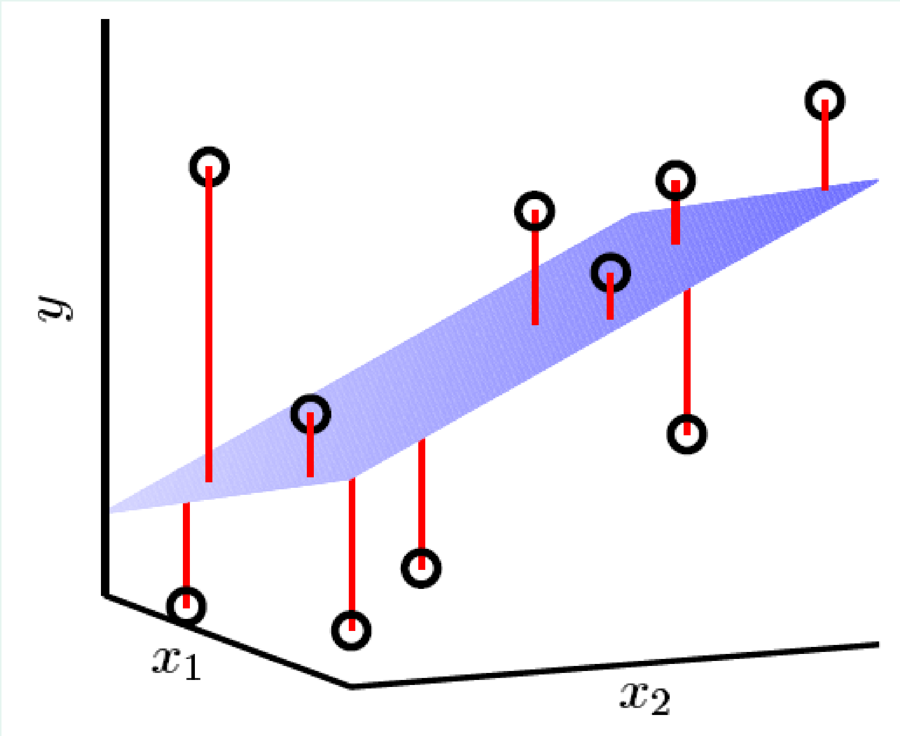
- linear regression hypothesis: $h(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$

Illustration of Linear Regression

$\mathbf{x} = (x) \in \mathbb{R}$



$\mathbf{x} = (x_1, x_2) \in \mathbb{R}^2$



linear regression:
find lines/hyperplanes with small residuals

The Error Measure

popular/historical error measure:

squared error $\mathrm{err}(\hat{y}, y) = (\hat{y} - y)^2$

in-sample

$$E_{\text{in}}(\mathbf{w}) = \frac{1}{N} \sum_{n=1}^{N} (\underbrace{h(\mathbf{x}_n)}_{\mathbf{w}^T \mathbf{x}_n} - y_n)^2$$

out-of-sample

$$E_{\text{out}}(\mathbf{w}) = \underset{(\mathbf{x}, y) \sim P}{\mathcal{E}} (\mathbf{w}^T \mathbf{x} - y)^2$$

next: how to minimize $E_{\text{in}}(\mathbf{w})$?

Fun Time

Consider using linear regression hypothesis $h(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$ to predict the credit limit of customers $\mathbf{x}$. Which feature below shall have a positive weight in a **good hypothesis** for the task?

1. birth month
2. monthly income
3. current debt
4. number of credit cards owned

Reference Answer: 2

Customers with higher monthly income should naturally be given a higher credit limit, which is captured by the positive weight on the 'monthly income' feature.

Matrix Form of $E_{in}(\mathbf{w})$

$$
\begin{aligned}
E_{\text{in}}(\mathbf{w}) &= \frac{1}{N}\sum_{n=1}^{N}(\mathbf{w}^T\mathbf{x}_n - y_n)^2 = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{x}_n^T\mathbf{w} - y_n)^2 \\
&= \frac{1}{N}\left\| \begin{matrix} \mathbf{x}_1^T\mathbf{w} - y_1 \\ \mathbf{x}_2^T\mathbf{w} - y_2 \\ \cdots \\ \mathbf{x}_N^T\mathbf{w} - y_N \end{matrix} \right\|^2
\end{aligned}
$$

l_2-norm

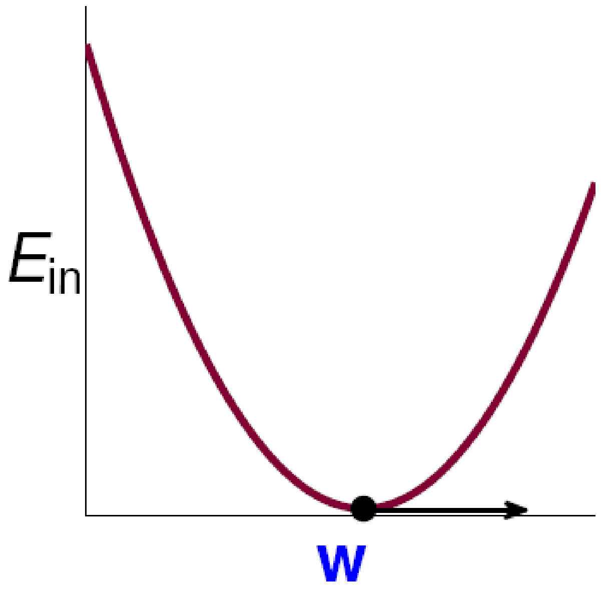
$$
\begin{aligned}
&= \frac{1}{N}\left\| \begin{bmatrix} --\mathbf{x}_1^T-- \\ --\mathbf{x}_2^T-- \\ \cdots \\ --\mathbf{x}_N^T-- \end{bmatrix} \mathbf{w} - \begin{bmatrix} y_1 \\ y_2 \\ \cdots \\ y_N \end{bmatrix} \right\|^2 \\
&= \frac{1}{N}\| \underbrace{\mathrm{X}}_{N\times(d+1)} \underbrace{\mathbf{w}}_{(d+1)\times 1} - \underbrace{\mathbf{y}}_{N\times 1} \|^2
\end{aligned}
$$

Norm

$$\boxed{\left\|x\right\|_n = \sqrt[n]{\sum_{i=1}^{n}\left|x_i\right|^n}}$$

- A norm is a function that assigns a strictly positive length or size to each vector in a vector space.
- $\boldsymbol{x} = (x_1, x_2, ..., x_n)$
- l_2-norm (Euclidean norm) $\left\|x\right\|_2 = \sqrt{x_1^2 + x_2^2 + x_3^2 + \ldots + x_N^2}$
- l_1-norm $\left\|x\right\|_1 = \sum_{i=1}^{n}\left|x_i\right| = \left|x_1\right| + \left|x_2\right| + \left|x_3\right| + \ldots.. + \left|x_N\right|$
- l_0-norm $\left\|x\right\|_0 = \sqrt[0]{\sum_{i=1}^{n}\left|x_i\right|^0} = \#\left(i \mid x_i \neq 0\right)$
- Infinite-norm $\left\|x\right\|_\infty = \max\left(\left|x_1\right|, \left|x_2\right|, \left|x_3\right|, \ldots., \left|x_n\right|\right)$

$$\min_{\mathbf{w}} E_{\text{in}}(\mathbf{w}) = \frac{1}{N}\|\mathrm{X}\mathbf{w} - \mathbf{y}\|^2$$



- $E_{\text{in}}(\mathbf{w})$: continuous, differentiable, **convex**
- necessary condition of 'best' $\mathbf{w}$

$$\nabla E_{\text{in}}(\mathbf{w}) \equiv \begin{bmatrix} \frac{\partial E_{\text{in}}}{\partial w_0}(\mathbf{w}) \\ \frac{\partial E_{\text{in}}}{\partial w_1}(\mathbf{w}) \\ \dots \\ \frac{\partial E_{\text{in}}}{\partial w_d}(\mathbf{w}) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \dots \\ 0 \end{bmatrix}$$

—not possible to 'roll down'

task: find $\mathbf{w}_{\text{LIN}}$ such that $\nabla E_{\text{in}}(\mathbf{w}_{\text{LIN}}) = \mathbf{0}$

The Gradient $\nabla E_{\text{in}}(\mathbf{w})$

$$E_{\text{in}}(\mathbf{w}) = \frac{1}{N}\|\mathrm{X}\mathbf{w} - \mathbf{y}\|^2 = \frac{1}{N}\left(\mathbf{w}^T \underbrace{\mathrm{X}^T\mathrm{X}}_{\mathrm{A}} \mathbf{w} - 2\mathbf{w}^T \underbrace{\mathrm{X}^T\mathbf{y}}_{\mathbf{b}} + \underbrace{\mathbf{y}^T\mathbf{y}}_{c}\right)$$

one w only

$$E_{\text{in}}(w) = \frac{1}{N}\left(aw^2 - 2bw + c\right)$$

$$\nabla E_{\text{in}}(w) = \frac{1}{N}\left(2aw - 2b\right)$$

simple! :-)

vector $\mathbf{w}$

$$E_{\text{in}}(\mathbf{w}) = \frac{1}{N}\left(\mathbf{w}^T\mathrm{A}\mathbf{w} - 2\mathbf{w}^T\mathbf{b} + c\right)$$

$$\nabla E_{\text{in}}(\mathbf{w}) = \frac{1}{N}\left(2\mathrm{A}\mathbf{w} - 2\mathbf{b}\right)$$

similar (**derived by definition**)

$$\nabla E_{\text{in}}(\mathbf{w}) = \frac{2}{N}\left(\mathrm{X}^T\mathrm{X}\mathbf{w} - \mathrm{X}^T\mathbf{y}\right)$$

Optimal Linear Regression Weights

task: find $\mathbf{w}_{\text{LIN}}$ such that $\frac{2}{N}\left(\mathrm{X}^T\mathrm{X}\mathbf{w} - \mathrm{X}^T\mathbf{y}\right) = \nabla E_{\text{in}}(\mathbf{w}) = \mathbf{0}$

invertible $\mathrm{X}^T\mathrm{X}$

- **easy!** unique solution

$$\mathbf{w}_{\text{LIN}} = \underbrace{\left(\mathrm{X}^T\mathrm{X}\right)^{-1}\mathrm{X}^T}_{\text{pseudo-inverse } \mathrm{X}^\dagger}\mathbf{y}$$

因為不一定反矩陣存在

- often the case because $N \gg d+1$

singular $\mathrm{X}^T\mathrm{X}$

- **many** optimal solutions
- one of the solutions

$$\mathbf{w}_{\text{LIN}} = \mathrm{X}^\dagger\mathbf{y}$$

by defining $\mathrm{X}^\dagger$ in other ways

practical suggestion:

use **well-implemented $\dagger$ routine**

instead of $\left(\mathrm{X}^T\mathrm{X}\right)^{-1}\mathrm{X}^T$

for numerical stability when **almost-singular**

Linear Regression Algorithm

1. from $\mathcal{D}$, construct input matrix X and output vector $\mathbf{y}$ by

$$\mathrm{X} = \underbrace{\begin{bmatrix} - - \mathbf{x}_1^T - - \\ - - \mathbf{x}_2^T - - \\ \cdots \\ - - \mathbf{x}_N^T - - \end{bmatrix}}_{N\times(d+1)} \quad \mathbf{y} = \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \cdots \\ y_N \end{bmatrix}}_{N\times 1}$$

2. calculate pseudo-inverse $\underbrace{\mathrm{X}^\dagger}_{(d+1)\times N}$
3. return $\underbrace{\mathbf{w}_{\mathrm{LIN}}}_{(d+1)\times 1} = \mathrm{X}^\dagger \mathbf{y}$

simple and efficient
with **good $\dagger$ routine**

範例

給定5組(X, Y)數據如下：

X	2	1	4	5	3
Y	1	3	7	6	3

(1) 求Y對X的迴歸直線方程式

(2) 利用迴歸直線，預測$x=8$時，y值應為多少？

$$y = ax + b = w_1 x + w_0$$

$$1 = 2w_1 + w_0$$
$$3 = 1w_1 + w_0$$
$$7 = 4w_1 + w_0$$
$$6 = 5w_1 + w_0$$
$$3 = 3w_1 + w_0$$

$$\underbrace{\begin{bmatrix} 1 \\ 3 \\ 7 \\ 6 \\ 3 \end{bmatrix}}_{\mathrm{y}} = \underbrace{\begin{bmatrix} 2 & 1 \\ 1 & 1 \\ 4 & 1 \\ 5 & 1 \\ 3 & 1 \end{bmatrix}}_{\mathrm{X}} \cdot \underbrace{\begin{bmatrix} w_1 \\ w_0 \end{bmatrix}}_{\mathrm{W_{LIN}}}$$

$$\mathbf{w}_{\mathrm{LIN}} = (\mathrm{X}^T\mathrm{X})^{-1}\mathrm{X}^T \mathbf{y}$$

$$\mathrm{W_{LIN}} = \left(\begin{bmatrix} 2 & 1 & 4 & 5 & 3 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 2 & 1 \\ 1 & 1 \\ 4 & 1 \\ 5 & 1 \\ 3 & 1 \end{bmatrix} \right)^{-1} \cdot \begin{bmatrix} 2 & 1 & 4 & 5 & 3 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 3 \\ 7 \\ 6 \\ 3 \end{bmatrix} = \begin{bmatrix} 1.2 \\ 0.4 \end{bmatrix}$$

$$y = 1.2x + 0.4$$

Fun Time

After getting $\mathbf{w}_{\mathrm{LIN}}$, we can calculate the predictions $\hat{y}_n = \mathbf{w}_{\mathrm{LIN}}^T \mathbf{x}_n$. If all $\hat{y}_n$ are collected in a vector $\hat{\mathbf{y}}$ similar to how we form $\mathbf{y}$, what is the matrix formula of $\hat{\mathbf{y}}$?

1. $\mathbf{y}$
2. $\mathrm{XX}^T\mathbf{y}$
3. $\mathrm{XX}^\dagger\mathbf{y}$
4. $\mathrm{XX}^\dagger\mathrm{XX}^T\mathbf{y}$

$x_n^T W_{LIN}$

$\Rightarrow XX^{+}y$

Reference Answer: ?

HW#1 Exercise

Linear Classification vs. Linear Regression

Linear Classification

$$
\begin{aligned}
\mathcal{Y} &= \{-1, +1\} \\
h(\mathbf{x}) &= \text{sign}(\mathbf{w}^T\mathbf{x}) \\
\text{err}(\hat{y}, y) &= [\![\hat{y} \neq y]\!]
\end{aligned}
$$

NP-hard to solve in general

Linear Regression

$$
\begin{aligned}
\mathcal{Y} &= \mathbb{R} \\
h(\mathbf{x}) &= \mathbf{w}^T\mathbf{x} \\
\text{err}(\hat{y}, y) &= (\hat{y} - y)^2
\end{aligned}
$$

efficient analytic solution

[助教飛飛的重點提示]
感知器與線性迴歸
如同兩兄弟
①數學公式上的差異?
②模型表示上的差異?
③誤差評估的差異

$\{-1, +1\} \subset \mathbb{R}$: linear regression for classification?

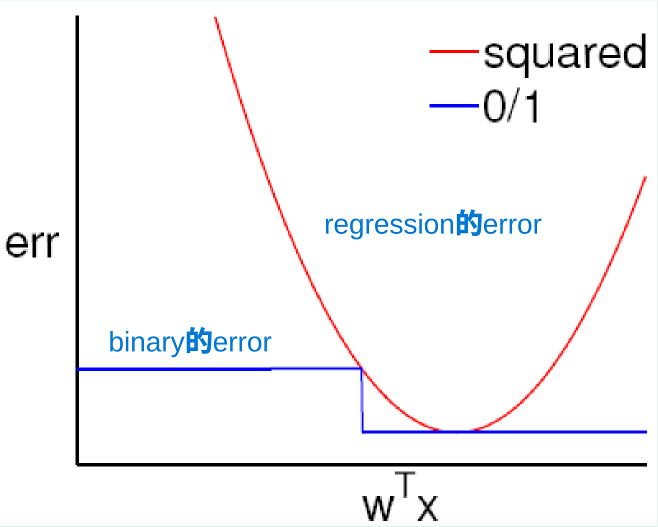
1. run LinReg on binary classification data $\mathcal{D}$ (**efficient**)
2. return $g(\mathbf{x}) = \text{sign}(\mathbf{w}_{\text{LIN}}^T\mathbf{x})$

but explanation of this **heuristic**?

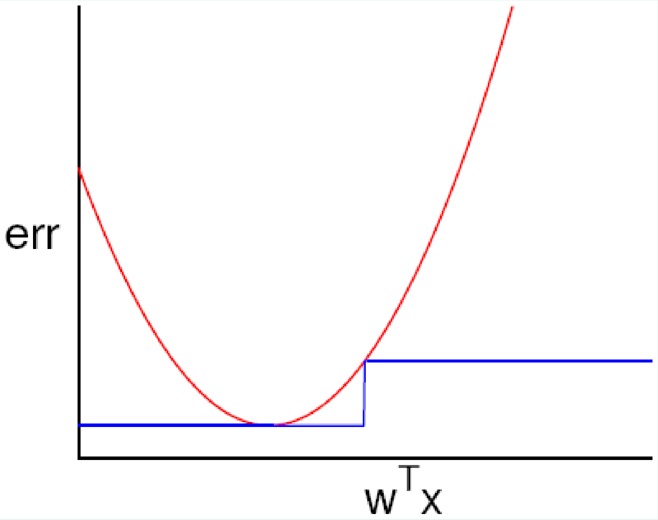
Relation of Two Errors

$$\text{err}_{0/1} = [\![\text{sign}(\mathbf{w}^T\mathbf{x}) \neq y]\!] \qquad \text{err}_{\text{sqr}} = \left(\mathbf{w}^T\mathbf{x} - y\right)^2$$

desired $y = 1$



desired $y = -1$



可以發現都會壓住binary，用誤差換取計算時間(regression較快)

$$\text{err}_{0/1} \leq \text{err}_{\text{sqr}}$$

Linear Regression for Binary Classification

$$\mathrm{err}_{0/1} \leq \mathrm{err}_{\mathrm{sqr}}$$

$$\begin{aligned} \text{classification } E_{\mathrm{out}}(\mathbf{w}) &\overset{\mathrm{VC}}{\leq} \text{classification } E_{\mathrm{in}}(\mathbf{w}) + \sqrt{\cdots\cdots} \\ &\leq \text{regression } E_{\mathrm{in}}(\mathbf{w}) + \sqrt{\cdots\cdots} \end{aligned}$$

- (loose) upper bound $\mathrm{err}_{\mathrm{sqr}}$ as $\widehat{\mathrm{err}}$ to approximate $\mathrm{err}_{0/1}$
- trade **bound tightness** for **efficiency**

$\mathbf{w}_{\mathrm{LIN}}$: useful baseline classifier,
or as **initial PLA/pocket vector**

Fun Time

Which of the following functions are upper bounds of the pointwise 0/1 error $[\![\text{sign}(\mathbf{w}^T\mathbf{x}) \neq y]\!]$ for $y \in \{-1, +1\}$?

1. $\exp(-y\mathbf{w}^T\mathbf{x})$
2. $\max(0, 1 - y\mathbf{w}^T\mathbf{x})$
3. $\log_2(1 + \exp(-y\mathbf{w}^T\mathbf{x}))$
4. all of the above

Reference Answer: 4

Plot the curves and you'll see. Thus, all three can be used for binary classification. In fact, all three functions connect to very important algorithms in machine learning and we will discuss one of them soon in the next lecture.

Stay tuned. :-)

Summary

How Can Machines Learn?

Linear Regression

- Linear Regression Problem
  use hyperplanes to approximate real values
- Linear Regression Algorithm
  analytic solution with pseudo-inverse
- Linear Regression for Binary Classification
  0/1 error $\leq$ squared error