Machine Learning

Lecture 7
Linear Regression

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?

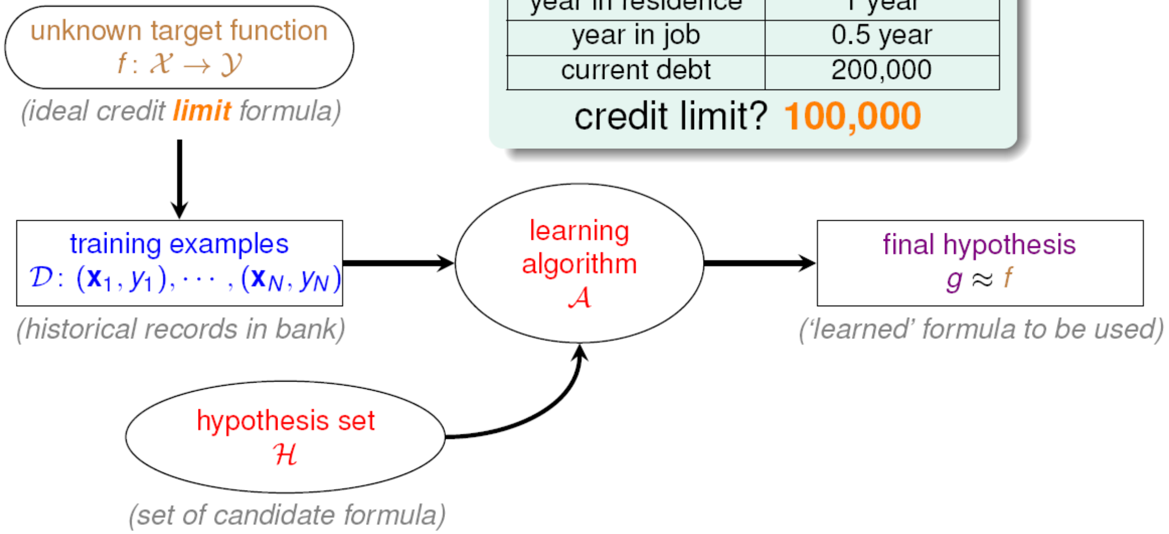
Linear Regression

- 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**

Linear Regression Hypothesis

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

門閥值

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

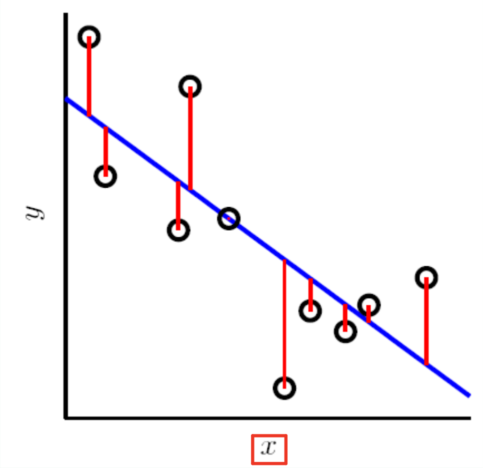
$$y \approx \sum_{i=0}^{d} w_i x_i$$

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

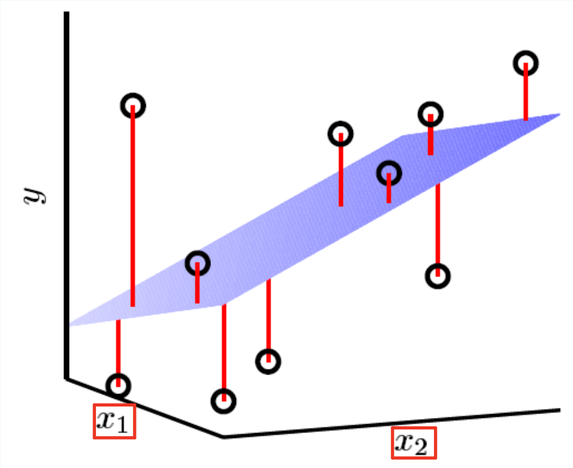
$h(\mathbf{x})$: like **perceptron**, but without the sign

Illustration of Linear Regression

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



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



linear regression:

高維度空間

find lines/hyperplanes with small residuals

The Error Measure

popular/historical error measure: 把err function求出後，再使用optimize function

越小越好

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

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?

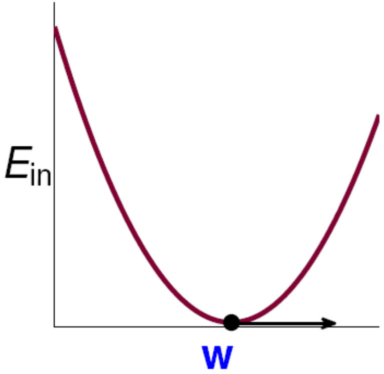
1. birth month
2. monthly income
3. current debt
4. number of credit cards owned

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

$$
\begin{aligned}
E_{in}(\mathbf{w}) &= \frac{1}{N}\sum_{n=1}^{N}(\mathbf{w}^T\mathbf{x}_n - y_n)^2 = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{x}_n^T\mathbf{w} - y_n)^2 \\
&= \frac{1}{N}\left\| \begin{array}{c} \overset{\text{預測}}{\boxed{\begin{array}{c}\mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \\ \mathbf{x}_N^T\end{array}}} \begin{array}{c}\mathbf{w} - \overset{\text{標準}}{y_1} \\ \mathbf{w} - y_2 \\ \ldots \\ \mathbf{w} - y_N\end{array} \end{array} \right\|^2 \\
&= \frac{1}{N}\left\| \begin{bmatrix} --\mathbf{x}_1^T-- \\ --\mathbf{x}_2^T-- \\ \ldots \\ --\mathbf{x}_N^T-- \end{bmatrix} \mathbf{w} - \begin{bmatrix} y_1 \\ y_2 \\ \ldots \\ y_N \end{bmatrix} \right\|^2 \\
&= \frac{1}{N}\| \underbrace{\mathrm{X}}_{N\times d+1} \underbrace{\mathbf{w}}_{d+1\times 1} - \underbrace{\mathbf{y}}_{N\times 1} \|^2
\end{aligned}
$$

多了0

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



- $E_{\text{in}}(\mathbf{w})$: 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 $\boxed{\nabla E_{\text{in}}(\mathbf{w}_{\text{LIN}}) = \mathbf{0}}$

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

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

one w only

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

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

simple! :-)

vector $\mathbf{w}$

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

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

similar (**derived by definition**)

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

Optimal Linear Regression Weights

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

$X^TXw = X^Ty \Rightarrow X^TX(X^TX)^{-1}w = (X^TX)^{-1}X^T$ 反向

考

invertible X^TX

- **easy!** unique solution

存在

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

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

singular X^TX

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

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

Linear regression

by defining $X^\dagger$ in other ways

practical suggestion:

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

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

for numerical stability when **almost-singular**

Linear Regression Algorithm

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

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

2. calculate pseudo-inverse $\underbrace{\mathrm{X}^\dagger}_{(d+1)\times N}$

3. return $\underbrace{\mathbf{w}_{\mathrm{LIN}}}_{(d+1)\times 1} = \underline{\mathrm{X}^\dagger\mathbf{y}}$

simple and efficient
with **good † routine**

Fun Time

預測
結果

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

收集成一個向量

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

Linear Classification vs. Linear Regression

Linear Classification

相同

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

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

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

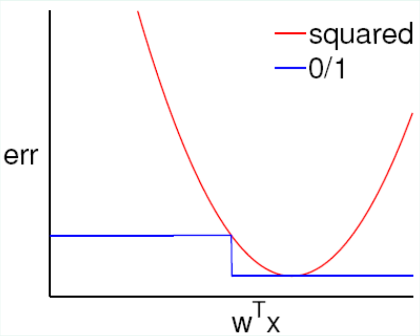
1. run LinReg on binary classification data $\mathcal{D}$ (**efficient**) 先求linear regression
2. return $g(\mathbf{x}) = \text{sign}(\mathbf{w}_{\text{LIN}}^T\mathbf{x})$ 再取sign

but explanation of this **heuristic**?

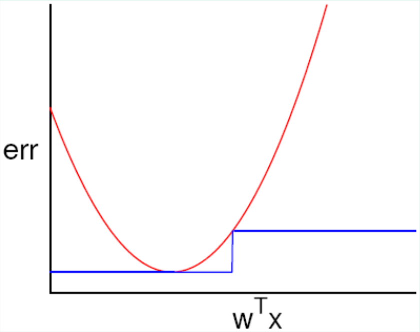
Relation of Two Errors

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

desired $y = 1$



desired $y = -1$



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

初始化條件 Upper bond

Linear Regression for Binary Classification

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

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

- (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 $y \in \boxed{\{-1, +1\}}$?

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

Summary

How Can Machines Learn?

Linear Regression

- Linear Regression Problem

  use hyperplanes to approximate real values
- Linear Regression Algorithm

  可以寫出一個公式 **analytic solution with pseudo-inverse**
- Linear Regression for Binary Classification

  0/1 error $\leq$ squared error