3월 21일 오후 11:59 마감

Assignment 02

Byung-Woo Hong 3월 14일 수업 댓글 추가

[Submission]

1. Write a python code, make a comment, and present results for each meaningful block of your code at Jupyter Notebook.
[New > Python 3]

2. Export your Jupyter Notebook file as PDF file at Jupyter Notebook.
[Download as > PDF via LaTeX (.pdf)]

3. Turn in your PDF file to the assignment at Google Classroom.
[Add > file]

****

[Taylor Approximation]

1. Define a differentiable function that maps from real number to real number.
2. Define a domain of the function.
3. Plot the function.
4. Select a point within the domain.
5. Mark the selected point on the function.
6. Define the first-order Taylor approximation at the selected point.
7. Plot the Taylor approximation with the same domain of the original function.

내 과제 할당됨

LaTeX

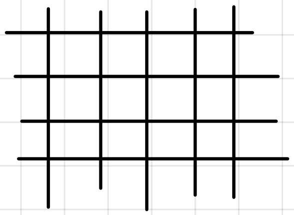
`$\frac{1}{x}$` $\Rightarrow$ $\frac{1}{x}$

\#

`# $f(x) = . . . . $`

f = ________________

$\boxed{f(x) = \sin x}$



① function 정의

② domain 정의

③ 그래프 보여주기

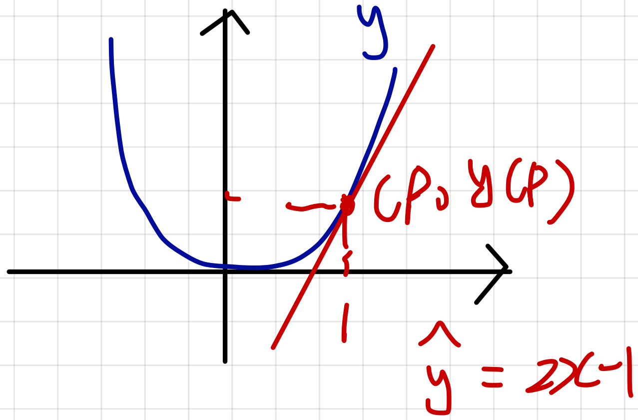
④ Taylor expression

ex) $y=x^2$ at 1

$\hat{y} = y(1) + y'(1)(x-1)$

$= 1 + 2\cdot 1\,(x-1)$

$= 2x-1$



3월 28일 오후 11:59 마감

Assignment 03

Byung-Woo Hong 오전 10:20 수업 댓글 추가

같은 label을 갖는 이미지를 보여줘라..?

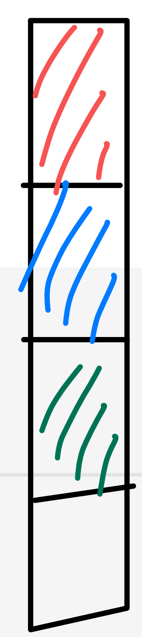
평균을 구해서 보여줘라.

이미지 보여줄 때 img값의 범위를
정해서 보여주기. (ex) 0~100)

1. Write codes with detailed comments and present useful results at Jupyter Notebook.
2. Export the Jupyter Notebook file as a PDF file.
3. Submit the PDF file to Classroom.

[Visualize average images]

1. Load MNIST training dataset.
2. Compute the average images for each label (digit) based on L2-norm.
3. Visualize the average images.



mnist_test.csv
쉼표로 구분된 값

mnist_train.csv
쉼표로 구분된 값

assignment03.py
텍스트

Assignment 04

Byung-Woo Hong 3월 29일 수업 댓글 추가

[K-means clustering]

algo 있음

group의 개수 ↑ group수↑ : 정확도↑

1. Apply K-means clustering to MNIST training dataset with different K = 5, 10, 15, 20 and present the following results for each K.
2. Visualize K centroid images for each category.
3. Plot the training energy per optimization iteration.
4. Plot the training accuracy per optimization iteration.
5. Plot the testing accuracy per optimization iteration.

최종 image만 plot

[energy]

\sum_{k = 1}^K \| x_i - c_{k_i} \|^2
where k_i denotes the category of x_i, and c_{k_i} denotes the centroid of category x_i.

[accuracy]

\frac{\sum_{k = 1}^K m_k}{N}
where N denotes the total number of data, and m_k denotes the number of data with majority for category k.

- (training energy) is computed on the training dataset.
- (training accuracy) is computed on the training dataset.
- (testing accuracy) is computed on the testing dataset.

내 과제

할당됨

내가 추가하거나 생성한 파일을 교사가 보거나 수정할 수 있습니다.

4월 11일 오후 11:59 마감

Assignment 05

Byung-Woo Hong 오전 9:04 수업 댓글 추가

[K-means clustering on Image]

1. Select any image with multiple distinctive regions with different colors.
2. Apply K-means clustering to the selected image with varying K (Your choice of 4 different K's).
3. Plot input image in color.
4. Plot the energy curve for each K.
5. Plot result image for each K.

[energy]

\sum_{k = 1}^K \| f_i - c_{k_i} \|^2,

f_i denotes image intensity at index i.
k_i denotes the category of f_i.
c_{k_i} denotes the centroid of category f_i.

[result image]

replace f_i with c_{k_i}.

내 과제 할당됨

5월 9일 오후 11:59 마감

Assignment 06

Byung-Woo Hong 오후 2:53 수업 댓글 추가

[K-means clustering on the spatial domain]

Apply K-means algorithm to the regular grid of a spatial domain in two dimension.
Let $\{ (0, 0), (0, 1), ..., (0, n-1), (1, 0), (1, 1), ..., (1, n-1), ..., (m-1, 0), (m-1, 1), ..., (m-1, n-1) \}$ be a regular grid.
Use two different distances based on L1-norm and L2-norm.
Display the clustering results using the color scheme based on the centroid.

내 과제 할당됨

내가 추가하거나 생성한 파일을 교사가 보거나 수정할 수 있습니다.

완료로 표시하거나 과제를 첨부하여 제출하세요.

추가 + 만들기 완료로 표시

비공개 댓글 추가...

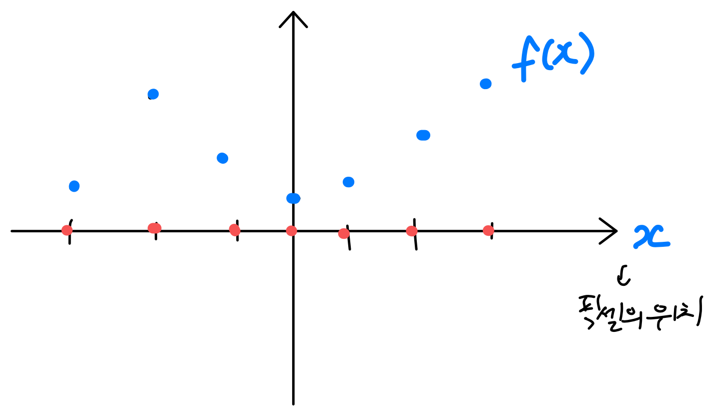
과제 #1

영상 $f: \Omega \to R$

Ω : domain of image f

↳ 시간, 픽셀의위치

codomain : 범위 색, ...



n

m

(0,0)	(0,1)	...	...	(0,n-1)
(1,0)	(1,1)	...	...	(1,n-1)
(m-1,0)	(m-1,1)	...	...	(m-1,n-1)

→ 2개의 matrix 필요

0	0	...
1	1	...
⋮		
m-1	m-1	...

x좌표 domain
matrix

0	1	2
0	1	2
⋮		
0	1	2

y 좌표 domain
matrix

1	2	3
4	5	6
7	8	9

$f(0,0) = 1$
$f(0,1) = 2$
$f(0,2) = 3$
$f(1,0) = 4$
$f(1,1) = 5$
⋮

$f : \Omega \to \mathbb{R}$
↑
픽셀들의 위치

$f^{-1}(5) = (1,1)$

codomain에 있는 range

$f(row, col) = val$

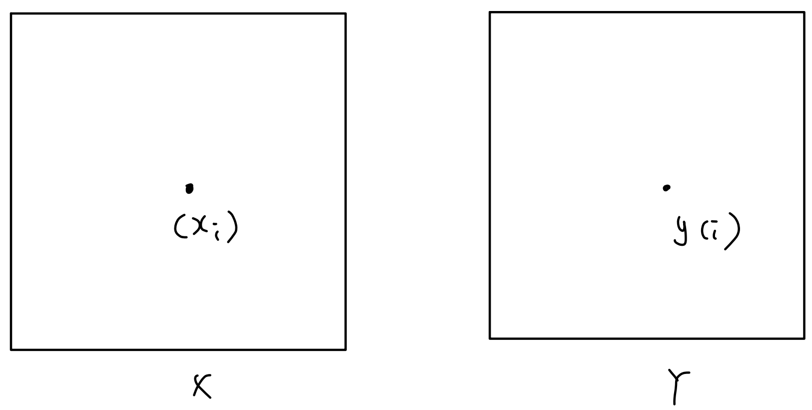
value와 index의 값이 필요

→ image와 동일한 size의 matrix를 2개 가지고 있어야 함.

index는 vector이다 (2차원 -vector)

→ domain을 어떻게 나타낼 것인가 (implementation)

→ 2개의 matrix 를 가지고 있어야함.

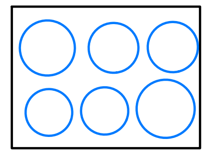


$L_2^2 = (c_x - x_i)^2 + (c_y - y_i)^2$

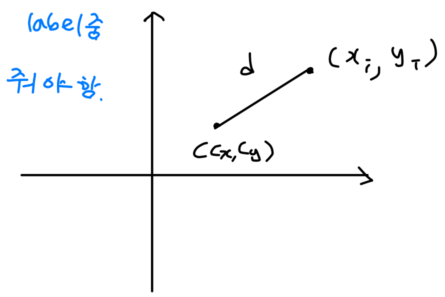
└ cluster centroid ┘

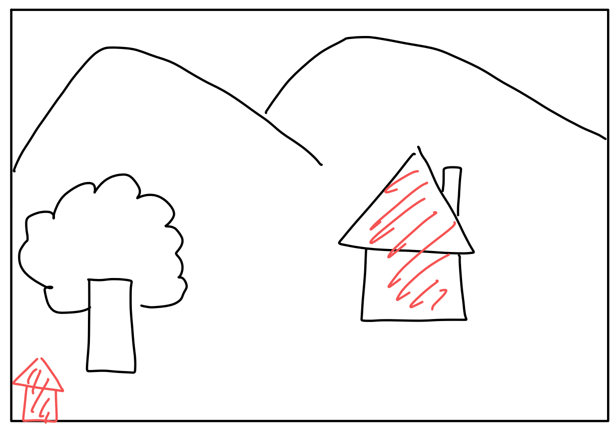
$L_1 = |c_x - x_i| + |c_y - y_i|$

→ clustering



초기화 : 좌표마다 random하게 label줌
& x_i와 y_i에 같은 label을 줘야함.





- 비슷한 색만으로 clustering 하면 두 개의 집이 하나의 cluster로 형성됨.
- 색 + 위치로 clustering 하면 두 집을 나누어 clustering 가능.

INPUT :

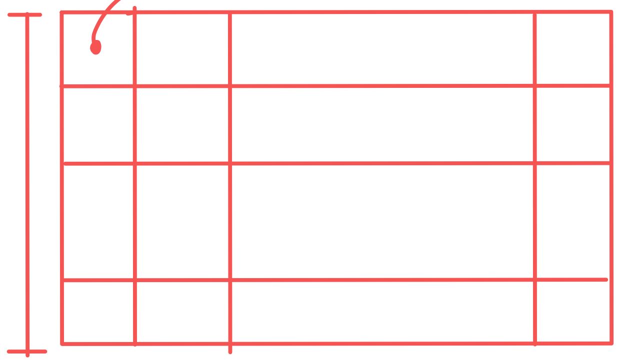
① # of rows

② # of columns

③ # of Clusters

④ $\begin{cases} L_2^2 \text{ - distance} \\ L_1 \text{ - distance} \end{cases}$

OUTPUT : (visualization)



cluster label은 인접 X

cluster value는 인접 O

ㄴ centroid의 값은 cluster의 value의 평균

$(C_x^{(1)}, C_y^{(1)})$

$(C_x^{(2)}, C_y^{(2)})$

⋮

$(C_x^{(k)}, C_y^{(k)})$

⇒ 같은 cluster 끼리는 똑같이 보이게 해라

각 cluster 마다

- (C_x, C_y)를 갖고 있음

④ 명암값 하면,

각 cluster 마다

- (R, G, B, C_x, C_y)를 가지고 있고, (5개)

이것들을 이용하여 clustering

$$\sum_i \left((C_x^{(k)} - x_i)^2 + (C_y^{(k)} - y_i)^2 \right)$$

1 ~ 100 사이

$$\min \boxed{\frac{1}{N} \sum_{i=1}^{N} (x_i - C_i^{(k_i)})^2 + (y_i - C_i^{(k_i)})^2}$$

k_i = cluster of x_i, y_i

$$\therefore \min \frac{1}{N} \sum_{i=1}^{N} \left[\| f(x_i, y_i) - \overset{(r,g,b)}{m^{(k_i)}} \|^2 + \alpha \left\{ \left(x_i - \overset{(x)}{C_x^{(k_i)}} \right)^2 + \left(y_i - \overset{(y)}{C_y^{(k_i)}} \right)^2 \right\} \right]$$

5월 16일 오후 11:59 마감

Assignment 07

Byung-Woo Hong 5월 6일 수업 댓글 추가

[Apply K-means algorithm to both image value and its spatial domain]

For a given input image (either gray or color), apply a K-means algorithm that is designed to take into consideration of both the image intensity and its spatial domain with varying parameters: the number of clusters and the trade-off between the intensity energy and the spatial energy.

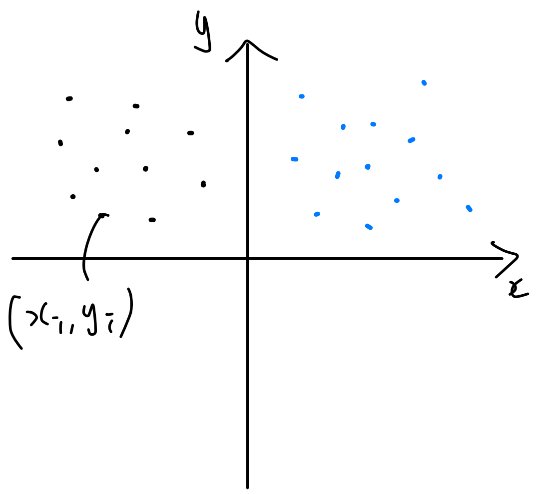
The objective function is given by:

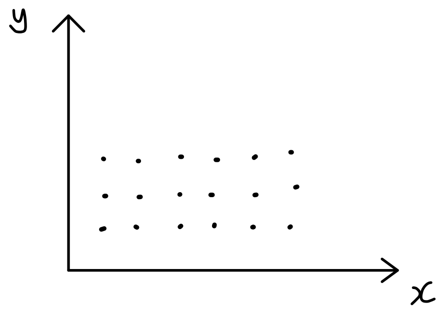
\sum_k \sum_\{ x \in I(k) \} [\| f(x) - m_k \|^2 + a * \| x - c_k \|^2]

where I(k) denotes the index set of x that belongs to cluster k, m_k denotes the centroid of image intensity for cluster k, c_k denotes the centroid of spatial location for cluster k, and a determines the importance between the image intensity and the spatial relation.

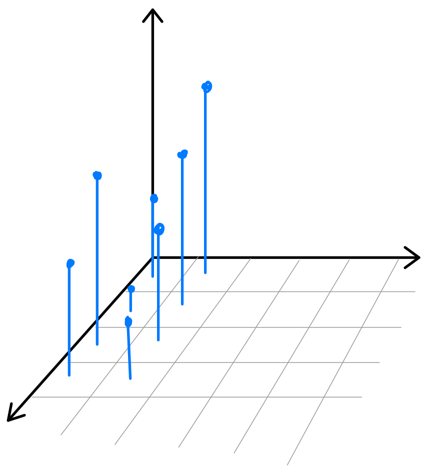
- Visualize the clustering results with varying k and a using the centroid color m_k for each cluster k.

- Visualize the energy curve for both the intensity energy and the spatial energy.





$f : \Omega \rightarrow \mathbb{R}$



domain + 값에 대해 cluster

1. Energy 정의

$z_i = (x_i, y_i)$ → 각각의 좌표 (i: 좌표의 index)

l_i = label of $f(z_i)$

$$\sum_i \|f(z_i) - m_{l_i}\|_2^2$$

$$\sum_i \|z_i - c_{l_i}\|_2^2$$

$$\sum_i \{\|f(z_i) - m_{l_i}\|_2^2 + \lambda \|z_i - c_{l_i}\|_2^2\}$$

$\lambda \to 0$ → 영상값만 가지고 cluster

$\lambda \to \infty$ → 위치만 가지고 cluster

⇒ 가까이 있어도 다른 영상 값이면 다른 cluster로, 같은 영상값이어도 멀리 있으면 다른 cluster가 되도록 하는 것이 목표

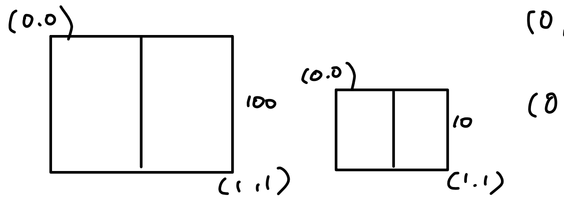
좌표와, 영상의 값이 같은 cluster를 갖는다.

Normalize 1) $z_i \in \Omega$, $\Omega = \{(0,0), (0,1), \cdots, (1,0), \cdots\}$

$$\frac{1}{|\Omega|} \sum_i \{\|f(z_i) - m_{l_i}\|_2^2 + \lambda \|z_i - c_{l_i}\|_2^2\}$$

i의 수
(모든 element의 수)

같은 이미지 사이즈만 다를 때,
같은 값을 갖도록 하기 위해 픽셀의 수 ($|\Omega|$)로 나눠줌



$(0, 1, 2, 3) \rightarrow (0, \frac{1}{3}, \frac{2}{3}, 1)$

$(0, 1.2, 3000) \rightarrow (0, \underset{\approx 0}{0}, \underset{\approx 0}{0}, 1)$

Normalize 2) linear scaling

$g(x) = \alpha f(x) + \beta$

$$\begin{cases} 1 = \alpha \cdot f_{max} + \beta \\ 0 = \alpha \cdot f_{min} + \beta \end{cases}$$

$$\frac{f(x) - \mu}{\sigma} = g(x)$$

g의 평균 = 0이 되고 표준편차가 1이 되도록 만들어줌.

data의 평균과 분산(표준편차) 고려

* 영상의 모든 값의 평균과 표준편차를 계산하여 나눠주어 normalize

주어진 x에 대해 비슷한 값을 갖게 되는 일종

⇒ whitening

* Image intensity → whitening

$$g(x) = \frac{f(x) - \mu}{\sigma}$$

$$\mu = \frac{\sum_i f(x_i)}{|\Omega|}, \quad \sigma = \frac{(f(x_i) - \mu)^2}{|\Omega|}$$

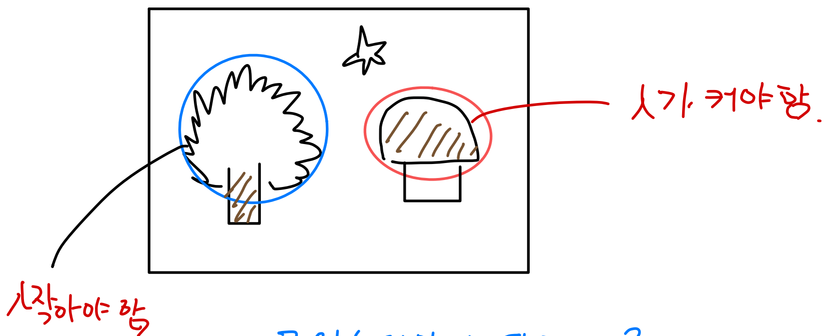
* Image Domain (coordinate)

→ 0 ~ 1 사이로 scaling

(uniform 하므로 다른 normalize/scaling 필요x)

* λ 와 k 변화 시키면서 값의 변화 알아보기

* λ 의 변화



클러스터마다 다른 λ를 적용하고 싶을 때...

⇒ λ는 적절하게 조절하자!

normalize를 잘하자!

* 출력하는 결과

input 이미지와 동일한 size에 이미지 출력

k_i로 모아진 픽셀들은 m_{ci}로 채워서

* m_{ci} : whitening 한 것의 평균 (변환된 값)

$f(z_i)$: whitening 한 (0 ~ 1)

평균이 0

$\Rightarrow f(x) = \sigma g(x) + \mu$

$\sigma m_{ci} + \mu$의 값을 plot 하면 됨.

ex)

(0, 1, 2, 3)

$\Downarrow$

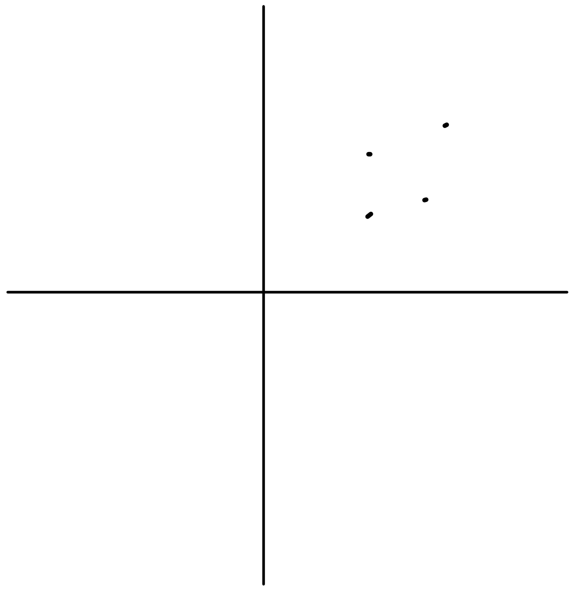
$\boxed{\left(-\frac{2}{4}, -\frac{1}{4}\right.}, \boxed{\left.\frac{1}{4}, \frac{2}{4}\right)}$

$-\frac{3}{8}$	$-\frac{3}{8}$	$\frac{3}{8}$	$\frac{3}{8}$
m_1	m_1	m_2	m_2

$\Downarrow$

0.5	0.5	2.5	2.5

이렇게 바꿔주어야 함.



$d = \sqrt{(x_1-x_2)^2 + (y_1-y_2)^2} + (r_1-r_2)^2 + (g_1-g_2)^2 + (b_1-b_2)^2$

5월 23일 오후 11:59 마감

Assignment 08

Byung-Woo Hong 5월 6일 수업 댓글 추가

[Polynomial fitting]

Solve a least square problem to find an optimal polynomial curve for a given set of two dimensional points.

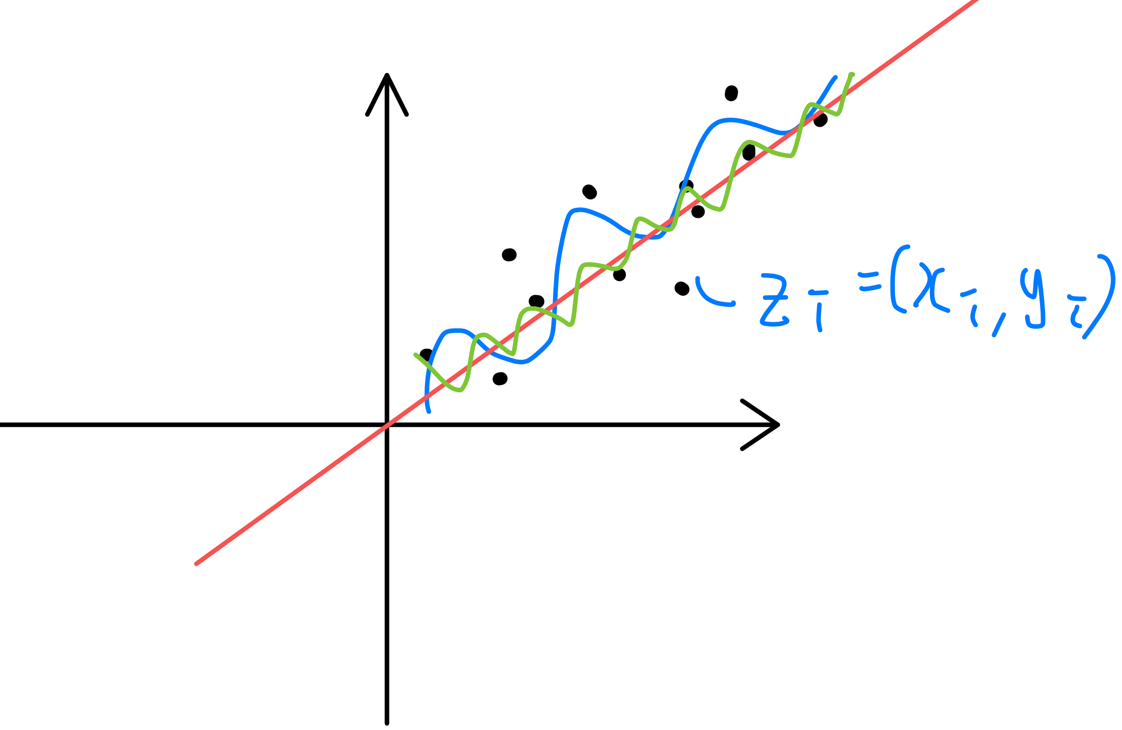
Demonstrate the effect of the degree of polynomial in fitting a given set of points.

- choose a polynomial curve and generate points along the curve with random noise

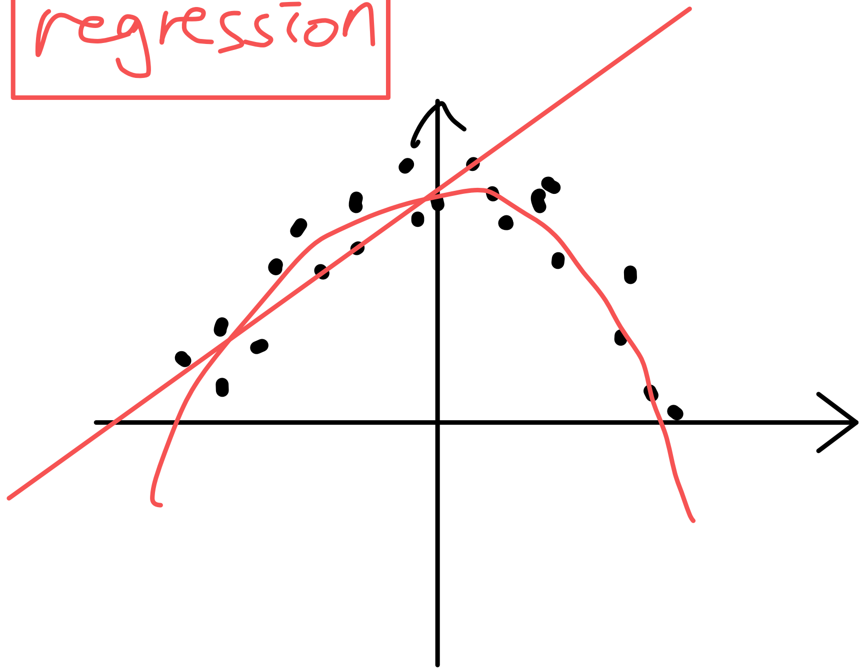
- plot the generated noisy points along with its original polynomial without noise

- plot the approximating polynomial curve obtained by solving a least square problem

- plot the approximating polynomial curve with varying polynomial degree



regression



$$\left.\begin{aligned} f_\theta(x) &= y \\ f_\theta(x) &= \theta_0 + \theta_1 x \end{aligned}\right] \text{직선}$$

$f_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2$ – 포물선

⋮

$f_\theta(x) = \theta_0 \cos(x) + \theta_1 \sin(x)$

* unknown의 수 < equation 수 여야 풀 수 ○

⇒ tall 이어야 함.

* parameter 수 ↑ : equation 수 ↑

* modeling → 함수 결정

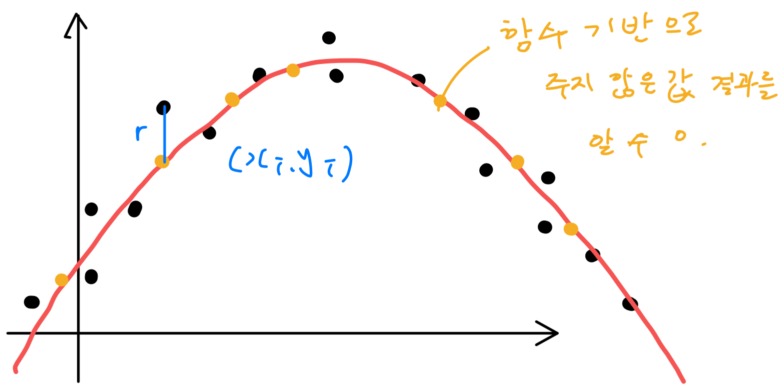
(함수의 form, parameter의 값 결정해야함)

모르면 일반적 form 만 알 수 ○

* 함수와 data 간의 distance (거리)를 계산하고,

이 distance를 최소화 하는 방향으로

parameter를 구한다.



$f_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2$

$(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)$

* least square

residual 의 합을 최소화 하는 해 가

→ pseudo-inverse 가 된다.

* data를 주면 함수를 return

→ 주지 않은 값에 대해서 값을 알 수 있음

< 해야 할 일 >

* 함수 정의 : $f_\theta(x)$

* 함수와 data 간의 distance 구하기

$r_i = f_\theta(x_i) - y_i$ ⇒ residual

(y 값에 대해서 안 계산함)

* 모든 residual의 (제곱의) 합.

$r_1^2 + r_2^2 + \cdots + r_n^2$

* $f_\theta(x)$

$$\underbrace{\begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ \vdots & \vdots & \vdots \\ 1 & x_n & x_n^2 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \end{bmatrix}}_{x} = \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}}_{b}$$

$Ax - b$ 의 element 하나가 r_i

$r_1^2 + r_2^2 + \cdots + r_n^2$

$= (Ax-b) \cdot (Ax-b)$

$= \|Ax-b\|_2^2$ → L2-norm square

$\|Ax-b\|_2^2$의 L2-Norm

$$
\begin{aligned}
&\|Ax-b\|^2 \\
&= (Ax-b)^T(Ax-b) \\
&= ((Ax)^T-b^T)(Ax-b) \\
&= (x^TA^T-b^T)(Ax-b) \\
&= x^TA^TAx - x^TA^Tb - b^TAx + b^Tb
\end{aligned}
$$

미분//

(A^TAx)
$+(x^TA^TA)^T$ $\quad A^Tb \quad A^Tb \quad 0$

* $x^Ty = [x_1 \cdots x_n]\begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$

x에 대해
미분하면? ⇒ y

* $y^Tx = [y_1 \cdots y_n]\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$

x에
대해 미분하면? ⇒ y

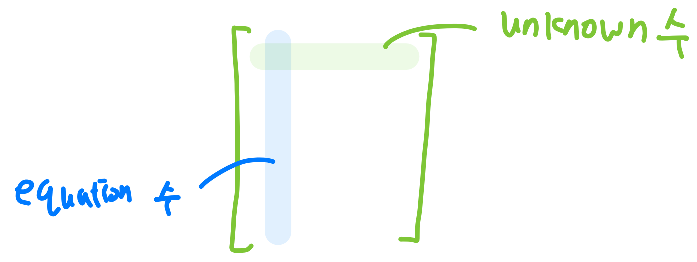
⇒ $\therefore 2A^TAx - 2A^Tb = 0$

$x = (A^TA)^{-1}A^Tb$

⇒ pesudo × b

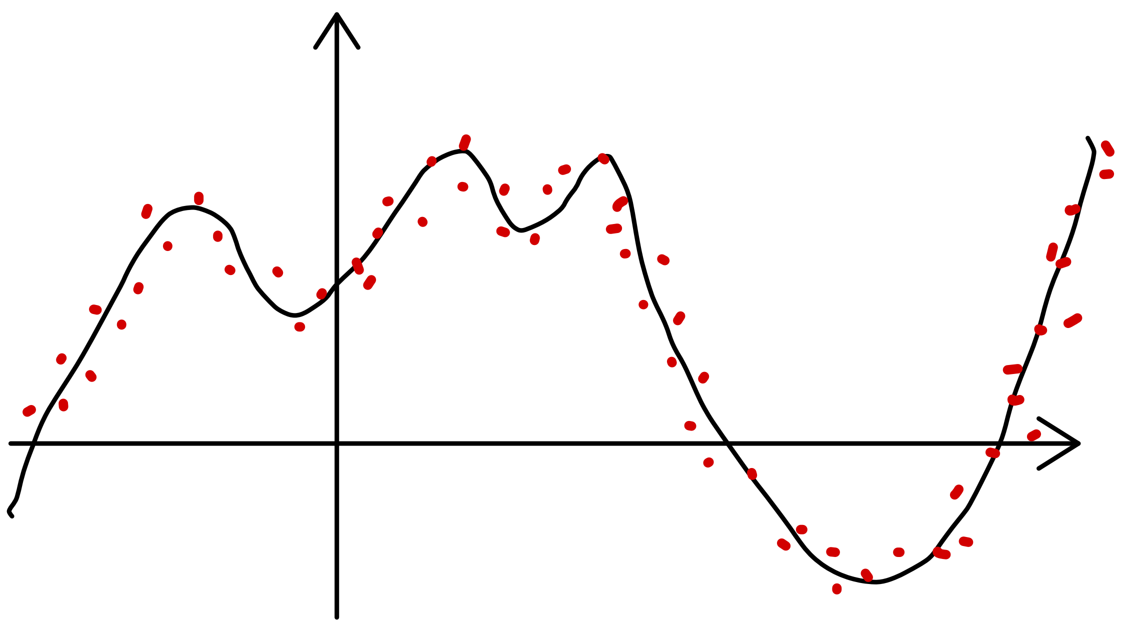
* row : unknown 의 수
col : equation의 수

unknown의 수 < equation 수
row < col
⇒ tall 이어야함.



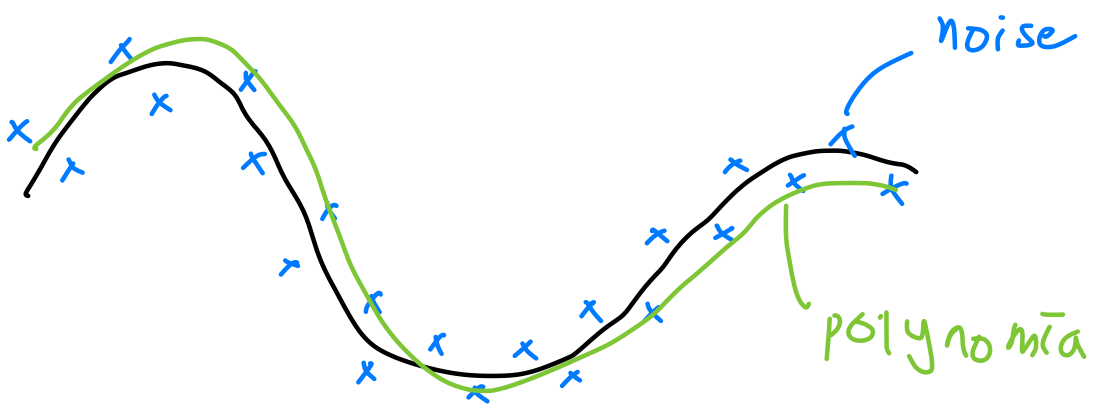
→ A가 linearly independent 해야 성립

* spatial domain



[ ]

$$z_i = x_i + \underbrace{\eta}_{noise}$$



polynomial 차수 다르게 해보면서
같이 plot

• Ax=b를 푸는데 ...

r은 polynomial 차수가 바뀌면서

결과, energy가 어떻게 되는지 (내려가는지)

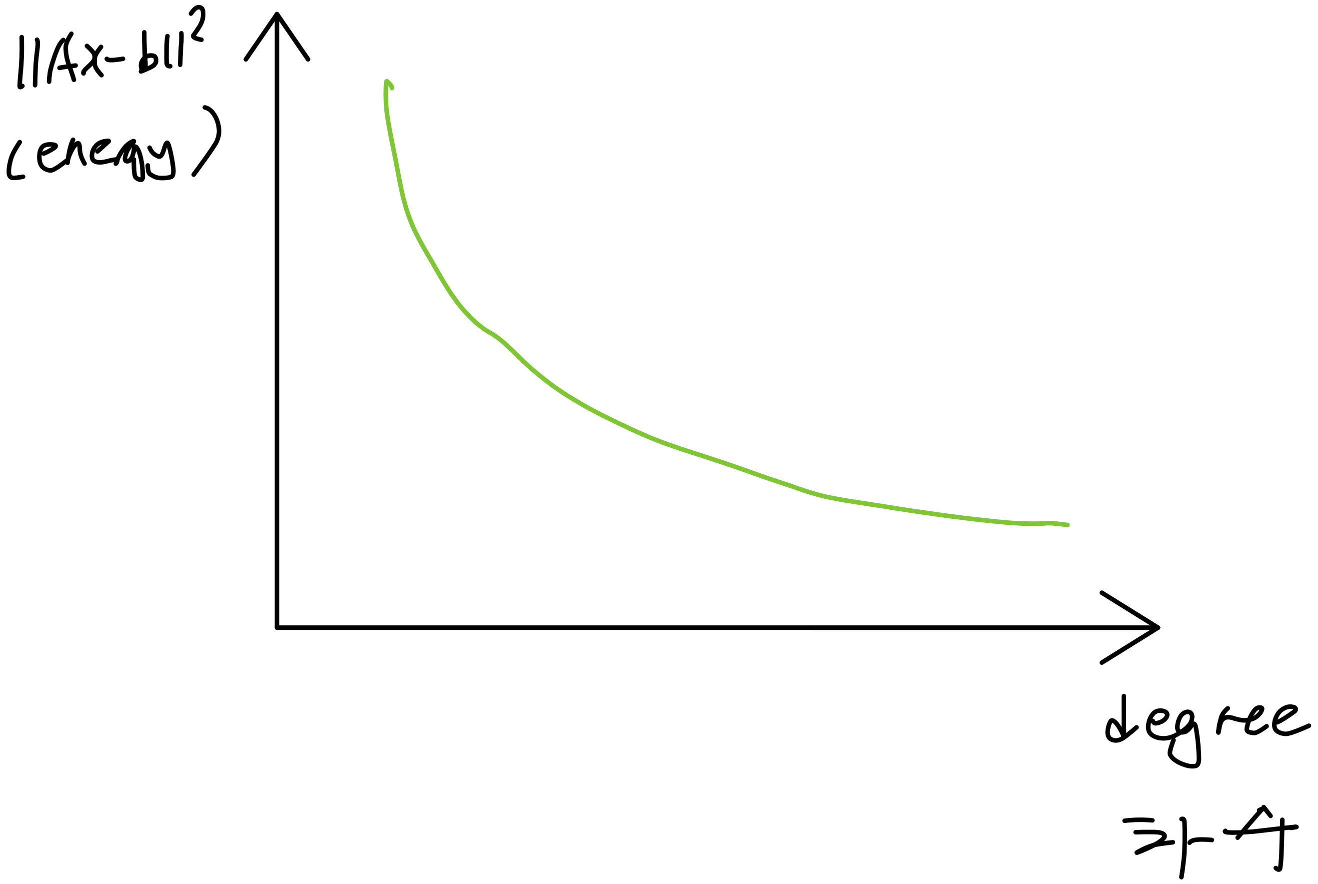
• Ax=b를 푸는 과정에서
curve가 조금 조금씩 바뀌는데

• iteration마다 θ값이 바뀌니까
→ polynomial 값 바뀜

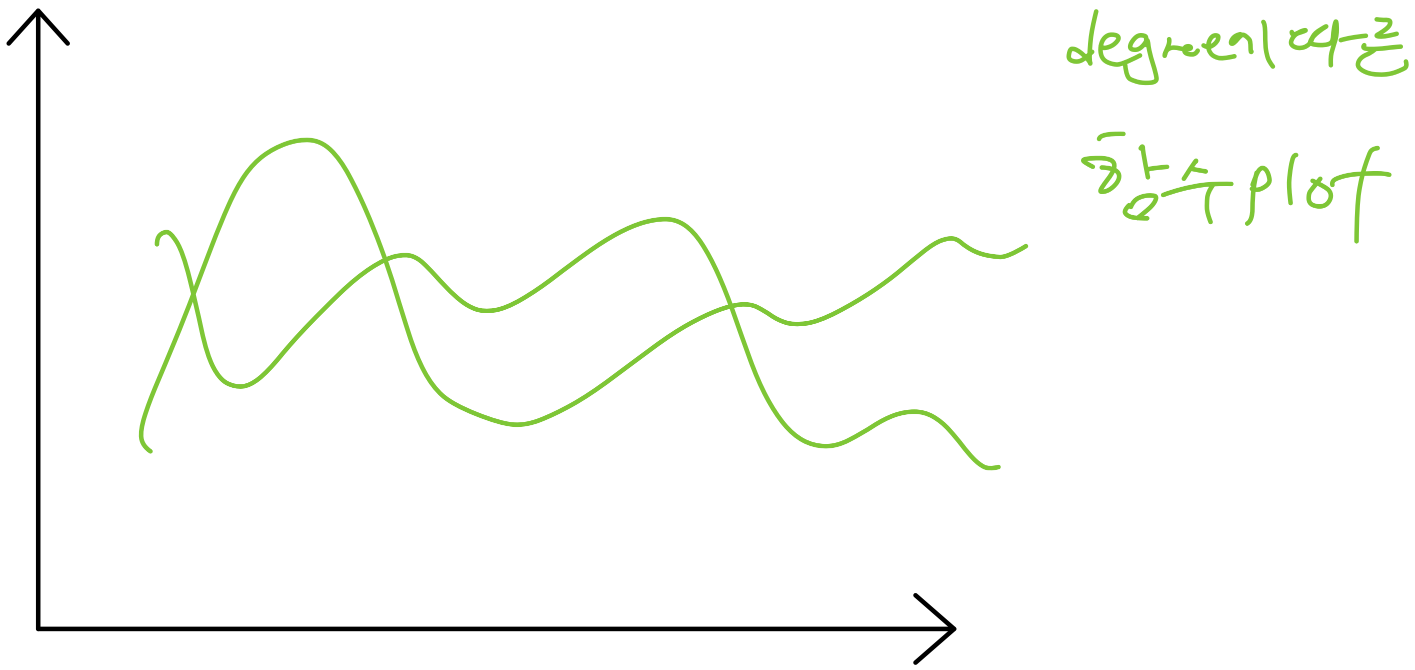
• 에너지 plot – least square

최종 Output

①



② 차수에 따른 fitting 된 함수 변화



이번에는

Iteration에 대해서는

아예 잊어!!

Assignment 09

Byung-Woo Hong 5월 25일 수업 댓글 추가

Build a binary classifier to classify digit 0 against all the other digits at MNIST dataset.

Let x = (x_1, x_2, ... , x_m) be a vector representing an image in the dataset.

The prediction function f_w(x) is defined by the linear combination of data (1, x) and the model parameter w:
f_w(x) = w_0 * 1 + w_1 * x_1 + w_2 * x_2 + ... + w_m * x_m
where w = (w_0, w_1, ... , w_m)

The prediction function f_w(x) should have the following values:
f_w(x) = +1 if label(x) = 0
f_w(x) = -1 if label(x) is not 0

The optimal model parameter w is obtained by minimizing the following objective function:
\sum_i (f_w(x^(i) - y^(i))^2

1. Compute an optimal model parameter using the training dataset
2. Compute (1) True Positive, (2) False Positive, (3) True Negative, (4) False Negative based on the computed optimal model parameter using (1) training dataset and (2) testing dataset.

Assignment 09

$x = (x_1, x_2, \cdots, x_m) \quad m = 28 \times 28$

$\hat{f}(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_m x_m$

$$\begin{cases} \text{if} \quad \ell(x^{(i)}) = 0, \text{ then } \quad y^{(i)} = +1 \\ \text{if} \quad \ell(x^{(i)}) \neq 0, \text{ then } \quad y^{(i)} = -1 \end{cases}$$

$x^{(1)} = (x_1^{(1)}, x_2^{(1)}, \cdots, x_m^{(1)})$ → 1번째이미지

$x^{(2)} = (x_1^{(2)}, x_2^{(2)}, \cdots, x_m^{(2)})$ → 2번째이미지

$\vdots$

$x^{(n)} = (x_1^{(n)}, x_2^{(n)}, \cdots, x_m^{(n)})$

$$\begin{bmatrix} 1 & x_1^{(1)} & x_2^{(1)} & \cdots & x_m^{(1)} \\ 1 & x_1^{(2)} & x_2^{(2)} & \cdots & x_m^{(2)} \\ \vdots & & & & \\ 1 & x_1^{(n)} & x_2^{(n)} & \cdots & x_m^{(n)} \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_m \end{bmatrix} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(n)} \end{bmatrix}$$

$\hat{f}(x) = \theta_0 + \theta_1 x_1 + \cdots + \theta_m x_m$

$\hat{f}(x) > 0 \rightarrow \ell(x) = 0$ → 0인 이미지 (이미지가 0)

$\hat{f}(x) < 0 \rightarrow \ell(x) \neq 0$ → 0이 아닌 이미지 (이미지가 1, 2, … 9)

$\hat{f}(x) = 0$?

① obtain optimal model

parameters : $\theta = (\theta_0, \theta_1, \cdots, \theta_m)$

using training dataset

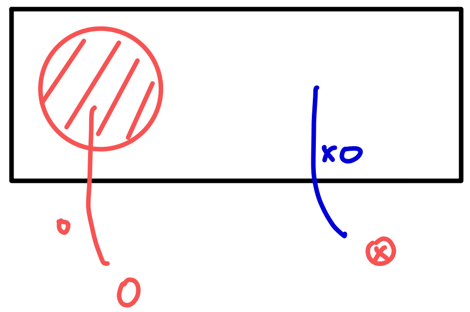
The model is defined by :

$\hat{f}(x) = \theta_0 + \theta_1 x + \cdots + \theta_m x_m$

where $x = (x_1, x_2, \cdots, x_m)$

② Compute the accuracy

using the test dataset



$\cdot \xrightarrow{?} 0$

True Positive

False positive

Answer \ Truth	O	⊗
O	True Positive (0을 0이라 함)	False Positive (0이 아닌것을 0이라고 함)
⊗	False Negative (0인데 0이 아니라고 함)	True Negative (0이 아닌데 0이 아님)

2) Accuracy

$y = \pm 1$ 이고, $f(x)$는 >0, <0, $=0$ 임

→ 계산한 값.

3) output

①

② 테이블 그리기

③ True positive

$(\hat{f}(x_1) - y_1)^2 + (f(x_2) - y_2)^2 + \cdots (f(x_n) - y_n)^2$

을 최소화 하는 $\theta_0, \cdots, \theta_m$ 구하기.

$2(\hat{f}(x_1) - y_1) \cdot \square$

$\therefore 2A^TAx - 2A^Tb = 0$

$x = (A^TA)^{-1}A^Tb$

$(m \times m)(m \times n)(n \times 1)$

⇒ pseudo × b

$m \times n$

$(1784 \times 60000)(60000 \times 1)$

$= (1784 \times 1)$

⇒ θ 구하기..?

$A^TAx = A^Tb$

(785×1)

(785×785) (785×1)

이거 구하기.

$(n \times m)(m \times 1) = (n \times 1)$

Assignment 10

Byung-Woo Hong 5월 31일 (5월 31일에 수정됨) 수업 댓글 추가

Build a binary classifier for each digit against all the other digits at MNIST dataset.

Let x = (x_1, x_2, ... , x_m) be a vector representing an image in the dataset.

The prediction function f_d(x; w) is defined by the linear combination of data (1, x) and the model parameter w for each digit d :
f_d(x; w) = w_0 * 1 + w_1 * x_1 + w_2 * x_2 + ... + w_m * x_m
where w = (w_0, w_1, ... , w_m)

The prediction function f_d(x; w) should have the following values:
f_d(x; w) = +1 if label(x) = d
f_d(x; w) = -1 if label(x) is not d

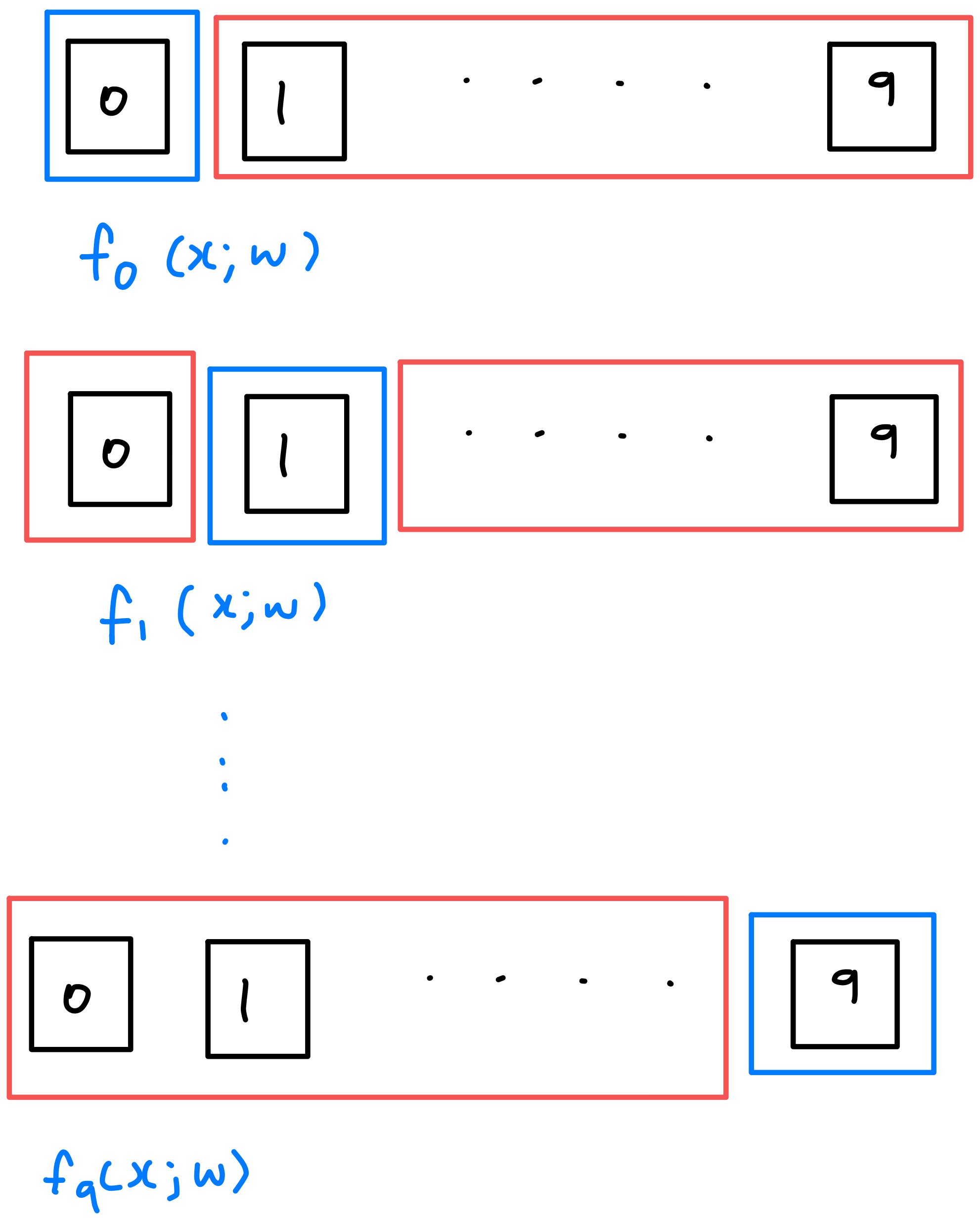
The optimal model parameter w is obtained by minimizing the following objective function for each digit d :
\sum_i (f_d(x^(i); w) - y^(i))^2

and the label of input x is given by:

argmax_d f_d(x; w)

1. Compute an optimal model parameter using the training dataset for each classifier f_d(x, w)
2. Compute (1) true positive rate, (2) error rate using (1) training dataset and (2) testing dataset.

Assignment 10



$$f_0(x;w) = \alpha_0$$

$$f_1(x;w) = \alpha_1$$

$$\vdots$$

$$f_9(x;w) = \alpha_9$$

$$l(x) = \boxed{\underset{d}{\mathrm{argmax}}}\, f_d(x;w)$$

Assignment 11

Byung-Woo Hong 5월 31일 (6월 1일에 수정됨) 수업 댓글 추가

Build a binary classifier based on k random features for each digit against all the other digits at MNIST dataset.

Let x = (x_1, x_2, ... , x_m) be a vector representing an image in the dataset.

The prediction function f_d(x; w) is defined by the linear combination of input vector x and the model parameter w for each digit d :

f_d(x; w) = w_0 * 1 + w_1 * g_1 + w_2 * g_2 + ... + w_k * g_k

where w = (w_0, w_1, ... , w_k) and the basis function g_k is defined by the inner product of random vector r_k and input vector x.

You may want to try to use g_k = max(inner production(r_k, x), 0) to see if it improves the performance.

The prediction function f_d(x; w) should have the following values:

f_d(x; w) = +1 if label(x) = d
f_d(x; w) = -1 if label(x) is not d

The optimal model parameter w is obtained by minimizing the following objective function for each digit d :
\sum_i (f_d(x^(i); w) - y^(i))^2

and the label of input x is given by:

argmax_d f_d(x; w)

1. Compute an optimal model parameter using the training dataset for each classifier f_d(x, w)
2. Compute (1) true positive rate, (2) error rate using (1) training dataset and (2) testing dataset.

Assignment 1

* feature / basis function 바꾸는 방법

: random하게 바꿔본다. → 의외로 잘 됨.

$m = 28 \times 28 = 784$

$$\hat{f}(x; w) = w_0 + w_1 x_1 + \cdots + w_m x_m$$

$$\begin{bmatrix} 1 \\ \boxed{\begin{matrix} x_1 \\ \vdots \\ x_m \end{matrix}} \end{bmatrix} \quad \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_m \end{bmatrix}$$

image

m+1개의 unknown

$$\begin{bmatrix} 1 & x_1^{(1)} & x_2^{(1)} & \cdots & x_m^{(1)} \\ 1 & x_1^{(2)} & x_2^{(2)} & \cdots & x_m^{(2)} \\ \vdots & \vdots & & & \\ 1 & x_1^{(n)} & x_2^{(n)} & \cdots & x_m^{(n)} \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_m \end{bmatrix} \approx \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(n)} \end{bmatrix}$$

feature

or

basis function

784 → m

$$\begin{bmatrix} r_1^{(1)} & r_2^{(1)} & \cdots & r_m^{(1)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} \bigcirc & \bigcirc & \cdots & \bigcirc \\ & & & \\ & & & \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_m \end{bmatrix}$$

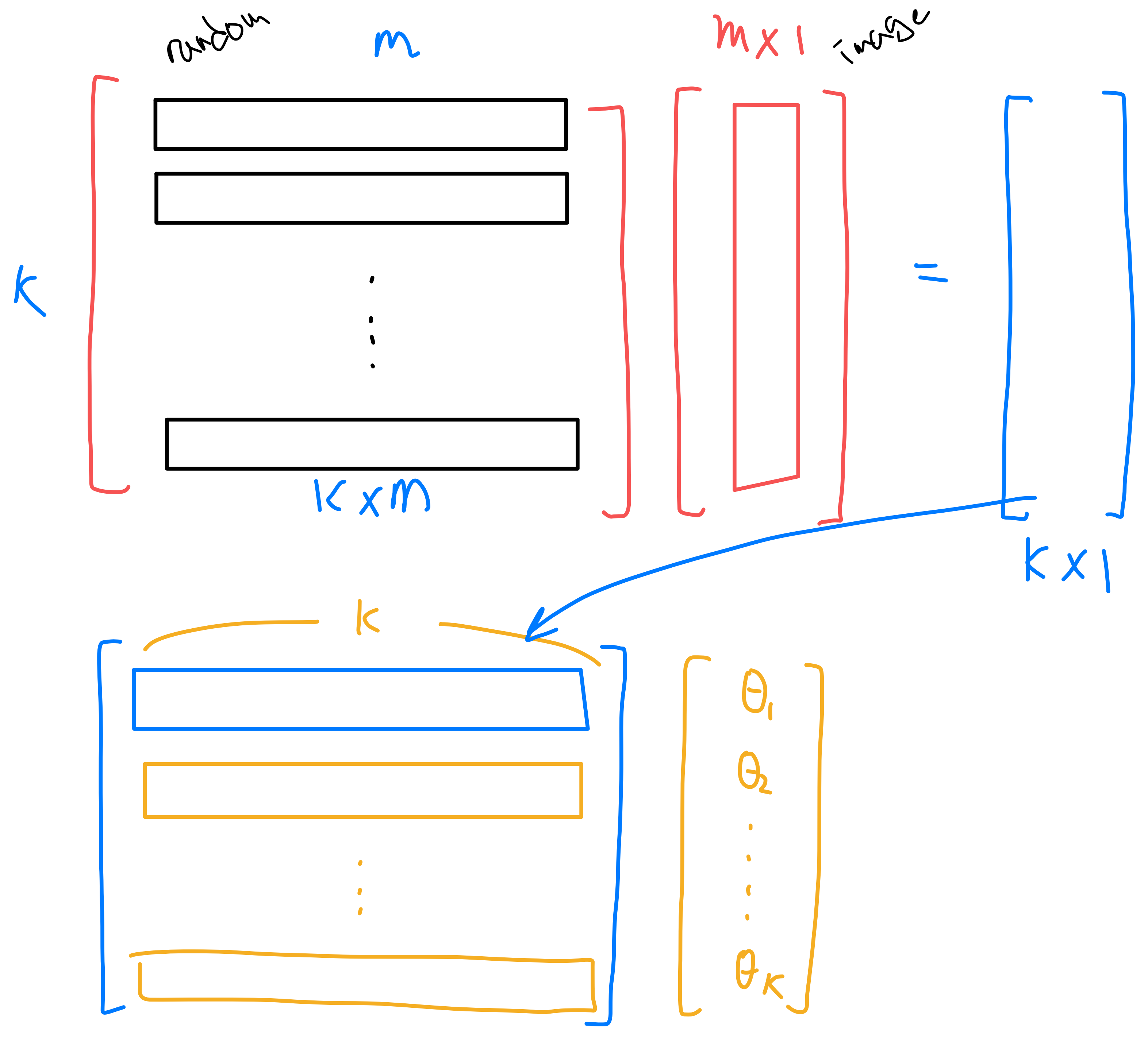
m= 0이고,

→ standard deviation은 내 맘대로 RN

$$r^{(i)} = (r_1^{(i)}, r_2^{(i)}, \cdots, r_m^{(i)})$$

$$x^{(j)} = (x_1^{(j)}, x_2^{(j)}, \cdots, x_m^{(j)})$$

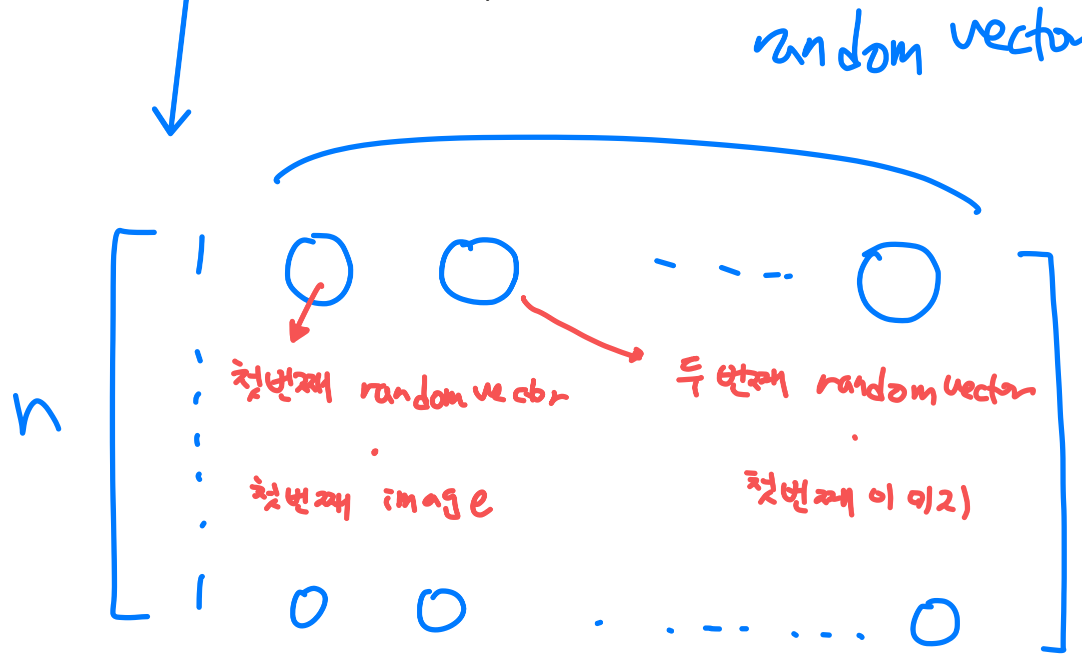
$$\begin{bmatrix} 1 & r^{(1)T}x^{(1)} & r^{(2)T}x^{(1)} & \cdots & r^{(m)T}x^{(1)} \\ & & & & \\ & & & & \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_m \end{bmatrix} \approx \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(n)} \end{bmatrix}$$



↑
여기가 A가 됨.

$$\|Ax - b\|^2$$

random vector의 수



k개의 random vector ⇒ unknown의 수 : k개

k

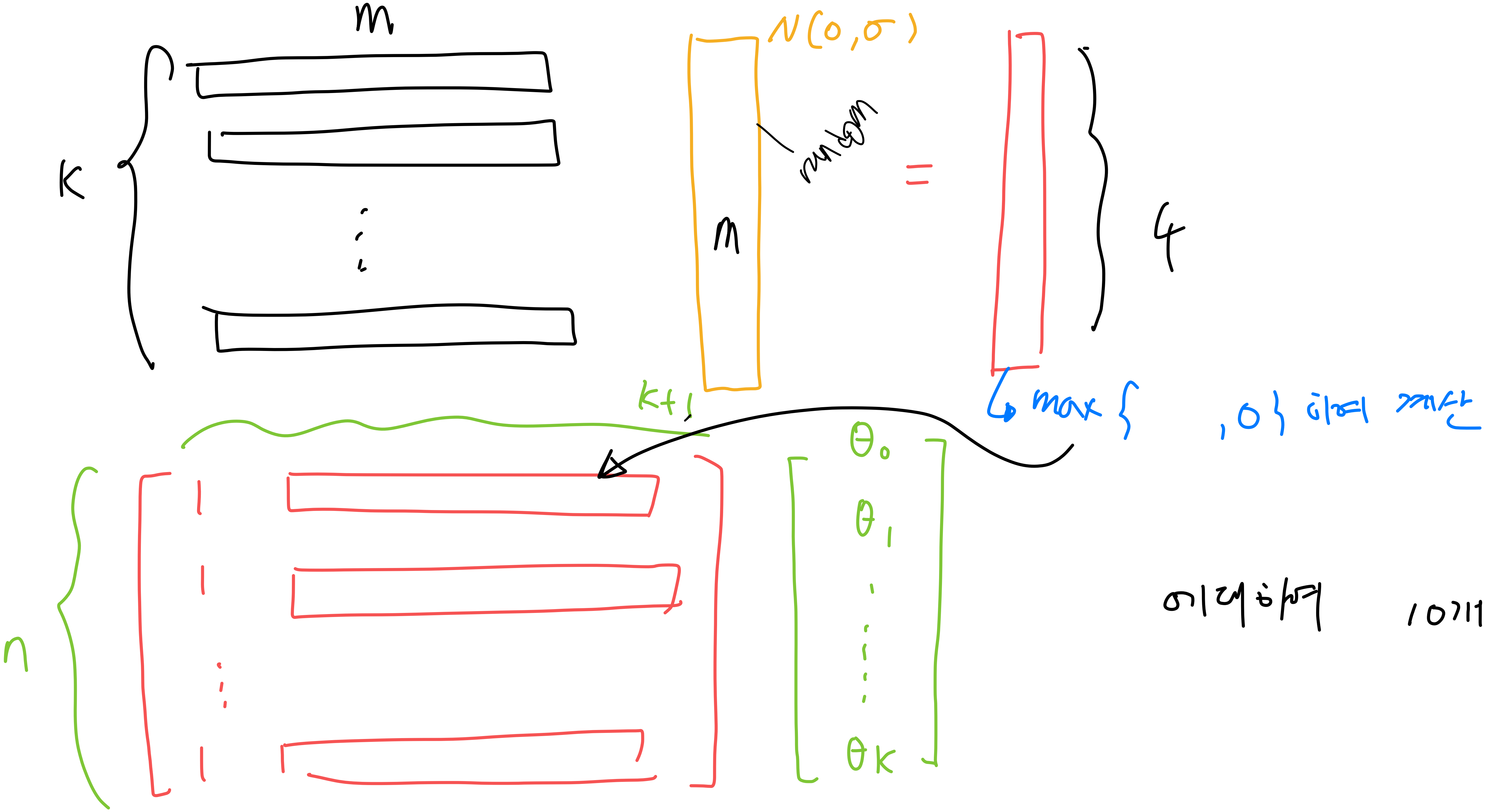
size가 m인 random vector를 k개 생성

→

input 이미지가 whitening

→ rv ⇒ 0/1 방향으로 결과가 나옴으로써

결과 성능 좋아짐.



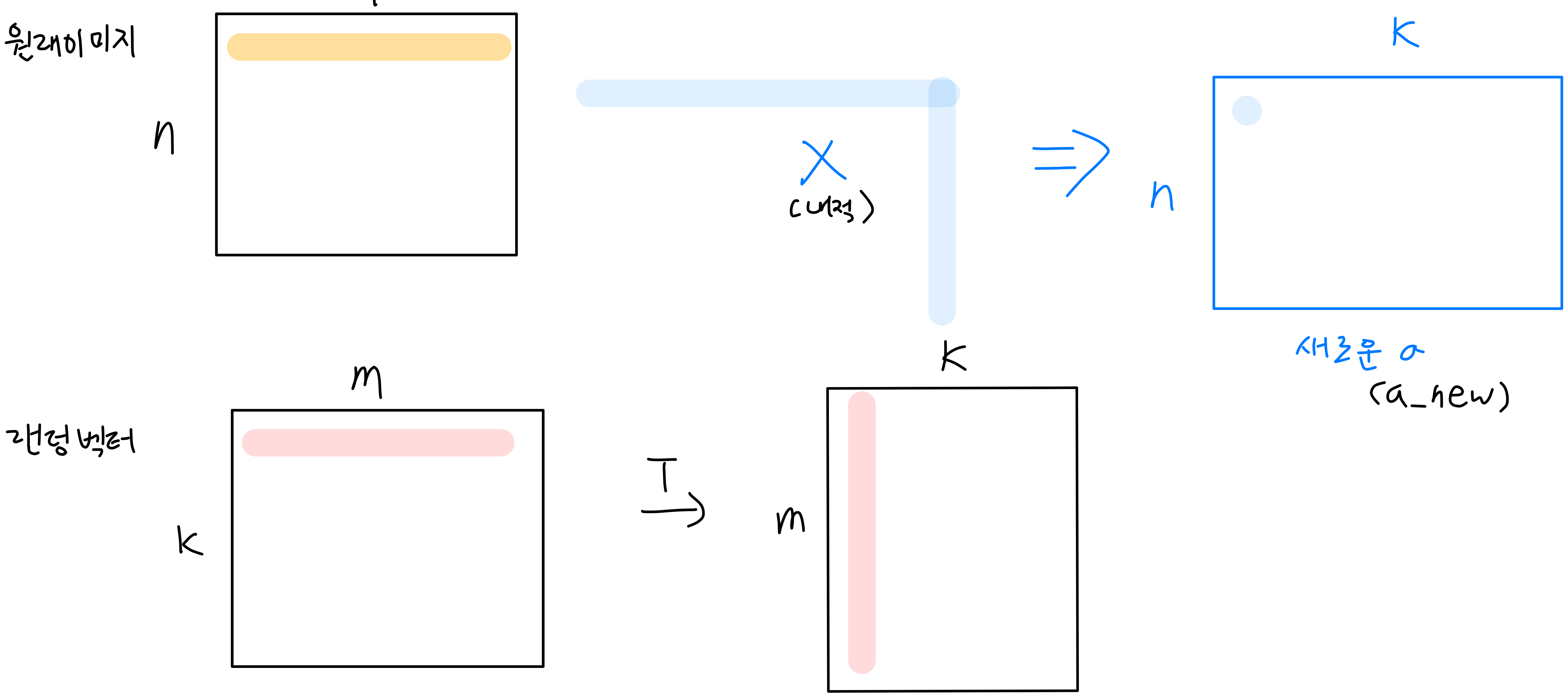
이대하여 10개에 대해 진행

$$\max\{\ ,0\}\ [1,\ 3,\ -2,\ 5,\ -n]$$

$$\Downarrow$$

$$[1,\ 3,\ 0,\ 5,\ 0]$$

①



② $a\_new \cdot \theta = b$ — 이전과 동일

$\theta \rightarrow$ Least Square로 계산

③ $y = a\_new \cdot \theta$ → 이전과 동일하게

↑ 요거 구하기

↓

이후로는 이전과정과 동일!!

but, test도 ①과 같은 방식으로 새로운 a를 이용하여 $y = a\_new \cdot \theta$ 계산