學號：B05602052　系級：　電機四　姓名：舒泓諭

1. 請使用不同的 Autoencoder model，以及不同的降維方式(降到不同維度)，討論其 reconstruction loss & public / private accuracy。（因此模型需要兩種，降維方法也需要兩種，但 clustrering 不用兩種。）

不同 Model 比較：

Convolution 完加入 Fully connected

```
class Autoencoder(nn.Module):
    def __init__(self):
        super(Autoencoder, self).__init__()
        self.encoder = nn.Sequential(#32*32
          nn.Conv2d(3, 32*C, 3, 2, 1), #16*16
          nn.SELU(),
          nn.Conv2d(32*C,64*C, 3, 2, 1),#8*8
          nn.SELU(),
          nn.Conv2d(64*C,128*C, 3, 2, 1),
          nn.SELU()

        )
        # define: decoder
        self.decoder = nn.Sequential(
          nn.ConvTranspose2d( 128*C,64*C, 2, 2),
          nn.ConvTranspose2d(64*C,32*C, 2, 2),
          nn.ConvTranspose2d(32*C, 3, 2, 2),
          nn.Tanh()
        )
        self.linear_i=nn.Linear(z_input,hidden)
        self.linear_o=nn.Linear(hidden,z_input)

    def forward(self, x):
        encoded = self.encoder(x)
        encoded=self.linear_i(encoded.view(encoded.size(0), -1))
        #print(encoded.shape)
        hold=self.linear_o(encoded)
        decoded = self.decoder(hold.reshape(-1,128,4,4))
        return hold, decoded
```

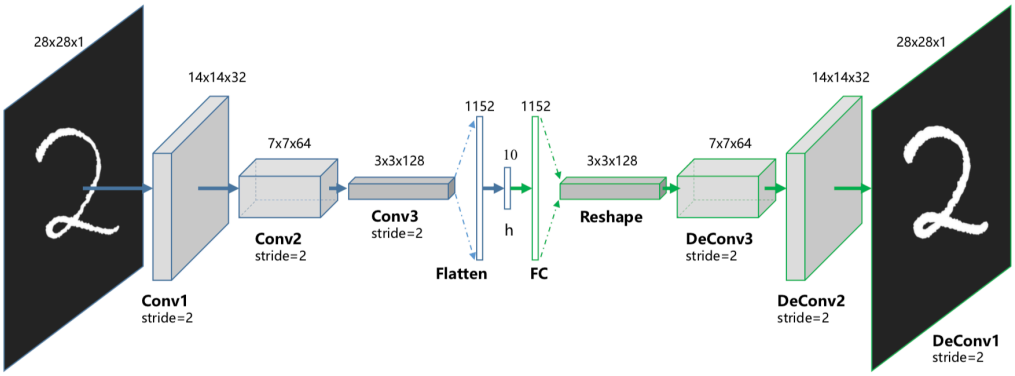
Convolution 完沒有加入 Fully connected

```
class Autoencoder(nn.Module):
    def __init__(self):
        super(Autoencoder, self).__init__()
        self.encoder = nn.Sequential(#32*32
          nn.Conv2d(3, 256, 3, 2, 1), #16*16
          nn.SELU(),
          nn.Conv2d(256,128, 3, 2, 1),#8*8
          nn.SELU(),
          nn.Conv2d(128,64, 3, 2, 1),  #4*4
          nn.SELU()

        )
        self.decoder = nn.Sequential(
          nn.ConvTranspose2d( 64,128, 2, 2),
          nn.ConvTranspose2d(128,256, 2, 2),
          nn.ConvTranspose2d(256, 3, 2, 2),
          nn.Tanh()

        )
    def forward(self, x):
        encoded = self.encoder(x)
        decoded = self.decoder(encoded)
        return encoded, decoded
```

Public	0.81518	0.75984
Private	0.81190	0.75740
reconstruction loss	0.0017	0.0006



架構參考自 Deep Clustering with Convolutional Autoencoders(Xifeng Guo2017)

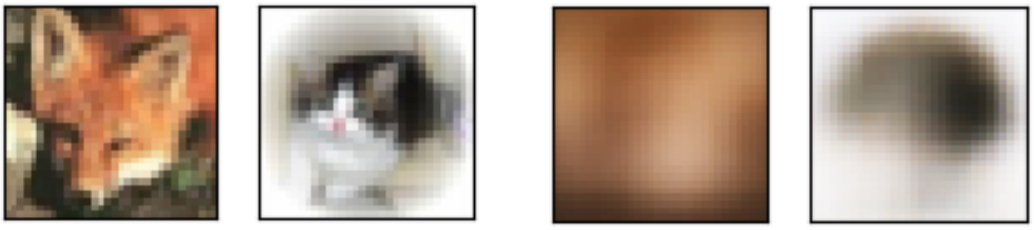
Encoder 和 Decoder 再經過 convolution 完之後，加入 fully connected layer，可以發現不論是 Publicc 還是 Private 分數都上升很多。此外，還可以發現 **reconstruction loss** 有經過 Fully connected layer，會比較難 reconstruct 回原本的圖型，其 reconstruction loss 會比較高，但也是因為這樣不論是 Publicc 還是 Private 分數都上升很多。

不同維度之比較：

降維至 12 維		**降維至 128 維**
Public	0.80555	0.72333
Private	0.80285	0.73126
reconstruction loss	`0.00006`	`0.00001`

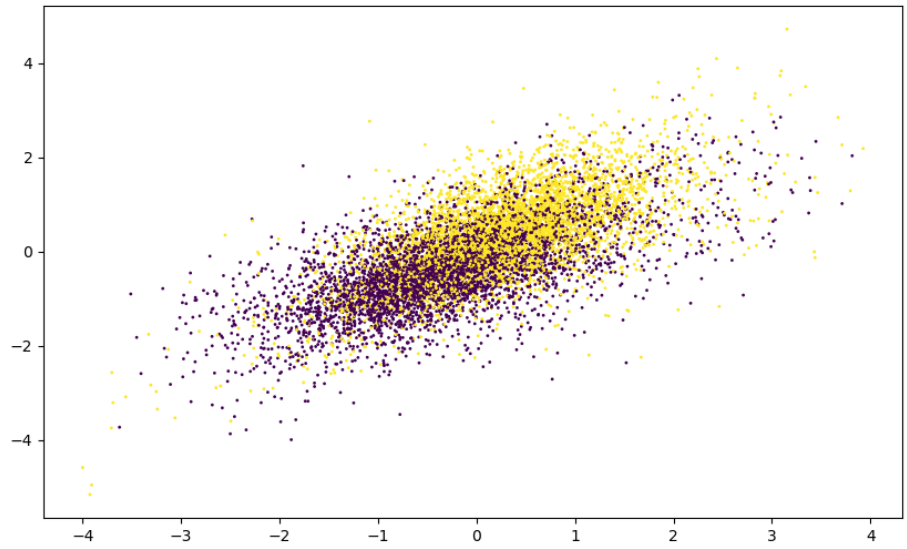
可以發現降維至 128 維 **reconstruction loss** 會降低很多，因為降至 128 維比較好把原本的 data reconstruct 回來，不過 Publicc 還是 Private 分數都會比降維至 12 維來的低。所以可以發現在這一個 task 中，找到比較重要的那幾個 eigen vector 會比，使用大量的 eigen vector 來的重要很多。此一現象可以在 Publicc 還有 Private score 發現。

2. 從 dataset 選出 2 張圖，並貼上原圖以及經過 autoencoder 後 reconstruct 的圖片。
   可以發現大致可以把圖片的的大致色塊解回來，但卻解不回清晰的輪廓以及邊界。



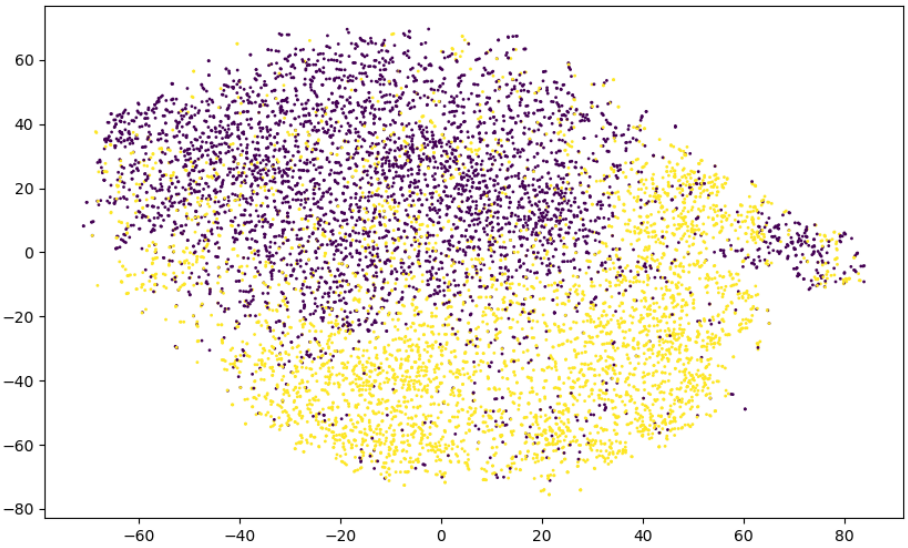
3. 在之後我們會給你 dataset 的 label。請在二維平面上視覺化 label 的分佈

Cluster 之前的:



Cluster 之後的:

可以發現在分類之後，黃色的資料會往右下角集中，紫色的資料會往左上角集中。



4. Principle Component Analysis (1%)

Given 10 samples in 3D: (1, 2, 3), (4, 8, 5), (3, 12, 9), (1, 8, 5), (5, 14, 2), (7, 4, 1), (9, 8, 9), (3, 8, 1), (11, 5, 6), (10, 11, 7)

(a) principal axes:

$$\boldsymbol{\lambda_1, \lambda_2, \lambda_3 = (15.2974434)(11.63052369)(5.47203291)}$$

$$\begin{pmatrix} \mathbf{0.6165947} \\ \mathbf{0.58881629} \\ \mathbf{0.52259579} \end{pmatrix}\begin{pmatrix} \mathbf{0.67817891} \\ \mathbf{-0.73439013} \\ \mathbf{0.02728563} \end{pmatrix}\begin{pmatrix} \mathbf{0.39985541} \\ \mathbf{0.33758926} \\ \mathbf{-0.85214385} \end{pmatrix}$$

(b) Please compute the principal components for each sample.

在算這一題時，會發現用程式親自算和 sklearn 的 pca 算出不一樣的答案，主要原因在於 PCA 在計算時，會先對資料平移平均數，所以 pca 解出來的 principal component 才會和手算的有誤差。

principal components for each sample:

$$\begin{pmatrix} \mathbf{3.36201464} \\ \mathbf{-0.70874446} \\ \mathbf{-1.48139761} \end{pmatrix}\begin{pmatrix} \mathbf{9.78988804} \\ \mathbf{-3.02597728} \\ \mathbf{0.03941652} \end{pmatrix}\begin{pmatrix} \mathbf{13.61894165} \\ \mathbf{-6.53257419} \\ \mathbf{-2.41865723} \end{pmatrix}\begin{pmatrix} \mathbf{7.94010395} \\ \mathbf{-5.06051399} \\ \mathbf{-1.16014972} \end{pmatrix}\begin{pmatrix} \mathbf{12.37159312} \\ \mathbf{-6.83599606} \\ \mathbf{5.02123906} \end{pmatrix}$$

$$\begin{pmatrix} \mathbf{7.19402383} \\ \mathbf{1.83697744} \\ \mathbf{3.29720109} \end{pmatrix}\begin{pmatrix} \mathbf{14.96324467} \\ \mathbf{0.47405978} \\ \mathbf{-1.36988181} \end{pmatrix}\begin{pmatrix} \mathbf{7.0829102} \\ \mathbf{-3.81329871} \\ \mathbf{3.0481365} \end{pmatrix}\begin{pmatrix} \mathbf{12.86219784} \\ \mathbf{3.95173109} \\ \mathbf{0.97349277} \end{pmatrix}\begin{pmatrix} \mathbf{16.30109667} \\ \mathbf{-1.10550298} \\ \mathbf{1.74702909} \end{pmatrix}$$

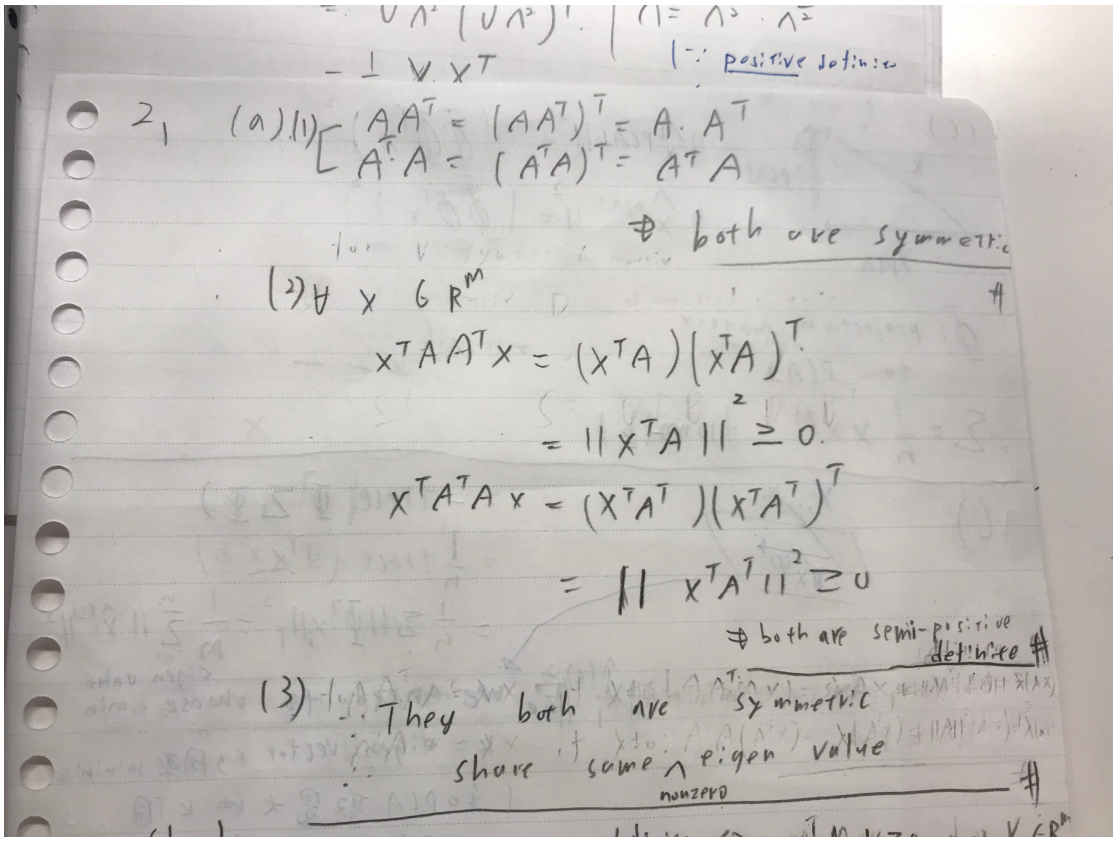
(c) What is the average reconstruction error if reduce dimension to 2D?

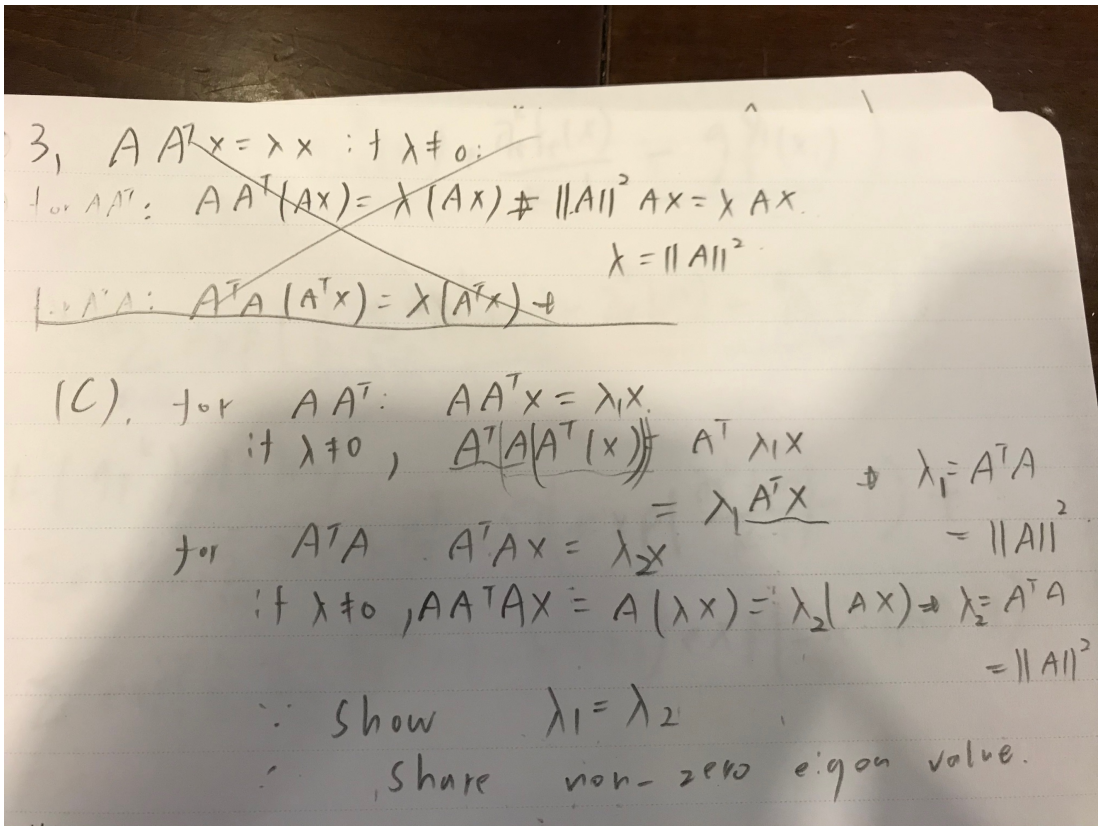
解回來之後把每一筆 data 的誤差相加後，再取平均。

6.064383051974749

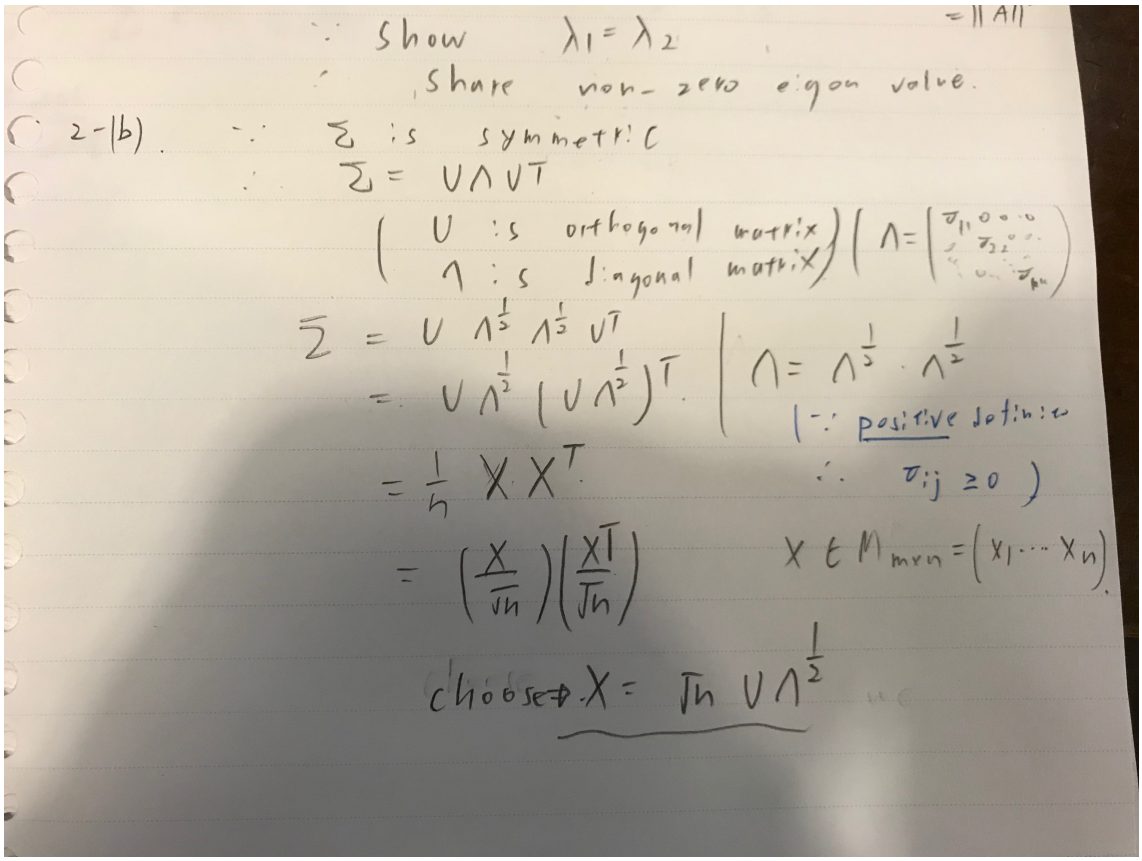
5.

a.

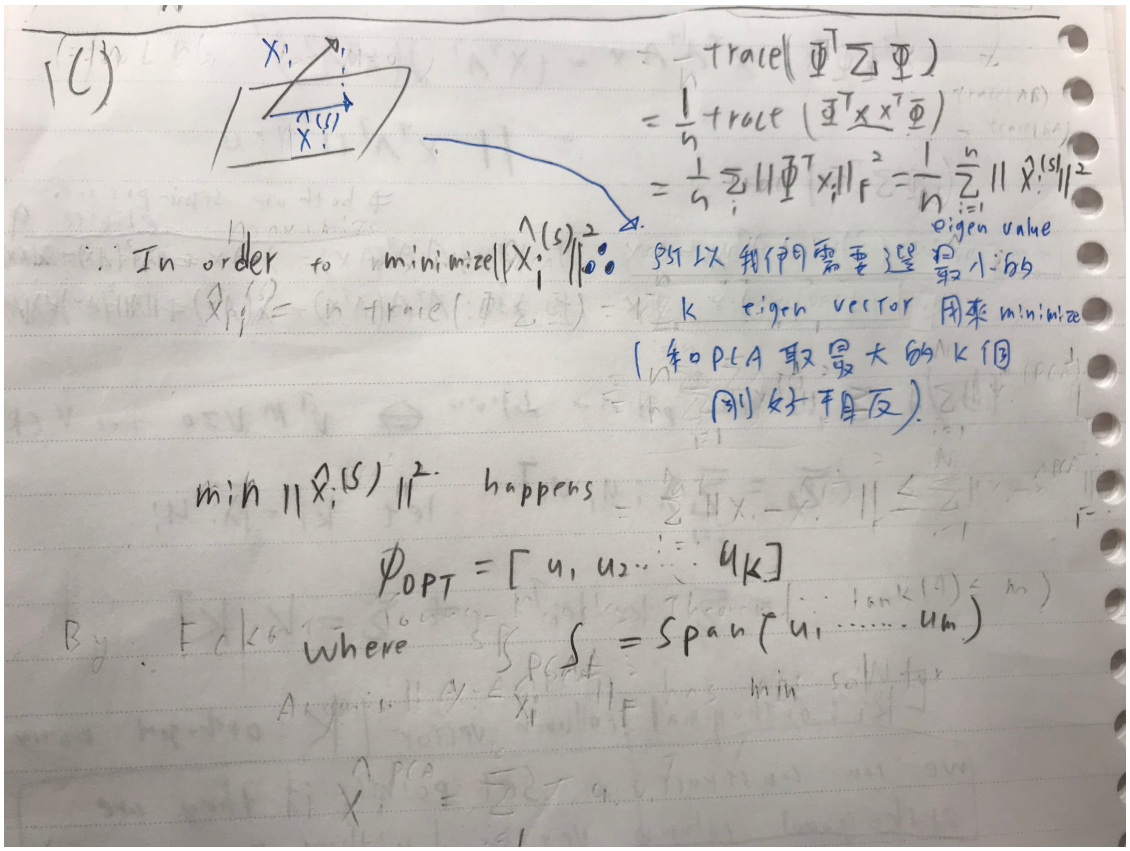

positive definite
2, (a)(1)
$AA^T = (AA^T)^T = A \cdot A^T$
$A^TA = (A^TA)^T = A^TA$
⇒ both are symmetric
(2) $\forall x \in R^m$
$x^TAA^Tx = (x^TA)(x^TA)^T$
$= \|x^TA\|^2 \geq 0$
$x^TA^TAx = (x^TA^T)(x^TA^T)^T$
$= \|x^TA^T\|^2 \geq 0$
⇒ both are semi-positive definite
(3) They both are symmetric
share same nonzero eigen value


3, $AA^Tx = \lambda x$: if $\lambda \neq 0$;
for AA^T: $AA^T(Ax) = \lambda(Ax) \Rightarrow \|A\|^2 Ax = \lambda Ax$
$\lambda = \|A\|^2$
for A^TA: $A^TA(A^Tx) = \lambda(A^Tx)$
(C). for AA^T: $AA^Tx = \lambda_1 x$
if $\lambda \neq 0$, $A^TA(A^Tx) = A^T\lambda_1 x$
$= \lambda_1 A^Tx$ ⇒ $\lambda_1 = A^TA$
$= \|A\|^2$
for A^TA $A^TAx = \lambda_2 x$
if $\lambda \neq 0$, $AA^TAx = A(\lambda x) = \lambda_2(Ax)$ ⇒ $\lambda_2 = A^TA$
$= \|A\|^2$
∴ Show $\lambda_1 = \lambda_2$
∴ share non-zero eigen value.

b.


∵ show $\lambda_1 = \lambda_2$
share non-zero eigen value.
2-(b) ∵ Σ is symmetric
$\Sigma = U \Lambda U^T$
(U : s orthogonal matrix
Λ : s diagonal matrix) ($\Lambda = \begin{pmatrix} \sigma_{11} & 0 & \cdots & 0 \\ 0 & \sigma_{22} & \cdots & \\ \vdots & & \ddots & \\ 0 & & & \sigma_{kk} \end{pmatrix}$
$\Sigma = U \Lambda^{\frac{1}{2}} \Lambda^{\frac{1}{2}} U^T$
$= U \Lambda^{\frac{1}{2}} (U \Lambda^{\frac{1}{2}})^T$ | $\Lambda = \Lambda^{\frac{1}{2}} \cdot \Lambda^{\frac{1}{2}}$
| ∵ positive definite
∴ $\sigma_{ij} \geq 0$)
$= \frac{1}{n} X X^T$
$= \left(\frac{X}{\sqrt{n}}\right)\left(\frac{X^T}{\sqrt{n}}\right)$ $X \in M_{m \times n} = (x_1 \cdots x_n)$
choose $X = \sqrt{n}\, U \Lambda^{\frac{1}{2}}$

c.


(c)
x_i
$\hat{x}_i^{(s)}$
$= \frac{1}{n}\text{trace}(\Phi^T \Sigma \Phi)$
$= \frac{1}{n}\text{trace}(\Phi^T X X^T \Phi)$
$= \frac{1}{n}\sum_i \|\Phi^T x_i\|_F^2 = \frac{1}{n}\sum_{i=1}^{n} \|\hat{x}_i^{(s)}\|^2$
eigen value
∴ In order to minimize $\|\hat{x}_i^{(s)}\|^2$
所以我們需要選最小的
k eigen vector 用來 minimize
(和PCA取最大的k個
剛好相反)
min $\|\hat{x}_i^{(s)}\|^2$ happens
$\Phi_{OPT} = [u_1, u_2 \cdots u_k]$
where $S = \text{Span}(u_1 \cdots u_m)$

6.

3. sketch of proof

For $\ell = 0, 1, \cdots \frac{T}{K} - 1$

For $k = 1, 2, \cdots, K$

$t = \ell K + k$

- Find $f_t \in F$ s.t. $\frac{\partial}{\partial a} L(g_{t-1}^1 \cdots g_{t-1}^k + a f_t \cdots g_{t-1}^K)$ is minimized

- Find a_t s.t. $L(g_{t-1}^1 \cdots g_{t-1}^k + a f_t, \cdots)$ is minimized

For k th class.

First we find f_t s.t. $L(\cdots g_t) < L(\cdots g_{t-1} \cdots)$

$$L(g_t^1 \cdots g_t^k \cdots g_t^K) = \sum_n \exp\left(\frac{1}{K-1}\left(\sum_{k \neq y_i} g_t^k(x_i) - g_t^{y_i}(x_i)\right)\right)$$

$$= \sum \exp\left(\frac{1}{K-1}\left(\sum_{k \neq y_i} \sum_{t=1}^{T} a_t^k f_t(x) - \sum_{t=1}^{T} a_t^{y_i} f_t(x)\right)\right)$$

接下來分成 $y_i \neq k$、$y_i = k$ 2 個 case 探討.

$$[AA^T = (AA^T)^T = A \cdot A^T$$
$$A^T A = (A^T A)^T = A^T \cdot A$$

$$= \sum_{y_i \neq k} \exp\left(\frac{1}{k-1} \sum_{k \neq y_i} g_{T-1}^{k}(x_i) + \frac{a_t^k f_t(x)}{k-1} - g_T^{y_i}(x_i) \right) +$$

$$\sum_{y_i = k} \exp\left(\frac{1}{k-1} \sum_{k \neq y_i} g_{T-1}^{k}(x_i) - a_t^k f_t(x) - g_T^{y_i}(x_i) \right)$$

$$\Delta L = L(g_T^k) - L(g_{T-1})^k$$

$$\simeq + \frac{a_T^k f_T(k)}{k-1} \quad (\because a_t^k \ll 1)$$

$$= \sum \exp\left(\frac{1}{k-1} \sum_{k \neq y_i} g_{T-1}^{k}(x_i) - g_{T-1}^{y_i}(x_i) \right] \left(\exp\left(\frac{a_T^k f_T(y_i)}{k-1} \right) - 1 \right) +$$

$$\sum_{y_i = k} \exp\left(\frac{1}{k-1} \sum_{k \neq y_i} g_{T-1}^{k}(x_i) - g_{T-1}^{y_i}(x_i) \right) \left(\exp\left[-a_t^k f_t(x_i) \right] - 1 \right)$$

$$\simeq - a_T^k f_T(x_i) \quad (\because a_T^k \ll 1)$$

$$\Delta L \simeq \sum \exp\left(\frac{1}{k-1} \sum g_{T-1}^{k}(x_i) - g_{T-1}^{y_i}(x_i) \right) \frac{a_t^k f_t(k)}{k-1}$$

$$= \sum \exp\left(\frac{1}{k-1} \sum g_{T-1}^{k}(x_i) - g_{T-1}^{y_i}(x_i) \right) a_t^k f_t(x_i)$$

$$\frac{\partial L}{\partial a_t} = \frac{f_t(k)}{k-1} \sum_{y_i \neq k} \exp\left(\frac{1}{k-1} \sum g_{T-1}^{k}(x_i) - g_{T-1}^{y_i}(x_i) \right)$$
$$- f_t(x_i) \sum_{y_i = k} \exp\left(\frac{1}{k-1} \sum g_{T-1}^{k}(x_i) - g_{T-1}^{y_i}(x_i) \right)$$

Q.E.D

We show $f_t \leftarrow f_{t-1} - \eta \frac{\partial L}{\partial a_t}$ ≒ like gradient descent

find a_t

$$\frac{\partial L}{\partial a_t} = \underbrace{\sum_{y_i \neq k} u_t^{\,n}}_{z_t \epsilon_t} e^{a_t} \cdot \frac{1}{k-1} - \underbrace{\sum_{y_i = k} u_t^{\,n}}_{z_t(1-\epsilon_t)} e^{-a_t} = 0$$

$$\Rightarrow z_t \epsilon_t e^{a_t} \frac{1}{k-1} - z_t (1-\epsilon_t) e^{-a_t} = 0$$

$$e^{2a_t} = \left(\frac{1-\epsilon_t}{\epsilon_t}\right)(k-1)$$

$$a_t = \ln \sqrt{\left(\frac{1-\epsilon_t}{\epsilon_t}\right)(k-1)}$$