學號:T10902303　系級:資工一　姓名:梁高浚

Link to model(for partB):

https://drive.google.com/file/d/1fCiq7VgNaoWy2kkZWXUT72PNgz8yprlr/view?usp=sharing

Kaggle Competetion

1. (1%) 請附上你在kaggle競賽上表現最好的降維以及分群方式，並條列五種不同降維維度的設定對應到的表現(public / private accuracy)，auto-encoder 和 PCA 只要任一維度不一樣即可算是一種組合。

   我在kaggle上最好的表現是使用autoencoder_best進行訓練後，用PCA降維至200維度，然後用TSNE和kmeans進行分群，在kaggle上的準確率達到0.80355，超過public leaderboard的strong baseline。
   autoencoder_best用了以下結構：

```
self.convolution_1 = nn.Conv2d(3 , 1024 , kernel_size = (3 , 3) , stride = (1 , 1) , padding = (1 , 1))
self.maxpool_1 = nn.MaxPool2d(2 , return_indices = True)
self.convolution_2 = nn.Conv2d(1024 , 256 , kernel_size = (3 , 3) , stride = (1 , 1) , padding = (1 , 1))
self.maxpool_2 = nn.MaxPool2d(2 , return_indices = True)
self.convolution_3 = nn.Conv2d(256 , 64 , kernel_size = (3 , 3) , stride = (1 , 1) , padding = (1 , 1))
self.maxpool_3 = nn.MaxPool2d(2 , return_indices = True)
self.convolution_4 = nn.Conv2d(64 , 16 , kernel_size = (3 , 3) , stride = (1 , 1) , padding = (1 , 1))
self.maxpool_4 = nn.MaxPool2d(2 , return_indices = True)
self.maxunpool_1 = nn.MaxUnpool2d(2)
self.deconvolution_1 = nn.ConvTranspose2d(16 , 64 , kernel_size = (3 , 3) , stride = (1 , 1) , padding = (1 , 1)
self.maxunpool_2 = nn.MaxUnpool2d(2)
self.deconvolution_2 = nn.ConvTranspose2d(64 , 256 , kernel_size = (3 , 3) , stride = (1 , 1) , padding = (1 , 1
self.maxunpool_3 = nn.MaxUnpool2d(2)
self.deconvolution_3 = nn.ConvTranspose2d(256 , 1024 , kernel_size = (3 , 3) , stride = (1 , 1) , padding = (1 ,
self.maxunpool_4 = nn.MaxUnpool2d(2)
self.deconvolution_4 = nn.ConvTranspose2d(1024 , 3 , kernel_size = (3 , 3) , stride = (1 , 1) , padding = (1 , 1
self.activate = nn.Tanh()
```

   另外，列出另外四種的降維方式如下：
   1. 與最佳模型一樣，除了沒有使用PCA降維: 0.78888
   2. 與最佳模型一樣，除了使用PCA降維至2000維: 0.76822
   3. autoencoder的結構更改為如下：

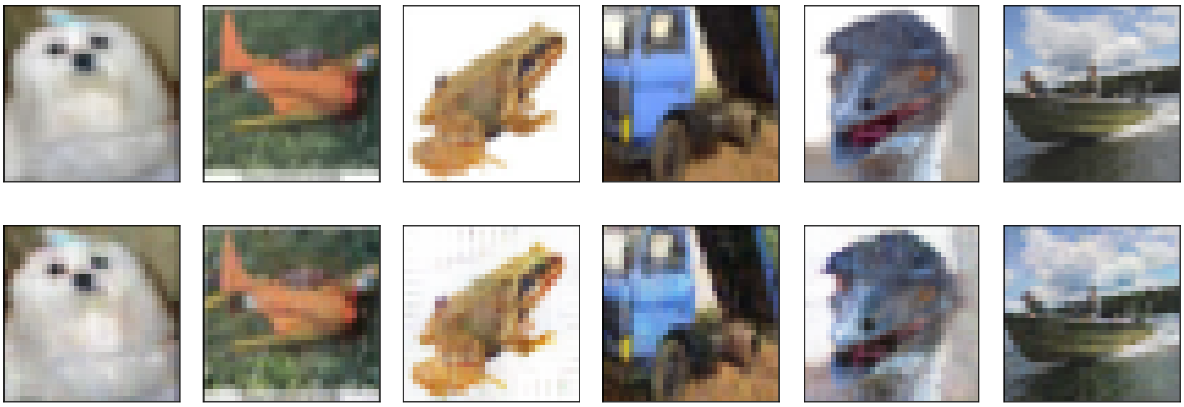
```
class Autoencoder_2(nn.Module):
    def __init__(self):
        super().__init__()
        self.convolution_1 = nn.Conv2d(3 , 1024 , kernel_size = (3 , 3) , stride = (1 , 1) , padding
        self.maxpool_1 = nn.MaxPool2d(2 , return_indices = True)
        self.convolution_2 = nn.Conv2d(1024 , 512 , kernel_size = (3 , 3) , stride = (1 , 1) , paddin
        self.maxpool_2 = nn.MaxPool2d(2 , return_indices = True)
        self.convolution_2_5 = nn.Conv2d(512 , 256 , kernel_size = (3 , 3) , stride = (1 , 1) , paddi
        self.maxpool_2_5 = nn.MaxPool2d(2 , return_indices = True)
        self.convolution_3 = nn.Conv2d(256 , 64 , kernel_size = (3 , 3) , stride = (1 , 1) , padding
        self.maxpool_3 = nn.MaxPool2d(2 , return_indices = True)
        self.convolution_4 = nn.Conv2d(64 , 16 , kernel_size = (3 , 3) , stride = (1 , 1) , padding =
        self.maxpool_4 = nn.MaxPool2d(2 , return_indices = True)
        self.maxunpool_1 = nn.MaxUnpool2d(2)
        self.deconvolution_1 = nn.ConvTranspose2d(16 , 64 , kernel_size = (3 , 3) , stride = (1 , 1)
        self.maxunpool_2 = nn.MaxUnpool2d(2)
        self.deconvolution_2 = nn.ConvTranspose2d(64 , 256 , kernel_size = (3 , 3) , stride = (1 , 1)
        self.maxunpool_2_5 = nn.MaxUnpool2d(2)
        self.deconvolution_2_5 = nn.ConvTranspose2d(256 , 512 , kernel_size = (3 , 3) , stride = (1 ,
        self.maxunpool_3 = nn.MaxUnpool2d(2)
        self.deconvolution_3 = nn.ConvTranspose2d(512 , 1024 , kernel_size = (3 , 3) , stride = (1 ,
        self.maxunpool_4 = nn.MaxUnpool2d(2)
        self.deconvolution_4 = nn.ConvTranspose2d(1024 , 3 , kernel_size = (3 , 3) , stride = (1 , 1)
        self.activate = nn.Tanh()
        return
```

(中間加多了一層convolution layer), 沒有使用PCA降維, 有使用TSNE & kmeans, 準確率為0.78400

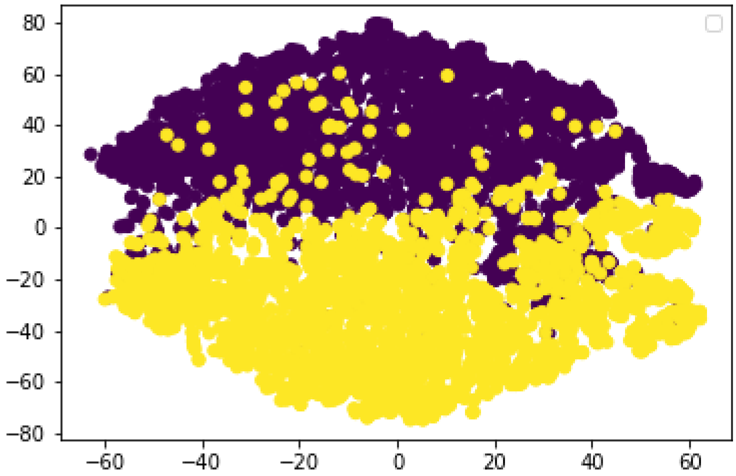
4. 與最佳模型一樣, 除了使用PCA降維至100維: 0.80088

2. (1%) 從 kaggle 的 dataset 選出 2 張圖, 並貼上原圖以及經過 autoencoder 後 reconstruct 的圖片;請將 visualization.npy 的檔案降維至二維平面並利用給定的 label 將資料上色 (前一半為 0;後半為 1)。

選出了index為1,2,3,4,5,6的圖片, 上方為原圖, 下方為經過autoencoder後重建的圖片。



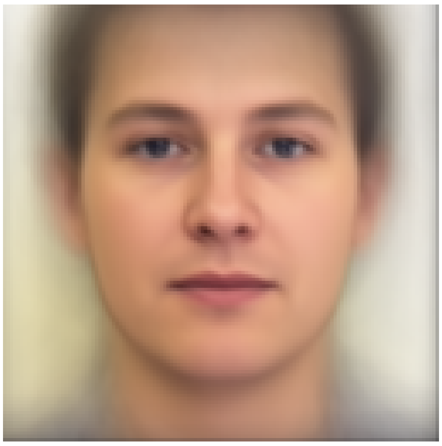
visualization.npy 的檔案降維至二維平面並利用給定的 label 將資料上色:



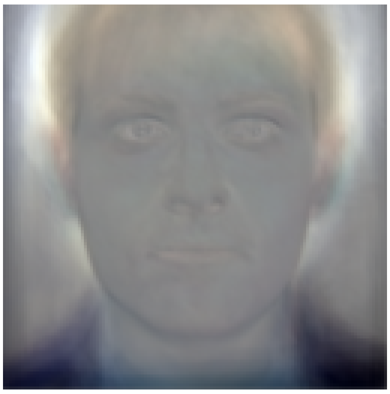
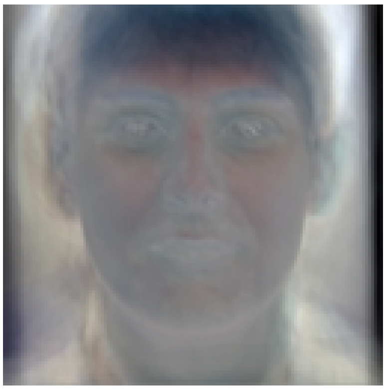
Eigenface

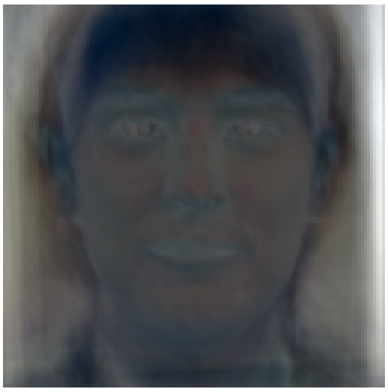
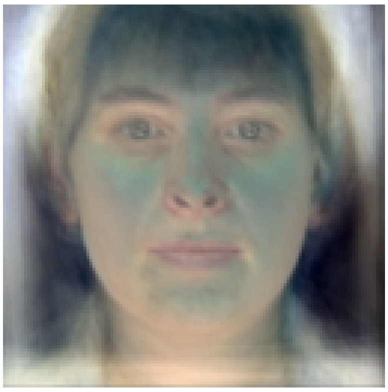
3. (1%) 請畫出所有臉的平均以及 Eigenvalue 最大的前五個 Eigenfaces。

所有臉的平均:

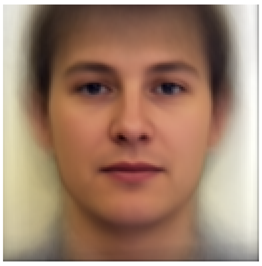
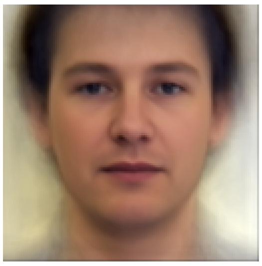
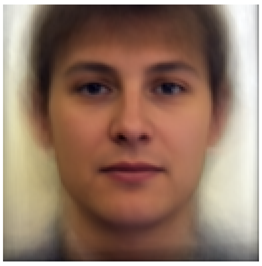
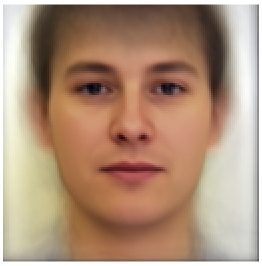


Eigenvalue 最大的前五個 Eigenfaces(由大到小排列):

4. (1%) 請從數據集中挑出任意五張圖片，並用上題前五大 Eigenfaces 進行 reconstruction，並畫出結果。

#20: , #45: , #101: , #132: , #223: 

5. (4%) Refer to math problem：
https://hackmd.io/@g4HRMJCzQL2hzLedRcbVPQ/SyCBoc1qt

Q1 Derive updated form of EM algorithm

a).

$$p(X;\theta) = \prod_{i=1}^{3} p(x_i;\theta) = \prod_{i=1}^{3} \sum_{k=1}^{2} \pi_k f_k(x_i)$$

Let $z_i \in \{1,2\}$ indicating which exponential distribution x_i is drawn from, it is the latent variable.

$z = \{z_1, z_2, z_3\}$ are collection of all latent variable.

$$\log p(X;\theta) = \sum_{i=1}^{3} p(z_i|X;\theta^{(t)}) \log(p(X;\theta))$$

$$= \sum_{i=1}^{3} p(z_i=k|X;\theta^{(t)}) (\log(X, z_i=k;\theta) - \log p(z_i=k|X;\theta))$$

$$= \underbrace{\sum_{i=1}^{3} p(z_i=k|X;\theta^{(t)}) \log(X, z_i=k;\theta)}_{Q(\theta|\theta^{(t)})} - \underbrace{\sum_{i=1}^{3} p(z_i=k|X;\theta^{(t)}) \log(p(z_i=k|X;\theta))}_{H(\theta|\theta^{(t)})}$$

For exponential distribution,

$$\log p(x_i, z_i=k;\theta) = \log\left(\frac{\pi_k}{\tau_k} e^{-\frac{x_i}{\tau_k}}\right) = \log\left(\frac{1}{\tau_k}\right) + \log\left(e^{-\frac{x_i}{\tau_k}}\right) + \log(\pi_k)$$

$$= -\log(\tau_k) - \frac{x_i}{\tau_k} + \log(\pi_k)$$

At E-step,

$$Q(\theta|\theta^{(t)}) = \sum_{i=1}^{3} p(z_i=k|x_i;\theta^{(t)}) \log(x_i, z_i=k;\theta)$$

$$= E_{z|x;\theta^{(t)}} [\log p(X, z;\theta)]$$

$$= \sum_{i=1}^{3} E_{z_i|x_i;\theta^{(t)}} [\log p(x_i, z_i;\theta)]$$

Posterior prob. dist. of latent variables z_i based on $\theta^{(t)}$:

$$p(z_i=k|x_i;\theta^{(t)}) = \frac{p(x_i, z_i=k;\theta^{(t)})}{\sum_{j=1}^{2} p(x_i, z_i=j;\theta^{(t)})} = \frac{\pi_k^{(t)} \frac{1}{\tau_k} e^{-\frac{x_i}{\tau_k}}}{\sum_{j=1}^{2} \pi_j^{(t)} \frac{1}{\tau_j} e^{-\frac{x_i}{\tau_j}}} = \delta_{ik}^{(t)}$$

Hence $$Q(\theta|\theta^{(t)}) = \sum_{i=1}^{3} \sum_{k=1}^{2} \delta_{ik}^{(t)} \left\{ -\log(\tau_k) - \frac{x_i}{\tau_k} + \log(\pi_k) \right\}$$

At M-step, choose $\theta^{(t+1)} = \underset{\theta \in \Theta}{\arg\max}\, Q(\theta|\theta^{(t)})$

We compute $\nabla_{\tau_k} Q(\theta|\theta^{(t)}) = 0$.

$$\nabla_{\tau_k} Q(\theta|\theta^{(t)}) = \sum_{i=1}^{3} \delta_{ik}^{(t)} \left\{ -\frac{1}{\tau_k} + \frac{x_i}{\tau_k^2} \right\}$$

$$\nabla_{\tau_k} Q(\theta|\theta^{(t)}) = \sum_{i=1}^{3} \delta_{ik}^{(t)} \left\{ \frac{x_i - \tau_k}{\tau_k^2} \right\}$$

So $\tau_k^{(t+1)}$ is $\dfrac{\sum_{i=1}^{3} \delta_{ik}^{(t)} x_i}{\sum_{i=1}^{3} \delta_{ik}^{(t)}}$

$$\pi_k^{(t+1)} = \frac{1}{N} \sum_{i=1}^{3} \delta_{ik}^{(t)}$$

c).

	i	1	2	3
	x_i	0	2	4
$f_1(x) = e^{-x}$	$p(x_i, z_i = 1; \theta^{(t)})$	$0.5e^{-0} = 0.5$	$0.5e^{-2} = 0.067668$	$0.5e^{-4} = 0.0091578$
$f_2(x) = \frac{1}{2}e^{-\frac{x}{2}}$	$p(x_i, z_i = 2; \theta^{(t)})$	$0.5(\frac{1}{2}e^{0}) = 0.25$	$0.5(\frac{1}{2}e^{-\frac{2}{2}}) = 0.09197$	$0.5(\frac{1}{2}e^{-\frac{4}{2}}) = 0.033834$
	$\sum_{j=1}^{2} p(x_i, z_i = j; \theta^{(t)})$	0.75	0.159638	0.0429918
	$\delta_{i1}^{(t)}$	0.6667	0.423884	0.2130127
	$\delta_{i2}^{(t)}$	0.3333	0.576116	0.7869873

d). $$\tau_1^{[1]} = \frac{0.6667 \times 0 + 0.423884 \times 2 + 0.2130127 \times 4}{0.6667 + 0.423884 + 0.2130127} = 1.303945$$

$$\tau_2^{[1]} = \frac{0.3333 \times 0 + 0.576116 \times 2 + 0.7869873 \times 4}{0.3333 + 0.576116 + 0.7869873} = 2.534881$$

$$\pi_1^{[1]} = \frac{1}{3}(0.6667 + 0.423884 + 0.2130127) = 0.434532$$

$$\pi_2^{[1]} = \frac{1}{3}(0.3333 + 0.576116 + 0.7869873) = 0.565468$$

Q2:

Q2.

a). Given a set of points $x_1, x_2, \ldots, x_{10} \in \mathbb{R}^3$, mean and covariance matrix:

$$\mu = \frac{1}{10} \sum_{n=1}^{10} x_n = \begin{pmatrix} 5.4 \\ 8 \\ 4.8 \end{pmatrix}$$

$$\Sigma = \frac{1}{10}\left(\sum_{n=1}^{10} (x_n - \mu)(x_n - \mu)^T\right) = \begin{pmatrix} 12.04 & 0.5 & 3.28 \\ 0.5 & 12.2 & 2.9 \\ 3.28 & 2.9 & 8.16 \end{pmatrix}$$

Then, perform orthogonal diagonalization, decompose $\Sigma = Q \Lambda Q^T$.

Then, we have:

$$Q = \begin{pmatrix} 0.616596 & 0.678179 & -0.399856 \\ 0.58815 & -0.73439 & -0.337589 \\ \underbrace{0.522591}_{v_1} & \underbrace{0.0272856}_{v_2} & \underbrace{0.852194}_{v_3} \end{pmatrix}$$

$$\Lambda = \begin{pmatrix} 15.2974 & 0 & 0 \\ 0 & 11.6305 & 0 \\ 0 & 0 & 5.47203 \end{pmatrix}$$

So, v_1, v_2, v_3 are eigenvectors of Σ, and they are principal axis.

Q2b). set W as Q^T:

$$W = \begin{pmatrix} 0.616596 & 0.58815 & 0.522196 \\ 0.678179 & -0.73439 & 0.0272856 \\ -0.399856 & -0.337589 & 0.852144 \end{pmatrix}$$

Compute principal components for each sample is just Wx_i, i from 1 to 10.

$$Wx_1 = \begin{pmatrix} 3.360684 \\ -0.7087442 \\ 1.481398 \end{pmatrix} \quad Wx_2 = \begin{pmatrix} 9.784564 \\ -3.025976 \\ -0.039416 \end{pmatrix}$$

$$Wx_3 = \begin{pmatrix} 13.610952 \\ -6.5365726 \\ 2.41866 \end{pmatrix} \quad Wx_4 = \begin{pmatrix} 7.934776 \\ -5.060513 \\ 1.160152 \end{pmatrix}$$

$$Wx_5 = \begin{pmatrix} 12.362272 \\ -1.8359938 \\ -5.021238 \end{pmatrix} \quad Wx_6 = \begin{pmatrix} 7.191368 \\ 1.8369786 \\ -3.297204 \end{pmatrix}$$

$$Wx_7 = \begin{pmatrix} 14.957928 \\ 0.4740614 \\ 1.36988 \end{pmatrix} \quad Wx_8 = \begin{pmatrix} 7.077584 \\ -3.8132974 \\ -3.048136 \end{pmatrix}$$

$$Wx_9 = \begin{pmatrix} 12.858882 \\ 3.9517326 \\ -0.973497 \end{pmatrix} \quad Wx_{10} = \begin{pmatrix} 16.293782 \\ -1.105508 \\ -1.747031 \end{pmatrix}$$

No. Date.

Q2 c). Average construction error:

Let $K = \begin{pmatrix} v_1^T \\ v_2^T \end{pmatrix} = \begin{pmatrix} 0.616596 & 0.58815 & 0.522596 \\ 0.678129 & -0.73439 & 0.0272856 \end{pmatrix}$

Error:

$$\frac{1}{10} \sum_{n=1}^{10} \| x_n \; K^T (K x_n) \|^2 = 6.0681063$$

Q3 a). $\because (AA^T)^T = (A^T)^T A^T = AA^T$

$(A^TA)^T = A^T (A^T)^T = A^TA$

So AA^T and A^TA are symmetric.

PSD: $\forall x \neq 0 \in \mathbb{R}^m$, $\forall y \neq 0 \in \mathbb{R}^n$, we have

$x^T (AA^T) x = (x^TA)(A^Tx) = (A^Tx)^T (A^Tx) = \|A^Tx\|^2 \geq 0$

$y^T (A^TA) y = (y^TA^T)(Ay) = (Ay)^T (Ay) = \|Ay\|^2 \geq 0$

So AA^T and A^TA are positive semi-definite.

Eigenvalues: Let $\lambda \neq 0$ be one of the eigenvalues of AA^T, and $v \in \mathbb{R}^m$ be the corresponding eigenvector.

we have $(AA^T - \lambda I) v = 0$

So, $(A^TA)(A^Tv) = A^T((AA^T)v) = A^T(\lambda v) = \lambda (A^Tv)$

therefore, λ is also one of the eigenvalues of A^TA, the corresponding eigenvector is A^Tv.

P.T.O.

Similarily, let $u \neq 0$ be one of the eigenvalues of A^TA. and set $k \in \mathbb{R}^n$ be the corresponding eigenvector.
$\therefore$ we have $(A^TA - \lambda I)k = 0$.
Therefore, $(AA^T)(Av) = A((A^TA)k) = A(uk) = u(Ak)$
so u is also one of the eigenvalues of AA^T, and Ak is the corresponding eigenvector.

$\Rightarrow$ From above, AA^T and A^TA have the same non-zero eigenvalue.

Q3b). We define a set of vector, namely $z_1, z_2, \dots, z_{2m} \in \mathbb{R}^m$,

$$\underset{z_1}{\begin{pmatrix} \sqrt{m} \\ 0 \\ \vdots \\ 0 \end{pmatrix}} \underset{z_2}{\begin{pmatrix} -\sqrt{m} \\ 0 \\ \vdots \\ 0 \end{pmatrix}} \underset{z_3}{\begin{pmatrix} 0 \\ \sqrt{m} \\ \vdots \\ 0 \end{pmatrix}} \underset{z_4}{\begin{pmatrix} 0 \\ -\sqrt{m} \\ \vdots \\ 0 \end{pmatrix}} \cdots \underset{z_{2m-1}}{\begin{pmatrix} 0 \\ 0 \\ \vdots \\ \sqrt{m} \end{pmatrix}} \underset{z_{2m}}{\begin{pmatrix} 0 \\ 0 \\ \vdots \\ -\sqrt{m} \end{pmatrix}}$$

then, the mean of $z_1, z_2, \dots z_{2m}$ is:

$$\frac{1}{2m}\sum_{k=1}^{2m} z_k = 0$$

$$\text{Covariance matrix} = \frac{1}{2m}\sum_{k=1}^{2m}(z_k - 0)(z_k - 0)^T = \frac{1}{2m}\sum_{k=1}^{2m} z_k z_k^T = I_m$$

then, since $\Sigma \in \mathbb{R}^{m\times m}$ is positive semi-definite, so $\exists A \in \mathbb{R}^{m\times m}$, s.t. $\Sigma = AA^T$ (e.g. we can perform Cholesky decomposition on AA^T to get Σ).

No. Date.

Let $x_k = Az_k + u$, then the mean of $x_1, x_2, \dots, x_{2m}$ is:

$$\frac{1}{2m}\sum_{k=1}^{2m}(Az_k+u) = A\left(\frac{1}{2m}\sum_{k=1}^{2m} z_k\right) + u = A\cdot 0 + u = u$$

Covariance matrix:

$$\frac{1}{2m}\sum_{k=1}^{2m}(x_k-u)(x_k-u)^T$$

$$= \frac{1}{2m}\sum_{k=1}^{2m}((Az_k+u)-u)((Az_k+u)-u)^T$$

$$= \frac{1}{2m}\sum_{k=1}^{2m}(Az_k)(Az_k)^T = \frac{1}{2m}\sum_{k=1}^{2m}(Az_k z_k^T A^T)$$

$$= A\left(\frac{1}{2m}\sum_{k=1}^{2m} z_k z_k^T\right)A^T = A\cdot I_m A^T = AA^T = \Sigma$$

(Q2c). Since Σ and $\Phi\Phi^T$ are symmetric, so we can perform orthogonal diagonalization on Σ and $\Phi\Phi^T$.

Let eigenvalues of Σ are $\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_m$, in descending order. and let eigenvalues of $\Phi\Phi^T$ as $\mu_1, \mu_2, \dots, \mu_m$, in descending order.

Let $\Phi = (v_1\ v_2\ \cdots\ v_k)$, since $\Phi^T\Phi = I_k$ (given), so $v_1, v_2, \dots, v_k$ are orthonormal vectors in $\mathbb{R}^m$.

Then, we can take a set of vectors $w_1, w_2, \dots, w_{m-k}$ in $\mathbb{R}^m$, to make $v_1, v_2, \dots, v_k, w_1, \dots, w_{m-k}$ as a set of orthonormal basis.

No. Date.

Since $(\Phi\Phi^T)v_i = \Phi(\Phi^T v_i) = \Phi \cdot e_i = v_i$,

so v_i is eigenvector of $\Phi\Phi^T$, and the corresponding eigenvalue is 1.

Since $(\Phi\Phi^T)w_i = \Phi(\Phi^T w_i) = \Phi \cdot 0 = 0 = 0 \cdot w_i$

so w_i is also eigenvector of $(\Phi\Phi^T)$, and the corresponding eigenvalue is 0.

so eigenvalue of $\Phi\Phi^T$: $\underbrace{u_1, u_2, \dots, u_k}_{=1}, \underbrace{u_{k+1}, u_{k+2}, \dots, u_m}_{=0}$

$$\begin{aligned}
\therefore \operatorname{Trace}(\Phi^T\Sigma\Phi) &= \operatorname{Trace}(\Phi^T(\Sigma\Phi)) \\
&= \operatorname{Trace}((\Sigma\Phi)\Phi^T) \\
&= \operatorname{Trace}(\Sigma(\Phi\Phi^T)) \\
&\geq \sum_{i=1}^{m} \lambda_i u_{m-i+1} \quad \text{(by Von Neumann's Trace Inequality)} \\
&= \lambda_1 u_m + \lambda_2 u_{m-1} + \dots + \lambda_{m-k} u_{k+1} + \lambda_{m-k+1} u_k + \dots + \lambda_m u_1 \\
&= \lambda_{m-k+1} + \lambda_{m-k+2} + \dots + \lambda_m
\end{aligned}$$

Now, after orthogonal diagonalization, we assume $\Sigma = Q\Lambda Q^T$.

and $Q = (\mu_1\ \mu_2 \cdots \mu_m)$, where $\mu_1, \mu_2, \dots, \mu_m$ are orthogonal and the corresponding eigenvalue of μ_i is λ_i.

We can take Φ as $(\mu_{m-k+1}\ \mu_{m-k+2} \cdots \mu_m) \in \mathbb{R}^{m\times k}$, and $\Phi^T\Phi = I_k$.

No. Date.

Now, we have:

$$\mathrm{Trace}(\Phi^T \Sigma \Phi)$$

$$= \mathrm{Trace}(\Phi^T(\Sigma\Phi))$$

$$= \mathrm{Trace}\left(\begin{pmatrix} \mu_{m-k+1}^T \\ \mu_{m-k+2}^T \\ \vdots \\ \mu_m^T \end{pmatrix} \cdot \Sigma (\mu_{m-k+1} \ \mu_{m-k+2} \ \cdots \ \mu_m)\right)$$

$$= \mathrm{Trace}\left(\begin{pmatrix} \mu_{m-k+1}^T \\ \mu_{m-k+2}^T \\ \mu_m^T \end{pmatrix} \cdot (\Sigma\mu_{m-k+1} \ \Sigma\mu_{m-k+2} \ \cdots \ \Sigma\mu_m)\right)$$

$$= \mathrm{Trace}\left(\begin{pmatrix} \mu_{m-k+1}^T \\ \mu_{m-k+2}^T \\ \vdots \\ \mu_m^T \end{pmatrix} \cdot (\lambda_{m-k+1}\mu_{m-k+1} \ \lambda_{m-k+2}\mu_{m-k+1} \cdots \lambda_m\mu_m)\right)$$

$$= \mathrm{Trace}\left(\begin{pmatrix} \lambda_{m-k+1}\|\mu_{m-k+1}\|^2 & & & \\ & \lambda_{m-k+2}\|\mu_{m-k+2}\|^2 & & \\ & & \ddots & \\ & & & \lambda_m\|\mu_m\|^2 \end{pmatrix}\right)$$

$$= \lambda_{m-k+1}\|\mu_{m-k+1}\|^2 + \lambda_{m-k+2}\|\mu_{m-k+2}\|^2 + \cdots + \lambda_m\|\mu_m\|^2$$

$$= \lambda_{m-k+1} + \lambda_{m-k+2} + \cdots + \lambda_m$$

So, when the column vector of Φ is $(\mu_{m-k+1} \ \mu_{m-k+2} \cdots \mu_m)$, $\mathrm{Trace}(\Phi^T\Sigma\Phi)$ can attain its minimum at $\lambda_{m-k+1} + \lambda_{m-k+2} + \cdots + \lambda_m$.