Solution to Homework 09

Name: Chen Shen

NetID: cs5236

1.

(a)

$$\mu = \begin{bmatrix} 1.5 & 2.5 & 3.5 \end{bmatrix}$$

(b)

$$\tilde{X} = X - \mu = \begin{bmatrix} 1.5 & -0.5 & -2.5 \\ 0.5 & 1.5 & 1.5 \\ -0.5 & -0.5 & -0.5 \\ -1.5 & -0.5 & 1.5 \end{bmatrix}$$

$$Q = \frac{1}{N}\tilde{X}^{\mathrm{T}}\tilde{X} = \begin{bmatrix} 1.25 & 0.25 & -1.25 \\ 0.25 & 0.75 & 0.75 \\ -1.25 & 0.75 & 2.75 \end{bmatrix}$$

(c) Eigenvalues are

$$\lambda_1 = 3.56166464, \lambda_2 = 1.1733803, \lambda_3 = 0.014955506$$

Eigenvectors are

$$v_1 = \begin{bmatrix} -0.45056922 \\ 0.19247228 \\ 0.87174641 \end{bmatrix}, v_2 = \begin{bmatrix} -0.66677184 \\ -0.72187235 \\ -0.18524476 \end{bmatrix}, v_3 = \begin{bmatrix} -0.59363515 \\ 0.66472154 \\ -0.45358856 \end{bmatrix}$$

(d)

$$\tilde{X} = AV^{\mathrm{T}}$$

$$A = \tilde{X}(V^{\mathrm{T}})^{-1} = \tilde{X}V = \begin{bmatrix} -2.95145599 & -0.17610969 & -0.0888421 \\ 1.37104342 & -1.69406159 & 0.0198819 \\ -0.30682473 & 0.78694448 & 0.19125108 \\ 1.8872373 & 1.0832268 & -0.12229089 \end{bmatrix}$$

(e)

$$X = \tilde{X} + \mu = Av + \mu = \begin{bmatrix} 3 & 2 & 1 \\ 2 & 4 & 5 \\ 1 & 2 & 3 \\ 0 & 2 & 5 \end{bmatrix}$$

(f) The largest two eigenvalues are

$$\lambda_1 = 3.56166464, \lambda_2 = 1.1733803$$

The reconstucted matrix should be

$$\tilde{X}_2 = \begin{bmatrix} 2.94726021 & 2.05905526 & 0.95970224 \\ 2.0118026 & 3.98678407 & 5.0090182 \\ 1.11353336 & 1.87287129 & 3.0867493 \\ -0.07259617 & 2.08128939 & 4.94453025 \end{bmatrix}$$

(g) The sum of reconstruction error squares = 0.059820245731225914
The sum of squares of skipped PCA coefficients = 0.0598202457312259
Generally, we can say they are equal to each other.

jupyter notebook used:

```
>>> import numpy as np
>>> import contextlib
>>>
>>> @contextlib.contextmanager
>>> def printoptions(*args, **kwargs):
>>>     original = np.get_printoptions()
>>>     np.set_printoptions(*args, **kwargs)
>>>     yield
>>>     np.set_printoptions(**original)
```

```
>>> X = [[3,2,1],
>>>      [2,4,5],
>>>      [1,2,3],
>>>      [0,2,5]]
>>> X = np.matrix(X)
```

```
>>> mu = X.mean(axis=0)
>>> Xt = X - mu
>>> print('tide X=')
>>> print(Xt)
```

```
tide X=
[[ 1.5 -0.5 -2.5]
 [ 0.5  1.5  1.5]
 [-0.5 -0.5 -0.5]
 [-1.5 -0.5  1.5]]
```

```
>>> Q = 1/4 * Xt.T * Xt
>>> print('Q=')
>>> print(Q)
```

```
Q=
[[ 1.25  0.25 -1.25]
 [ 0.25  0.75  0.75]
 [-1.25  0.75  2.75]]
```

```
>>> w, v = np.linalg.eig(Q)
>>> print('Eigenvalues=')
>>> print(w)
>>> print('Eigenvectors=')
>>> print(v)
```

```
Eigenvalues=
[3.56166464 1.1733803  0.01495506]
Eigenvectors=
[[-0.45056922 -0.66677184 -0.59363515]
 [ 0.19247228 -0.72187235  0.66472154]
 [ 0.87174641 -0.18524476 -0.45358856]]
```

```
>>> A = Xt * np.linalg.inv(v.T)
>>> print('PCA coefficients=')
>>> print(A)
```

```
PCA coefficients=
[[-2.95145599 -0.17610969 -0.0888421 ]
 [ 1.37104342 -1.69406159  0.0198819 ]
 [-0.30682473  0.78694448  0.19125108]
 [ 1.8872373   1.0832268  -0.12229089]]
```

```
>>> Xr = A * v.T + mu
>>> print('Reconstructed X=')
>>> with printoptions(suppress=True):
>>>     print(Xr)
```

```
Reconstructed X=
[[3. 2. 1.]
 [2. 4. 5.]
 [1. 2. 3.]
 [0. 2. 5.]]
```

```
>>> tran = [[1,0,0],
>>>         [0,1,0],
>>>         [0,0,0]]
>>> tran = np.matrix(tran)
>>> v2 = v * tran
>>> A2 = A * tran
>>> Xr2 = A2 * v2.T + mu
>>> print('Reconstructed X (with two largest eigenvalue)=')
>>> print(Xr2)
```

```
Reconstructed X (with two largest eigenvalue)=
[[ 2.94726021  2.05905526  0.95970224]
 [ 2.0118026   3.98678407  5.0090182 ]
 [ 1.11353336  1.87287129  3.0867493 ]
 [-0.07259617  2.08128939  4.94453025]]
```

```
>>> e1 = np.power(Xr2 - X, 2).sum()
>>> e2 = np.power(np.ravel(A[:,-1]), 2).sum(axis=0)
>>> print('The sum of reconstruction error squares=')
>>> print(e1)
>>> print('The sum of squares of skipped PCA coefficients=')
>>> print(e2)
```

```
The sum of reconstruction error squares=
0.059820245731225914
The sum of squares of skipped PCA coefficients=
0.0598202457312259
```

2.

Based on the defination of SVD, one matrix X can be decomposed as

$$X = USV^{\mathrm{T}}$$

So we have

$$
\begin{aligned}
X^{\mathrm{T}}X &= VSU^{\mathrm{T}}USV^{\mathrm{T}} \\
&= VS^2V^{\mathrm{T}} \\
&= VS^2V^{-1}
\end{aligned}
$$

where U is an $m \times m$ unitary matrix, S is a diagonal $m \times n$ matrix with non-negative real numbers on the diagonal, V is an $n \times n$ unitary matrix.

Besides, $X^{\mathrm{T}}X$ is symmetric and non-negative, it can be spectral decomposed as

$$X^{\mathrm{T}}X = Q\Lambda Q^{-1}$$

Since the combination is exclusive once the order of eigenvalue is determined, we can have

$$V = Q$$

$$\Lambda = S^2$$

Thus, the covariance matrix of the coefficient data is diagonal, and the variance of the coefficients are equal to the eigenvalues.

3.

Generally, it is beneficial if we kepp all the coefficients, since we can model the original data better unless overfitting. However, in a paticular case, the benefit of keeping a subset of coefficients is to reduce computing and train our model faster, which matters in a real trial. We should keep the coefficients best reflecting the features, which are the coefficients corresponding to the largest eigenvalues.

4.

The problem is that the PCA coefficients with large variances will kill those with small variances. We can fix it through scaling the principle components or the PCA coefficients to have unit variances.