Sheet 1 "PCA"

February 20, 2019

0.1 Name : Amr Ashraf Ibrahim Elzawawy

0.2 ID: 3788

---

1 Question 1 on Data Matrix

1.1 Given the Data Matrix D below, answer the following questions

ID	a1	a2	a3	a4
1	10	60	10	90
2	20	50	40	70
3	30	50	30	40
4	20	50	20	60
5	10	60	30	10

a. What is number of dimensions?
Ans : Four dimensions.

---

b. What are the types of the attributes?
Ans : Numeric Integer.

---

c. What is the distance between x1 and x3?
Ans :

$$Distance = \sqrt{\sum_{i=1}^{d}(a_i - b_i)^2} = \sqrt{\sum_{i=1}^{d}(x1_i - x3_i)^2}$$

Distance = $\sqrt{(10-30)^2 + (60-50)^2 + (10-30)^2 + (90-40)^2}$ = $10\sqrt{34}$ = 58.3 units of length

d. What is the length of x2?
Ans :

$$Length = \sqrt{\sum_{i=1}^{d} a_i^2} = \sqrt{\sum_{i=1}^{d} x2_i^2}$$

Length = $\sqrt{(20)^2 + (50)^2 + (40)^2 + (70)^2} = 10\sqrt{94}$ = 96.95 units of length

---

e. What is the cos(angle) between x2 and x4?

Ans :

$$cos\theta = \frac{a^T b}{\|a\|\|b\|} = \frac{x2^T x4}{\|x2\|\|x4\|}$$

First, calculate $x2^T x4$ =

$$\begin{pmatrix} 20 & 50 & 40 & 70 \end{pmatrix} \begin{pmatrix} 20 \\ 50 \\ 20 \\ 60 \end{pmatrix} = 20 * 20 + 50 * 50 + 40 * 20 + 70 * 60 = 7900$$

Then, calculate $\|x2\|\|x4\| = (10\sqrt{94})(10\sqrt{69})$ = 8053.57

Thus, $cos\theta = \frac{7900}{8053.57}$ = 0.98

And, Angle between x2 and x4 = 11.48 degrees

---

f. Do we need attribute scaling?

Ans :

Since the range of values of raw data varies widely, in some machine learning algorithms, objective functions will not work properly without normalization/scaling. For example, the majority of classifiers calculate the distance between two points by the Euclidean distance. If one of the attributes has a broad range of values, the distance will be governed by this particular attribute. Therefore, the range of all attribute should be normalized/scaled so that each attribute contributes approximately proportionately to the final distance. Another reason why attribute scaling is applied is that gradient descent converges much faster with attribute scaling than without it.

---

g. Compute the attribute scaled data matrix after scaling each attribute linearly between 0 and 1

Ans:

The calculation to normalize a single value for a column is: *scaled value = (value - min) / (max - min)*

New Attribute scaled data matrix is,

ID	a1	a2	a3	a4
1	0	1	0	1
2	0.5	0	1	0.75
3	1	0	0.67	0.375
4	0.5	0	0.33	0.625
5	0	1	0.67	0

---

h. Repeat parts c,d,e on the scaled data matrix in part (g)
Ans :

$$Distance = \sqrt{\sum_{i=1}^{d}(a_i - b_i)^2} = \sqrt{\sum_{i=1}^{d}(x1_i - x3_i)^2}$$

Distance = $\sqrt{(0-1)^2 + (1-0)^2 + (0-0.67)^2 + (1-0.375)^2}$ = 1.685 units of length

---

$$Length = \sqrt{\sum_{i=1}^{d} a_i^2} = \sqrt{\sum_{i=1}^{d} x2_i^2}$$

Length = $\sqrt{(0.5)^2 + (0)^2 + (1)^2 + (0.75)^2}$ = 1.3463 units of length

---

$$cos\theta = \frac{a^T b}{\|a\|\|b\|} = \frac{x2^T x4}{\|x2\|\|x4\|}$$

First, calculate $x2^T x4 = \begin{pmatrix} 0.5 & 0 & 1 & 0.75 \end{pmatrix} \begin{pmatrix} 0.5 \\ 0 \\ 0.33 \\ 0.625 \end{pmatrix} = 0.5*0.5 + 0*0 + 1*0.33 + 0.75*0.625 = 1.04875$
Then, calculate $\|x2\|\|x4\|$ = (1.3463)(0.86575) = 1.1656
Thus, $cos\theta = \frac{1.04875}{1.1656}$ = 0.89975
And, Angle between x2 and x4 = 25.87 degrees

---

2 Question 2 on Data matrix

2.1 Given the Data Matrix D above submit your python code and its output that will do the following

Import libraries needed

```
In [2]: import numpy as np
```

Store Matrix D

```
In [3]: matD = np.array([[10,60,10,90],[20,50,40,70],[30,50,30,40],[20,50,20,60],[10,60,30,10]]
```

a. Compute the norm of each instance. (5x1)

```
In [19]: np.sum(np.abs(matD)**2,axis=-1)**(1./2)
```

```
Out[19]: array([109.08712115,  96.95359715,  76.81145748,  83.06623863,
                 68.556546  ])
```

b. Compute the Cosine similarity matrix (5x5) matrix
Import needed libraries

```
In [5]: from sklearn.metrics.pairwise import cosine_similarity

In [6]: cosine_similarity(matD)

Out[6]: array([[1.        , 0.93604728, 0.85927665, 0.97114627, 0.65519967],
               [0.93604728, 1.        , 0.95338522, 0.98093136, 0.76728608],
               [0.85927665, 0.95338522, 1.        , 0.95604709, 0.8735402 ],
               [0.97114627, 0.98093136, 0.95604709, 1.        , 0.77264358],
               [0.65519967, 0.76728608, 0.8735402 , 0.77264358, 1.        ]])
```

c. Compute the Euclidean Distance matrix of the instances (5x5)
Import needed libraries

```
In [7]: from sklearn.metrics.pairwise import euclidean_distances

In [20]: euclidean_distances(matD)

Out[20]: array([[ 0.        , 38.72983346, 58.30951895, 34.64101615, 82.46211251],
                [38.72983346,  0.        , 33.1662479 , 22.36067977, 62.44997998],
                [58.30951895, 33.1662479 ,  0.        , 24.49489743, 37.41657387],
                [34.64101615, 22.36067977, 24.49489743,  0.        , 52.91502622],
                [82.46211251, 62.44997998, 37.41657387, 52.91502622,  0.        ]])
```

3 Question 3 on Principal Component Analysis

3.1 Given Data matrix D above. Consider a1, a2 and a4 only

a. Write down the new data matrix D3 (5x3)

ID	a1	a2	a4
1	10	60	90
2	20	50	70
3	30	50	40
4	20	50	60
5	10	60	10

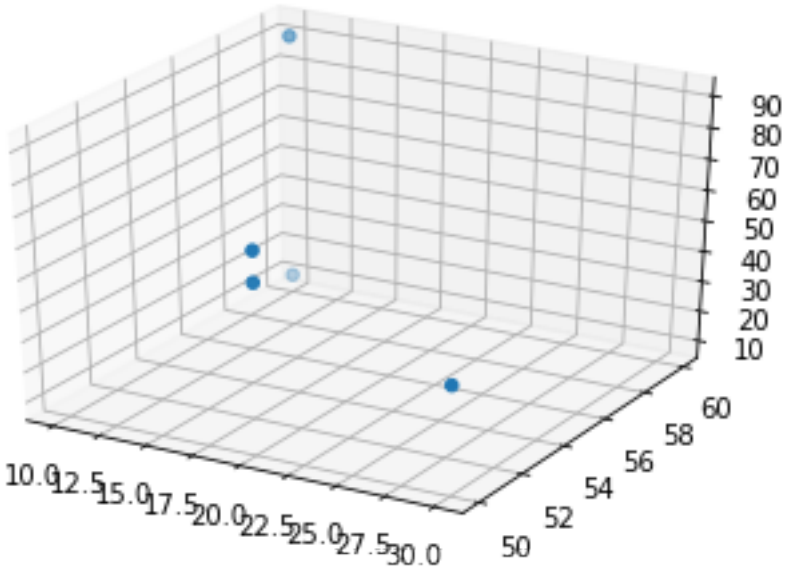
b. Plot the data using 3d scatter plots
Import need libraries

```
In [21]: from mpl_toolkits.mplot3d import Axes3D
         import matplotlib.pyplot as plt

In [22]: ax = plt.axes(projection='3d')
```

```
matD3 = np.array([[10,60,90],[20,50,70],[30,50,40],[20,50,60],[10,60,10]])
dim1 = matD3[:,0]
dim2 = matD3[:,1]
dim3 = matD3[:,2]
ax.scatter3D(dim1,dim2,dim3)
```

Out[22]: <mpl_toolkits.mplot3d.art3d.Path3DCollection at 0x7fa1dfa59a58>



c. Compute the mean vector (3x1)

Ans :

We compute the mean vector by calculating the mean of each attribute along all instances.

Mean of a1 = $\frac{(10+20+30+20+10)}{5} = 18$

Mean of a2 = $\frac{(60+50+50+50+60)}{5} = 54$

Mean of a4 = $\frac{(90+70+40+60+10)}{5} = 54$

Mean vector is $\begin{pmatrix} 18 & 54 & 54 \end{pmatrix}$

It can be also computed using np.mean(matD3, axis=0)

---

d. Compute centered data matrix Z by subtracting mean vector from the Data Matrix. (5x3)

Ans:

Simply subtract mean vector from each row in D3 matrix.

$$Z = \begin{pmatrix} -8 & 6 & 36 \\ 2 & -4 & 16 \\ 12 & -4 & -14 \\ 2 & -4 & 6 \\ -8 & 6 & -44 \end{pmatrix}$$

e. Compute Covariance matrix COV (3x3)

```
In [133]: matZ = np.array([[-8,6,36],[2,-4,16],[12,-4,-14],[2,-4,6],[-8,6,-44]])
          covMat = np.cov(matZ,rowvar=False, bias=True)
          covMat

Out[133]: array([[ 56., -32., -12.],
                 [-32.,  24., -16.],
                 [-12., -16., 744.]])
```

f. Use python solvers to find eigenvalues (Diagonal 3x3 matrix) and eigen vectors (3x3) matrix. Take care of the eigenvalues order.

Ans :

After we computed the covariance matrix. We apply the python solver to get eigenvalues and vectors.

```
In [134]: (eigVal,eigVect) = np.linalg.eig(covMat)
          print("Eigen Values Array : ")
          print(eigVal)
          #eigVal is an array, we need to convert it into diagonal 3*3 matrix
          eigValMat = np.diag(eigVal)
          print("Eigen Vectors Matrix : ")
          print(eigVect)
          print("Eigen Values Matrix : ")
          print(eigValMat)

Eigen Values Array :
[ 75.77229965   3.68692565 744.5407747 ]
Eigen Vectors Matrix :
[[ 0.85025739 -0.52611085 -0.01642407]
 [-0.5263604  -0.8499905  -0.02146802]
 [ 0.00266574 -0.02689832  0.99963462]]
Eigen Values Matrix :
[[ 75.77229965   0.           0.        ]
 [  0.           3.68692565   0.        ]
 [  0.           0.         744.5407747 ]]
```

g. Verify $U^T \Lambda U = COV$.

```
In [135]: eigVect.T @ eigValMat  @ eigVect

Out[135]: array([[ 55.80541937, -32.29904625,   0.96755382],
                 [-32.29904625,  24.17564786, -19.29756198],
                 [  0.96755382, -19.29756198, 744.01893277]])
```

h. Compute the explained variance by the eigenvector corresponding to the largest eigenvalue. Do you think one eigenvector is good enough?

Ans :

The eigenvector required is

$$X = \begin{pmatrix} -0.01642407 \\ -0.02146802 \\ 0.99963462 \end{pmatrix}$$

To calculate the explained variance we compute,

$$\frac{\lambda_x}{\sum_{i=1}^{d} \lambda_i} = \frac{744.5407747}{744.5407747 + 3.68692565 + 75.77229965} = 0.90$$

One Eigenvector is not enough, as using only one eigenvector will give much error in transformation of axes.Instead Using the largest 2 eigen vectors will yeild 99% and this is much much less error.

---

i. Compute the projection matrix P to go to 2-dimensions. Consider the top two eigenvectors of matrix U according to eigenvalues.(3x2)

Ans : Considering top 2 eigenvectors of matrix U*UˆT only produces P as follows.

$$P = \begin{pmatrix} 0.85025739 & -0.01642407 \\ -0.5263604 & -0.02146802 \\ 0.00266574 & 0.99963462 \end{pmatrix} \begin{pmatrix} 0.85025739 & -0.5263604 & 0.00266574 \\ -0.01642407 & -0.02146802 & 0.99963462 \end{pmatrix}$$

---

j. Project the instances into a 2-Dimension space. $x = P^T x$

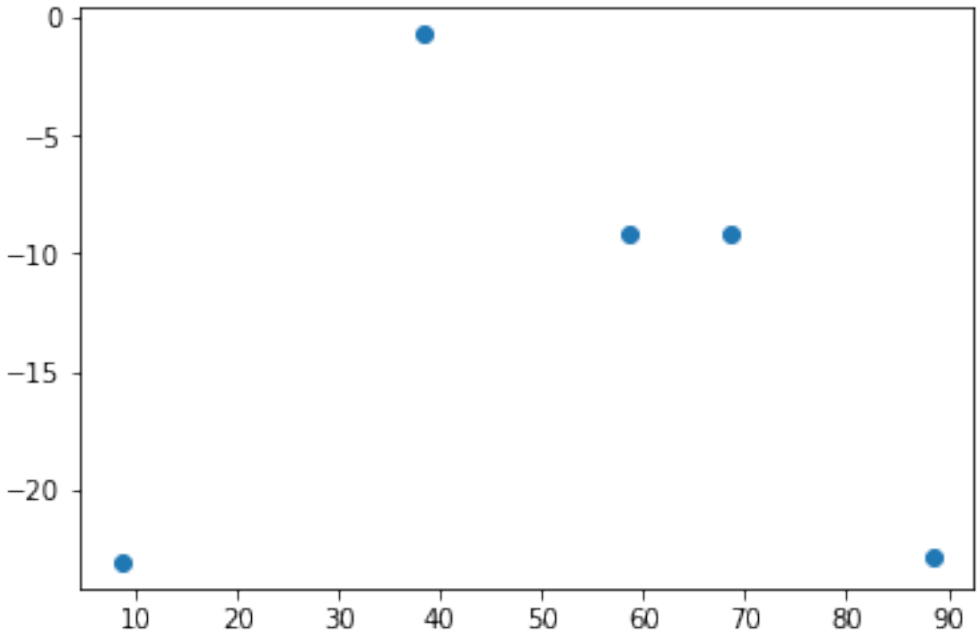
Ans : First define P matrix in python then we get x

```
In [154]: matU = np.array([[-0.01642407,0.85025739],[-0.02146802,-0.5263604],[0.99963462,0.002
          matP = matU @ matU.T
          x1 = matD3[0,:]
          x2 = matD3[1,:]
          x3 = matD3[2,:]
          x4 = matD3[3,:]
          x5 = matD3[4,:]
          new_x1 = matP @ x1   #instace 1 projected.
          new_x2 = matP @ x2   #instace 2 projected.
          new_x3 = matP @ x3   #instace 3 projected.
          new_x4 = matP @ x4   #instace 4 projected.
          new_x5 = matP @ x5   #instace 5 projected.
          matProjection = np.array([new_x1,new_x2,new_x3,new_x4,new_x5])
```

```
In [153]: A_x1 = matU.T @ x1
          A_x2 = matU.T @ x2
          A_x3 = matU.T @ x3
          A_x4 = matU.T @ x4
          A_x5 = matU.T @ x5
```

```
matD2 = np.array([A_x1,A_x2,A_x3,A_x4,A_x5])
ax = plt.axes()
ax.scatter(matD2[:,0],matD2[:,1])
```

```
Out[153]: <matplotlib.collections.PathCollection at 0x7fa1deecbc18>
```



4 Question 4 : Midterm1 Question Fall 2017

4.1 Given the data below , answer the following questions

A. Compute 3x3 Covariance matrix of the 5 tuples dataset we have.

Ans:

First we store the original given matrix

$$X = \begin{pmatrix} 0.5 & 4.5 & 2.5 \\ 2.2 & 1.5 & 0.1 \\ 3.9 & 3.5 & 1.1 \\ 2.1 & 1.9 & 4.9 \\ 0.5 & 3.2 & 1.2 \end{pmatrix}$$

```
In [155]: matX = np.array([[0.5,4.5,2.5],[2.2,1.5,0.1],[3.9 ,3.5 ,1.1], [2.1,1.9,4.9],[0.5,3.2
          meanVect = np.mean(matX, axis=0)
          matZ = matX - meanVect
```

```
In [156]: np.cov(matZ,rowvar=False,bias=True)
```

```
Out[156]: array([[ 1.6064, -0.4148, -0.2764],
                 [-0.4148,  1.1936, -0.0432],
                 [-0.2764, -0.0432,  2.7424]])
```

B. The trace of the covariance matrix is the sum of the eigenvalues of the matrix.
1. Compute the three eigenvalues of the covariance matrix if

$$\frac{\lambda_a}{\lambda_c} = 0.505$$

and

$$\frac{\lambda_b}{\lambda_c} = 0.647$$

Ans:
Following the rule above.

$$Traceof Covariancematrix = (1.6964 + 1.1936 + 2.7424) = 5.6324$$

$$\lambda_a + \lambda_b + \lambda_c = 5.6324$$

By Substituting with given relations

$$0.505\lambda_c + 0.647\lambda_c + \lambda_c = 5.6324$$

$$\lambda_c = 3.4804$$

and,

$$\lambda_a = 1.757602$$

and,

$$\lambda_b = 2.518188$$

2.Determine the explained variance using only using λ_b, λ_c
Ans:
To calculate the explained variance we compute,

$$\frac{\lambda_c + \lambda_b}{\sum_{i=1}^{d} \lambda_i} = \frac{3.4804 + 2.518188}{3.4804 + 2.518188 + 1.757602} = 0.77$$
