

CSCI 347
Homework 04

Show your work. Include any code snippets you used to generate an answer, using comments in the code to clearly indicate which problem corresponds to which code

1. (2 points) Consider matrix A and vector v . Compute the matrix-vector product Av .

$$A = \begin{pmatrix} 2 & 1 \\ 1 & 3 \end{pmatrix}, v = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

$$Av = \begin{pmatrix} 2 * (-1) + 1(1) \\ 1 * (-1) + 3(1) \end{pmatrix} Av = \begin{pmatrix} -1 \\ 2 \end{pmatrix}$$

2. Consider matrix A and data set D :

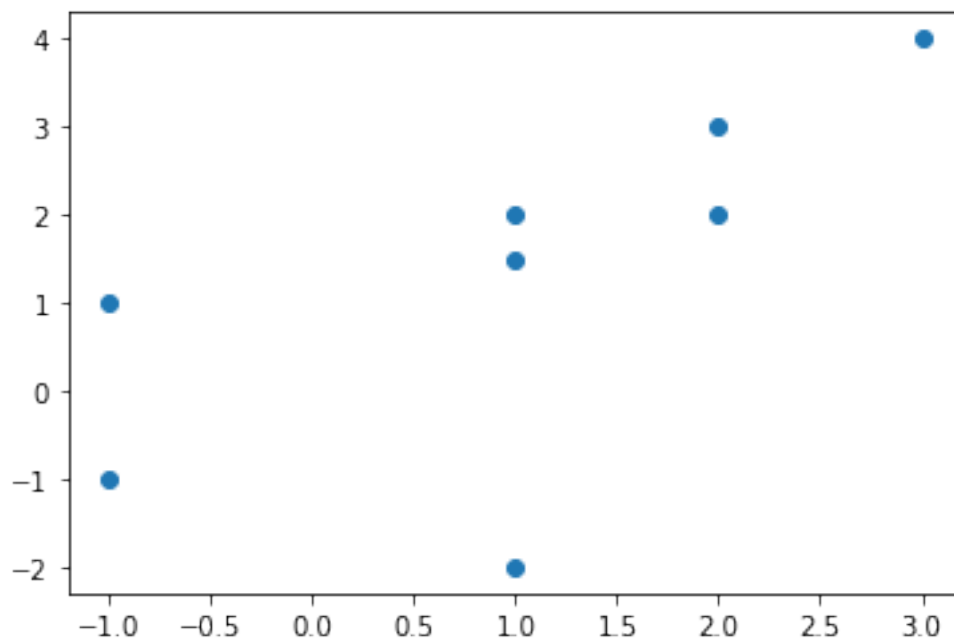
$$A = \begin{pmatrix} \frac{\sqrt{3}}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{\sqrt{3}}{2} \end{pmatrix}, D = \begin{pmatrix} 1 & 1.5 \\ 1 & 2 \\ 3 & 4 \\ -1 & -1 \\ -1 & 1 \\ 1 & -2 \\ 2 & 2 \\ 2 & 3 \end{pmatrix}$$

- (a) (2 points) Let X_1 and X_2 be the first and second attributes of the data, respectively. Use Python to create a scatter plot of the data, where the x -axis is X_1 and the y -axis is X_2 .

```
import matplotlib.pyplot as plt
```

```
data = [(1, 1.5),  
        (1, 2),  
        (3, 4),  
        (-1, -1),  
        (-1, 1),  
        (1, -2),  
        (2, 2),  
        (2, 3)]
```

```
df = pd.DataFrame(data)  
plt.scatter(df[0], df[1])  
plt.show()
```



- (b) (4 points) Treating each row as a 2-dimensional vector, apply the linear transformation A to each row. In other words, let x_i be the i -th row of D . For each x_i , find the matrix-vector product Ax_i . For example, $x_2 = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$.

```
D = np.array([(1, 1.5),
              (1, 2),
              (3, 4),
              (-1, -1),
              (-1, 1),
              (1, -2),
              (2, 2),
              (2, 3)])
```

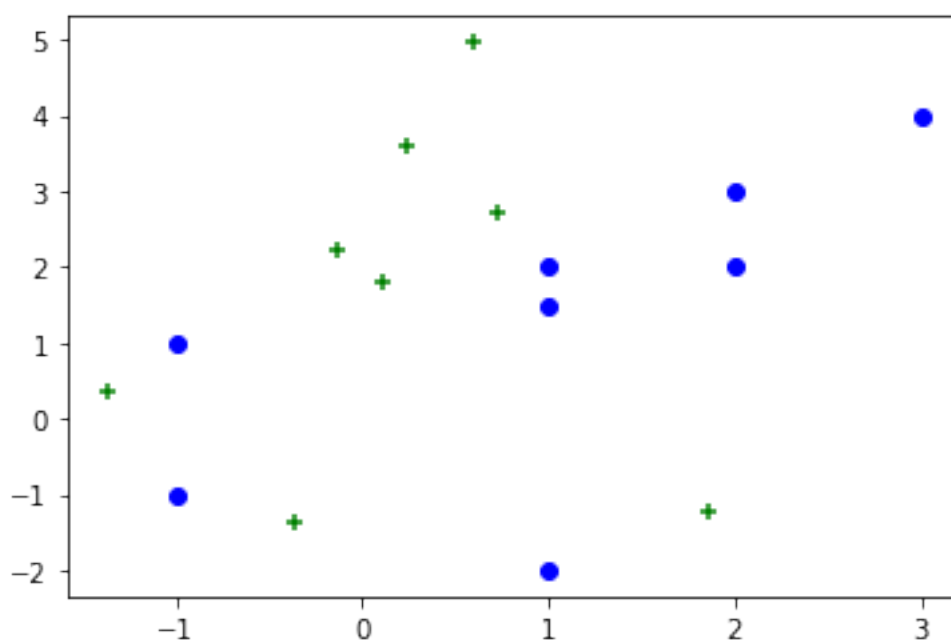
```
A = np.array([(math.sqrt(3)/2), (-1/2)],
              [(1/2), (math.sqrt(3)/2)])
```

```
linear_transformed_data = A.dot(D.T)
```

$$transformed = \begin{pmatrix} 0.1160254 & 1.79903811 \\ -0.1339746 & 2.23205081 \\ 0.59807621 & 4.96410162 \\ -0.3660254 & -1.3660254 \\ -1.3660254 & 0.3660254 \\ 1.8660254 & -1.23205081 \\ 0.73205081 & 2.73205081 \\ 0.23205081 & 3.59807621 \end{pmatrix}$$

- (c) (3 points) Use Python to create a plot showing both the original data and the transformed data, with the x -axis still corresponding to X_1 and the y -axis corresponding to X_2 . Use different colors and markers to differentiate between the original and transformed data. That is, each transformed data point in the plot should be one matrix-vector product Ax_i , which is a 2-dimensional vector. Each original point in the plot should have the same coordinates as it did in part 2.1.

```
fig = plt.figure()
ax = fig.add_subplot(1, 1, 1)
ax.scatter(D[:, 0], D[:, 1], color = 'blue')
ax.scatter(L.T[:, 0], L.T[:, 1], color = 'green', marker = '+')
plt.show()
```



- (d) (1 point) Write down the multi-dimensional mean of the data. (Remember that this should be a 2-dimensional vector)

```
mm = np.mean(D, axis = 0)
mmL1 = L.T[:, 0].sum() / 8
mmL2 = L.T[:, 1].sum() / 8
```

$$multiMean = \begin{pmatrix} 1 & 1.3125 \\ 0.2098 & 1.6367 \end{pmatrix}$$

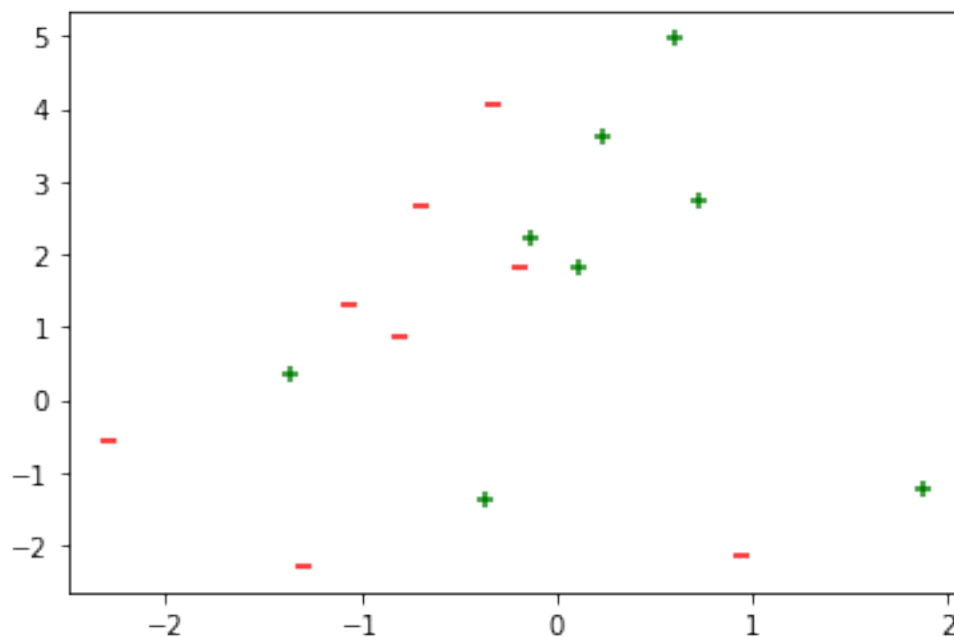
- (e) (2 points) Mean-center the data. Write down the mean-centered data matrix.

```
centering = lambda x: x- x.mean()
Z = centering(L)
print(Z)
print(Z.mean()) #extremely close to 0
```

```
[[ -0.80719147 -1.05719147 -0.32514066 -1.28924228 -2.28924228  0.94280853
  -0.19116607 -0.69116607]
 [ 0.87582123  1.30883393  4.04088474 -2.28924228 -0.55719147 -2.15526768
  1.80883393  2.67485934]]
-1.6653345369377348e-16
```

- (f) (2 points) Use Python to create a scatter plot showing both the original data and the mean-centered data, where the x -axis is X_1 and the y -axis is X_2 . Use different colors and markers to differentiate between the original and mean-centered data.

```
fig = plt.figure()
ax = fig.add_subplot(1, 1, 1)
ax.scatter(Z.T[:, 0], Z.T[:, 1], color = 'red', marker = '_')
ax.scatter(L.T[:, 0], L.T[:, 1], color = 'green', marker = '+')
plt.show()
```



- (g) (3 points) Write down the covariance matrix of the data matrix D . Use estimated covariance.

```
print(np.cov(D))
```

```
[[ 0.125  0.25  0.25  0.   0.5  -0.75  0.   0.25 ]
 [ 0.25  0.5  0.5  0.   1.   -1.5  0.   0.5 ]
 [ 0.25  0.5  0.5  0.   1.   -1.5  0.   0.5 ]
 [ 0.    0.    0.    0.    0.    0.    0.    0. ]
 [ 0.5   1.   1.   0.   2.   -3.   0.   1. ]
 [-0.75 -1.5  -1.5  0.  -3.   4.5  0.  -1.5 ]
 [ 0.    0.    0.    0.    0.    0.    0.    0. ]
 [ 0.25  0.5  0.5  0.   1.   -1.5  0.   0.5 ]]
```

- (h) (3 points) Write down the covariance matrix of the centered data matrix Z . Use estimated covariance.

```
print(np.cov(Z))
```

$$\text{cov}(Z) = \begin{pmatrix} 0.87269389 & 0.0954086 \\ 0.0954086 & 5.05141325 \end{pmatrix}$$

- (i) (3 points) Write down the covariance matrix of the data after applying standard normalization.

```
Z_normalized = MinMaxScaler().fit_transform(Z)
print(np.cov(Z_normalized))
```

$$\text{cov}(Z_{normalized}) = \begin{pmatrix} 0.21428571 & -0.21428571 \\ -0.21428571 & 0.21428571 \end{pmatrix}$$

Acknowledgements: Homework problems adapted from assignments of Veronika Strnadova-Neeley.