APM120, quiz #2, 2021, **solutions**
Applied Linear Algebra and Big Data
Last updated: Sunday 11th April, 2021, 11:49

your name: _____, HUID: _____

Read these instructions carefully: Please solve all problems, **deriving, calculating and showing explicitly all stages of your solution and explaining each step of each question**. The number of points for each question is noted below, the total number of points is 110 and the final score is min(100, your points). Use a non-programmable calculator to convert your answers to decimal number format, carrying out calculations to **three significant digits**. Please box your final answers. Limit your essay responses to no more than the specified number of words, longer responses will be truncated.

Total time, including solving/ scanning via genius scan or a similar app/ uploading/tagging each question on gradescope (as in homework): **3 hours**, strictly enforced. Gradescope will terminate the session and submit uploaded material after 3 hours.

Enjoy!

1. (34 pts) Consider the linear set of equations $\mathsf{A}\mathbf{x} = \mathbf{b}$, and the SVD: $\mathsf{A} = \mathsf{U}\Sigma\mathsf{V}^T$,

```
A=[    1,     1;
      -2,    -2;
       3,     3];
b=[    2;
       3;
       4];
U=[-0.267, -0.956,  0.12;
    0.535, -0.0439, 0.844;
   -0.802,  0.289,  0.523];
Sigma=[ 5.29,     0;
           0,     0;
           0,     0];
V=[-0.707, -0.707;
   -0.707,  0.707];
```

(a) 50 words: (i) In general, if there are more equations than unknowns ($N < M$) and the rank of the matrix is equal to the number of unknowns ($r = N < M$), do you expect a solution to exist? Why? How can one define what a useful and unique solution is in such a case? (ii) How is that solution calculated? (Write the equation to be solved and explain briefly, in words only, how it was derived)

(b) Calculate $\mathsf{A}^T\mathsf{A}$ and its rank. Explain why the standard over-determined solution (for $r = N < M$) cannot be used in this case.

(c) Find two different solutions for $\mathbf{x}$ that have the smallest possible norm of the residual vector $\mathbf{r} = \mathsf{A}\mathbf{x} - \mathbf{b}$, and calculate that residual explicitly for these two solutions. *Numerical check:* $r_3 = -2.29$.

(d) What assumptions does one need to make in order to find a unique solution for the specific problem $\mathsf{A}\mathbf{x} = \mathbf{b}$ given above? How is it calculated? Solve for the optimal solution $\mathbf{x}$ in this specific case. *Numerical check:* $x_2 = 0.286$.

Solution:

(a) **(9)** (i) The equations are likely contradictory, so there is no solution. Can look for the solution that minimizes the norm of the residuals. (ii) the solution solves $\mathsf{A}^T\mathsf{A}\mathbf{x} = \mathsf{A}^T\mathbf{b}$; this was derived using least squares, minimizing the norm of the residuals, $\mathbf{r}^T\mathbf{r}$.

(b) **(9)** $\mathsf{A}^T\mathsf{A}$ and its rank:

```
ATA=[   14,    14;
        14,    14];
% it's rank is clearly 1.
% also:
```

```
ATb=[    8;
         8];
```

The standard over-determined solution cannot be used bec A^TA cannot be inverted.

(c) **(8)** Two different solutions that have smallest possible norm

```
x1=[ 0.571;
        0];
x2=[    0;
     0.571];
r1=[-1.43;
    -4.14;
    -2.29];
r2=[-1.43;
    -4.14;
    -2.29];
```

(d) **(8)** Need to look for a solution that both minimizes the residuals and has the smallest possible norm. Solution is obtained from $\mathsf{A}^T\mathsf{A}\mathbf{x} = \mathsf{A}^T\mathbf{b}$ using SVD and the pseudo inverse as $\mathbf{x} = (\mathsf{A}^T\mathsf{A})^\dagger\mathbf{b} = \mathsf{A}^\dagger\mathbf{b}$. In this case,

```
A_pinv=[ 0.0357, -0.0714,  0.107;
         0.0357, -0.0714,  0.107];
x=[ 0.286;
    0.286];
r=[-1.43;
   -4.14;
   -2.29];
```

2. (33 pts) Consider the data set X representing normalized monthly variations in the numbers of individuals infected by COVID (first line) and flu (second line) over $N = 5$ months, and the data set Y representing the variations in the number of people wearing masks when leaving their home (first line) and the those working from home (second line). Let $\mathsf{C} = \mathsf{X}\mathsf{Y}^T/N$, where,

```
X=[   1.68,   -0.32,   -2.42,   -0.42,    1.48;
      1.44,   -0.56,   -2.36,   -0.36,    1.84];
Y=[  -1.44,    0.16,    2.76,   -0.04,   -1.44;
      1.76,   -0.64,   -2.04,   -0.84,    1.76];
C=[ -2.253,   ??;
        ??,   2.25];
U=[-0.7033, -0.7109;
   -0.7109,  0.7033];
Sigma=[  4.488,       0;
             0,  0.01454];
V=[ 0.7114,    0.7028;
   -0.7028,    0.7114];
```

(a) **(i)** Calculate the missing elements of the covariance matrix C using the above information. (*Suggestion:* calculate also elements that are already given to verify that your calculation is correct). **(ii)** 30 words: Interpret *all elements* of the covariance matrix, noting and explaining their signs and magnitude. **(iii)** 50 words: What do the U SVD vectors represent in Maximum Covariance Analysis in general? The V vectors? What can the singular values be used for in a general MCA analysis and how (give the formula)? **(iv)** 30 words: Explain why multivariate PCA may sometimes fail to extract the correct relation between two data sets. Why can MCA help? What information is lost in MCA?

(b) 100 words for all items combined: **(i)** Calculate the explained fraction of the total covariance for each mode. **(ii)** which U and V vectors matter here? Analyze the structure of the relevant singular U vectors, interpret them, and discuss their importance for this particular problem, discussing explicitly the interpretation in terms of what each of the X and Y variables represents. **(iii)** Interpret the corresponding V vectors in view of your interpretation of the U vectors. *Note:* in (ii and iii) here, do not just indicate that there is a positive/negative correlation between this or that, but rather tell a brief(!) story of what this means, and speculate why this may be the case.

Solution:

(a) **(i) (3)** the covariance matrix and its SVD,

```
%                mask     home
C= covid    [ -2.253,    2.211;
```

```
   flu       -2.262,   2.25];
U=[-0.7033, -0.7109;
   -0.7109,  0.7033];
Sigma=[  4.488,       0;
             0,  0.01454];
V=[ 0.7114,  0.7028;
   -0.7028,  0.7114];
```

(ii) (3) Interpret the covariance matrix: first column means that more mask wearing means less COVID (C_{11}) and less flu (C_{21}); second column shows that more home work means more COVID (C_{12}) and more flu (C_{22}). All magnitudes are comparable, so all these covariance elements seem equally important.

(iii) (3) U vectors represent the structure of X that is most correlated with the structure in Y that is shown by the V vectors. Singular values tell us what fraction of the total covariance is explained by the i SVD U-V pair, as given by $\sigma_i^2 / \sum_j \sigma_j^2$.

(iv) (3) Multivariate PCA may sometimes fail for example if the co-variability is smaller than the variability of each mode, and it is not orthogonal to the independent variability. MCA helps by extracting only the co-variability. It cannot be used to tell us anything about the independent variability of X and Y.

(b) **(i) (5)** explained fraction of the total covariance

```
total_covariance=[  20.14];
fraction_covariance=[      1;
                       1.05e-05];
```

(ii) (8) Given the fraction covariance, only the first U vector and first V vector matter. $\mathbf{u}_1$ shows that COVID and flu respond very similarly to mask wearing and home-staying.

(iii) (8) $\mathbf{v}_1$ shows that mask wearing has the opposite effect from home staying! This suggests that people who wear masks outside get infected at home.

3. (33 pts) Consider the data set X below, composed of $N = 8$ two-dimensional vectors.

   (a) Plot the data, label each data point by its index number (1 to N).

   (b) Cluster the data into two clusters ($k = 2$) and then three clusters ($k = 3$), by simply examining your data scatter plot.

   (c) Go explicitly through the first few steps of using k-means to cluster the data in X for $k = 3$: **(i)** Set the first representative point to be data vector number 1 and calculate the two additional ones needed for $k = 3$. **(ii)** Go over the first three non-representative data points only, adjusting the clusteroids at every iteration. When assigning data vectors to a cluster, you may estimate which clusteroid is nearest without an explicit distance calculation, when possible.

   (d) Plot an elbow plot for $k = 1, 2, 3$ based on the maximum cluster diameter (use the `distances` matrix given below to calculate the diameters). What is the appropriate number of clusters based on this plot?

   (e) 30 words: What is the cost of k-means clustering? Very briefly outline the explanation for this estimate. Why is hierarchical clustering more expensive?

   (f) How is the data covariance matrix calculated? Interpret the off-diagonal elements of the data covariance matrix C given below. Use the inverse covariance C^{-1} given below to calculate the missing four Mahalanobis distances below (`distances_M`) between all N data vectors.

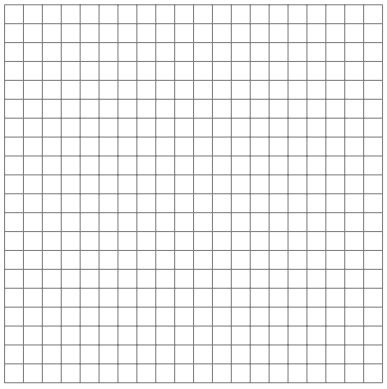
   (g) (i) Calculate the cluster diameters for $k = 1, 2, 3$ based on your already calculated clustering but using the Mahalanobis distance. (ii) Re-plot the elbow plot using the Mahalanobis distances, and discuss the effects of using these distance on the number of clusters based on the elbow plot. Base your discussion on the data structure seen in your scatter plot of the data points, and on its effect on the Mahalanobis distance.

```
X=[  2.5,   9.5,     9,     5,     8,     6,   8.5,   5.5;
     2.5,   9.5,   8.5,   5.5,     8,     6,     9,     5];
C=[ 5.06,     5;
       5,  5.06];
Cinv=[ 8.05, -7.95;
      -7.95,  8.05];
distances=[    0,   9.9,  8.85,  3.91,  7.78,  4.95,  8.85,  3.91;
               9.9,     0,  1.12,  6.02,  2.12,  4.95,  1.12,  6.02;
               8.85,  1.12,     0,     5,  1.12,  3.91,  0.707,  4.95;
               3.91,  6.02,     5,     0,  3.91,  1.12,  4.95,  0.707;
               7.78,  2.12,  1.12,  3.91,     0,  2.83,  1.12,  3.91;
               4.95,  4.95,  3.91,  1.12,  2.83,     0,  3.91,  1.12;
               8.85,  1.12,  0.707,  4.95,  1.12,  3.91,     0,     5;
               3.91,  6.02,  4.95,  0.707,  3.91,  1.12,     5,     0];
```

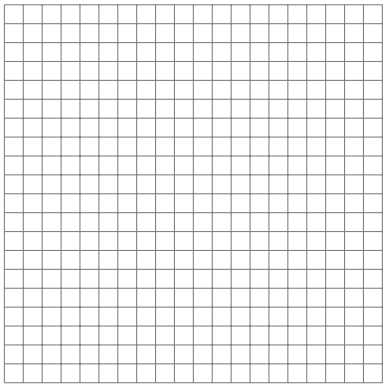
```
distances_M=[  0,     3.12,  3.12,  1.87,  2.45,  1.56,  3.12,  1.87;
               3.12,     0,  1.45,  2.36,  0.67,  1.56,  1.45,  2.36;
               3.12,  1.45,     0,  3.23,  1.45,  1.87,  ????,  1.56;
               1.87,  2.36,  3.23,     0,  1.87,  1.45,  1.56,  ????;
               2.45,  0.67,  1.45,  1.87,     0,  0.89,  1.45,  1.87;
               1.56,  1.56,  1.87,  1.45,  0.89,     0,  1.87,  1.45;
               3.12,  1.45,  ????,  1.56,  1.45,  1.87,     0,  3.23;
               1.87,  2.36,  1.56,  ????,  1.87,  1.45,  3.23,     0];
```

Use labeled axes similar to these:

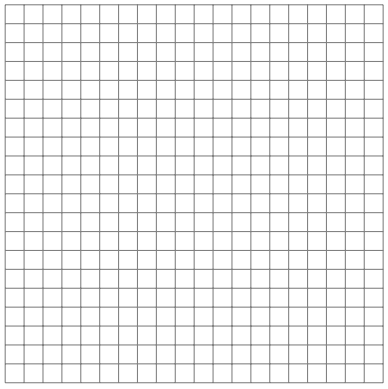
Data & clustering



elbow plot 1

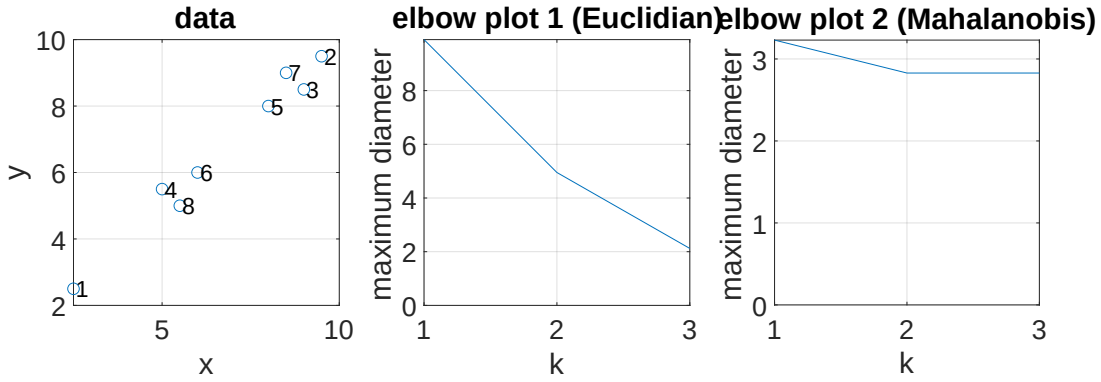


elbow plot 2



Solution:

(a) **(3)** Plot data, label each data point by its index number (1 to N).



(b) **(3)** cluster the data by inspection:
$k = 2$: (2,3,5,7); (1,4,6,8)
$k = 3$: 1; (2,3,5,7); (4,6,8)

(c) **(7)** k-means Clustering: should be done by hand for only 3 points for $k = 3$, here is a Matlab solution for all points and all k, emulating a hand-calculation.

Finding initial representatives: for $k = 2$, the second initial representative point is clearly data vector 2. When $k = 3$, the third one is number 6. Now need to go over the remaining points, and adjust the clusteroid every time.

Here is the detailed k-means clustering solution, step by step, the part required in exam is in blue.

```
==========================================================================
k=1:
data:
X=[  2.50 ,   9.50 ,   9.00 ,   5.00 ,   8.00 ,   6.00 ,   8.50 ,   5.50 ;
     2.50 ,   9.50 ,   8.50 ,   5.50 ,   8.00 ,   6.00 ,   9.00 ,   5.00 ];
distances:
         0    9.8995    8.8459    3.9051    7.7782    4.9497    8.8459    3.9051
    9.8995         0    1.1180    6.0208    2.1213    4.9497    1.1180    6.0208
    8.8459    1.1180         0    5.0000    1.1180    3.9051    0.7071    4.9497
    3.9051    6.0208    5.0000         0    3.9051    1.1180    4.9497    0.7071
    7.7782    2.1213    1.1180    3.9051         0    2.8284    1.1180    3.9051
    4.9497    4.9497    3.9051    1.1180    2.8284         0    3.9051    1.1180
    8.8459    1.1180    0.7071    4.9497    1.1180    3.9051         0    5.0000
    3.9051    6.0208    4.9497    0.7071    3.9051    1.1180    5.0000         0

 -------------- step 1: find clusteroids
specified k=1, specified point 1 to be the first representative.
initial centroids are:1
clusteroid_coordinates=[   2.5 ;
                           2.5 ];

 -------------- step 2: assign points
distances to representative points and cluster assignment:

data point 2, distance to clusteroids: 9.89949
nearest clusteroid is #1
cluster members now:
cluster 1:  1 2
updated clusteroids:
clusteroid_coordinates=[     6 ;
                             6 ];
data point 3, distance to clusteroids: 3.90512
nearest clusteroid is #1
cluster members now:
cluster 1:  1 2 3
updated clusteroids:
clusteroid_coordinates=[     7 ;
                          6.83 ];
data point 4, distance to clusteroids: 2.4037
nearest clusteroid is #1
cluster members now:
cluster 1:  1 2 3 4
```

```
updated clusteroids:
clusteroid_coordinates=[   6.5 ;
                           6.5 ];
data point 5, distance to clusteroids: 2.12132
nearest clusteroid is #1
cluster members now:
cluster 1:  1 2 3 4 5
updated clusteroids:
clusteroid_coordinates=[   6.8 ;
                           6.8 ];
data point 6, distance to clusteroids: 1.13137
nearest clusteroid is #1
cluster members now:
cluster 1:  1 2 3 4 5 6
updated clusteroids:
clusteroid_coordinates=[  6.67 ;
                          6.67 ];
data point 7, distance to clusteroids: 2.96742
nearest clusteroid is #1
cluster members now:
cluster 1:  1 2 3 4 5 6 7
updated clusteroids:
clusteroid_coordinates=[  6.93 ;
                             7 ];
data point 8, distance to clusteroids: 2.45781
nearest clusteroid is #1
cluster members now:
cluster 1:  1 2 3 4 5 6 7 8
updated clusteroids:
clusteroid_coordinates=[  6.75 ;
                          6.75 ];
k=1, max diameter=9.89949
======================================================================
k=2:
data:
X=[  2.50 ,   9.50 ,   9.00 ,   5.00 ,   8.00 ,   6.00 ,   8.50 ,   5.50 ;
     2.50 ,   9.50 ,   8.50 ,   5.50 ,   8.00 ,   6.00 ,   9.00 ,   5.00 ];
distances:
         0    9.8995    8.8459    3.9051    7.7782    4.9497    8.8459    3.9051
    9.8995         0    1.1180    6.0208    2.1213    4.9497    1.1180    6.0208
    8.8459    1.1180         0    5.0000    1.1180    3.9051    0.7071    4.9497
    3.9051    6.0208    5.0000         0    3.9051    1.1180    4.9497    0.7071
    7.7782    2.1213    1.1180    3.9051         0    2.8284    1.1180    3.9051
    4.9497    4.9497    3.9051    1.1180    2.8284         0    3.9051    1.1180
    8.8459    1.1180    0.7071    4.9497    1.1180    3.9051         0    5.0000
    3.9051    6.0208    4.9497    0.7071    3.9051    1.1180    5.0000         0

 -------------- step 1: find clusteroids
specified k=2, specified point 1 to be the first representative.
search clusteroid #2, calc max of min(distance from each data point to existing clusteroids):
looking for clusteroid #2, min dist(point #2,clusteriods)=9.89949
```

```
looking for clusteroid #2, min dist(point #3,clusteriods)=8.8459
looking for clusteroid #2, min dist(point #4,clusteriods)=3.90512
looking for clusteroid #2, min dist(point #5,clusteriods)=7.77817
looking for clusteroid #2, min dist(point #6,clusteriods)=4.94975
looking for clusteroid #2, min dist(point #7,clusteriods)=8.8459
looking for clusteroid #2, min dist(point #8,clusteriods)=3.90512
point 2 has max (min distance), choose it to be the clusteroid #2.
initial centroids are:1 2
clusteroid_coordinates=[   2.5 ,    9.5 ;
                           2.5 ,    9.5 ];

 -------------- step 2: assign points
distances to representative points and cluster assignment:

data point 3, distance to clusteroids: 8.8459  1.11803
nearest clusteroid is #2
cluster members now:
cluster 1:  1
cluster 2:  2 3
updated clusteroids:
clusteroid_coordinates=[   2.5 ,   9.25 ;
                           2.5 ,      9 ];
data point 4, distance to clusteroids: 3.90512  5.50568
nearest clusteroid is #1
cluster members now:
cluster 1:  1 4
cluster 2:  2 3
updated clusteroids:
clusteroid_coordinates=[  3.75 ,   9.25 ;
                             4 ,      9 ];
data point 5, distance to clusteroids: 5.83631  1.60078
nearest clusteroid is #2
cluster members now:
cluster 1:  1 4
cluster 2:  2 3 5
updated clusteroids:
clusteroid_coordinates=[  3.75 ,   8.83 ;
                             4 ,   8.67 ];
data point 6, distance to clusteroids: 3.0104  3.89087
nearest clusteroid is #1
cluster members now:
cluster 1:  1 4 6
cluster 2:  2 3 5
updated clusteroids:
clusteroid_coordinates=[   4.5 ,   8.83 ;
                          4.67 ,   8.67 ];
data point 7, distance to clusteroids: 5.89727  0.471405
nearest clusteroid is #2
cluster members now:
cluster 1:  1 4 6
cluster 2:  2 3 5 7
```

```
updated clusteroids:
clusteroid_coordinates=[   4.5 ,   8.75 ;
                         4.67 ,   8.75 ];
data point 8, distance to clusteroids: 1.05409  4.96236
nearest clusteroid is #1
cluster members now:
cluster 1:  1 4 6 8
cluster 2:  2 3 5 7
updated clusteroids:
clusteroid_coordinates=[  4.75 ,   8.75 ;
                          4.75 ,   8.75 ];
k=2, max diameter=4.94975
================================================================================
k=3:
data:
X=[  2.50 ,   9.50 ,   9.00 ,   5.00 ,   8.00 ,   6.00 ,   8.50 ,   5.50 ;
     2.50 ,   9.50 ,   8.50 ,   5.50 ,   8.00 ,   6.00 ,   9.00 ,   5.00 ];
distances:
         0    9.8995    8.8459    3.9051    7.7782    4.9497    8.8459    3.9051
    9.8995         0    1.1180    6.0208    2.1213    4.9497    1.1180    6.0208
    8.8459    1.1180         0    5.0000    1.1180    3.9051    0.7071    4.9497
    3.9051    6.0208    5.0000         0    3.9051    1.1180    4.9497    0.7071
    7.7782    2.1213    1.1180    3.9051         0    2.8284    1.1180    3.9051
    4.9497    4.9497    3.9051    1.1180    2.8284         0    3.9051    1.1180
    8.8459    1.1180    0.7071    4.9497    1.1180    3.9051         0    5.0000
    3.9051    6.0208    4.9497    0.7071    3.9051    1.1180    5.0000         0
  -------------- step 1: find clusteroids
specified k=3, specified point 1 to be the first representative.
search clusteroid #2, calc max of min(distance from each data point to existing clusteroids):
looking for clusteroid #2, min dist(point #2,clusteriods)=9.89949
looking for clusteroid #2, min dist(point #3,clusteriods)=8.8459
looking for clusteroid #2, min dist(point #4,clusteriods)=3.90512
looking for clusteroid #2, min dist(point #5,clusteriods)=7.77817
looking for clusteroid #2, min dist(point #6,clusteriods)=4.94975
looking for clusteroid #2, min dist(point #7,clusteriods)=8.8459
looking for clusteroid #2, min dist(point #8,clusteriods)=3.90512
point 2 has max (min distance), choose it to be the clusteroid #2.
search clusteroid #3, calc max of min(distance from each data point to existing clusteroids):
looking for clusteroid #3, min dist(point #3,clusteriods)=1.11803
looking for clusteroid #3, min dist(point #4,clusteriods)=3.90512
looking for clusteroid #3, min dist(point #5,clusteriods)=2.12132
looking for clusteroid #3, min dist(point #6,clusteriods)=4.94975
looking for clusteroid #3, min dist(point #7,clusteriods)=1.11803
looking for clusteroid #3, min dist(point #8,clusteriods)=3.90512
point 6 has max (min distance), choose it to be the clusteroid #3.
initial centroids are:1 2 6
clusteroid_coordinates=[   2.5 ,    9.5 ,      6 ;
                           2.5 ,    9.5 ,      6 ];

 -------------- step 2: assign points
distances to representative points and cluster assignment:
```

```
data point 3, distance to clusteroids: 8.8459  1.11803  3.90512
nearest clusteroid is #2
cluster members now:
cluster 1:  1
cluster 2:  2 3
cluster 3:  6
updated clusteroids:
clusteroid_coordinates=[   2.5 ,   9.25 ,      6 ;
                           2.5 ,      9 ,      6 ];
data point 4, distance to clusteroids: 3.90512  5.50568  1.11803
nearest clusteroid is #3
cluster members now:
cluster 1:  1
cluster 2:  2 3
cluster 3:  6 4
updated clusteroids:
clusteroid_coordinates=[   2.5 ,   9.25 ,    5.5 ;
                           2.5 ,      9 ,   5.75 ];
data point 5, distance to clusteroids: 7.77817  1.60078  3.36341
nearest clusteroid is #2
cluster members now:
cluster 1:  1
cluster 2:  2 3 5
cluster 3:  6 4
updated clusteroids:
clusteroid_coordinates=[   2.5 ,   8.83 ,    5.5 ;
                           2.5 ,   8.67 ,   5.75 ];
 point 7, distance to clusteroids: 8.8459  0.471405  4.42295
nearest clusteroid is #2
cluster members now:
cluster 1:  1
cluster 2:  2 3 5 7
cluster 3:  6 4
updated clusteroids:
clusteroid_coordinates=[   2.5 ,   8.75 ,    5.5 ;
                           2.5 ,   8.75 ,   5.75 ];
data point 8, distance to clusteroids: 3.90512  4.96236  0.75
nearest clusteroid is #3
cluster members now:
cluster 1:  1
cluster 2:  2 3 5 7
cluster 3:  6 4 8
updated clusteroids:
clusteroid_coordinates=[   2.5 ,   8.75 ,    5.5 ;
                           2.5 ,   8.75 ,    5.5 ];
k=3, max diameter=2.12132
k=1, max_diameter=9.89949
k=2, max_diameter=4.94975
k=3, max_diameter=2.12132
```

And again, using Matlab, and including the Mahalanobis distance question:

```
Clustering using k-means with k=2:
Best total sum of distances = 17
idx=[    1,     2,     2,     1,     2,     1,     2,     1];
data points in cluster i=1:
Xi=[  2.5,     5,     6,   5.5;
      2.5,   5.5,     6,     5];
distances_in_cluster=[    0,  3.91,  4.95,  3.91;
                       3.91,     0,  1.12,  0.707;
                       4.95,  1.12,     0,  1.12;
                       3.91,  0.707,  1.12,     0];
distances_in_cluster_M=[    0,  1.87,  1.56,  1.87;
                         1.87,     0,  1.45,  2.83;
                         1.56,  1.45,     0,  1.45;
                         1.87,  2.83,  1.45,     0];
data points in cluster i=2:
Xi=[  9.5,     9,     8,   8.5;
      9.5,   8.5,     8,     9];
distances_in_cluster=[    0,  1.12,  2.12,  1.12;
                       1.12,     0,  1.12,  0.707;
                       2.12,  1.12,     0,  1.12;
                       1.12,  0.707,  1.12,     0];
distances_in_cluster_M=[    0,  1.45,  0.669,  1.45;
                         1.45,     0,  1.45,  2.83;
                         0.669,  1.45,     0,  1.45;
                         1.45,  2.83,  1.45,     0];


Clustering using k-means with k=3:
Best total sum of distances = 3.5
idx=[    3,     1,     1,     2,     1,     2,     1,     2];
data points in cluster i=1:
Xi=[  9.5,     9,     8,   8.5;
      9.5,   8.5,     8,     9];
distances_in_cluster=[    0,  1.12,  2.12,  1.12;
                       1.12,     0,  1.12,  0.707;
                       2.12,  1.12,     0,  1.12;
                       1.12,  0.707,  1.12,     0];
distances_in_cluster_M=[    0,  1.45,  0.669,  1.45;
                         1.45,     0,  1.45,  2.83;
                         0.669,  1.45,     0,  1.45;
                         1.45,  2.83,  1.45,     0];
```

```
data points in cluster i=2:
Xi=[    5,     6,   5.5;
      5.5,     6,     5];
distances_in_cluster=[    0,  1.12,  0.707;
                       1.12,     0,  1.12;
                       0.707,  1.12,     0];
distances_in_cluster_M=[    0,  1.45,  2.83;
                         1.45,     0,  1.45;
                         2.83,  1.45,     0];
data points in cluster i=3:
Xi=[  2.5;
      2.5];
distances_in_cluster=[    0];
distances_in_cluster_M=[    0];
```

(d) **(3)** elbow plots: above. The appropriate number of clusters based on Euclidean distances seems to be 3.

(e) **(4)** 30 words: cost of k-means: N*k, because we need to calculate the distance from N points to k clusteroids. Hierarchical clustering is more expensive because we need to calculate the distance between all points at each step.

(f) **(5)** Data covariance matrix is XX^T/N; Interpret the off-diagonal elements: the two coordinates are strongly correlated (large off-diagonal terms) because the data points extend along the diagonal of $x = y$. Calculating missing Mahalanobis distances:

```
distances_M=[0,  3.12,  3.12,  1.87,  2.45,  1.56,  3.12,  1.87;
             3.12,     0,  1.45,  2.36,  0.669,  1.56,  1.45,  2.36;
             3.12,  1.45,     0,  3.23,  1.45,  1.87,  2.83,  1.56;
             1.87,  2.36,  3.23,     0,  1.87,  1.45,  1.56,  2.83;
             2.45,  0.669,  1.45,  1.87,     0,  0.892,  1.45,  1.87;
             1.56,  1.56,  1.87,  1.45,  0.892,     0,  1.87,  1.45;
             3.12,  1.45,  2.83,  1.56,  1.45,  1.87,     0,  3.23;
             1.87,  2.36,  1.56,  2.83,  1.87,  1.45,  3.23,     0];
```

(g) (i) **(4)** cluster diameters for $k = 1, 2, 3$ using the Mahalanobis distance: look for points that are furthest away in each cluster based on Mahalanobis distance. These are not the same points that are furthest away in terms of Euclidean distances! We find, for example, that with $k = 2$ the largest distances within the two clusters are between points 7 and 3 in one cluster and points 4 and 8 in the other. These are the points that seem closer in Euclidean distances. The reason is that the Mahalanobis distance compresses the distances along the diagonal direction.

(ii) **(4)** Elbow plot using the Mahalanobis distances: above. Based on this, only two clusters! Point 1 belongs with 4,6,8 given that the distances along the diagonal are stretched in this data set.

4. (10 pts) Calculate an approximation for the distances between pairs of random vectors in the L_2 norm that is valid at the limit of a large data dimension N. Assume the random vectors are normally distributed along each dimension, such that the probability of finding a value x_i of the ith coordinate of a vector is $P(x_i) = \frac{1}{\sqrt{\pi}} e^{-x_i^2}$. Note that $\int_{-\infty}^{\infty} x^2 \exp(-x^2) = \sqrt{\pi}/2$.

Hint: This question is not asking you to repeat a calculation from HW or class, but to think independently based on material covered in class and HW. You may make assumptions regarding whichever information you find needed yet unspecified.

Solution:

The distance is given by,

$$d^2 = \sum_{i=1}^{N} (x_i - y_i)^2 = N \langle (x_i - y_i)^2 \rangle = N \int_{-\infty}^{\infty} dx_i \int_{-\infty}^{\infty} dy_i \, \frac{1}{\sqrt{\pi}} e^{-x_i^2} \frac{1}{\sqrt{\pi}} e^{-y_i^2} (x_i - y_i)^2.$$

This used the fact that for large N, the sum over i is the same as N times the average distance, and the average is written as the probability distribution function times the quantity being averaged. Using $(x_i - y_i)^2 = x_i^2 - 2x_i y_i + y_i^2$, the integral over $-2x_i y_i$ should vanish because it has equal and opposite contributions over the different quadrants of the plane. The integrals over each coordinate i are the same, so that the sum can be replaced by a factor N. The integral can then be written as,

$$d^2 = N \int_{-\infty}^{\infty} dx_i \int_{-\infty}^{\infty} dy_i \, \frac{1}{\sqrt{\pi}} e^{-x_i^2} \frac{1}{\sqrt{\pi}} e^{-y_i^2} (x_i^2 + y_i^2).$$

This can be broken into two integrals, one over x_i^2 and one over y_i^2. The two are identical, so can be replaced with just one of them with a factor 2,

$$\begin{aligned} d^2 &= 2N \int_{-\infty}^{\infty} dx_i \int_{-\infty}^{\infty} dy_i \, \frac{1}{\sqrt{\pi}} e^{-x_i^2} \frac{1}{\sqrt{\pi}} e^{-y_i^2} x_i^2 \\ &= 2N \times \int_{-\infty}^{\infty} dx_i \frac{1}{\sqrt{\pi}} e^{-x_i^2} x_i^2 \times \int_{-\infty}^{\infty} dy_i \frac{1}{\sqrt{\pi}} e^{-y_i^2} \\ &= 2N \times \frac{1}{2} \times 1 = N. \end{aligned}$$

where we used the fact that the integral over the PDF is one. The final results means that $d = \sqrt{N}$.