IITM-CS5691 : Pattern Recognition and Machine Learning — Release Date: Apr 5, 2021
Assignment 2 — Due Date: Apr 18, 2021 (11.59pm)

Roll No: CS18B013 — Name: PRASHANTH
Collaborators (if any):
References (if any):

- Use LaTeX to write-up your solutions (in the solution blocks of the source LaTeX file of this assignment), and submit the resulting single pdf file at GradeScope by the due date. (Note: **No late submissions** will be allowed, other than one-day late submission with 10% penalty or four-day late submission with 30% penalty! Within GradeScope, indicate the page number where your solution to each question starts, else we won't be able to grade it! You can join GradeScope using course entry code **5VDNKV**).
- For the programming question, please submit your code (rollno.ipynb file and rollno.py file in rollno.zip) directly in moodle, but provide your results/answers in the pdf file you upload to GradeScope.
- Collaboration is encouraged, but all write-ups must be done individually and independently, and mention your collaborator(s) if any. Same rules apply for codes written for any programming assignments (i.e., write your own code; we will run plagiarism checks on codes).
- If you have referred a book or any other online material for obtaining a solution, please cite the source. Again don't copy the source *as is* - you may use the source to understand the solution, but write-up the solution in your own words.
- Points will be awarded based on how clear, concise and rigorous your solutions are, and how correct your code is. Overall points for this assignment would be **min**(your score including bonus points scored, 50).

1. (10 points) [SINGULARLY PCA!] Consider a dataset of N points with each datapoint being a D-dimensional vector in $\mathbb{R}^D$. Let's assume that:
   - we are in a high-dimensional setting where $D >> N$ (e.g., D in millions, N in hundreds).
   - the $N \times D$ matrix X corresponding to this dataset is already mean-centered (so that each column's mean is zero, and the covariance matrix seen in class becomes $S = \frac{1}{N}X^TX$).
   - the rows (datapoints) of X are linearly independent.

   Under the above assumptions, please attempt the following questions.

   (a) (3 points) Whereas X is rectangular in general, XX^T and X^TX are square. Show that these two square matrices have the same set of non-zero eigenvalues. Further, argue briefy why these equal eigenvalues are all positive and N in number, and derive the multiplicity of the zero eigenvalue for both these matrices.
   (Note: The square root of these equal positive eigenvalues $\{\lambda_i := \sigma_i^2\}_{i=1,\ldots,N}$ are called the singular values $\{\sigma_i\}_{i=1,\ldots,N}$ of X.)

Solution: Where as X is rectangular in general,XX^T and X^TX are square. Show that these two square matrices have the same set of non-zero eigenvalues.

$$XX^T\psi = \lambda\psi \therefore \lambda \text{ is not equal to } 0$$
$$X^TXX^T\psi = \lambda X^T\psi \text{ Multiply } X^t \text{ on both sides}$$
$$X^TX(X^T\psi) = \lambda(X^T\psi)$$
$$\text{Let } P = XX^T$$
$$Z^TXX^TZ = (Z^TX)(Z^TX)^T$$

$$\boxed{|\,|Z^TX||^2 >= 0}$$

$$P = \lambda \Rightarrow \psi^TY\psi = \lambda * ||\psi||^2$$

The Rank for $XX^T and X^TX has N$. the multiplicity of the zero eigenvalue for XX^T=0 and for XX^T is $D - N$.

(b) (2 points) We can choose the set of eigenvectors $\{u_i\}_{i--=1,\ldots,N}$ of XX^T to be an orthonormal set and similarly we can choose an orthonormal set of eigenvectors $\{v_j\}_{j=1,\ldots,D}$ for X^TX. Briefly argue why this orthonormal choice of eigenvectors is possible. Can you choose $\{v_i\}$ such that each v_i can be computed easily from u_i and X alone (i.e., without having to do an eigenvalue decomposition of the large matrix X^TX; assume $i = 1, \ldots, N$ so that $\lambda_i > 0$ and $\sigma_i > 0$)?
(Note: $\{u_i\}$,$\{v_i\}$ are respectively called the left,right singular vectors of X, and computing them along with the corresponding singular values is called the Singular Value Decomposition or SVD of X.)

Solution: We can choose the set of eigenvectors $\{u_i\}_{i--=1,\ldots,N}$ of XX^T to be an orthonormal set and similarly we can choose an orthonormal set of eigenvectors $\{v_j\}_{j=1,\ldots,D}$ for X^TX. Let us solve the equar=tion which is as folows.

$$|v_i|^2 = v_i^Tv_i$$

$$\boxed{\because v_i = \tfrac{1}{\sigma_i}X^Tu_i}$$

$$v_i^Tv_i = \tfrac{1}{\sigma_i^2}u_i^TX^TXu_i$$
$$\tfrac{1}{\sigma_i^2}u_i^TX^TXu_i = \tfrac{1}{\sigma_i^2}u_i^T\sigma_i^2u_i$$
$$\tfrac{1}{\sigma_i^2}u_i^T\sigma_i^2u_i = 1$$
$$v_i^Tv_j = \tfrac{1}{\sigma_i\sigma_j}u_i^TXX^Tu_j$$
$$\tfrac{1}{\sigma_i\sigma_j}u_i^TXX^Tu_j = 0$$

(c) (2 points) Applying PCA on the matrix X would be computationally difficult as it would in-

volve finding the eigenvectors of $S = \frac{1}{N}X^TX$, which would take $O(D^3)$ time. Using answer to the last question above, can you reduce this time complexity to $O(N^3)$? Please provide the exact steps involved, including the exact formula for computing the normalized (unit-length) eigenvectors of S.

Solution: eigenvectors for XX^T is $O(N^3)$ –> (1).
for $\frac{1}{\sigma_i}X^Tu_i\frac{\mu_i}{sigma_i}$ is $O(N^2D)$ –> (2)
By adding (1) and (2) we get: $O(N^3) + O(N^2D) = O(N^2D)$

(d) (3 points) Exercise 12.2 from Bishop's book helps prove why minimum value of the PCA squared error J, subject to the orthonormality constraints of the set of principal axes/directions $\{u_i\}$ that we seek, is obtained when the $\{u_i\}$ are eigenvectors of the data covariance matrix S. That exercise introduces a modified squared error $\widetilde{J}$, which involves a matrix H of Langrange multipliers, one for each constraint, as follows:

$$\widetilde{J} = Tr\left\{\widehat{U}^TS\widehat{U}\right\} + Tr\left\{H(I - \widehat{U}^T\widehat{U})\right\}$$

where $\widehat{U}$ is a matrix of dimension $D \times (D - M)$ whose columns are given by u_i. Now, any solution to minimizing $\widetilde{J}$ should satisfy $S\widehat{U} = \widehat{U}H$, and one specific solution is that the columns of $\widehat{U}$ are the eigenvectors of S, in which case H is a diagonal matrix. Show that any general solution to $S\widehat{U} = \widehat{U}H$ also gives the same value for $\widetilde{J}$ as the above specific solution.
(Hint: Show that H can be assumed to be a symmetric matrix, and then use the eigenvector expansion i.e., diagonalization of H.)

Solution:

2. (10 points) [TO SQUARE OR NOT SQUARE WITH K-MEANS]

(a) (3 points) If instead of squared Euclidean distance, you use ℓ_1 norm in the objective function of (hard) K-means, then what are the new assignment and update equations? If your data contains outliers, would you prefer this over the regular K-means? Justify.

Solution:

(b) (2 points) Consider a Gaussian mixture model with scalar covariance matrices: $\Sigma_r = \sigma^2 I$ where σ is a fixed parameter, r represents the mixture-component/cluster, and I the identity matrix. Show that for this model as σ tends to zero, the EM-based soft K-means algorithm (i.e., its assignment/update equations) become the same as the hard K-means algorithm.

Solution:

(c) (5 points) We will see how K-means clustering can be done in polynomial time if the data points are along a line (1-dimensional or 1D).

i. (1 point) Consider four datapoints: $x_1 = 1, x_2 = 3, x_3 = 6$, and $x_4 = 7$; and desired number of clusters $k = 3$. What is the optimal K-means clustering in this case?

Solution: At $x = 1, x = 3, x = 6, x = 7$. The optimal K-means clustering in this case $0 + 0 + (0.5)^2 + (0.5)^2 = 0.5$

ii. (1 point) You might think that the iterative K-means algorithm seen in class converges to global optima with 1D datapoints. Show that it can get stuck in a suboptimal cluster assignment for the problem in part (i).

Solution: The three means at $x = 3, x = 6, and x = 7$. The first cluster Points consist x_1 and x_2 next cluster is x_3 and next is x_4, The point x_1 and x_2 are close to the first mean Similarly for point x_3 is close to second mean hence, there is no update for the this mean and for x_4 as well. so therefore we can say that erative K-means can get stuck in a suboptimal cluster.

iii. (3 points) Suppose we sort our data such that $x_1 \leqslant x_2 \leqslant \cdots \leqslant x_n$. Show then that any optimal K-means clustering partitions the points into contiguous intervals, i.e. prove that each cluster in an optimal clustering consists of points $x_a, x_{a+1}, \ldots, x_b$ for some $1 \leqslant a \leqslant b \leqslant n$.

Solution: Cluster assignment is used to assign data to the clusters that were previously generated by some clustering methods such as K-means, This algorithm requires that the corresponding clustering procedures save cluster information, or cluster model, which also includes the control parameters for consistency. It assumes that new data is from similar distribution as previous data, and will not update the cluster information. For clusters generated by K-means, distances between new data and cluster centers are calculated, and then the new data is assigned to the cluster with the smallest distance. if we consider a center p_i therefore every point x and p_i is close to the cluster center. the all points from x_0 to x_1 is assigned to clusters. Hence, that any optimal K-means clustering partitions the points into contiguous intervals that is each each cluster in an optimal clustering consists of points.

iv. (1 point) [BONUS] Show a $O(kn^2)$ dynamic programming algorithm of k-means when the data is 1-dimensional.

Solution:

3. (10 points) [THINKING HIERARCHICALLY...] Consider some of the most common metrics of dis-

tance between two clusters $A = \{a_1, a_2, \ldots, a_m\}$ and $B = \{b_1, b_2, \ldots, b_n\}$.

- Minimum distance between any pair of points from the two clusters

$$\min_{a \in A, b \in B} \|a - b\|$$

- Maximum distance between any pair of points from the two clusters,

$$\max_{a \in A, b \in B} \|a - b\|$$

- Average distance between *all* pairs of points from the two clusters,

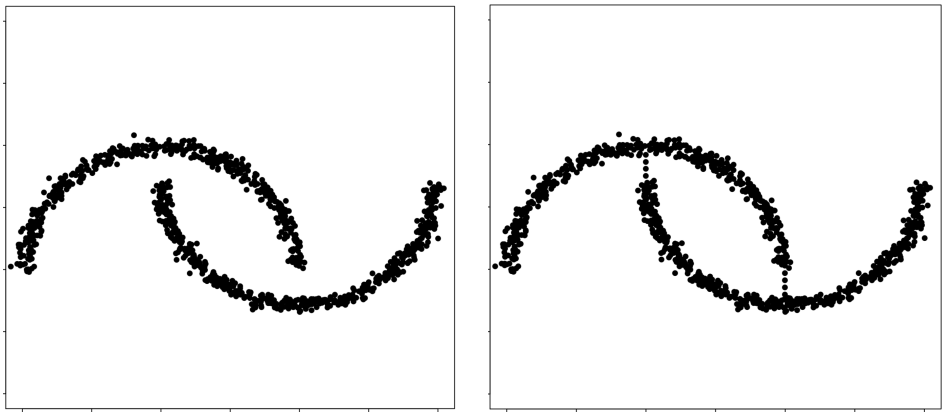
$$\frac{1}{m \cdot n} \sum_{i=1}^{m} \sum_{j=1}^{n} \|a_i - b_j\|$$

As discussed in class, we can obtain clusters by *cutting* the hierarchical tree with a line that crosses at required number of points (K).

(a) (2 points) Which of the three distance/dissimilarity metrics described above would most likely result in clusters most similar to those given by K-means? (Consider the hierarchical clustering method as described in class and further *cut* the tree to obtain K clusters. Assume K is power of 2.) Explain briefly.

Solution: Clustering is a technique that groups similar objects such that the objects in the same group are more similar to each other than the objects in the other groups. The group of similar objects is called a Cluster. Hierarchical cluster analysis or HCA is an unsupervised clustering algorithm which involves creating clusters that have predominant ordering from top to bottom. The average linkage clustering is a method of calculating distance between clusters in hierarchical cluster analysis . The linkage function specifying the distance between two clusters is computed as the average distance between objects from the first cluster and objects from the second cluster. This performs similar to K-means so the average linkage would be most similar to K-means. The remaining types of linkages will produce clusters of different shapes and cannot be able to split convex shapes to a good extent.

(b) (3 points) Which among the three metrics discussed above (if any) would lead hierarchical clustering to correctly separate the two moons in Figure 1a? How would your answer change (if at all) in case of Figure 1b? Explain briefly.



(a) (b)

Figure 1: (a) Standard moon crescent distribution. (b) Moon crescent distribution with data points in adjoining area.

Solution: For the given figure 1a, The great option to lead hierarchical clustering to correctly separate the two moons is by Minimum distance between any pair of points from the two clusters. I believe that minimum linkage will separate the two moons.
For the given figure 1.b, Any of the three methods will not work for this particular figure(1.b), because
a) if we consider minimum clustering, the datapoints may have in different clusters.
b) If we consider of maximum linkage, we have the points in the same cluster, this points have maximum distance comparing with other points in the same cluster.
c) we cannot use Average linkage as well. so, the these three option are not suitable for this particular figure(1.b)

(c) (3 points) Consider the distance matrix in Table 1. Show the hierarchy of clusters created by the minimum distance hierarchical clustering algorithm, along with the intermediate steps. Finally, draw the dendrogram with edge lengths indicated.
(Note: You can draw the dendrogram on paper and upload the screenshot.)

Table 1: Distance between nodes

	A	B	C	D	E
A	0	0.73	6.65	4.61	5.24
B	0.73	0	4.95	2.90	3.45
C	6.65	4.95	0	2.24	1.41
D	4.61	2.90	2.24	0	1
E	5.24	3.45	1.41	1	0

Solution: The hierarchy of clusters created bythe minimum distance hierarchical clustering algorithm, along with the intermediate steps. In the given table AB has min distance

	AB	C	D	E
AB	0	4.95	2.90	3.45
C	4.95	0	2.24	1.41
D	2.90	1.41	0	1
E	3.45	1.41	1	0

1) Now, let us change the matrix by deleting particular rows and columns and add column and row AB, then the new outcome of the matrix is as follows.

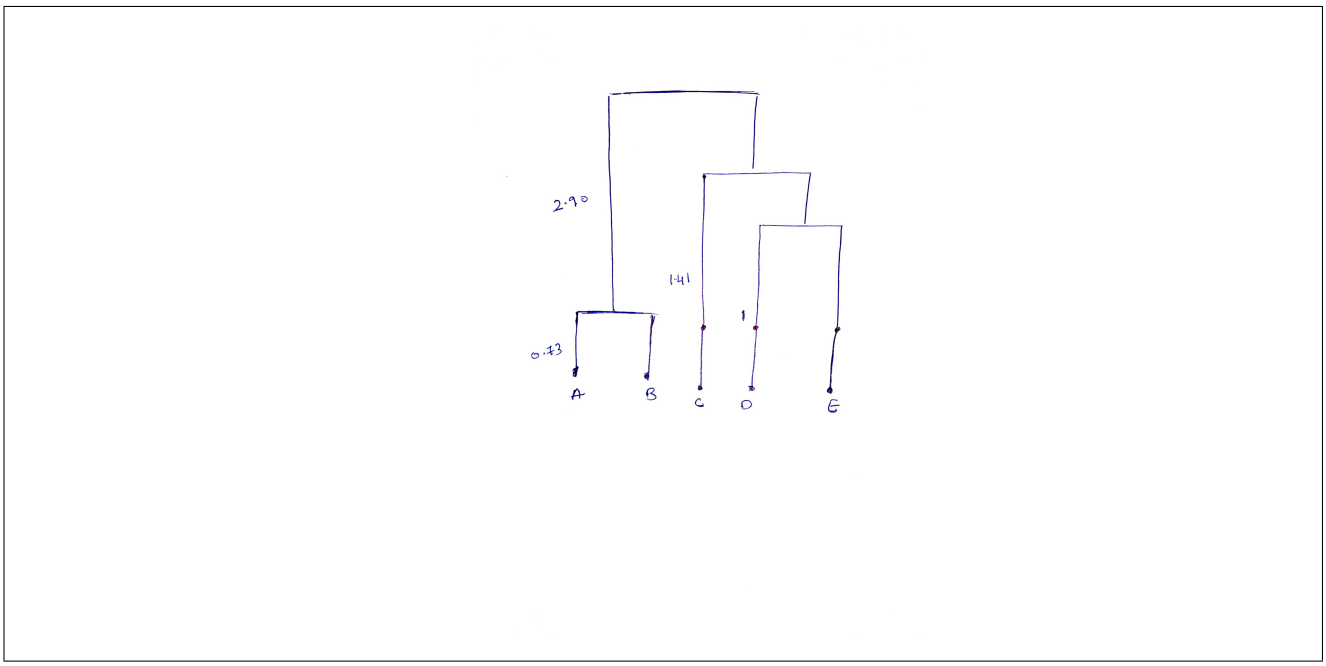
	AB	C	DE
AB	0	4.95	2.90
C	4.95	0	1.41
DE	2.90	1.41	0

$$\boxed{c_i =\because distance = min(dis(A), dis(B))}$$

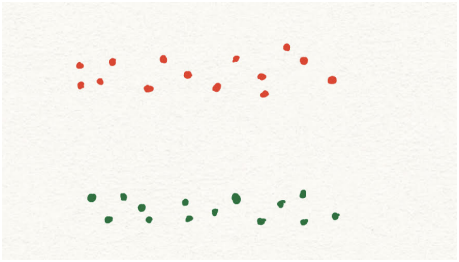
2) C and DE are closest updating table, new outcome matrix is as follows

	AB	CDE
AB	0	2.90
CDE	2.90	0

3) The diagram of dendrogram with edge lengths indicated.which is as follows



(d) (2 points) Which distance metric (minimum, maximum and average) is more likely to generate following results given in Figure 3a for K = 2? Why?



(a)

Figure 3: Result produced for K = 2 clusters. Red points belong to one cluster and green to the other.

Solution: We can use minimum distance metrics to get the clusters as shown above. Because , consider any two nearest points which belong to the same group(above/below). As there is a sufficient gap in between the two given groups, the minimum metric would make sure that all points in the same group would consider distances from the matrix which are actually reduced after using minimum metrics. Hence, minimum metrics has the most chance to produce clusters as shown

4. (10 points) [CUTTING SPECTRAL APART] One of the several ways to express a given dataset is by using a *graph*. Each of the N datapoints in the dataset can be thought of as a vertex/node in a graph

and any two datapoints can be connected in the graph with an edge whose non-negative weight W_{ij} indicates the similarity between the ith and jth datapoints. We will look at methods to partition this graph into two clusters, especially one that gained early prominence in computer vision. These methods can be recursively applied to partition the graph into any required number of clusters.

(a) (1 point) A graph cut is a technique that separates a given graph into two disjoint sets of vertices and the degree of similarity (formally called the *cut cost*) between the two sets is given by the sum of weights of the edges between the sets (i.e., edges whose two endpoint nodes lie in different sides of the partition). The obvious method to separate the data into two is by choosing a partition that has the minimum cut cost. What do you think is/are the drawback(s) of this method? (Hint: Think about the sizes of the two sets in the partition.)

Solution: A graph-cut is a grouping technique in which the degree of dissimilarity between these two groups is computed as the total weight of edges removed between these 2 pieces. By minimizing this cut value, one can optimally bi-partition the graph and achieve good segmentation. The drawback is, The minimum cut criteria occasionally supports cutting isolated nodes in the graph due to the small values achieved by partitioning such nodes. It is dependent on distance and independent on cluster size if there is less number of data points and less distance this is going to make as a cluster so hence this may lead to sub optimal clustering.

(b) (2 points) Due to the above drawback(s), we use a variation of the min cut method called normalized cut to partition the graph into two. The problem of finding the minimum-cost normalized cut can be reduced to this problem:

$$min_y \frac{y^T(D-W)y}{y^TDy} \text{ subject to } y \in \{1,-c\}^N \text{ and } y^TD\mathbf{1} = 0$$

where y_i takes one of the two discrete values $\{1,-c\}$ to indicate which side of the cut/partition the ith datapoint belongs to, W is the symmetric $N \times N$ similarity (non-negative edge-weights) matrix, D is a diagonal matrix called the degree matrix with $d_{ii} = \Sigma_j W_{ij}$, and $\mathbf{1}$ is a vector whose entries are all ones. This expression (not including the constraints) is called the *Generalized Rayleigh's Quotient* (GRQ). The matrix in the numerator, $D - W$ is the called the Laplacian Matrix, denoted by L. Prove that the Laplacian matrix is a singular matrix.

Solution: According to the given question, y_i akes one of the two discrete values1,cto indicate which side of the cut/partition the ith datapoint belongs to,W is the symmetric N×N similarity (non-negative edge weights) matrix, D is a diagonal matrix called the degree matrix with $d_{ii} = \Sigma_j W_{ij}$ and 1 is a vector whose entries are all ones. The matrix in the numerator,$D - W$ is the called the Laplacian Matrix, denoted by L. So we have to prove that the Laplacian matrix is a singular matrix.

$$\boxed{d_{ii} = \Sigma_j W_{ij}}$$

$D - W$ is the called the Laplacian Matrix Now the sum becomes which is as follows.

$$\boxed{\text{sum-}W_{00} - W_{01} - W_{02} \ldots (1\text{stRow})}$$

The determinant becomes.

$$\boxed{\text{C1->C1+C2+C3+C4....CN}}$$

The determinant of D-A is zero.

$$\boxed{\therefore \text{ The laplacian matrix D-A is a singular matrix.}}$$

(c) (3 points) As the above minimization problem is NP-hard with the two constraints, we first let go of both constraints. Then, the above GRQ can be minimized over $y \in \mathbb{R}^N$ by solving the generalized eigenvalue system $(D - W)y = \lambda Dy$. Show that this equation can be expressed in the form $(ALA)z = \lambda z$, by expressing A, z in terms of D, W, y. Compute the eigenvector corresponding to the smallest eigenvalue of the matrix $M = ALA$.

Solution:

(d) (4 points) Now, let's bring back the constraint that $y^T D\mathbf{1} = 0$ (the constraint that y takes only two discrete values can remain relaxed as a final real-valued solution $\{y_i\}$ can be clustered using 2-means for instance to identify the desired partition). Prove that the GRQ above (subject to $y^T D\mathbf{1} = 0$) is minimized when y is the eigenvector corresponding to the second smallest eigenvalue of the generalized eigenvalue system $(D - W)y = \lambda Dy$.
(Hint: You can use the following fact. Let A be a real symmetric matrix. Under the constraint that x is orthogonal to the $(j-1)$ eigenvectors corresponding to the $(j-1)$ smallest eigenvalues of A, the Rayleigh's quotient $\frac{x^T Ax}{x^T x}$ is minimized when x is the eigenvector corresponding to the j^{th} smallest eigenvalue.)

Solution:

5. (10 points) [LIFE IN LOWER DIMENSIONS...] You are provided with a dataset of 1797 images in a folder here - each image is 8x8 pixels and provided as a feature vector of length 64. You will try

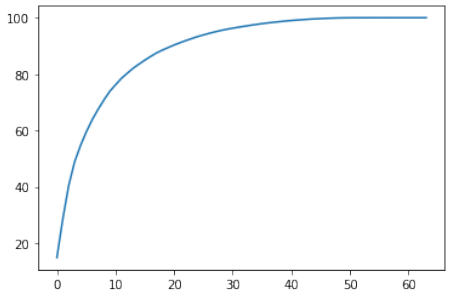
your hands at transforming this dataset to a lower-dimensional space, and clustering the images in this reduced space.

Please use the template .ipynb file in the same folder to prepare your solution. Provide your results/answers in the pdf file you upload to GradeScope, and submit your code separately in this moodle link. The code submitted should be a rollno.zip file containing two files: rollno.ipynb file (including your code as well as the exact same results/plots uploaded to Gradescope) and the associated rollno.py file.

Write the code from scratch for both PCA and clustering. The only exception is the computation of eigenvalues and eigenvectors for which you could use the numpy in-bulit function.

(a) (3 points) Run PCA algorithm on the given dataset. Plot the cumulative percentage variance explained by the principal components. Report the number of principal components that contribute to 90% of the variance in the dataset.

Solution:



If we observe clearly from above graph 90% of the variance is at 20th index, we can say that the variance is at M = 21. We can conclude that 21 principal components that contribute to 90% of the variance of the dataset.

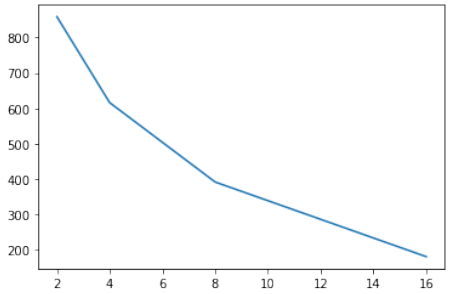
(b) (3 points) Perform reconstruction of data using the dimensionality-reduced data considering the number of dimensions [2,4,8,16]. Report the Mean Square Error (MSE) between the original data and reconstructed data, and interpret the optimal dimension $\widehat{d}$ based on the MSE values.

Solution: The MSE values for each M as follows
1) For M=2 it is 858.9447808487328
2) For M=4 it is 616.1911300562693
3) For M=8 it is 391.7947361149765
4) For M=16 it is 180.93970325737862
So therefore, The MSE value is minimum for M = 16, and hence 16 is the optimal dimension D.

(c) (3 points) Apply K-means clustering on the reduced dataset from last subpart (b) (i.e., the $\mathbb{R}^{64}$

to $\mathbb{R}^{\hat{d}}$ reduced dataset; pick the initial k points as cluster centers during initialization). Report the optimal choice of K you have made from the set [1...15]. Which method did you choose to find the optimum number of clusters? And explain briefy why you chose that method.
Also, show the 2D scatter plot (consider only the first two dimensions of optimal $\hat{d}$) of the datapoints based on the cluster predicted by K-means (use different color for each cluster).

Solution:



The sum of squared distances decreases along with increasing K. to determine this optimal value of k. The values of K will be 9,10. So, here we have used elbow method.

(d) (1 point) Summarise and explain your observations from the above experiments. Is the PCA+K-means clustering consistent with how your brain would cluster the images?

Solution: The intuition is that PCA seeks to represent all n data vectors as linear combinations of a small number of eigenvectors, and does it to minimize the mean-squared reconstruction error. In contrast, K-means seeks to represent all n data vectors via small number of cluster centroids, i.e. to represent them as linear combinations of a small number of cluster centroid vectors where linear combination weights must be all zero except for the single 1. This is also done to minimize the mean-squared reconstruction error. No doubtedly, For clustering the images PCA and K-means are absolute right method to use. so, to consider all for clustering would be more expensive. so Instead, we are using PCA to reduce the dimensions into few important principal ones. After that applying K-means is not that heavy. So, hence PCA+K-means clustering consistent with our brain would cluster the images.