CS37300 Homework	— Homework 5—	Last update: April 4, 2024
Due: April 17, 2024		

Name: _____ PUID: _____

Instructions and Policy: You are allowed to study with others and use online resources for reference, however, the work you turn in must be your own. This means do not copy/paste from Stack Exchange (or from another student.) If you have worked closely with other students, provide their name(s) and a brief (at most one paragraph) description of the interaction; if we feel this oversteps the bounds, we will discuss it with you. Each student should write up their own solutions independently.

The requirements below are supposed to be followed in this and further homework assignments.

- For your theoretical submission:
  - YOU MUST INCLUDE YOUR NAME IN THE HOMEWORK.
  - The answers MUST be submitted via Gradescope.
  - Please write clearly and concisely - clarity and brevity will be rewarded. Refer to known facts as necessary.
  - Theoretical questions MUST include the intermediate steps to the final answer.
- For your programming answers:
  - The Python scripts will be submitted separately via Gradescope
  - Zero points will be given in any question where the Python code answer doesn't match the answer on Gradescope.
  - If the answer is/includes a plot, it should be added to your theoretical submission unless otherwise specified.

Your code is REQUIRED to run on Python 3.11 in an environment detailed in Homework 0 under the Python Installation section. While your code may run in other environments, any points lost due to environmental issues will not be regraded.

Please make sure you don't use any libraries that are not imported for you. If such a library is used, you will get 0 pt for the coding part of the assignment.

If your code doesn't run on Gradescope, then even if it compiles on another computer, it will still be considered not running and the respective part of the assignment will receive 0 pt.

Theoretical Questions (8 + 20 + 16 = 44 pts)

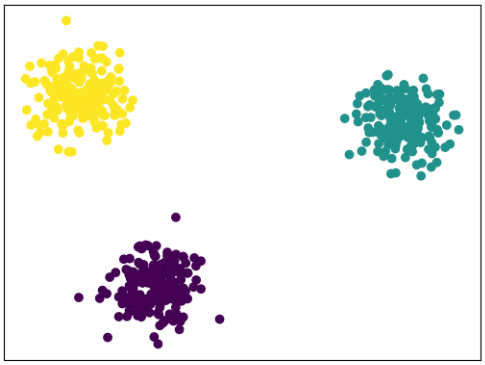
Please submit your answers on Gradescope.

Q0 (8 pts): True or False questions

Answer the following as True or False with a justification or example. Points are uniformly distributed wihtin the questions.

1. (**4 pts**) K-means++ initialization guarantees the convergence of K-means algorithm to the globally optimal solution by iteratively selecting centroids with probabilities proportional to their squared distances from the existing centroids.

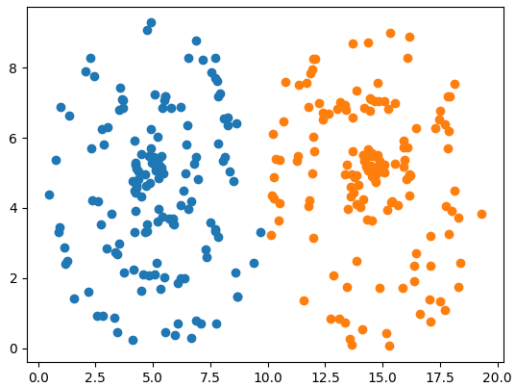
2. (**4 pts**) k-means score is minimized when $k = 3$ for the following dataset:



Q1 (20 pts): Clustering

1. (**5 pts**) There are two clusters of data points ($k = 2$) in the figure below. Which of the following clustering method(s) is(are) most likely to produce the following results? Choose the most likely method(s) and explain why it/they will work better than others briefly. Use several sentences to explain your answer.

   A. Hierarchical clustering with single link.
   B. Hierarchical clustering with complete link.
   C. Hierarchical clustering with average link.
   D. K-means.
   E. GMM (with no assumption on the covariance matrices).

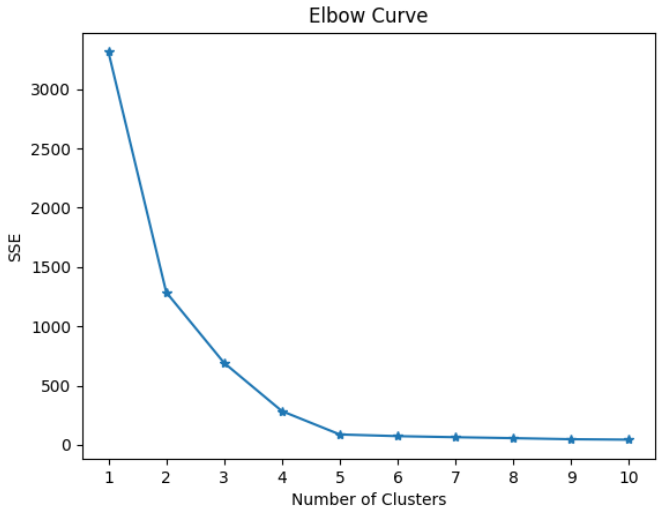


2. (**3 pts**) Finding the right k number in k-means clustering is important and fundamental. There are two popular methods to determine the optimal k number:

   (a) The elbow method:

   - Step 1: Run the algorithm for different choices of k and record the Sum of Squared Errors (SSE).
   - Step 2: Plot the k-SSE curve, then choose the k value at which an increase in k will cause a very small decrease in the error sum, while a decrease will sharply increase the error sum (the elbow of the curve).

   Based on the definition, what is the appropriate k number of the following graph?



(b) (**3 pts**) The silhouette coefficient:

The silhouette coefficient quantifies how well a data point fits into its assigned cluster based on

(1) how far away the data point is from points in other clusters

(2) how close the data point is to other points in the cluster

To compute the silhouette coefficient for data point i, we first compute the following two values:

$$a(i) = \frac{1}{|C_I| - 1} \sum_{j \in C_I, i \neq j} d(i, j)$$

$$b(i) = \min_{J \neq I} \frac{1}{|C_J|} \sum_{j \in C_J} d(i, j),$$

where C_I and C_J denote clusters I and J and i and j denote data points, with point i belonging to cluster C_I and point j belonging to cluster C_J. Then, the silhouette coefficient (whose value ranges between -1 and 1) is given by:

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}.$$

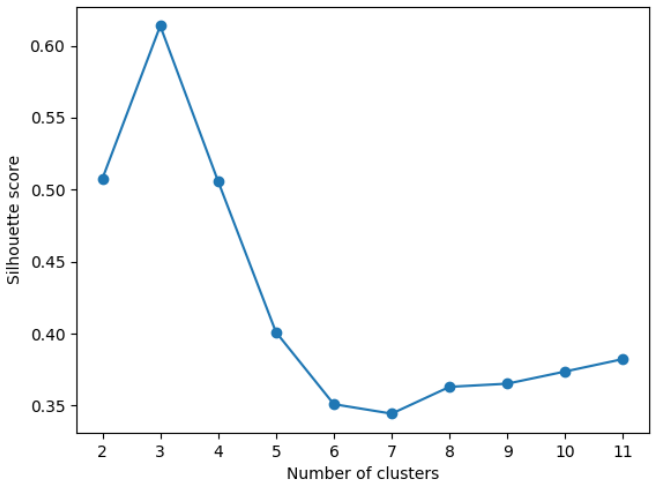
Larger numbers indicate that samples are closer to points in their own clusters than they are to points in other clusters.

This is used to choose the number of clusters by plotting the average silhouette coefficient of all samples, for different choices of k, and choosing the highest score.

Answer the following question: Consider a scenario where a data point "x" belongs to cluster "I" with a low average distance to the points within its own cluster and a higher average distance to points in a different cluster. Choose the correct option for x. How would the silhouette coefficient for data point "x" be affected by these distance characteristics?

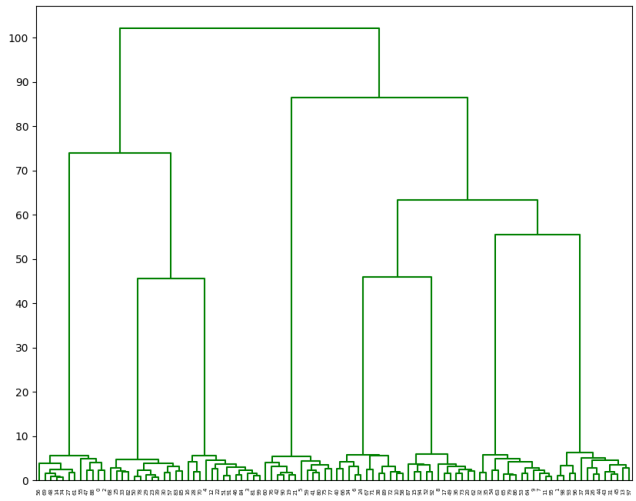
i. $a(x) > b(x)$
ii. $a(x) = b(x)$
iii. $a(x) < b(x)$

(c) (**3 pts**) For the following plot of the silhouette coefficient method, which value of k should we choose?



3. One way to choose the optimal number of clusters using the dendrogram is to find the region with maximum distance (vertically) which does not include a new branching (split). Then the number of clusters can be determined by the number of branches within that region.

   (a) ((**3 pts**)) Based on the above method, find the appropriate number of clusters for the dendrogram given below (you may directly draw on the figure below).



(b) (**3 pts**) Explain why the above method is a reasonable way to choose the number of clusters. (Hint: use the meaning of height in dendrogram.)

Q2 (16 pts): PCA

Find the first two principal components of the given 2D points

$$x_1 = (3, 0), \quad x_2 = (1, 2), \quad x_3 = (4, 3), \quad x_4 = (0, -1).$$

1. **(4 pts)** Construct data matrix X for these points and then standardize it to $\tilde{X}$ (subtract x_{ij} with $\bar{x_j}$) so that the mean of each feature is 0.

2. **(4 pts)** Calculate the covariance matrix Σ. Note: The covariance matrix is given by

$$\Sigma = \frac{1}{n}\tilde{X}^T\tilde{X}$$

where $\tilde{X}$ is the standardized data matrix, and n is the sample size.

3. **(4 pts)** Calculate the two eigenvalues of the covariance matrix Σ. Which eigenvalue corresponds to the **first** principal component? Please also explain your choice in the answer.

4. **(4 pts)** Please calculate the principal component associated with the smaller eigenvalue from the previous question, for the given 2D points. The equation for computing an eigenvector is given below.

$$\Sigma v = \lambda v$$

Please normalize the length of the principal component to 1 (i.e., its L2 norm is 1).

Programming Part (46 pts)

Overview

In this assignment, you will implement a PCA algorithm using `numpy`.

Files

You will fill in functions in `pca.py`. After you finish, you will submit these as well as any other files mentioned in the assignment to Gradescope.

Asides from these three files, you will also find utils.py in the same folder. You do not need to modify it. utils.py contains some helper functions.

Packages

You will need the following packages for this assignment:

- `numpy`
- `scikit-learn`
- `torchvision`
- `matplotlib`

These packages should be installed if you have installed Anaconda. If you are not using Anaconda, you can install them using `pip` or `conda`. For example, you can run `pip install numpy` in the terminal to install numpy.

Note that for this assignment, you should not use any other packages. You may see some other packages in the skeleton code, but they are only used for testing and grading. If you use any other packages, you may lose points. If you are not sure whether you can use a package, please ask the course staff.

Evaluation

Unless otherwise specified, your code will be evaluated by a autograder on Gradescope using the same environment as detailed below. You should make sure your code runs without errors in this environment. Otherwise, you may lose points. You can submit your code to Gradescope as many times as you want. We will only grade the latest submission.

There will be two types of test cases in the autograder: public (local) and hidden. Public test cases are visible to you, and you can see the results after you submit your code. You can also run the public test cases locally by running `python <filename>.py` in the same folder as your code. Passing all public test cases **does not** guarantee you will get full credit for the assignment.

Hidden test cases, however, are not visible to you, and you will not be able to see the results until your grade is published. The final score you get for this assignment is the sum of the scores of all public and hidden test cases.

Getting Help

If you have any questions about this assignment, please contact the course staff for help, preferably during office hours. You can also post your questions on the course forum. However, please do not post any code publicly. When asking questions, you should describe your problem in words and post the relevant code snippet. Please avoid showing a screenshot of your code to TAs and ask "why my code is not working".

1 PCA

In this part, you will implement the PCA algorithm using `numpy`. During testing, we will use the PCA algorithm to reduce the dimensionality of the Fashion MNIST dataset and see how well the reduced dataset can be used to classify the images.

You will write your code in `pca.py`.

1.1 PCA - Constructor 4 pts

Under `class PCA`, fill in the `__init__` method so that it initializes `self.n_components` to the number of components to keep.

1.2 PCA - Covariance Matrix 4 pts

Under `class PCA`, fill in the `cov` method so that it returns the covariance matrix of the input data. You should use the formula for covariance matrix from the lecture slides or `np.cov`.

1.3 PCA - Eigenvectors and Eigenvalues 4 pts

Under `class PCA`, fill in the `eig` method so that it returns the eigenvectors and eigenvalues of the input matrix. It is preferred that you use `np.linalg.eigh` to compute the eigenvectors and eigenvalues.

1.4 PCA - Fit 6 pts

Under `class PCA`, fill in the `fit` method so that it computes the eigenvectors and eigenvalues of the covariance matrix of the input data. You should use the `cov` and `eig` methods you implemented above. You should also sort the eigenvectors and eigenvalues in descending order of eigenvalues. You should store the first `self.n_components` eigenvectors in `self.components_`.

Then, calculate the explained variance ratio of the first `self.n_components` components and store it in `self.explained_variance_ratio_` and the first 784 components and store it in `self.explained_variance_ratio_784_`. Recall that the explained variance ratio of a component is the ratio of the variance of that component to the total variance of the data. Mathematically, it is defined as follows:

$$\text{explained variance ratio of component } i = \frac{\lambda_i}{\sum_{j=1}^{d} \lambda_j}$$

where λ_i is the eigenvalue of component i and d is the dimensionality of the data.

1.5 PCA - Transform 6 pts

Under `class PCA`, fill in the transform method so that it transforms the input data using the first `self.n_components` eigenvectors. You should use the `self.components_` attribute you computed in the `fit` method. You should return the transformed data.

Before transforming, you should center the data by subtracting the mean of the data from each data point. The use the following formula to transform the data:

$$\text{transformed data} = \text{data} \times \text{components}^\top$$

where data is the centered data, components is the matrix of eigenvectors, and components$^\top$ is the transpose of the matrix of eigenvectors.

1.6 PCA - Inverse Transform 6 pts

Under `class PCA`, fill in the `inverse_transform` method so that it transforms the input data back to the original space. You should use the `self.components_` attribute you computed in the `fit` method. You should return the reconstructed data.

Remember to add the mean of the data back to each data point after transforming.

1.7 PCA - Analysis of Explained Variance Ratio 4 + 6 = 10 pts

Once you have implemented the PCA algorithm, you need to run the program to see how well the reduced dataset can be used to classify the images. You will see a plot of the explained variance ratio of the first 50 components. Based on the graph and the output of program, fill in the `report_my_judgement` method (4 pts) so that it returns an integer that describes the best number of components to keep in your opinion. Return the number of components you think is best.

You should include the plot of the explained variance ratio in your theoretical writeup (6 pts).

1.8 PCA - Overall Check 6 pts

During test, we will run the program on the test set and see how well the reduced dataset can be used to classify the images. On your side, you should see that the KNN classifier can achieve an accuracy of at least 0.8 on the reduced dataset, which should be pretty comparable to the accuracy of the KNN classifier on the original dataset.

Also generated by the program is a plot of the 5 images in the original dataset and the reconstructed image. That should give you a sense of how well the PCA algorithm works, and you should see that the reconstructed images are pretty similar to the original images. Include the plot of the reconstructed images in your theoretical writeup.

Submission

You will submit the following files to Gradescope:

- `pca.py`
- `compare.png`
- `explained_variance_ratio.png`