ISYE 6740 Fall 2024
Homework 1 (100 points + 10 bonus points)

In this homework, the superscript of a symbol x^i denotes the index of samples (not raising to ith power); this is a convention in this class. Please follow the homework submission instructions in the syllabus.

1 Concept questions [30 points]

Please provide a brief answer to each question, and provide math arguments if asked.

1. (5 points) What's the main difference between supervised and unsupervised learning? Give two benefit and drawback for supervised and unsupervised learning, respectively.

   Answer:
   In supervised learning, we use labeled training data, and in unsupervised learning, we don't have label for data points.
   Supervised learning:
   Benefit: It provides accurate results. It can perform classification and regression tasks.
   Drawback: It's hard to handle big data. It needs labeled data for training process.
   Unsupervised learning:
   Benefit: It can handle very large datasets. It doesn't need labeled data.
   Drawback: It's difficult to evaluate the performance. It may cause higher risk of inaccurate results.

2. (5 points) Consider a dataset with data points each having 3 features, e.g., $x^1 = \{$"Atlanta", "house", $500k\}$, and $x^2 = \{$"San Francisco", "house", $300k\}$. Define a proper similarity function $d(x^i, x^j)$ for this kind of data, and argue why it is a reasonable choice. (Hint: The feature vector consists of categorial and real-valued features; for categorical variables, it is better to convert them into one-hot-keying binary vectors and use Hamming distance, and for real-valued features, you may use Euclidean distance, for instance. And then you can combine the similarity measure in some way.)

   Answer:
   For the first two categorical variables, we convert them into one-hot-keying binary vectors: For city: Atlanta could be 1, San Francisco could be 0 For housing type: house could be 1 Then, x^1: [1, 1, 500]. x^2: [0, 1, 300].
   Hamming distance: 1
   Assuming weighting (α, β, γ) for (city, housing type, price)
   Similarity function d(x^i, x^j):

   $$d(x^1, x^2) = \alpha \cdot 1\,(\text{city}) + \beta \cdot 0\,(\text{housing type}) + \gamma \cdot 200\,(\text{price}) = \alpha + 200\gamma$$

3. (5 points) Show that the clustering assignment problem

   $$\pi(i) = \arg \min_{j=1,\ldots,k} \|x^i - c^j\|^2$$

   is equivalent to solving

   $$\pi(i) = \arg \min_{j=1,\ldots,k} (c^j)^T \left(\frac{1}{2} c^j - x^i \right).$$

Note that the second approach will facilitate "vectorized" operation and be implemented in our demo code.

Answer:
in the following step, when we find minimum, we can ignore the following term because it remain constant

$$\left(x^i\right)^T x^i$$

$$\begin{aligned}
\pi(i) &= \arg\min_{j=1,\ldots,k} \|x^i - c^j\|^2 \\
&= \arg\min_{j=1,\ldots,k} \left(x^i - c^j\right)^T \left(x^i - c^j\right) \\
&= \arg\min_{j=1,\ldots,k} \left(x^i\right)^T x^i - \left(x^i\right)^T c^j - \left(c^j\right)^T x^i + \left(c^j\right)^T c^j \\
&= \arg\min_{j=1,\ldots,k} \left(x^i\right)^T x^i - 2\left(x^i\right)^T c^j + \left(c^j\right)^T c^j \\
&= \arg\min_{j=1,\ldots,k} \left(x^i\right)^T x^i - 2\left(c^j\right)^T x^i + \left(c^j\right)^T c^j \\
&= \arg\min_{j=1,\ldots,k} 2\left(c^j\right)^T \left(\frac{1}{2}c^j - x^i\right)
\end{aligned}$$

is equivalent to

$$\arg\min_{j=1,\ldots,k} (c^j)^T \left(\frac{1}{2}c^j - x^i\right).$$

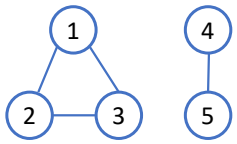
4. (5 points) Why do different initializations for k-means lead to different results?

   Answer:
   The objective function for k-means is non-convex optimization, there are multiple local optimum. After setitng initial centroids, the k-means algorithm iterates between assigning data points to the nearest centroid and recalculating the centroids based on the current cluster assignments. The algorithm stops when the centroids stop moving, but where it converges depends on the initial positions of the centroids. So different starting points might lead to different results.

5. (5 points) Why k-means will (be guaranteed to) stop after a finite number of iterations?

   Answer:
   Since we have fixed number of data points and clusters, there is a finite number of ways to perform enumeration and combinations of these points to the centroids, with each iteration decreasing the sum of squares, leading to eventual convergence when no further changes occur in cluster centroids.

6. (5 points) Consider the following simple graph



   Write down the graph Laplacian matrix and find the eigenvectors associated with the zero eigenvalues. Explain how you find out the number of disconnected clusters in the graph and identify these disconnected clusters using these eigenvectors.

Answer:
Graph Laplacian matrix:

$$L = D - A = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} - \begin{bmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 2 & -1 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ -1 & -1 & 2 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix}$$

Eigendecomposition of the Laplacian:

$$LV = V\Lambda$$

Λ is the diagonal matrix of eigenvalues.
V is the matrix of eigenvectors corresponding to these eigenvalues.

From the eigenvalues, we can know that the number of zero eigenvalues is 2, so there are 2 disconnected clusters in the graph.

Get the following eigenvalues and eigenvectors:

$$\Lambda = \begin{bmatrix} 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad V = \begin{bmatrix} 0.8165 & -0.5774 & 0.3096 & 0 & 0 \\ -0.4082 & -0.5774 & -0.8091 & 0 & 0 \\ -0.4082 & -0.5774 & 0.4995 & 0 & 0 \\ 0 & 0 & 0 & 0.7071 & 0.7071 \\ 0 & 0 & 0 & -0.7071 & 0.7071 \end{bmatrix}$$

Based on these eigenvectors in matrix V, the first three nodes (1, 2, 3) are part of one cluster. The last two nodes (4, 5) form another cluster.

Python Codes:

```
import numpy as np
L = np.array([[ 2, -1, -1,  0,  0],
              [-1,  2, -1,  0,  0],
              [-1, -1,  2,  0,  0],
              [ 0,  0,  0,  1, -1],
              [ 0,  0,  0, -1,  1]])
eigenvalues, eigenvectors = np.linalg.eig(L)
print(np.round(eigenvalues, 4))
print(np.round(eigenvectors, 4))
```

2 Math of k-means clustering [25 points]

Given m data points $\mathrm{x}^i \in \mathbb{R}^n$, $i = 1, \ldots, m$, K-means clustering algorithm groups them into k clusters by minimizing the distortion function over $\{r^{ij}, \mu^j\}$

$$J = \sum_{i=1}^{m} \sum_{j=1}^{k} r^{ij} \|\mathrm{x}^i - \mu^j\|^2, \tag{1}$$

where $r^{ij} = 1$ if x^i belongs to the j-th cluster and $r^{ij} = 0$ otherwise.

1. (10 points) Derive mathematically that using the squared Euclidean distance $\|\mathrm{x}^i - \mu^j\|^2$ as the dissimilarity function, the centroid that minimizes the distortion function J for given assignments r^{ij} are given by

$$\mu^j = \frac{\sum_i r^{ij} \mathrm{x}^i}{\sum_i r^{ij}}.$$

That is, μ^j is the center of j-th cluster.
Hint: You may start by taking the partial derivative of J with respect to μ^j, with r^{ij} fixed.

Answer:
J is the distortion function, when the slop is 0, it would reach the minimum value. Then we want the derivative of the objective J to 0

$$\frac{\partial J}{\partial \mu^j} = \frac{\partial \sum_{i=1}^{m} r^{ij} \|\mathrm{x}^i - \mu^j\|^2}{\partial \mu^j} = 0$$

$$\sum_{i=1}^{m} 2r^{ij}(x^i - \mu^j) = 0$$

$$\sum_{i=1}^{m} 2r^{ij}x^i - \mu^j \sum_{i=1}^{m} 2r^{ij} = 0$$

$$\mu^j = \frac{\sum_i r^{ij} x^i}{\sum_i r^{ij}}$$

2. (10 points) Derive mathematically what should be the assignment variables r^{ij} be to minimize the distortion function J, when the centroids μ^j are fixed.

Answer:
r^{ij} is binary and $\mathrm{r}^{ij} \in \{1, 2\}$, $\sum_{j=1}^{k} r^{ij} = 1$
If we want to minimize the distortion function

$$J = \sum_{i=1}^{m} \sum_{j=1}^{k} r^{ij} \|\mathrm{x}^i - \mu^j\|^2$$

We need to get

$$\min \|x^i - \mu^j\|^2$$

The smallest J happens when each x^i is to assign it to the centroid μ^j that is closest to x^i. Thus, r^{ij} should be 1 for the centroid μ^j that minimizes $\|x^i - \mu^j\|^2$, and 0 for all other centroids.

$$r^{ij} = \begin{cases} 1 & \text{if } j = \arg\min_{j'} \|x^i - \mu^{j'}\|^2, j' = 1, \ldots, k \\ 0 & \text{otherwise} \end{cases}$$

3. (5 points) For the question above, now suppose we change the similar score to a "quadratic" distance (also known as Mahalanobis distance) for given and fixed positive definite matrix $\Sigma \in \mathbb{R}^{n \times n}$, and the distortion function becomes:

$$J = \sum_{i=1}^{m} \sum_{j=1}^{k} r^{ij} (\mathrm{x}^i - \mu^j)^T \Sigma (\mathrm{x}^i - \mu^j),$$

Derive what μ^j and r^{ij} becomes in this case.

Answer:
Derive μ^j:

$$J = \sum_{i=1}^{m} \sum_{j=1}^{k} r^{ij} (\mathrm{x}^i - \mu^j)^T \Sigma (\mathrm{x}^i - \mu^j)$$

$$\frac{\partial J}{\partial \mu^j} = \frac{\partial \sum_{i=1}^{m} r^{ij}(\mathrm{x}^i - \mu^j)^T \Sigma(\mathrm{x}^i - \mu^j)}{\partial \mu^j} = \sum_{i=1}^{m} r^{ij} 2\Sigma(\mathrm{x}^i - \mu^j)$$

Let

$$\sum_{i=1}^{m} r^{ij} 2\left(\Sigma x^i - \Sigma \mu^j\right) = 0$$

Since Σ is positive definite (hence invertible), we can divided both sides by Σ

$$\mu^j = \frac{\sum_i r^{ij} x^i}{\sum_i r^{ij}}$$

Derive r^{ij}:
To minimize $(\mathrm{x}^i - \mu^j)^T \Sigma(\mathrm{x}^i - \mu^j)$

$$r^{ij} = \begin{cases} 1 & \text{if } j = \arg\min_{j'} (\mathrm{x}^i - \mu^j)^T \Sigma(\mathrm{x}^i - \mu^j), j' = 1, \ldots, k \\ 0 & \text{otherwise} \end{cases}$$

3 Image compression using clustering [20 points]

In this programming assignment, you are going to apply clustering algorithms for image compression. This can also be viewed as an example of segmenting colors in an automated fashion using K-means clustering.

Your task is to implement *K-means* for this purpose. **It is required you implement the algorithms yourself rather than calling k-means from a package. However, it is ok to use standard packages for supplementary tasks, e.g., file i/o, linear algebra, and visualization.**

Formatting instruction

As a starting point, we suggest the following input/output signature for your k-means algorithm.

Input

- `pixels`: the input image representation. Each row contains one data point (pixel). For image dataset, it contains 3 columns, each column corresponding to Red, Green, and Blue components. Each component has an integer value between 0 and 255.

- `k`: the number of desired clusters.

Output

- `class`: cluster assignment of each data point in pixels. The assignment should be 1, 2, 3, etc. For $k = 5$, for example, each cell of the class should be either 1, 2, 3, 4, or 5. The output should be a column vector with `size(pixels, 1)` elements.

- `centroid`: location of k centroids (or representatives) in your result. With images, each centroid corresponds to the representative color of each cluster. The output should be a matrix with K rows and 3 columns. The range of values should be [0, 255], possibly floating point numbers.

Hand-in

Both of your code and report will be evaluated. Upload the code as a zip file, and the report as a pdf, separately from the zip file. In your report, answer the following questions:

1. (10 points) Use k-means with squared-ℓ_2 norm as a metric for `coastal-abstract.jpeg` and `football.bmp` and also choose a third picture of your own to work on. We recommend the size of 320×240 or smaller. Run your k-means implementation with these pictures, with several different $k = 2, 4, 8, 16, 32$.

   Comment: Your algorithm will segment the image into k regions in the RGB color space. For each pixel in the input image, the algorithm returns a label corresponding to a cluster.

   Run your k-means implementation (with squared-ℓ_2 norm) with random initialization centroids. Please try multiple times and report the best one for each k (in terms of image quality).

   Answer:

2. (5 points) Please report how long it takes to converge for each k (report the number of iterations and elapsed time in seconds).

   Answer:

   kmeans_football.png:
   k: 2, time: 0.71s, iterations: 30

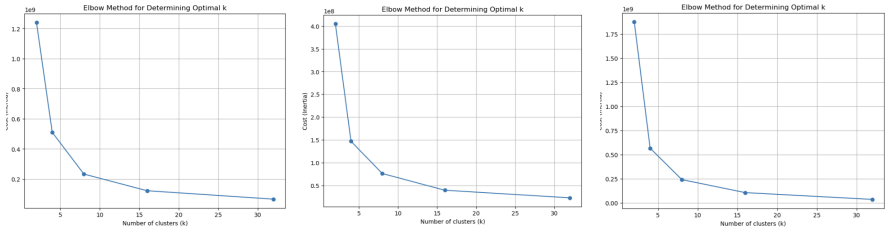
k: 4, time: 2.76s, iterations: 67
k: 8, time: 3.87s, iterations: 52
k: 16, time: 12.60s, iterations: 84
k: 32, time: 25.08s, iterations: 82

kmeans_coastal-abstract.png:
k: 2, time: 0.20s, iterations: 29
k: 4, time: 0.41s, iterations: 32
k: 8, time: 1.50s, iterations: 68
k: 16, time: 5.98s, iterations: 137
k: 32, time: 7.51s, iterations: 83

kmeans_Grammy.png:
k: 2, time: 0.16s, iterations: 16
k: 4, time: 0.17s, iterations: 10
k: 8, time: 1.42s, iterations: 43
k: 16, time: 4.22s, iterations: 73
k: 32, time: 7.28s, iterations: 61

3. (5 points) Describe a method to find the best k. What is your best k?

Answer:



K-means is a clustering method, we are able to use the elbow method which is a heuristic used in determining the number of clusters in a data set. These three plots are elbow_football, elbow_coastal-abstract and elbow_Grammy. The elbow method suggests that the best k is typically at the point where the cost (inertia) starts to decrease more slowly, forming an "elbow" in the curve. This is where adding more clusters provides diminishing returns in terms of reducing the cost. Based on the elbow method, we can choose the k=8.

Note

- You may see errors caused by empty clusters when you use too large k. Your implementation should treat this exception as well. That is, do not terminate even if you have an empty cluster, but automatically decrement to a smaller number of clusters.

- We recommend you test your code with several different pictures so that you can detect some problems that might happen occasionally.

- If we detect plagiarism from any other student's code or from the web, you will not be eligible for any credit for the entire homework, not just for this problem.

4 MNIST Dataset clustering [25 points]

This question is to compare different classifiers and their performance for multi-class classifications on the complete MNIST dataset at `http://yann.lecun.com/exdb/mnist/`. You can find the data file **mnist_10digits.mat** in the homework folder. The MNIST database of handwritten digits has a training set of 60,000 examples and a test set of 10,000 examples. Use the number of clusters $K = 10$.

We suggest you "standardize" the features (pixels in this case) by dividing the values of the features by 255 (thus mapping the range of the features from [0, 255] to [0, 1]).

We are going to use *purity* score as a performance metric: each cluster is assigned to the class which is most frequent in the cluster, and then the accuracy of this assignment is measured by the number of correlated assigned samples and divided by the size of the cluster:

$$\text{purity}_i = \frac{\text{corrected assigned samples}_i}{\text{size of cluster}_i}$$

for the cluster i.

1. (15 points) Use the squared-ℓ_2 norm as a metric for clustering (you may base it on the code you had for Question 2 and make necessary changes.) Report the *purity* score for each cluster.

   Answer:

   Purity Scores for each cluster (0-9): 0.5313171630334664, 0.5268094142629623, 0.907359586830213, 0.4268099547511312, 0.6239301896291324, 0.5279870828848224, 0.35721493440968716, 0.7808690215692404, 0.8970241918218796, 0.8593668007696345

2. (10 points) Now try your k-means with the Manhattan distance (or ℓ_1 distance) and repeat the same steps in Part (1). Please note that the assignment of data points should be based on the Manhattan distance, and the cluster centroid (by minimizing the sum of deviance – as a result of using the Manhattan distance) will be taken as the "median" of each cluster. Report the *purity* score for each cluster. Comment on which metric gives the better result?

   Answer:

   Purity Scores for each cluster (0-9): 0.6409356725146199, 0.44725487278678766, 0.5995013599274706, 0.4793692897141173, 0.39017543859649123, 0.4500792393026941, 0.3693774203157581, 0.6978470675575352, 0.42123248661578766, 0.9310193321616872

   Overall purity score with euclidean distance: 0.59085

   Overall purity score with manhattan distance: 0.5173333333333333

   The k-means clustering with euclidean distance gives the better result.

5 Political blogs dataset [bonus, 10 points]

We will study a political blog dataset first compiled for the paper Lada A. Adamic and Natalie Glance, "The political blogosphere and the 2004 US Election", in Proceedings of the WWW-2005 Workshop on the Weblogging Ecosystem (2005). It is assumed that blog-site with the same political orientation are more likely to link to each other, thus, forming a "community" or "cluster" in a graph. In this question, we will see whether or not this hypothesis is likely to be true based on the data.

- The dataset nodes.txt contains a graph with $n = 1490$ vertices ("nodes") corresponding to political blogs.
- The dataset edges.txt contains edges between the vertices. You may remove isolated nodes (nodes that are not connected to any other nodes) in the pre-processing.

We will treat the network as an undirected graph; thus, when constructing the adjacency matrix, make it symmetrical by, e.g., set the entry in the adjacency matrix to be one whether there is an edge between the two nodes (in either direction).

In addition, each vertex has a 0-1 label (in the 3rd column of the data file) corresponding to the true political orientation of that blog. We will consider this as the true label and check whether spectral clustering will cluster nodes with the same political orientation as possible.

1. (5 points) Use spectral clustering to find the $k = 2, 5, 10, 30, 50$ clusters in the network of political blogs (each node is a blog, and their edges are defined in the file edges.txt). Find majority labels in each cluster for different k values, respectively. For example, if there are $k = 2$ clusters, and their labels are $\{0, 1, 1, 1\}$ and $\{0, 0, 1\}$ then the majority label for the first cluster is 1 and for the second cluster is 0. **It is required you implement the algorithms yourself rather than calling from a package.**

   Now compare the majority label with the individual labels in each cluster, and report the *mismatch rate* for each cluster, when $k = 2, 5, 10, 30, 50$. For instance, in the example above, the mismatch rate for the first cluster is 1/4 (only the first node differs from the majority), and the second cluster is 1/3.

2. (5 points) Tune your k and find the number of clusters to achieve a reasonably small *mismatch rate*. Please explain how you tune k and what is the achieved mismatch rate. Please explain intuitively what this result tells about the network community structure.