

# ISYE 6740 Summer 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  $i$ th 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.
2. (5 points) Consider a dataset with data points each having 3 features, e.g.,  $x^1 = \{\text{"Atlanta"}, \text{"house"}, 500k\}$ , and  $x^2 = \{\text{"San Francisco"}, \text{"house"}, 300k\}$ . 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 categorical 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.)
3. (5 points) Show that the clustering assignment problem

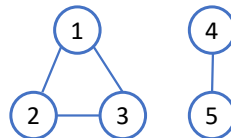
$$\pi(i) = \arg \min_{j=1, \dots, k} \|x^i - c^j\|^2$$

is equivalent to solving

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

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

4. (5 points) Why do different initializations for k-means lead to different results?
5. (5 points) Why  $k$ -means will (be guaranteed to) stop after a finite number of iterations?
6. (5 points) Consider the following simple graph



Write down the graph Laplacian matrix and find the eigenvectors associated with the zero eigenvalues. If using code to find the eigenvectors, you still must write out your work and the Laplacian matrix. Explain how you find out the number of disconnected clusters in the graph and identify these disconnected clusters using these eigenvectors.

## 2 Math of k-means clustering [25 points]

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

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

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

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

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

That is,  $\mu^j$  is the center of  $j$ -th cluster.

Hint: You may start by taking the partial derivative of  $J$  with respect to  $\mu^j$ , with  $r^{ij}$  fixed.

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

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

Derive what  $\mu^j$  and  $r^{ij}$  becomes in this case.

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

1. (10 points) Use  $k$ -means with squared- $\ell_2$  norm as a metric for `Glockenbronze.png` (a micrograph of bronze revealing a cast dendritic structure) and `football.bmp` and also choose a third picture of your own to work on. We recommend the size of  $320 \times 240$  or smaller. Run your  $k$ -means implementation with these pictures, with these specific several different  $k$  values  $k = 2, 5, 10, 20, 30$ . You must report on each of the three images for all  $k$  values in your final answer.

*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- $\ell_2$  norm) with random initialization centroids. Please experiment with multiple random initializations and report the best one for each  $k$  (in terms of image quality).

2. (5 points) Please report how long it takes to converge for each  $k$  (report the number of iterations and elapsed time in seconds for every  $k$  and every Image).
3. (5 points) Describe a method to find the best  $k$ . What is your best  $k$ ?

## Note

- Your code must run across all  $k$  values and all images in less than 10 minutes total. Hint: Vectorization will make this requirement trivial.
- 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. Think about what this means in the context of the image compression problem.
- Setting a random seed will ensure reproducibility and is good best practice.
- Ensure that you are using an appropriate stopping criteria in accordance with the  $k$ -means algorithm presented in the lecture. Placing a cap on iterations as your criteria is not valid.
- 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- $\ell_2$  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.
2. (10 points) Now try your  $k$ -means with the Manhattan distance (or  $\ell_1$  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?

## 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 (Same as purity score from the image compression problem) 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* (Also known as *misclassification 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.

#### Note

- Do not calculate your overall mismatch rate by averaging the mismatch rate of different clusters. Overall mismatch rate should be calculated by

$$\frac{\sum(\text{Elements Mismatched in Each Cluster})}{\sum(\text{Total Elements across all clusters})}$$