50.007 Machine Learning, Spring 2023
Homework 2

Due 25 Feb 2023, 11:59 pm

1. K-Means [30 points]

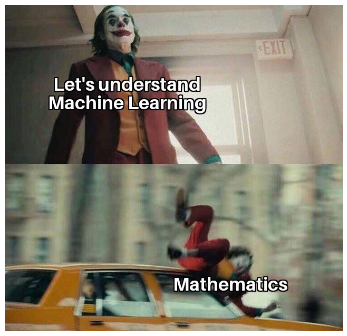
Consider the image "hw2_img.jpg". The image has 675 rows and 1200 columns which can be represented as a 3-d array with one axis for the height, width and colour channels. We can rearrange this 3-d array into a 2-d array of shape (810000, 3), where each row represents one pixel in the image and each column represents one of the three colour channels. In this assignment, we will explore clustering methods, applying them in particular to the problem of dividing the pixels of the image into a small number of similar clusters. Consider the K-means clustering algorithm, as described in class. In particular, consider a version in which the inputs to the algorithm are:

- The set of data to be clustered. (i.e., the vectors $x^{(1)}, x^{(2)}, x^{(3)}, \ldots$)
- The desired number of clusters, K.
- Initial centroids for the K clusters.

Then the algorithm proceeds by alternating: (1) assigning each instance to the class with the nearest centroid, and (2) recomputing the centroids of each class—until the assignments and centroids stop changing. Please use squared Euclidean distance (Lecture 5, Eq. 2) as the metric for clustering.

There are many implementations of K-means publicly available. However, please implement K-Means on your own. Then, use your implementation to cluster the data in the file mentioned above ("hw2_img.jpg"), using K = 8, and the initial centroids as given below in the table:

R	G	B
255	255	255
255	0	0
128	0	0
0	255	0
0	128	0
0	0	255
0	0	128
0	0	0



You may use any image IO library to load and convert the image into an integer array. The Pillow package has good compatibility with NumPy so you may choose to use that.

Turn in your code, as well as a report on all of the following:

(a) How many clusters there are in the end. (A cluster can "disappear" in one iteration of the algorithm if no vectors are closest to its centroid.)

(b) The final centroids of each cluster.

(c) The number of pixels associated to each cluster.

(d) Plot the sum of squared Euclidean distance of each pixel to the nearest centroid (Lecture 5, Eq. 8) against the iteration number of the algorithm.

Visualize your result by replacing each pixel with the centroid to which it is closest, and displaying the resulting image.

2. K-Mediods [10 points]

In clustering, Euclidean distance is not the only way to measure the distance between two points/vectors. l_p norms is a family of distance measures that are parameterized by $p \geq 1$. The l_p norm of a vector is:

$$\|x\|_p = \left(\sum_j |x_j|^p \right)^{\frac{1}{p}}.$$

Euclidean distance is the l_2 norm of the vector difference between two points, i.e.,

$$\|x - y\|_2 = \left(\sum_j |x_j - y_j|^2 \right)^{\frac{1}{2}}.$$

The Manhattan distance is the l_1 norm of the vector difference between two points, i.e.,

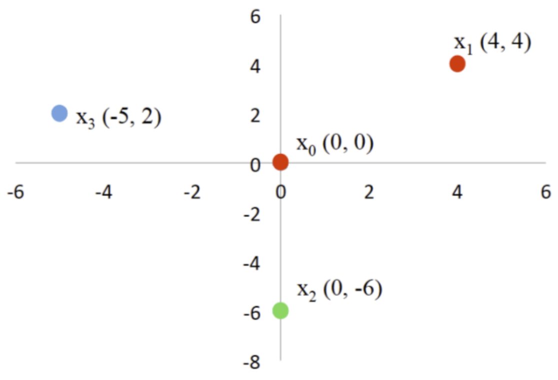
$$\|x - y\|_1 = \sum_j |x_j - y_j|.$$

The l_∞ distance is the maximum absolute element in the vector difference between two points, i.e.,

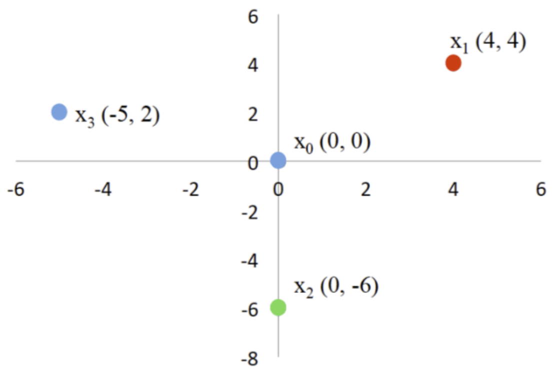
$$\|x - y\|_\infty = \max_j |x_j - y_j|.$$

The following figures (points in the same cluster have the same color) are produced by the k-medoids algorithm for $k = 3$ clusters using l_1 , l_2 , and l_∞ distance measures. Indicate which distance measure is used for each figure.

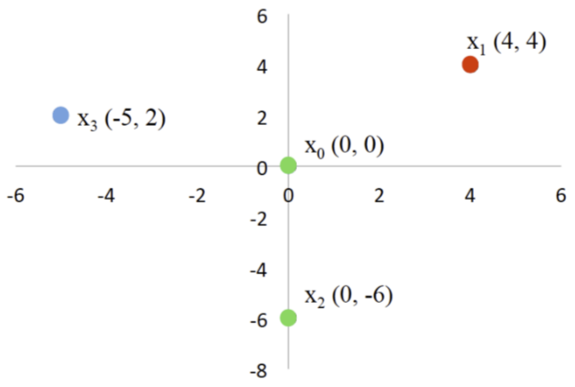
A.



B.



C.



Question 3 [20 points] Download and install the widely used SVM implementation LIBSVM (`https://github.com/cjlin1/libsvm`, or `https://www.csie.ntu.edu.tw/~cjlin/libsvm/`; clicking on either link takes you to the webpage). We expect you to install the package on your own – this is part of learning how to use off-the-shelf machine learning software. Read the documentation to understand how to use it.

Download `promoters` folder. In that folder are `training.txt` and `test.txt`, which respectively contain 74 training examples and 32 test examples in LIBSVM format. The goal is to predict whether a certain DNA sequence is a promoter[1] or not based on 57 attributes about the sequence (this is a binary classification task).

Run LIBSVM to classify promoters with different kernels (0-3), using default values for all other parameters. What is your test accuracy for each kernel choice?

[1]A promoter is a region of DNA that facilitates the transcription of a particular gene. The ability to predict promoters is of practical importance in searching for new promoter sequences.

Question 4 [20 points] Suppose we are looking for a maximum margin linear classifier through the origin, i.e., the bias $b = 0$. This means that we have to minimize

$$\frac{1}{2}\|\mathbf{w}\|^2 \text{ subject to } y^t\mathbf{w}\cdot\mathbf{x}^t \geq 1, t = 1, ..., n.$$

(a) [15 points] Suppose there are two training examples $\mathbf{x}^{(1)} = (1,1)^T$ and $\mathbf{x}^{(2)} = (1,0)^T$ with labels $y^{(1)} = 1$ and $y^{(2)} = -1$. What is the $\mathbf{w}$ in this case, and what is the margin γ?

(b) [15 points] How will the parameters $\mathbf{w}$ and the margin γ change in the previous question if the bias/offset parameter b is allowed to be non-zero?

Question 5 [20 points] In this problem, we consider constructing new kernels by combining existing kernels. Recall that for some function $K(\mathbf{x}, \mathbf{z})$ to be a kernel, we need to be able to write it as an inner product of vectors from some high-dimensional feature space:

$$K(\mathbf{x}, \mathbf{z}) = \phi(\mathbf{x})^T\phi(\mathbf{z})$$

Mercer's theorem gives a necessary and sufficient condition for a function K to be a kernel: its corresponding kernel matrix has to be symmetric and positive semidefinite, where the elements of a kernel matrix are inner products between all pairs of examples.
Suppose that $K_1(\mathbf{x}, \mathbf{z})$ and $K_2(\mathbf{x}, \mathbf{z})$ are kernels over $\mathcal{R}^n \text{x} \mathcal{R}^n$. For each of the cases below, state whether K is also a kernel. If it is, prove it. If it is not, give a counter example. (*Hints: You can use either Mercer's theorem or the definition of a kernel, as needed.*).

1. $K(\mathbf{x}, \mathbf{z}) = K_1(\mathbf{x}, \mathbf{z})K_2(\mathbf{x}, \mathbf{z})$
2. $K(\mathbf{x}, \mathbf{z}) = aK_1(\mathbf{x}, \mathbf{z}) - bK_2(\mathbf{x}, \mathbf{z})$, where $a, b > 0$ are real numbers
3. $K(\mathbf{x}, \mathbf{z}) = tanh(\alpha K_1(\mathbf{x}, \mathbf{z}) + \mathbf{C})$, where $\alpha, \mathbf{C} > 0$ are real numbers
4. $K(\mathbf{x}, \mathbf{z}) = f(\mathbf{x})f(\mathbf{z})$, where $f : \mathcal{R}^n \rightarrow \mathcal{R}$ be any real valued function of x.