

## Homework 5

*Release Date: April 15, 2025**Due Date: April 29, 2025*

- HW5 will count for 10% of the grade. This grade will be split between the written (30 points) and programming (30 points) parts.
- All written homework solutions are required to be formatted using  $\text{\LaTeX}$ . Please use the template [here](#). Do not modify the template. **This** is a good resource to get yourself more familiar with  $\text{\LaTeX}$ , if you are still not comfortable.
- You will submit your solution for the written part of HW5 as a single PDF file via Gradescope. The deadline is **11:59 PM ET**. Contact TAs on Ed if you face any issues uploading your homeworks.
- Collaboration is permitted and encouraged for this homework, though each student must understand, write, and hand in their own submission. In particular, it is acceptable for students to discuss problems with each other; it is not acceptable for students to look at another student's written Solutions when writing their own. It is also not acceptable to publicly post your (partial) solution on Ed, but you are encouraged to ask public questions on Ed. If you choose to collaborate, you must indicate on each homework with whom you collaborated.
- Please refer to the notes and slides posted on the website if you need to recall the material discussed in the lectures.

# 1 Written Questions (30 points)

## Problem 1: $k$ -means Clustering (16 points)

Recall the  $k$ -means clustering problem with data  $x_1, \dots, x_m$ . Our goal was to find  $k$  clusters denoted by  $C_1, \dots, C_k \subseteq [m]$  such that  $\cup_{i=1}^k C_i = [m]$  (cover all data points) and for all  $i \neq j$ ,  $C_i \cap C_j = \emptyset$  (and are disjoint).

**1.1 (5 points)** To measure the “goodness” of the cluster we defined

$$Z(C_1, \dots, C_k) = \sum_{l=1}^k \frac{1}{2|C_l|} \sum_{i,j \in C_l} \|x_i - x_j\|_2^2.$$

Show that we can equivalently express  $Z$  as

$$Z(C_1, \dots, C_k) = \sum_{l=1}^k \sum_{i \in C_l} \|x_i - \mu_l\|_2^2$$

where  $\mu_l = \frac{1}{|C_l|} \sum_{i \in C_l} x_i$  is the centroid of cluster  $l$ .

**1.2 (3 points)** In Lloyd’s algorithm, for every cluster  $C$ , we choose the cluster center to be the centroid  $\mu$  where  $\mu = \frac{1}{|C|} \sum_{i \in C} x_i$ . However, there may be other ways of choosing the cluster centers for  $k$ -means, for a cluster  $C$ . Let  $Z(C, z) = \sum_{i \in C} \|x_i - z\|_2^2$  be the “goodness” of cluster  $C$  with center  $z$ . Prove that  $\mu$  is the optimal center for the cluster  $C$ , that is,

$$Z(C, \mu) = \min_z Z(C, z).$$

**1.3 (3 points)** Suppose we pick the center of the cluster  $z$  uniformly randomly from the points in cluster  $C$ , such that,  $\Pr[z = x_i] = \frac{1}{|C|}$  for each  $i \in C$ . Let us denote this distribution over centers as  $\rho$ . Show that we do not lose too much by choosing a random cluster center according to  $\rho$  (similar to what  $k$ -means++ does) compared to choosing the centroid. In particular, prove that

$$\mathbb{E}_{z \sim \rho}[Z(C, z)] = 2Z(C, \mu).$$

**1.4 (5 points)** Show that the EM algorithm to solve Gaussian Mixture Models reduces to  $k$ -means if we set  $\Sigma_l = \sigma^2 I$  for all  $l \in [k]$  and take the limit  $\sigma \rightarrow 0$ .

**Optional (zero points)** The Lloyd’s algorithm we discussed in class does not always find the optimal  $k$ -means solution, and it is hard in general to find the optimal solution efficiently. However, when the input is in 1-dimensions, the problem can be solved optimally and efficiently. Design an  $O(km^2)$  dynamic programming algorithm for solving the  $k$ -means problem in single dimension.

*Hint: If we sort the data points  $x_1 \leq x_2 \leq \dots \leq x_m$  in increasing order, then the optimal clusters correspond to intervals of the points, that is, cluster contains all points between some index  $i$  and  $j$ ,  $\{x_i, x_{i+1}, \dots, x_j\}$ .*

## Problem 2: PCA (14 points)

Consider data points  $x_1, \dots, x_m$  such that each feature has been normalized to have mean 0 and variance 1 (as discussed in class). One way to look at PCA is to take the maximizing variance point of view, as we did in class, that is, by finding  $u$  such that it maximizes the following objective:

$$\max_{u: \|u\|_2=1} \frac{1}{m-1} \sum_{i=1}^m (x_i^\top u)^2.$$

**2.1 (4 points)** Let  $S = \frac{1}{m-1} \sum_{i=1}^m x_i x_i^\top$  be the sample covariance (notice the Bessel's correction). Recall from the lecture notes that the first principal component,  $u_1$ , is the eigenvector of  $S$  corresponding to the largest eigenvalue  $\lambda_1$ . The second principal component,  $u_2$ , maximizes the variance subject to being orthogonal to  $u_1$ . Show that  $u_2$  is the eigenvector of  $S$  corresponding to the second largest eigenvalue,  $\lambda_2$  (assume  $\lambda_1 > \lambda_2$ ), and that the maximum variance achieved is  $\lambda_2$ , that is,

$$\max_{u: \|u\|_2=1, u^\top u_1=0} \frac{1}{m-1} \sum_{i=1}^m (x_i^\top u)^2 = \lambda_2.$$

*Hint: Use the eigenvalue decomposition for  $S$  and the constraints on  $u$ .*

**2.2 (4 points)** Consider another alternate view of finding a useful direction  $u$  by instead minimizing the reconstruction error

$$\min_{u: \|u\|_2=1} \frac{1}{m} \sum_{i=1}^m \|x_i - (u^\top x_i)u\|_2^2.$$

Show that this is actually equivalent to maximizing variance, that is

$$\arg \max_{u: \|u\|_2=1} \frac{1}{m-1} \sum_{i=1}^m (x_i^\top u)^2 = \arg \min_{u: \|u\|_2=1} \frac{1}{m} \sum_{i=1}^m \|x_i - (u^\top x_i)u\|_2^2.$$

**2.3 (6 points)** Let us generalize this to  $k$  useful directions. In particular, show that the following two optimizations are equivalent:

$$\underbrace{\arg \max_{U \in \mathbb{R}^{k \times d}: UU^\top = I} \frac{1}{m-1} \sum_{i=1}^m \|U x_i\|_2^2}_{\text{maximizing variance}} = \underbrace{\arg \min_{U \in \mathbb{R}^{k \times d}: UU^\top = I} \frac{1}{m} \sum_{i=1}^m \|x_i - U^\top U x_i\|_2^2}_{\text{minimizing reconstruction error}}.$$

## 2 Programming Questions (30 points)

Use the link [here](#) to access the Google Colaboratory (Colab) file for this homework. Be sure to make a copy by going to “File”, and “Save a copy in Drive”. As with the previous homeworks, this assignment uses the PennGrader system for students to receive immediate feedback. As noted on the notebook, please be sure to change the student ID from the default ‘99999999’ to your 8-digit PennID.

Instructions for how to submit the programming component of HW 5 to Gradescope are included in the Colab notebook. You may find this [PyTorch linear algebra reference](#) and this [general PyTorch reference](#) to be helpful in perusing the documentation and finding useful functions for your implementation.