Problem 1

In this problem, you will explore the differences between low-rank approximation (PCA) and least-squares approximation.

(a) First consider one-dimensional data lying with the affine relationship $y = ax + b$. Using the script `prob1a.m`, fit the given data in `xx,yy` using a least-squares estimate for $a, b$ and plot the resulting line. Next, fit the same data using a rank-1 approximation, as described on Pg. 6.12-6.13 of the notes. **Turn in** the resulting plot.

**Hint:** To obtain a correct low-rank approximation, you must first center the data by removing its mean. This can be done efficiently using the `mean` and `repmat` functions. Don’t forget to add the mean back when you go to plot.

(b) Next, we’ll fit a two-dimensional plane using each method. Complete the least-squares and low-rank approximations in `prob1b.m`. Although it is difficult to see, the resulting planes are different. The takeaway here is that we aren’t limited to fitting one-dimensional data! **Turn in** the resulting plot.

**Note:** There is no offset for this part, so no need to center.

Problem 2

In this problem, we’ll implement a simple algorithm for subspace clustering called $K$-subspaces (KSS). This is an unsupervised version of the nearest-subspace classifier we implemented in class. The algorithm is similar in spirit to the $K$-means clustering algorithm, in that it alternates between cluster assignment and estimating the cluster archetypes, with the key difference being that the cluster archetypes are subspaces instead of points.

Recall that the union of subspaces (UoS) model says that we have $N$ points lying on a union of $K$ subspaces of dimension $r$. However, we don’t know which points lie on which subspace, so we need to determine both the subspace assignments and the subspaces themselves. To do this, we’ll take an alternating approach. First, initialize with $K$ random subspaces. Now alternate between the following two steps until convergence (which is guaranteed)

1. Estimate labels based on nearest subspace (as in class demo).
2. Estimate subspaces using PCA.

The pseudocode for the KSS algorithm is given in Algorithm 1.

(a) Implement the KSS algorithm for subspace clustering in the provided `KSS.m` file. Code for generating random subspaces is provided on line 13 of `prob2.m`. Test your implementation on the provided synthetic data using the provided `prob2.m` script for $r \in \{1, 5, 10, 49\}$. **Turn in** a plot of clustering error vs. $r$. 

1
Algorithm 1 \(K\)-subspaces (KSS) algorithm for subspace clustering

1: **Input:** \(X = \{x_1, x_2, \ldots, x_N\} \subset \mathbb{R}^D\): data, \(K\): number of clusters/subspaces, \(r\): rank of subspaces, \(\text{maxIter}\): maximum number of iterations
2: **Output:** \(y \in \{1, \ldots, K\}^N\): vector of labels for points in \(X\)
3: \(U_1, \ldots, U_K \overset{iid}{\sim} \text{Unif}(\text{St}(D, r))\) \(\quad\) Draw \(K\) random subspace bases
4: \(y_{\text{curr}}^i \leftarrow \arg \min_{k \in \{1, \ldots, K\}} \|x_i - U_k U_k^T x_i\| \quad \forall i \in \{1, \ldots, N\}\) \(\quad\) Cluster by nearest subspace
5: initialize \(y_{\text{prev}}\) randomly
6: **while** \(y_{\text{prev}} \neq y_{\text{curr}}\) and \(\text{numIter} < \text{maxIter}\) **do**
7: \(y_{\text{prev}} \leftarrow y_{\text{curr}}\)
8: \(U_k \leftarrow \text{PCA}\left\{x_i \in X : y_{\text{curr}}^i = k\right\}, r\) \(\quad\) Estimate subspaces
9: \(y_{\text{curr}}^i \leftarrow \arg \min_{k \in \{1, \ldots, K\}} \|x_i - U_k U_k^T x_i\| \quad \forall i \in \{1, \ldots, N\}\) \(\quad\) Cluster by nearest subspace
10: **end while**

(b) For \(r = 5\), plot (using `imagesc`) the Gram matrix for \(X\) and turn in the resulting plot. What structure do you see? This is due to the fact that points lying on a low-rank subspace tend to have large inner product relative to points not on low-dimensional subspaces. This fact is the basis for the Thresholded Subspace Clustering algorithm as well as Coherence Pursuit (project papers).

(c) **Optional:** Test your algorithm on the handwritten digits dataset given in class.

Problem 3

In this problem, you will implement the (classical) Multidimensional Scaling (MDS) algorithm for embedding points into a low-dimensional space given only a notion of distance between points. We derived this algorithm beginning on Pg. 6.31 of the notes. In that case, we considered only Euclidean distances, but in practice we could use any notion of dissimilarity. In this problem, we’ll use the cosine similarity, i.e., inner products, to get a notion of distance between points on a union of subspaces. To convert this to a distance, we define

\[
d(x_i, x_j) = 1 - \frac{x_i \cdot x_j}{\|x_i\| \|x_j\|},
\]

which is between 0 and 1 due to the normalization of points.

(a) Implement the MDS algorithm derived in class by completing the `MDS.m` function included. To test your algorithm, we’ll consider again the MNIST digits used in the class demo. Run your `MDS.m` on the `mnistSubset.mat` data using only digits 1 and 2. Turn in a plot of the data embedded in three-dimensional space. Plot the points from each class in a different color (easily done using `hold all`).

(b) Now we’ll compare to the embedding we get by using PCA for dimensionality reduction. Let \(X\) be the matrix whose columns are the vectors from classes 1 and 2 only. Using PCA as described in Section 6.2 of the notes, embed these points into three-dimensional space. Turn in a plot of the embedded data, again using a different color for each of the two classes. How does this compare to what you get from MDS? Note that PCA operates on the data, while MDS operates only on the pairwise distances.
Tool Belt

Recall that the goal of the “tool belt” is to identify the properties, identities, and tricks that you use most often when solving matrix problems. Ideally, these will be of use to you when solving machine learning problems outside of this class.

Please list 0-2 additional tools for your tool belt here based on what you learned in class, from the lecture notes, or from this assignment. Based on what you’ve learned since the beginning of class, is there anything you would remove from previous weeks?