B01538234 # Homework 11 CS1420 – Spring 2021 April 11, 2021

Problem 1

a. A halfspace model would classify this data well. This is because it is linearly seperable in 2D, and halfspaces can correctly classify linearly seperable sets of data.

b. If the principal component here (red) was used to reduce the data to a single dimension, a halfspace model would not be able to classify it well. This is because the halfspace that correctly classifies this data (blue) is roughly parallel to the first principal component of the data, so after the data is reduced to that single dimension, the data would not be linearly sepearable anymore.

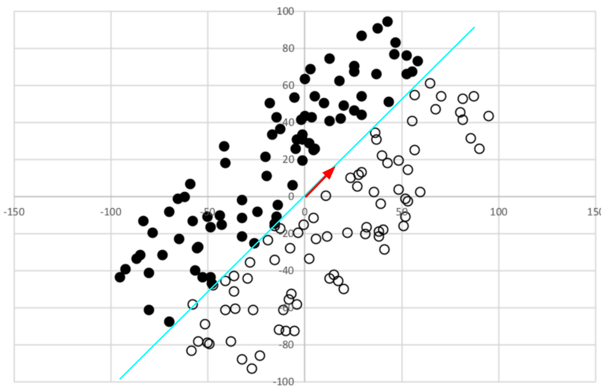

Figure 1: Data with red first Princpal Component and blue Halfspace Classifier

c. It is possible to project the data into a one dimensional linear subspace (purple) that is still linearly sepearable. The space has to be orthogonal to the halfspace classifier (blue) as shown in the figure so that when the data is reduced none of the dimensionality that separates the data is lost.

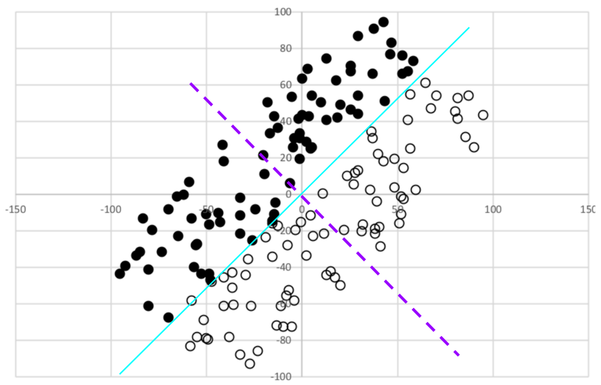

Figure 2: The blue halfspace will still classify the data well after being projected into the purple linear subspace

d. PCA doesn't consider the labels of the data when identifying the principal components. This means it is possible to have it choose linear subspaces that eliminates information vital for classification. If applied naively, PCA could remove components of the data that are actually very important when classifying.

Report

- The plot does match my expectations. The numbers are a bit blurry, but there is a clear number defined in subplot, indicating that each of the centroids was optimized for a specific digit.

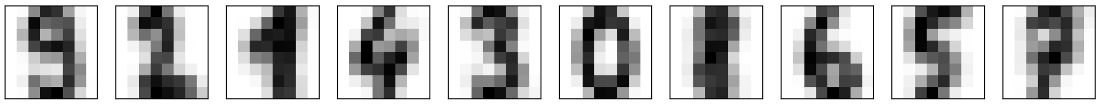

Figure 3: `plot_kmeans()` output. Shows numbers [9, 2, 1, 4, 3, 0, 8, 6, 5, 7]

- Many other non-Euclidian metric can be used for clustering algorithms. For example, Kernel-based methods can be used to transform the data from the input space to a feature space. Taking the norm of the difference between transformed data can capture different information about the features of the centroid. Different kernels can capture different types of feautres or feature mappings deemed relevant for the problem. Weighted Euclidian distances can also be used. This gives a strogner weight or bias to distances depending on certain features. This can be used to skew centroid in a certain perdetermined way.

- When $K < 10$, I expect most of the centroid to appear as they do when $K = 10$. This is because the centroid will still cluster around the digits. I would also expect one or two of these centroids to be fuzzy, or not as clear. This centroid will likely be the average of all the other $10 - K$ digits that the other centroids did not center on. When $K > 10$ I expect to see all 10 clearly defined centroids as when $K = 10$. The remainder I expect to be more fuzzy since they will not have a meaningful place in the input space left to center on. It is also possible that some of these 'extra' centroid will end up centering very close to the others, resulting in some repeat digits in the centroid images.