Assignment #4 - SOLUTIONS

CS181-S22

Due: 11:59pm March 25, 2022

Homework 4: SVM, Clustering, and Ethics

Introduction

This homework assignment will have you work with SVMs, clustering, and engage with the ethics lecture. We encourage you to read Chapters 5 and 6 of the course textbook.

Please submit the **writeup PDF to the Gradescope assignment 'HW4'**. Remember to assign pages for each question.

Please submit your **LaTeX file and code files to the Gradescope assignment 'HW4 - Supplemental'**.

Problem 1 (Fitting an SVM by hand, 10pts)

For this problem you will solve an SVM by hand, relying on principled rules and SVM properties. For making plots, however, you are allowed to use a computer or other graphical tools.

Consider a dataset with the following 7 data points each with $x \in \mathbb{R}$ and $y \in \{-1, +1\}$:

$$\{(x_i, y_i)\}_{i=1}^{7} = \{(-3, +1), (-2, +1), (-1, -1), (0, +1), (1, -1), (2, +1), (3, +1)\}$$

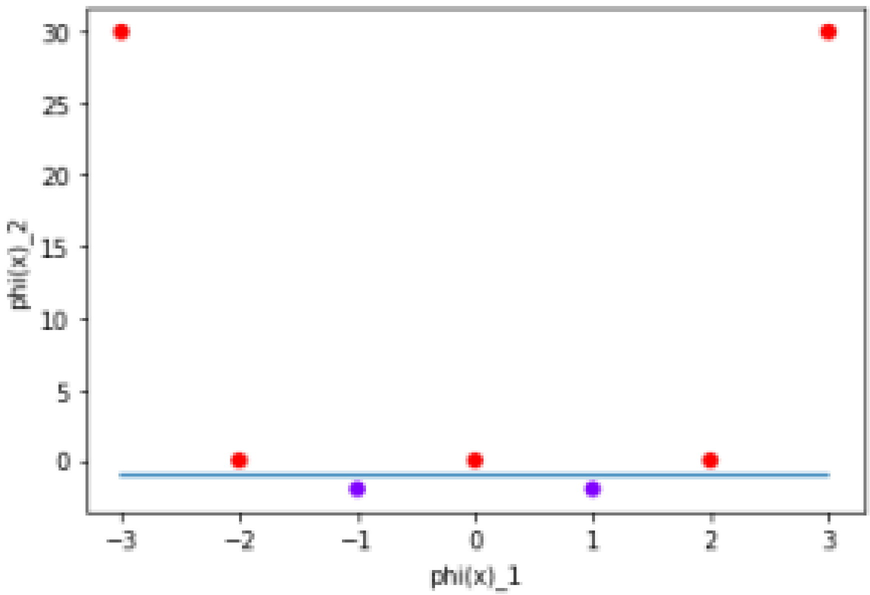
Consider mapping these points to 2 dimensions using the feature vector $\boldsymbol{\phi}(x) = (x, -\frac{8}{3}x^2 + \frac{2}{3}x^4)$. The hard margin classifier training problem is:

$$\begin{aligned} &\min_{\mathbf{w}, w_0} \frac{1}{2}\|\mathbf{w}\|_2^2 \\ \text{s.t.} \quad & y_i(\mathbf{w}^\top \boldsymbol{\phi}(x_i) + w_0) \geq 1, \ \forall i \in \{1, \ldots, n\} \end{aligned}$$

Make sure to follow the logical structure of the questions below when composing your answers, and to justify each step.

1. Plot the transformed training data in $\mathbb{R}^2$ and draw the optimal decision boundary of the max margin classifier. You can determine this by inspection (i.e. by hand, without actually doing any calculations).
2. What is the value of the margin achieved by the optimal decision boundary found in Part 1?
3. Identify a unit vector that is orthogonal to the decision boundary.
4. Considering the discriminant $h(\boldsymbol{\phi}(x); \mathbf{w}, w_0) = \mathbf{w}^\top \boldsymbol{\phi}(x) + w_0$, give an expression for *all possible* $(\mathbf{w}, w_0)$ that define the decision boundary. Justify your answer.
5. Consider now the training problem for this dataset. Using your answers so far, what particular solution to $\mathbf{w}$ will be optimal for the optimization problem?
6. What is the corresponding optimal value of w_0 for the $\mathbf{w}$ found in Part 5 (use your result from Part 4 as guidance)? Substitute in these optimal values and write out the discriminant function $h(\boldsymbol{\phi}(x); \mathbf{w}, w_0)$ in terms of the variable x .
7. Which points could possibly be support vectors of the classifier? Confirm that your solution in Part 6 makes the constraints above tight—that is, met with equality—for these candidate points.
8. Suppose that we had decided to use a different feature mapping $\boldsymbol{\phi}'(x) = (x, -4x^2 + \frac{1}{2}x^4)$. Does this feature mapping still admit a separable solution? How does its margin compare to the margin in the previous parts? Based on this, which set of features might you prefer and why?

Solution

1. 

2. The margin is the minimum, unsigned, normalized orthogonal distance from any correctly classified point to the decision boundary. In this case, we can consider the unit orthogonal vector $(0, 1)$, and the points $(-2, 0), (-1, -2), (0, 0), (1, -2)$ and $(2, 0)$ (in $\mathbb{R}^2$) are all distance 1 away from the decision boundary (which is defined by $\boldsymbol{\phi}(x)_2 = -1$).

3. A vector $\mathbf{w}$ that is perpendicular to the decision boundary is $[0\ 1]^\top$.

4. Using vector $\mathbf{w} = [0\ 1]^\top$, and setting discriminant $h(\boldsymbol{\phi}(x); \mathbf{w}, w_0) = \mathbf{w}^\top \boldsymbol{\phi}(x) + w_0 = \boldsymbol{\phi}(x)_2 + w_0 = 0$ for $\boldsymbol{\phi}(x)_2 = -1$, we need $w_0 = 1$. Now, recognizing that the decision boundary is invariant under scalar multiplication, the set of all possible $(\mathbf{w}, w_0)$ that define the optimal decision boundary is $([0\ \beta]^\top, \beta)$, with $\beta > 0$.

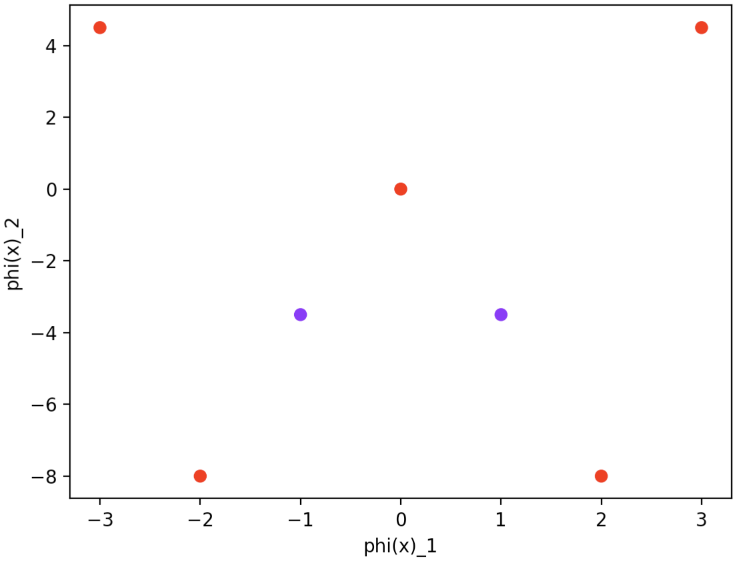
5. We want to find the solution that satisfies $y_i(\mathbf{w}^\top \boldsymbol{\phi}(x_i) + w_0) \geq 1$ and minimizes $||\mathbf{w}||_2^2$. For this, we set $y_i(\mathbf{w}^\top \boldsymbol{\phi}(x_i) + w_0) = 1$ on any support vector (these are the examples closest to the decision boundary.)

   Consider for example $x_2 = (-2, +1)$ with $\phi(x_2) = [-2\ 0]^\top$, and $y_2 = +1$. We need
   $$y_2(\mathbf{w}^\top \boldsymbol{\phi}(x_2) + w_0) = (1)([0\ \beta][-2\ 0]^\top + \beta) = 1,$$
   and thus $\beta = 1$. Therefore, the optimal solution to the training problem has $\mathbf{w} = [0\ 1]^\top$.

6. We know that $\beta = 1$, and thus from the general solution $([0\ \beta]^\top, \beta)$ we have $w_0 = 1$. The discriminant function is $h(\boldsymbol{\phi}(x); \mathbf{w}, w_0) = [0\ 1]^\top \boldsymbol{\phi}(x) + 1 = \boldsymbol{\phi}(x)_2 + 1 = -\frac{8}{3}x^2 + \frac{2}{3}x^4 + 1$.

7. The points that could be support vectors for this problem are those on the margin boundary (since this is a hard margin classifier), which in the original space are examples $(-2, +1), (-1, -1), (0, +1), (+1, -1), (2, +1)$. Plugging in each of these examples to the discriminant function, we find that $y_i(\mathbf{w}^\top \boldsymbol{\phi}(x_i) + w_0) = 1$ for each of these examples, and thus the constraints are binding. Note that not all of these points are necessarily support vectors; lying on the margin boundary is a necessary but not sufficient condition. Support vectors are those that have $a_n > 0$ in the solution to the dual optimization problem.

8. It is not linearly separable! Based on this, we would prefer the first feature set as it allows us to to use hard-margin SVM to find the decision boundary (the *purpose* of using kernel functions is to make our data more separable). Note that if students answered the previous version of the problem with the feature mapping $(x, -\frac{31}{12}x^2 + \frac{7}{12}x^4)$, they would have said the data is linearly separable.


4
2
0
−2
−4
−6
−8
phi(x)_2
−3
−2
−1
0
1
2
3
phi(x)_1

Problem 2 (K-Means and HAC, 20pts)

For this problem you will implement K-Means and HAC from scratch to cluster image data. You may use `numpy` but no third-party ML implementations (eg. `scikit-learn`).

We've provided you with a subset of the MNIST dataset, a collection of handwritten digits used as a benchmark for image recognition (learn more at http://yann.lecun.com/exdb/mnist/). MNIST is widely used in supervised learning, and modern algorithms do very well.

You have been given representations of MNIST images, each of which is a 784×1 greyscale handwritten digit from 0-9. Your job is to implement K-means and HAC on MNIST, and to test whether these relatively simple algorithms can cluster similar-looking images together.

The code in `T4_P2.py` loads the images into your environment into two arrays – `large_dataset`, a 5000x784 array, will be used for K-means, while `small_dataset`, a 300x784 array, will be used for HAC. In your code, you should use the ℓ_2 norm (i.e. Euclidean distance) as your distance metric.

Important: Remember to include all of your plots in your PDF submission!

Checking your algorithms: Instead of an Autograder file, we have provided a similar dataset, `P2_Autograder_Data`, and some visualizations, `HAC_visual` and `KMeans_visual`, for how K-means and HAC perform on this data. Run your K-means (with $K = 10$ and `np.random.seed(2)`) and HAC on this second dataset to confirm your answers against the provided visualizations. Do **not** submit the outputs generated from `P2_Autograder_Data`. Load this data with `data = np.load('P2_Autograder_Data.npy')`.

1. Starting at a random initialization and $K = 10$, plot the K-means objective function (the residual sum of squares) as a function of iterations and verify that it never increases.

2. For $K = 10$ and for 3 random restarts, print the mean image (aka the centroid) for each cluster. There should be 30 total images. Code that creates plots for parts 2, 3, and 4 can be found in `T4_P2.py`.

3. Repeat Part 2, but before running K-means, standardize or center the data such that each pixel has mean 0 and variance 1 (for any pixels with zero variance, simply divide by 1). For $K = 10$ and 3 random restarts, show the mean image (centroid) for each cluster. Again, present the 30 total images in a single plot. Compare to Part 2: How do the centroids visually differ? Why?

4. Implement HAC for min, max, and centroid-based linkages. Fit these models to the `small_dataset`. For each of these 3 linkage criteria, find the mean image for each cluster when using 10 clusters. Display these images (30 total) on a single plot.

   How do the "crispness" of the cluster means and the digits represented compare to mean images for k-means? Why do we only ask you to run HAC once?

   Important Note: For this part ONLY, you may use `scipy`'s `cdist` function to calculate Euclidean distances between every pair of points in two arrays.

5. For each of the HAC linkages, as well as one of the runs of your k-means, make a plot of "Number of images in cluster" (y-axis) v. "Cluster index" (x-axis) reflecting the assignments during the phase of the algorithm when there were $K = 10$ clusters.

   Intuitively, what do these plots tell you about the difference between the clusters produced by the max and min linkage criteria?

   Going back to the previous part: How does this help explain the crispness and blurriness of some of the clusters?

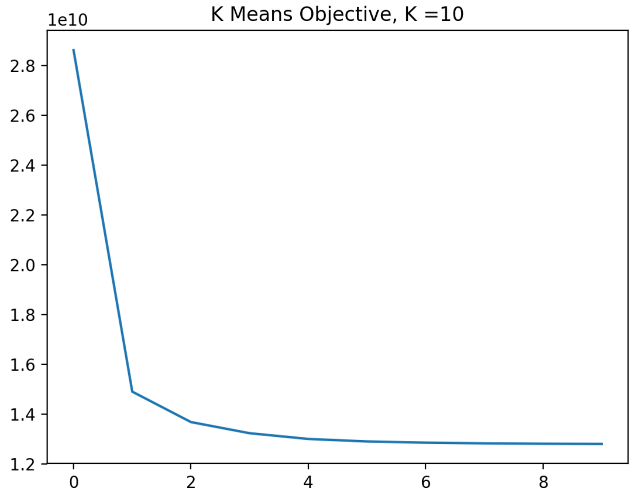
Problem 2 (cont.)

6. For your K-means with $K = 10$ model and HAC min/max/centroid models using 10 clusters on the `small_dataset` images, use the `seaborn` module's `heatmap` function to plot a confusion matrix between each pair of clustering methods. This will produce 6 matrices, one per pair of methods. The cell at the ith row, jth column of your confusion matrix is the number of times that an image with the cluster label j of one method has cluster i in the second method. Which HAC is closest to k-means? Why might that be?
7. Suppose instead of comparing the different clustering methods to each other, we had decided to compute confusions of each clustering method to the *true* digit labels (you do *not* have to actually compute this). Do you think how well the clustering match the true digits is reasonable evaluation metric for the clustering? Explain why or why not.

Solution

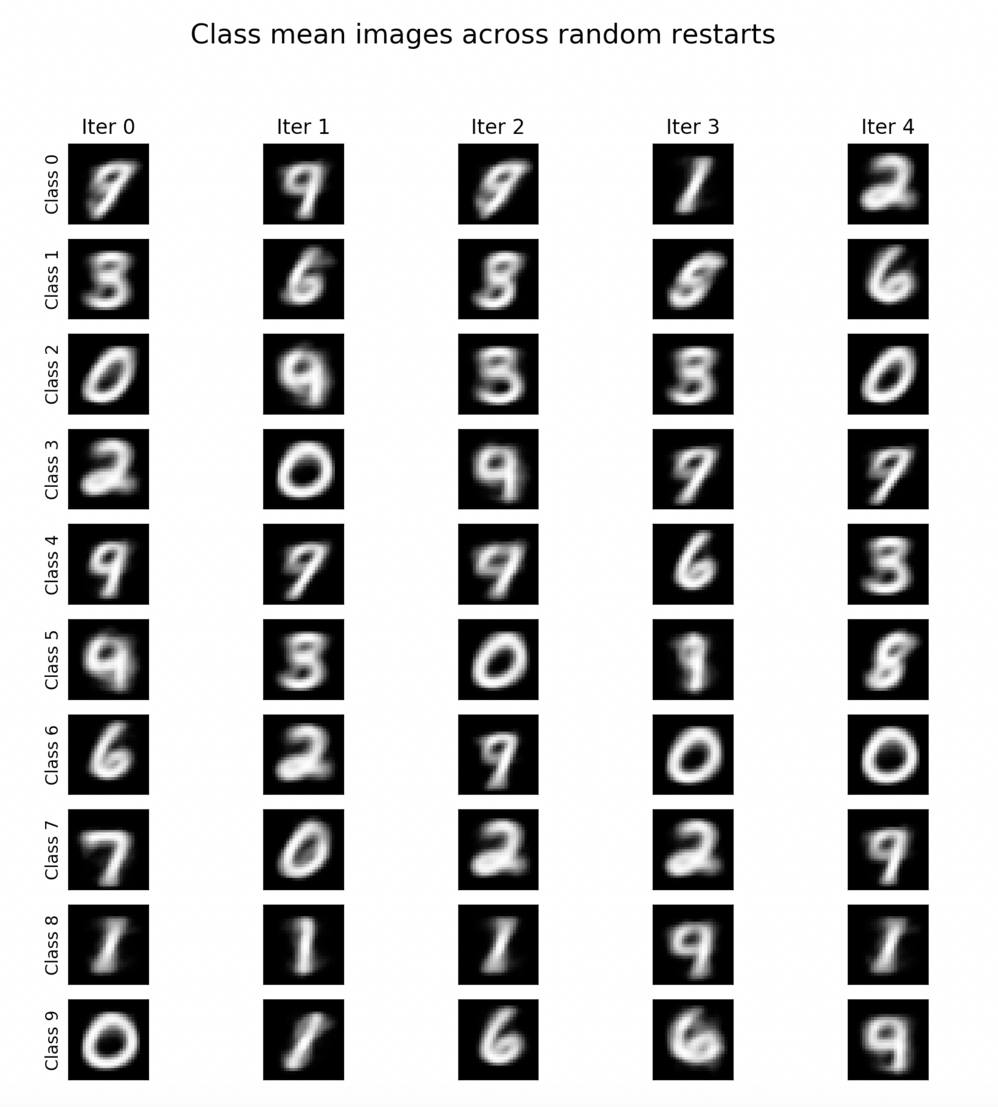
Part 1

You should have found that the K-means objective never increases, which you showed with a plot visualization.



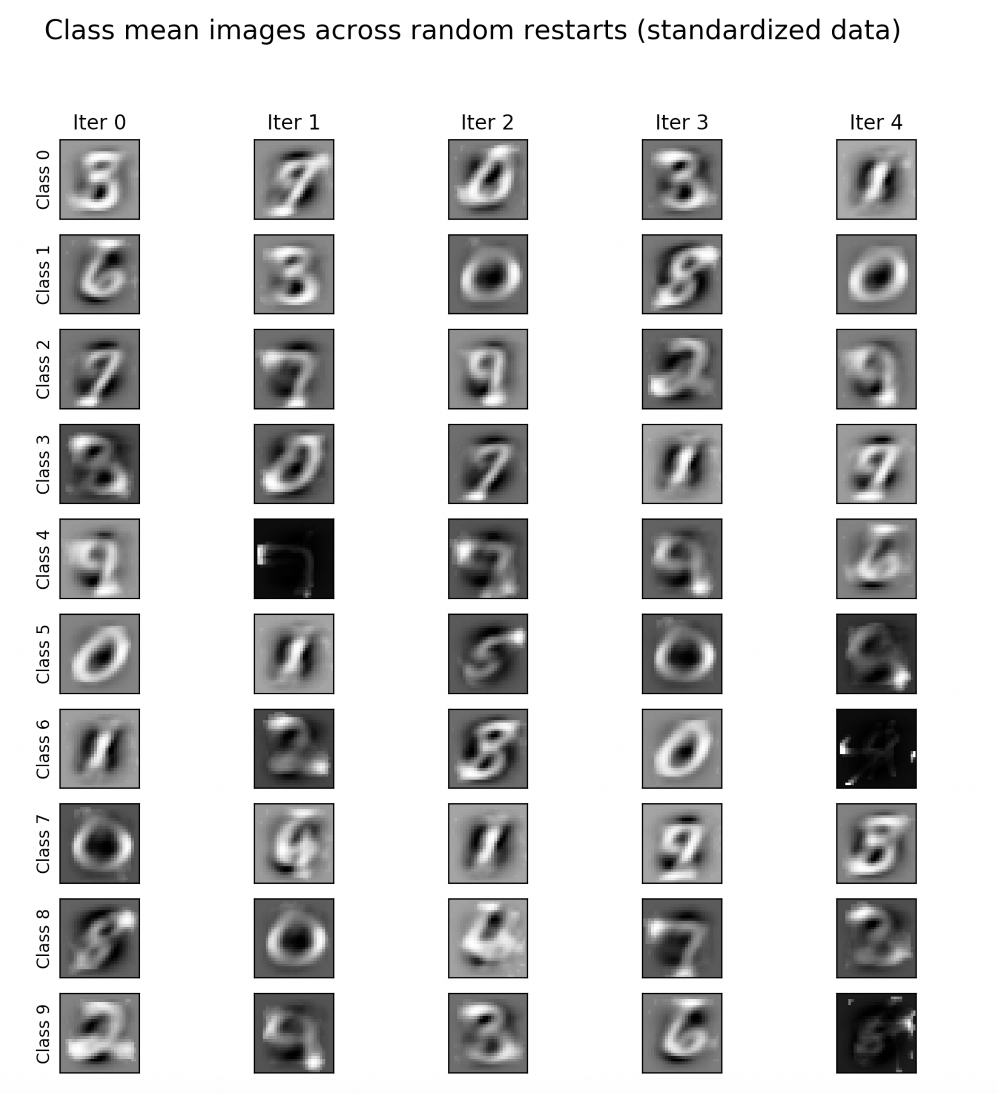
Part 2

You should find that with $K = 10$, you get close to a cluster for each digit! You should find that the mean images vary, but mostly just look like blobs with slight formations of various digits in them. If you reduce the value of K, you would likely have found that similar looking digits tend to merge together, as one would expect, and if you increase it, then the same digit can have two classes, perhaps because of two different handwriting styles.



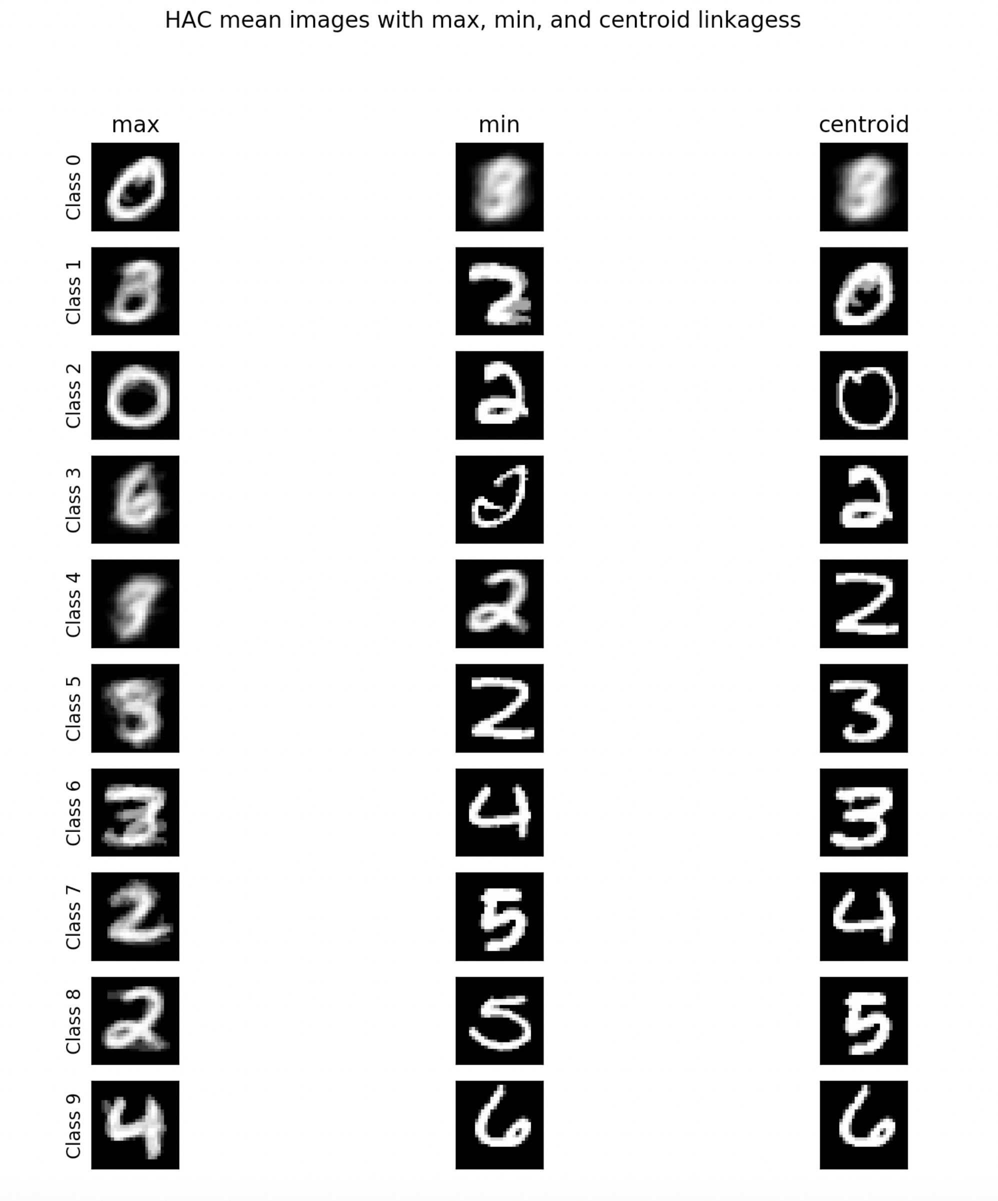
Part 3

By standardizing the data, you should observe that the model is able to associate digits with clusters but not as cleanly. You may notice that the backgrounds of images are more blended, that the images make more use of negative space, or that certain numbers (3 and 8) seem to be blending together.



Part 4

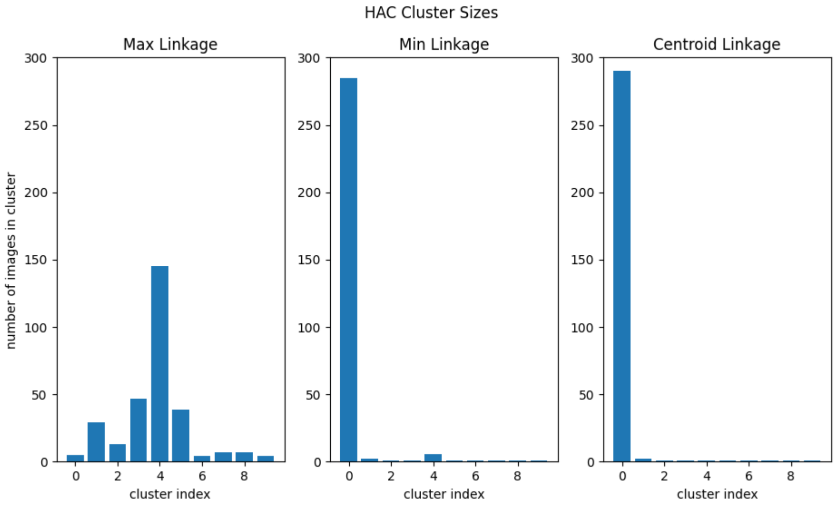
It appears that HAC performs worse than K-means in terms of clustering the images since we can see that the same digits are often being assigned to the same class. In terms of ”crispness”, the mean images in HAC do seem to be more ”crisp” than the blurry mean images in the K-means clustering algorithm. HAC is a deterministic algorithm and thus only needs to be run once (compared to the K-means algorithm which is non-deterministic).

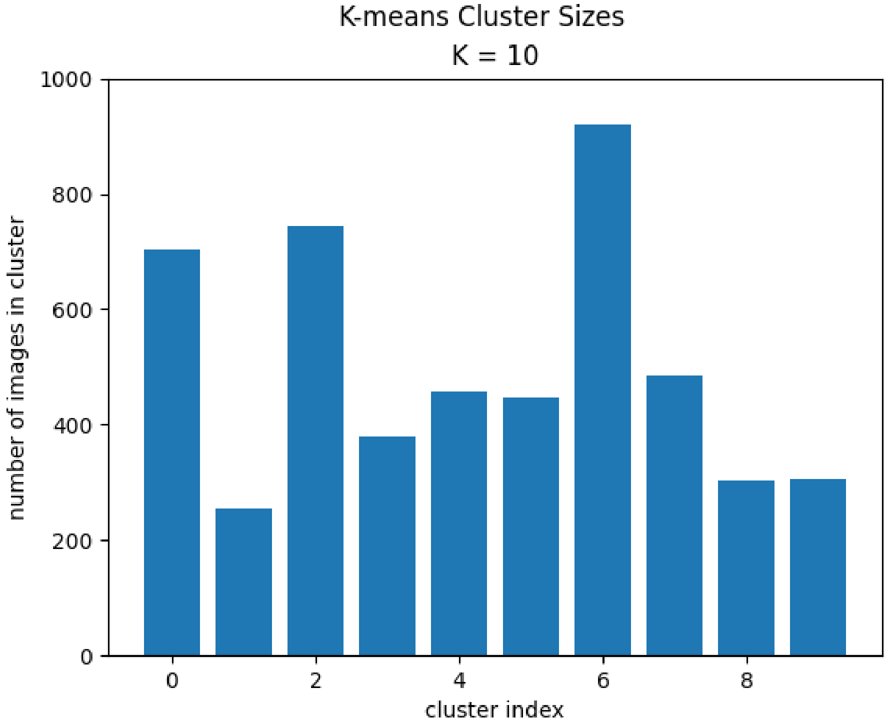


Part 5

The plot below shows that the min-linkage criterion puts nearly all the images in the same cluster, while the max-linkage criterion spreads out the images more evenly across the clusters. It makes sense that the min-linkage clusters will be more 'stringy' because we merge clusters based on the *minimum* distance between them, so we may end up merging clusters whose centroids are fairly far apart. On the other hand, the max-linkage criterion will account for the size/internal variance of each cluster by merging based on the maximum distance between clusters.

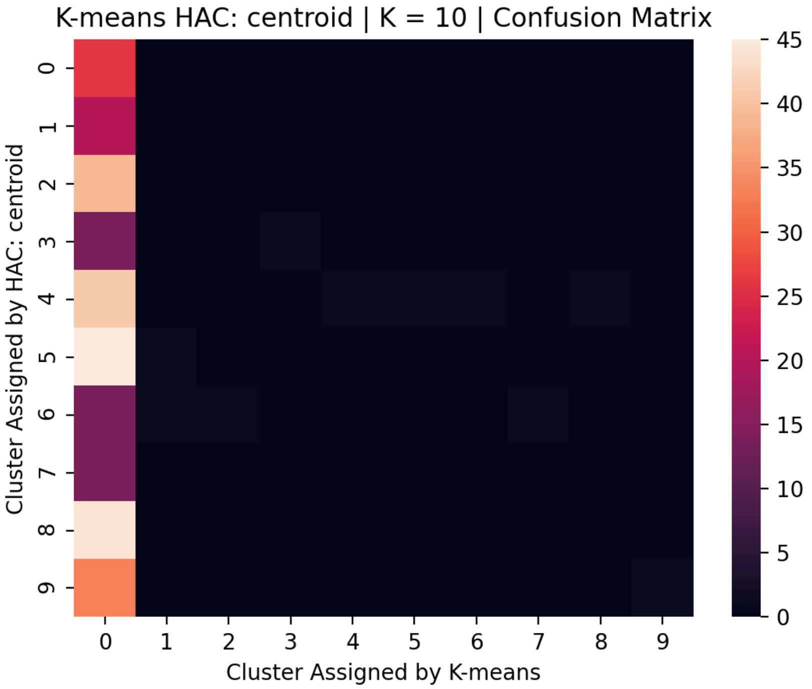
As described, the max-linkage criterion prefers compact clusters and usually does a better job of separating the digits into clusters than the min-linkage. Thus, the mean image formed for each cluster still has some semblance to a digit but is blurry because, as seen from the graphs, each cluster has a significant number of data points. On the other hand, the min-linkage prefers long, stringy clusters. The first cluster captures most of the data points as it starts forming a chain (most of the data points do lie close to each other). This large cluster results in the formation of an extremely blurry image as the cluster doesn't exclusively contain any one particular digit. But the rest of the clusters only have a few datapoints and thus the min-linkage clusters are formed between very similar data points giving rise to a very crisp image of digits.

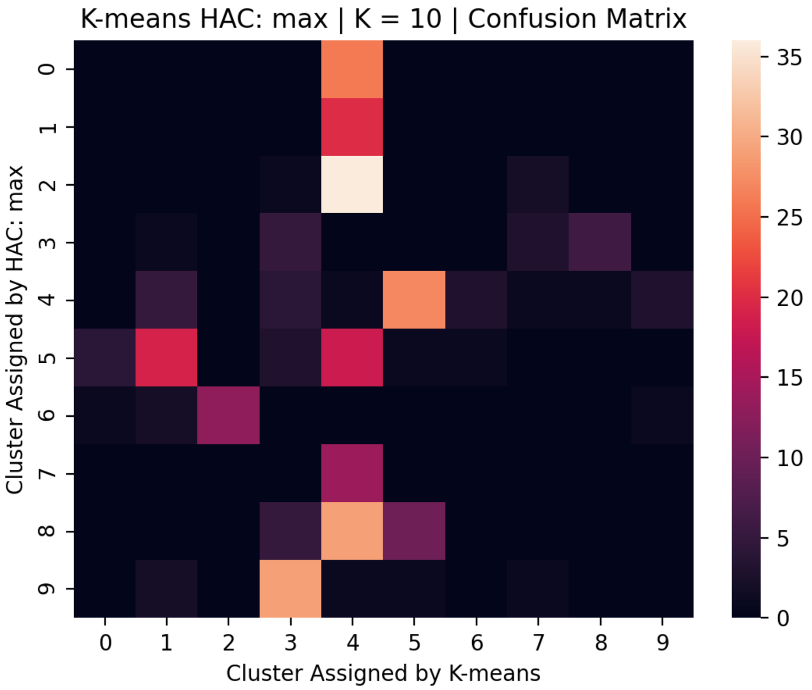


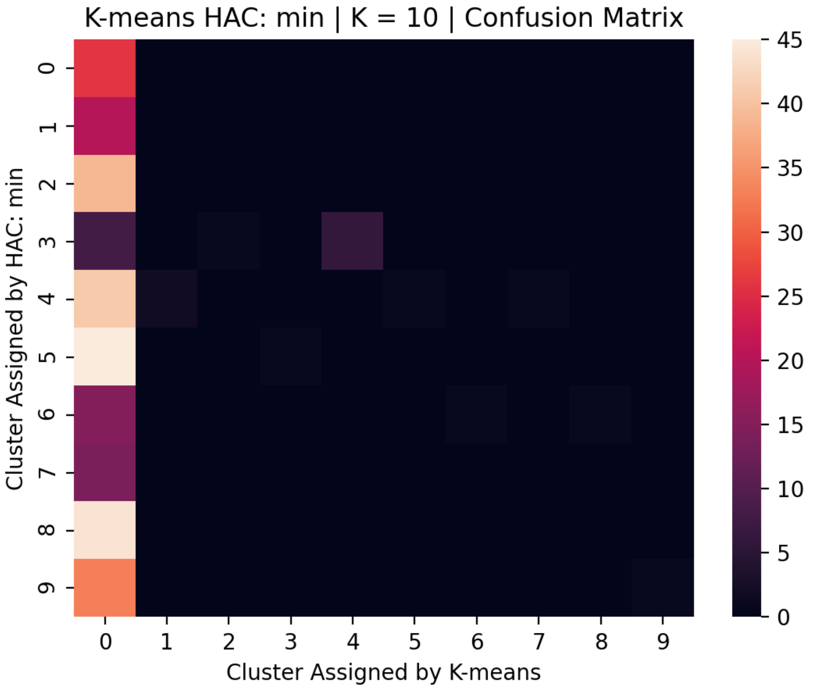


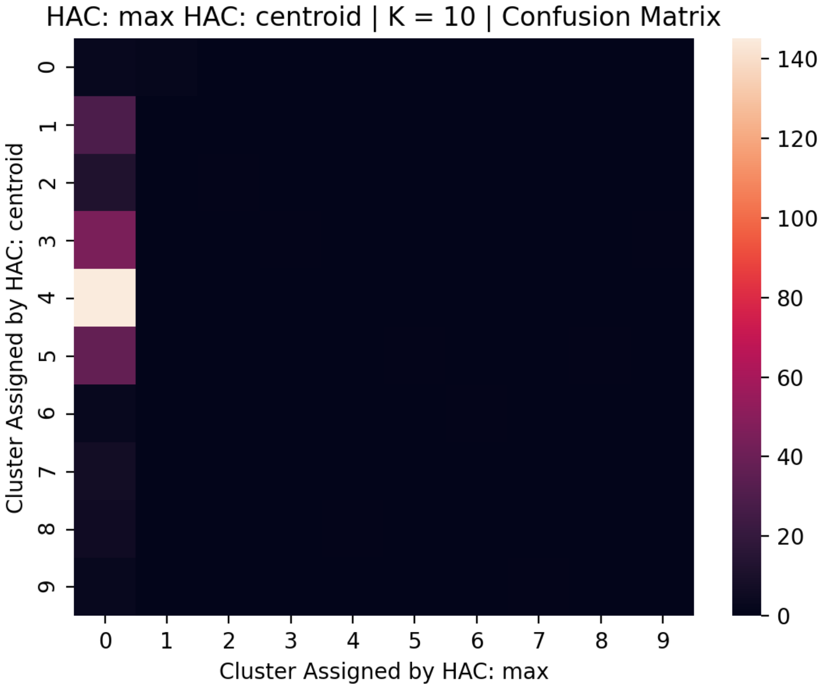
Part 6

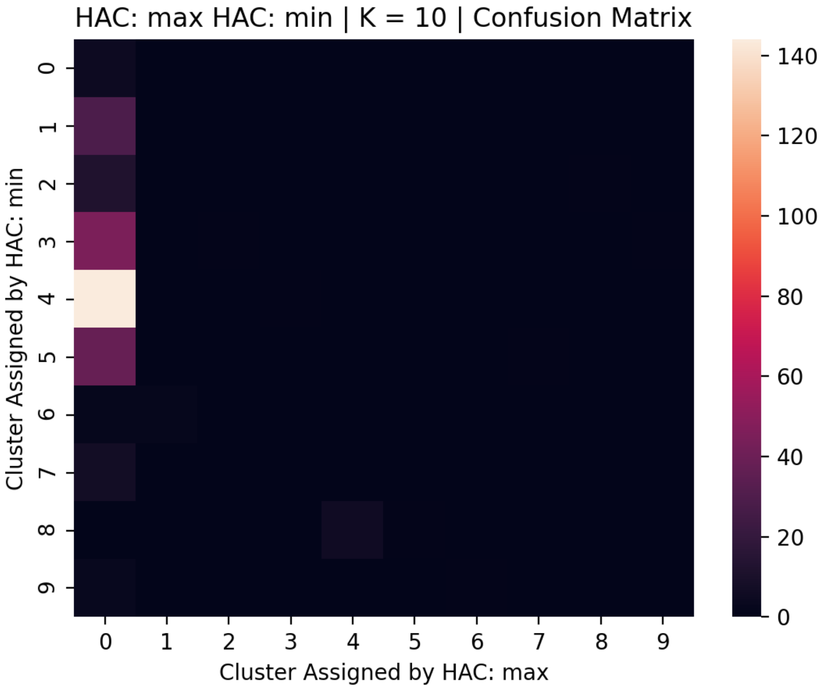
The confusion matrices for all 6 model pairs using the setup in the staff solutions are shown below. HAC with max linkage seems to be the closest to k-means. This makes sense given our results from part 4 where we note that HAC seems to spread out the images more evenly across the clusters due to the fact that merges are based on the maximum distance between clusters which accounts for the size/internal variance of each cluster. This is most similar to the behavior of k-means.

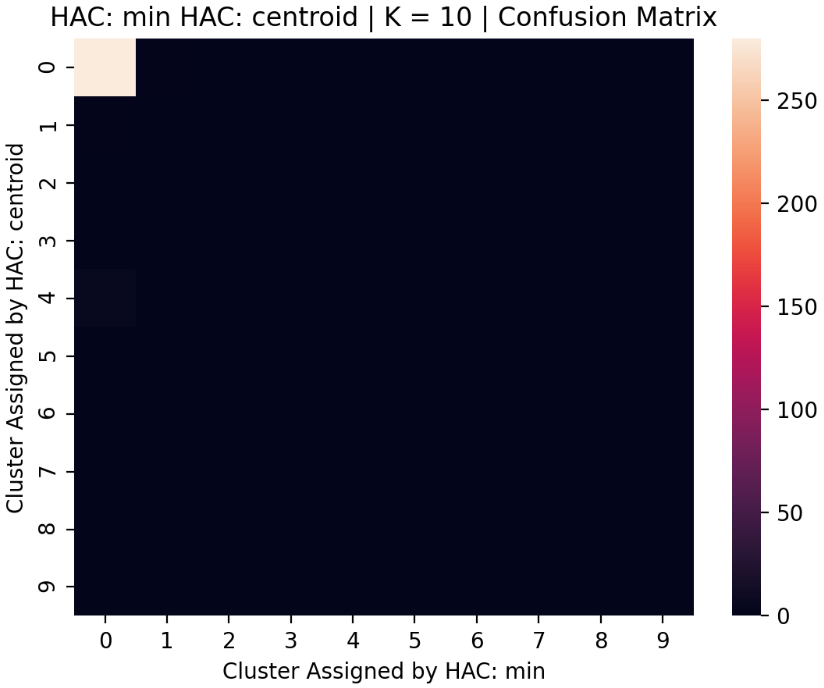

K-means HAC: centroid | K = 10 | Confusion Matrix
Cluster Assigned by HAC: centroid
0
1
2
3
4
5
6
7
8
9
Cluster Assigned by K-means
0
1
2
3
4
5
6
7
8
9
45
40
35
30
25
20
15
10
5
0


K-means HAC: max | K = 10 | Confusion Matrix
Cluster Assigned by HAC: max
0
1
2
3
4
5
6
7
8
9
Cluster Assigned by K-means
0
1
2
3
4
5
6
7
8
9
35
30
25
20
15
10
5
0


K-means HAC: min | K = 10 | Confusion Matrix
Cluster Assigned by HAC: min
Cluster Assigned by K-means
0
1
2
3
4
5
6
7
8
9
0
5
10
15
20
25
30
35
40
45


HAC: max HAC: centroid | K = 10 | Confusion Matrix
Cluster Assigned by HAC: centroid
Cluster Assigned by HAC: max
0
1
2
3
4
5
6
7
8
9
0
20
40
60
80
100
120
140





Part 7

In general, this matching seems like a reasonable evaluation metric for clustering. While we may not know beforehand which cluster is supposed to match with which label, being able to recognize the degree to which there exists a 1-to-1 correspondence between true labels and assigned clusters can give us insight into which clustering algorithms performed better.

Problem 3 (Ethics Assignment, 5pts)

Select a real-life outcome in Artificial Intelligence or Machine Learning that you believe is morally wrong. You can select your own outcome from the news or select one of the outcomes in the options below:

- COMPAS, a case management tool predicting recidivism that flagged “blacks are almost twice as likely as whites to be labeled a higher risk but not actually re-offend” (Angwin 2016).
- An NLP algorithm filled in the inference “Man is to ____ as woman is to ____” with “Man is to computer programmer as woman is to homemaker” (Bolukbasi et al, 2016).
- http://www.survivalofthebestfit.com/game: a game that exemplifies algorithmic bias in resume screening
- IBM Diversity in faces: insufficient training data for darker-skinned faces
- Other Unfair Algorithms: Algorithms of Oppression (a really good book with tons of examples), VI-SPDAT, Allegheny Family Screening Tool

Draw a causal chain that resulted in this outcome and circle the choice points that were the largest contributors to the outcome. At each morally relevant choice point, write two alternative decisions that could have prevented the outcome.

Solution

5 points
The students drew a causal chain that was sufficiently complete, including each important event relevant to the outcome and circled the choice points relevant to the outcome. The alternative decisions at each choice point were specific, clear, and would have actually prevented the outcome. The student’s decisions of alternatives were creative, and the student displayed a strong understanding of causal chains.

4 points
The students drew a causal chain that was sufficiently complete, including each important event relevant to the outcome and circled the choice points relevant to the outcome. The student did present specific and clear alternative decisions but those decisions would not actually have prevented the outcome. The student displayed a strong understanding of the causal chain, but did not think thoroughly about how alternate decisions would have impacted the result.

3 points
The students drew a causal chain that was sufficiently complete, including each important event relevant to the outcome and circling each relevant choice point. However, the student did not present two clear alternative decisions at each choice point. The alternatives were either under-specified or clearly failing to prevent the outcome.

2 points
The student drew a causal chain that was incomplete, missing important events that contributed to the outcome.

1 point
The student did not draw a causal chain.

Name

Collaborators and Resources

Whom did you work with, and did you use any resources beyond cs181-textbook and your notes?

Calibration

Approximately how long did this homework take you to complete (in hours)?