CS181-S22

Assignment #4

Due: 11:59pm EST, 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:

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

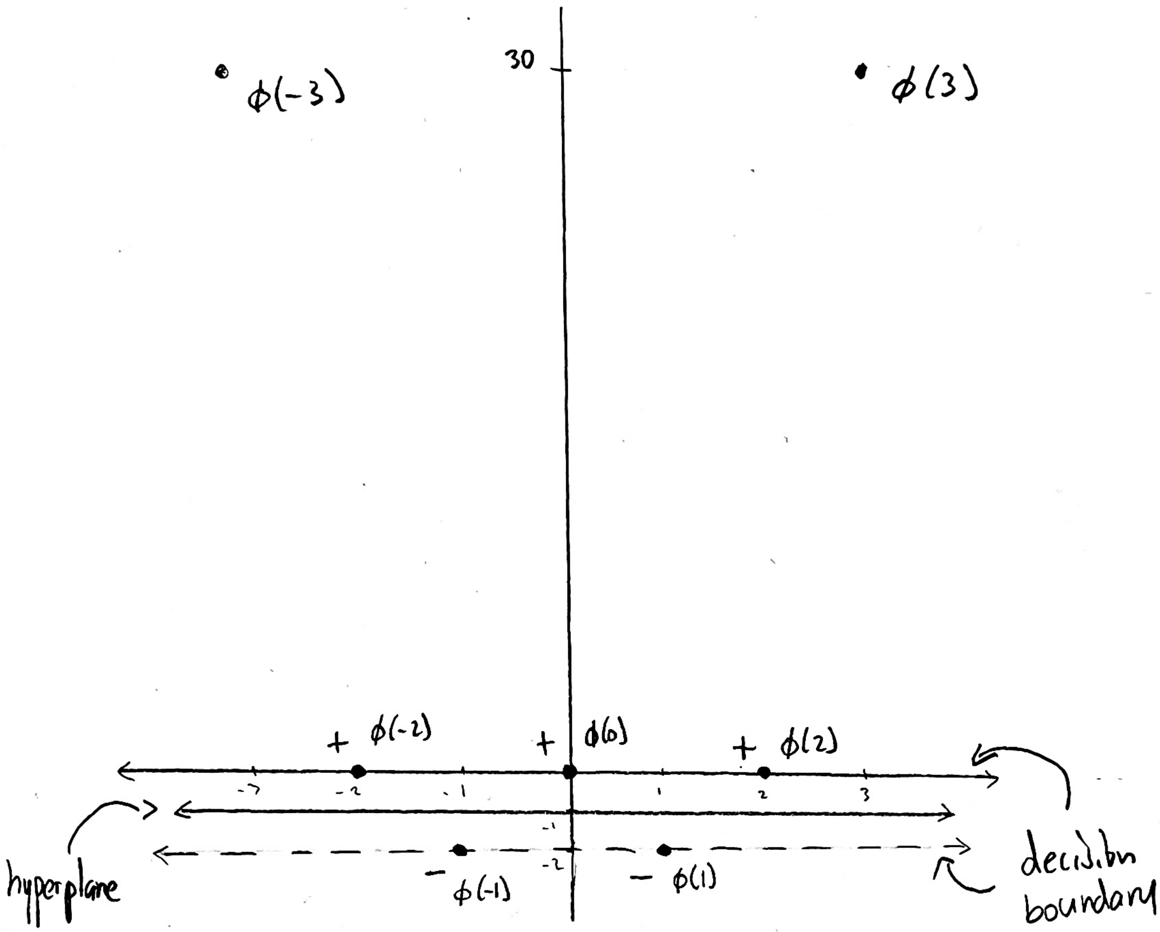
Make sure to follow the logical structure of the questions below when composing your answers, and to justify each step.

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 optimal decision boundary from 1.1. Justify your answer.

   Hint: The boundary is where the discriminant is equal to 0. Use what you know from 1.1 and 1.3 to solve for $\mathbf{w}$ in terms of w_0. (If you solve this problem in this way, then w_0 corresponds to your free parameter to describe the set of all possible $(\mathbf{w}, w_0)$.)
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 $\boldsymbol{\phi}'(x) = (x, -\frac{31}{12}x^2 + \frac{7}{12}x^4)$. 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. The following is a plot of the training data and the optimal decision boundary.



2. The value of the margin achieved is 1.

3. A unit vector orthogonal to the decision boundary is $(0, 1)$.

4. The set of all possible $(\mathbf{w}, w_0)$ that define the optimal decision boundary are all those such that, given $\mathbf{w} = (w_1, w_2)$, satisfy $w_1 = 0$ and $w_2 = w_0$.

5. The solution $\mathbf{w} = (0, 1)$ is optimal for the optimization problem.

6. $w_0 = 1$. Substituting these optimal values into our discriminant function gives

$$h(\boldsymbol{\phi}(x)) = \boldsymbol{\phi}_2(x) + 1$$

7. The points $(-2, +1), (-1, -1), (0, +1), (1, -1), (2, +1)$ could be support vectors of the classifier. For each of these points, plugging the features into the discriminant function gives the exact corresponding y value for those features (e.g., the point $\{x_i, y_i\} = (-2, +1)$ has $\boldsymbol{\phi}(x_i) = (-2, 0)$, and $h(\boldsymbol{\phi}(x_i)) = 0 + 1 = +1$, which is y_i).

8. This feature mapping still admits a separable solution. The margin for this feature mapping is half as big as the margin in the previous parts. Based on this, I would prefer the previous set of features, as the larger margin of 1 allows there to be support vectors with discriminant values of exactly either -1 or +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?

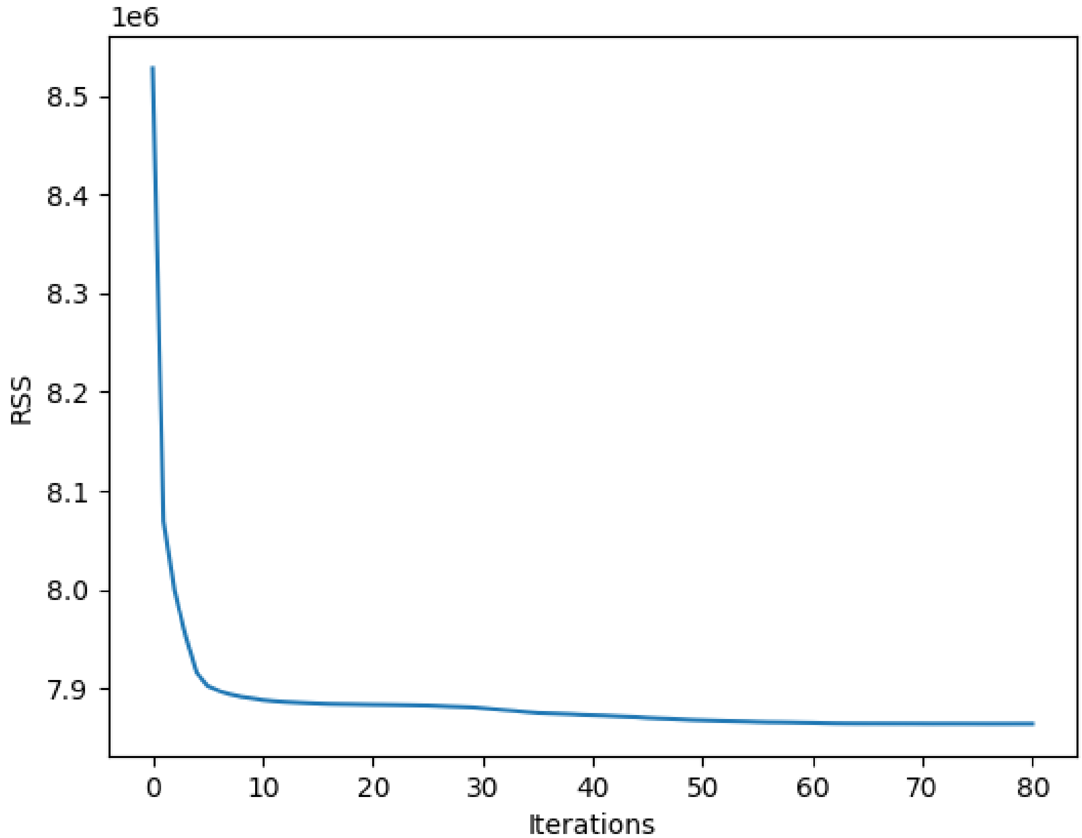
Problem 2 (cont.)

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?

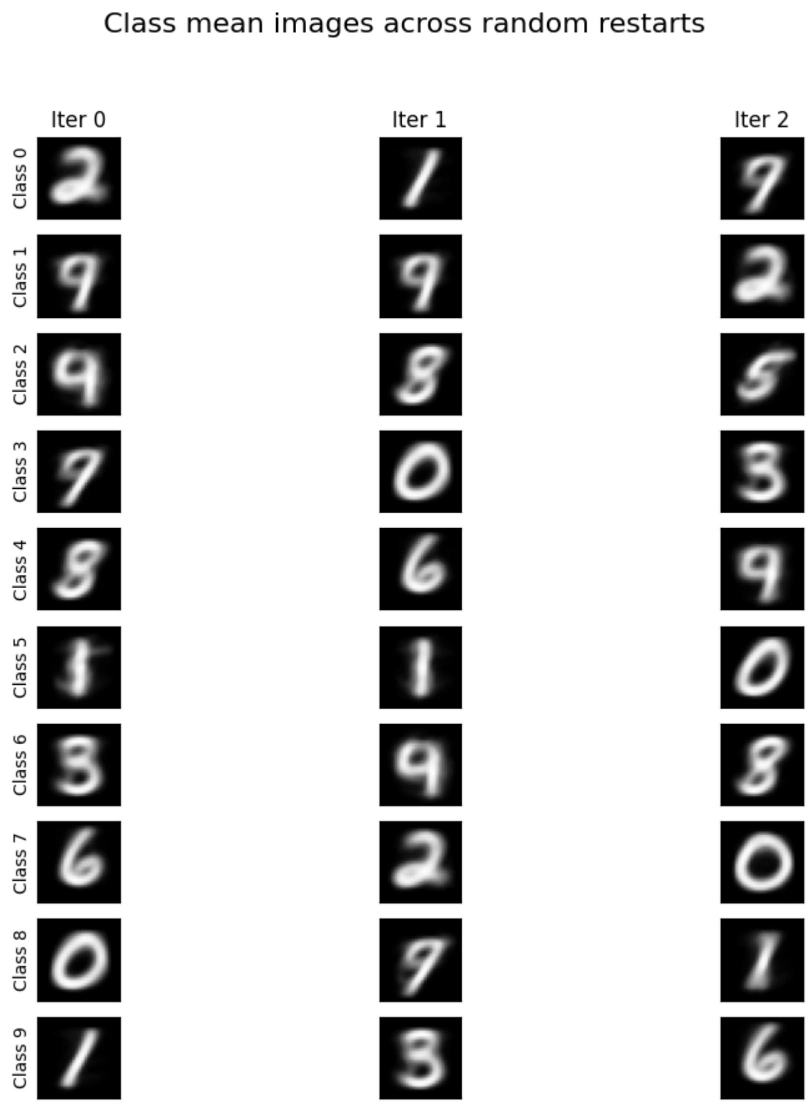
7. Suppose instead of comparing the different clustering methods to each other, we had decided to compute confusions of each clustering method to the *true* digit labels (you do *not* have to actually compute this). Do you think how well the clustering match the true digits is reasonable evaluation metric for the clustering? Explain why or why not.

Solution

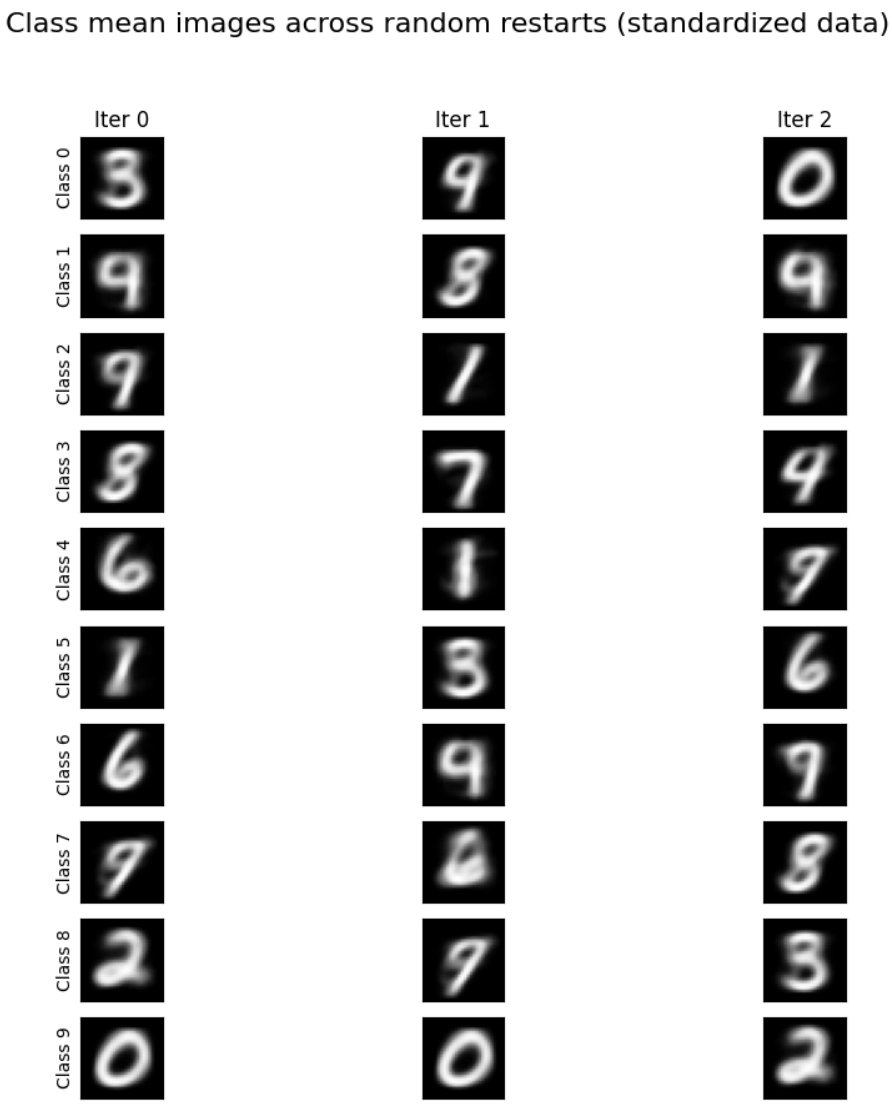
1. The following is the plot for this problem



2. The following is the plot for this problem

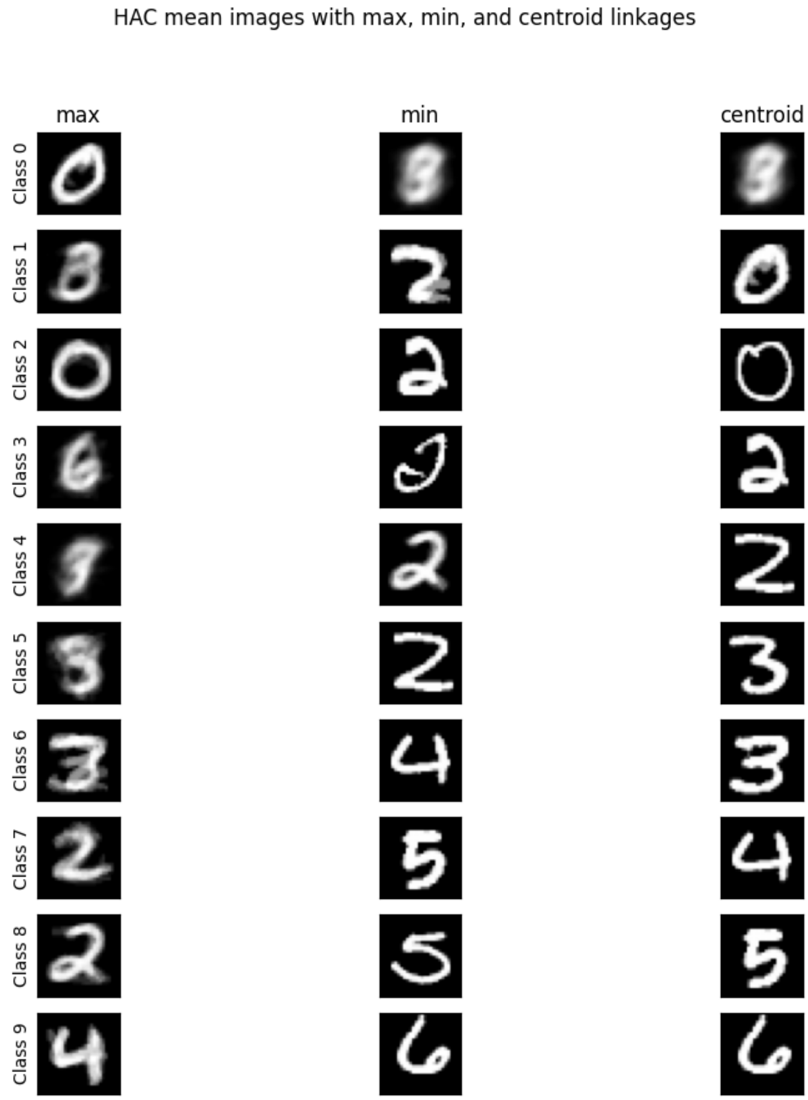


3. The following is the plot for this problem



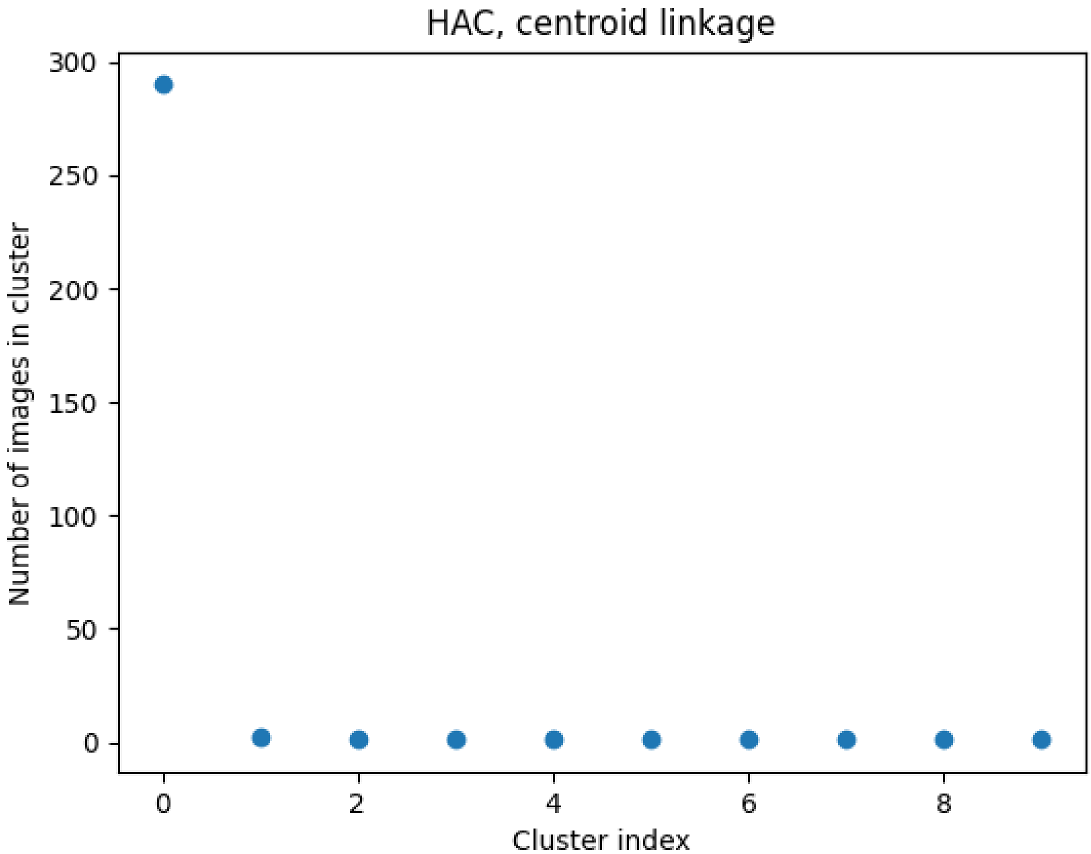
Compared to Part 2, the centroids differ in that they are assigned to different underlying numbers. In terms of blurriness, crispness, etc., there is not a noticeable difference.

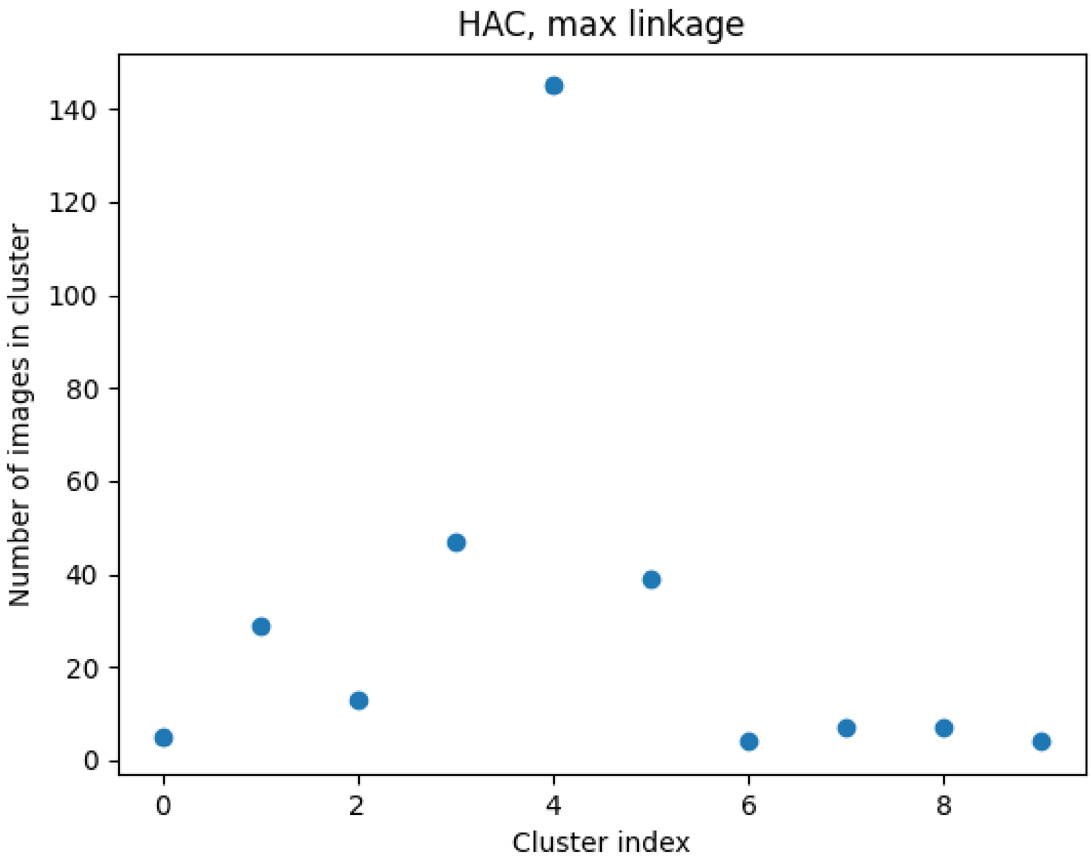
4. The following is the plot for this problem



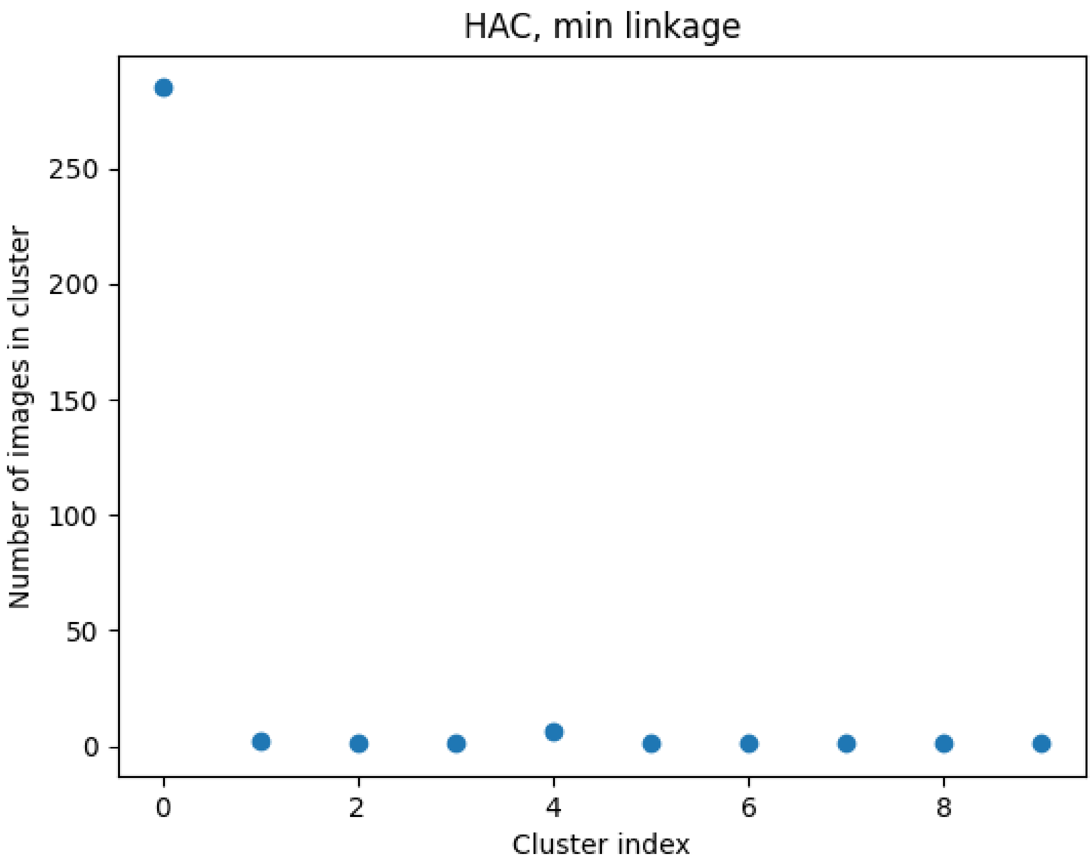
The means determined using min and centroid linkage HAC are significantly more crisp than those determined using KMeans. The means in max linkage HAC are similar in crispness to those from KMeans. HAC was only ran once as it has the same outcome every time for a given data set (there is no randomness as in the picking of cluster centers in KMeans).

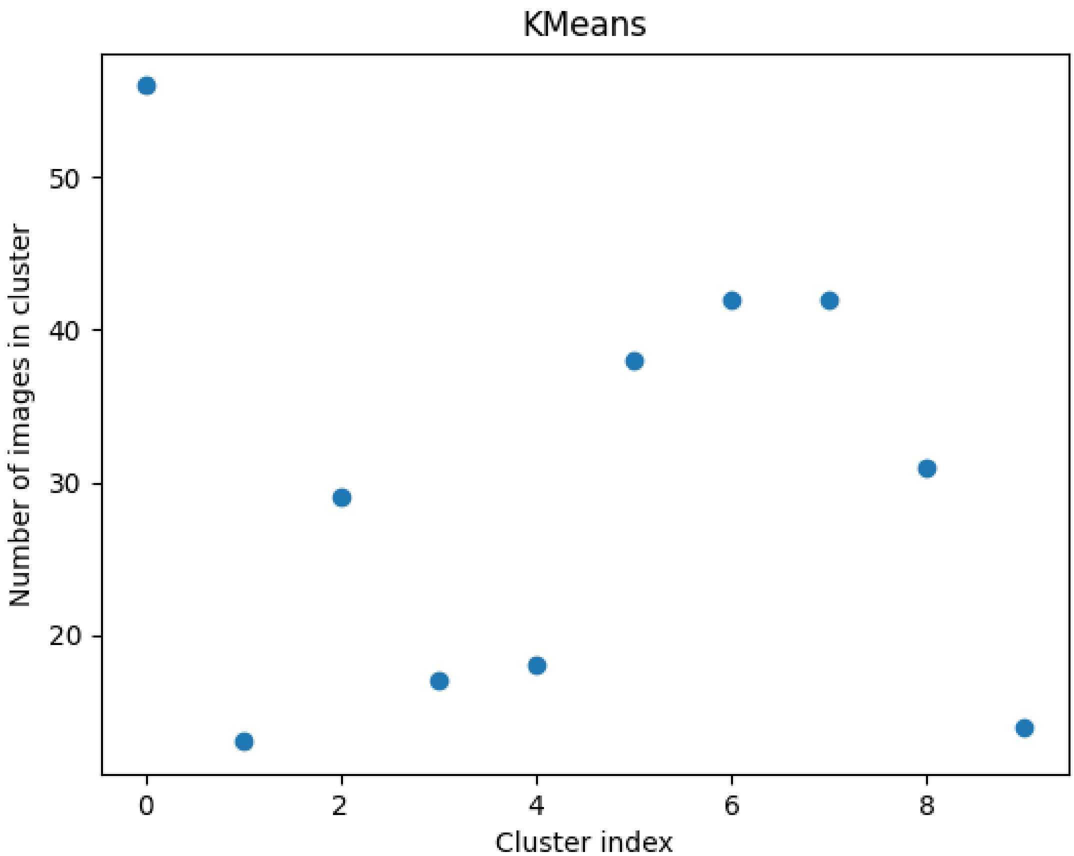
5. The following are the plots for this problem





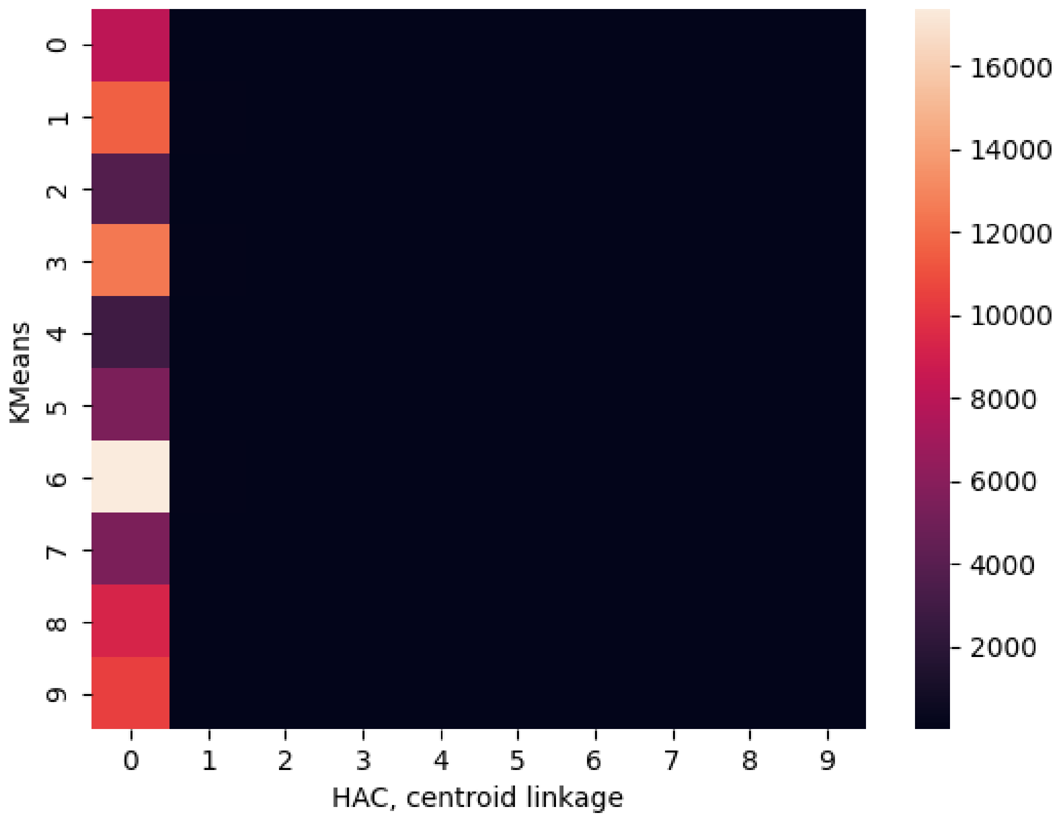
HAC, max linkage
Number of images in cluster
Cluster index
0
20
40
60
80
100
120
140
0
2
4
6
8

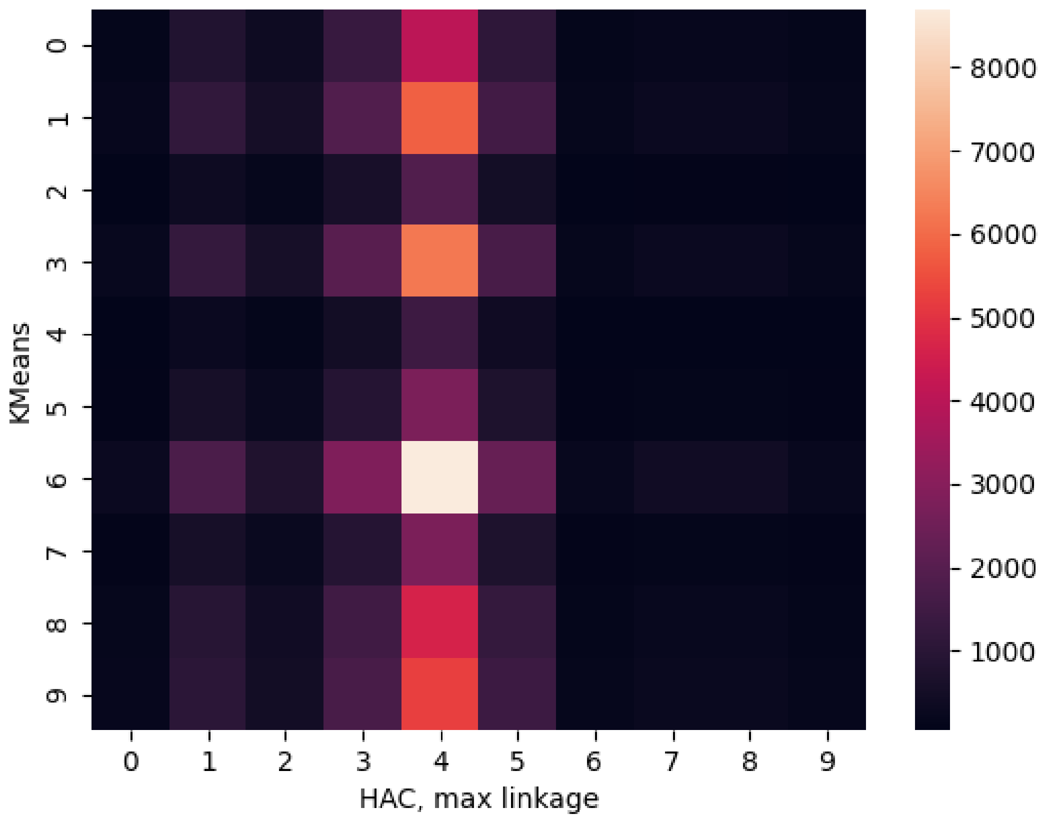

HAC, min linkage
Number of images in cluster
250
200
150
100
50
0
0
2
4
6
8
Cluster index



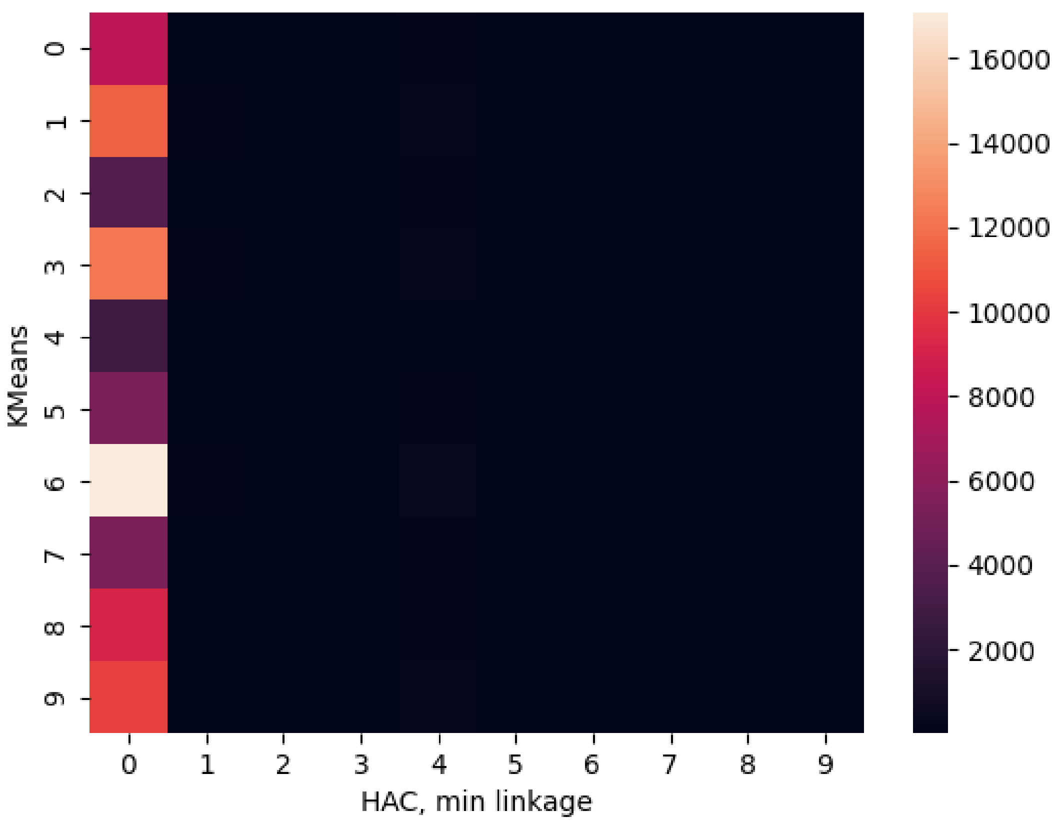
These plots show that the clusters produced by the max linkage criteria are on more similar in size to each other, where those produced by the min linkage criteria are very different in size (one is much larger than the others). This explains the crispness of the clusters as a small cluster is going to be more crisp due to it being the average of less images (introducing less blur into the image).

6. The following are the plots for this problem

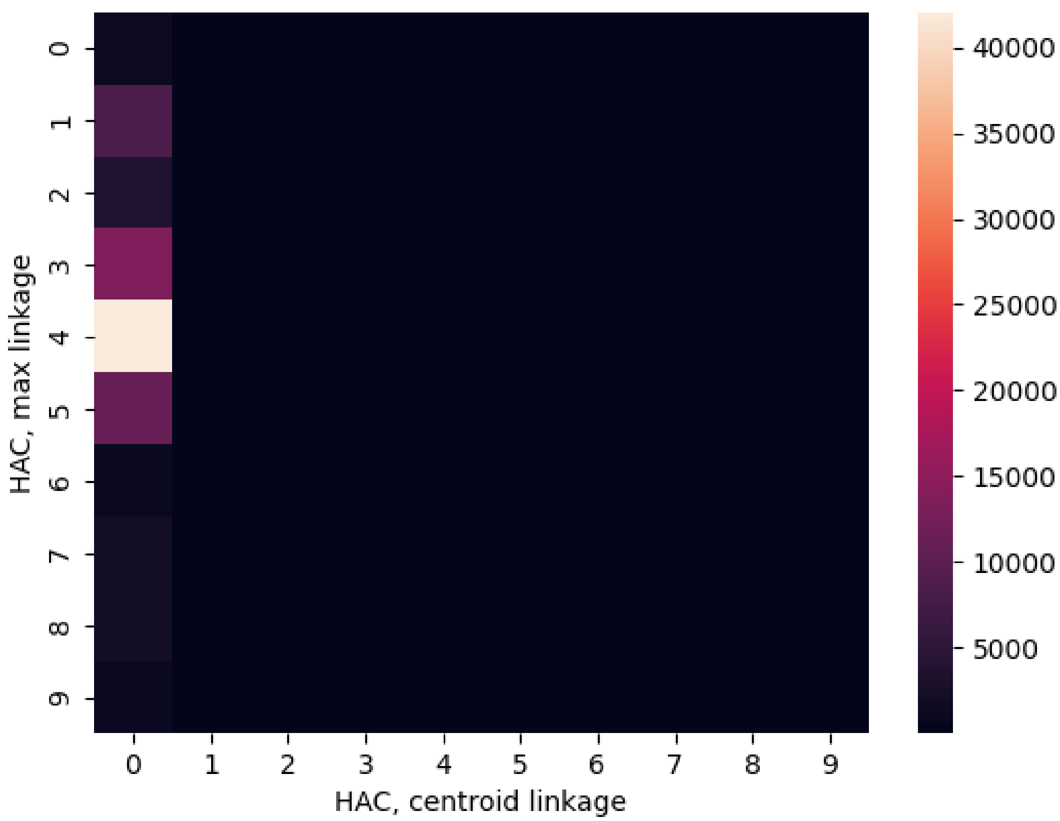




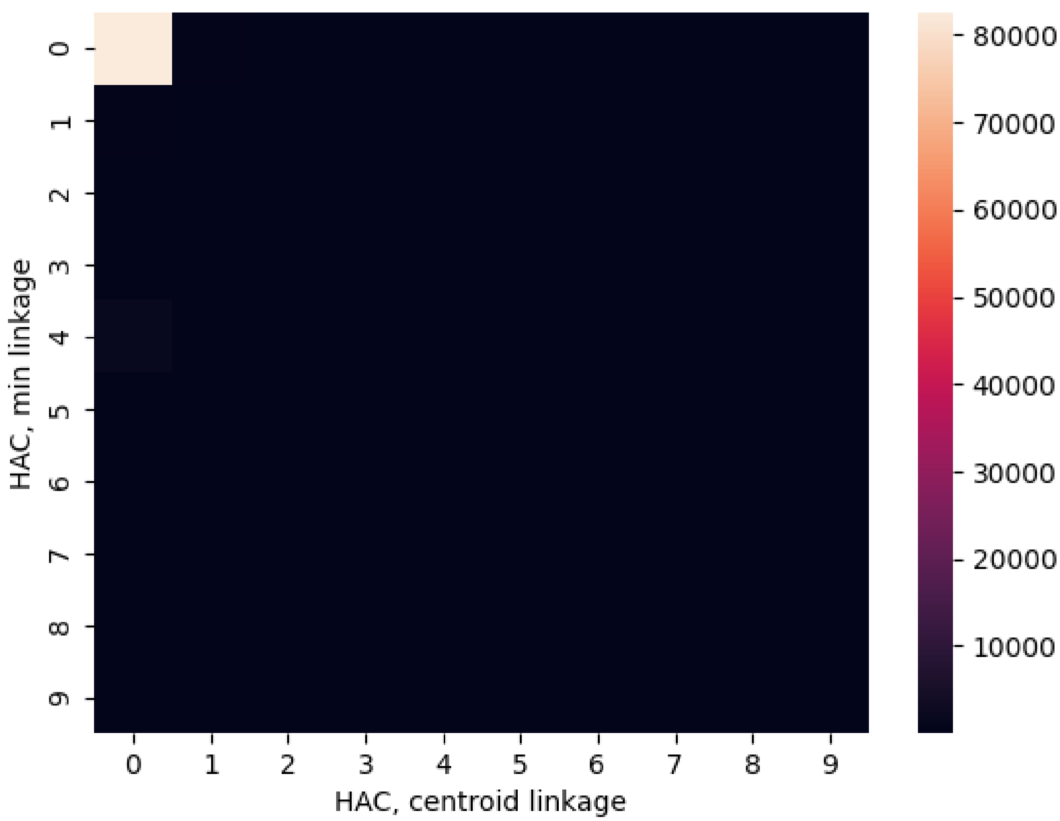
0
1
2
3
4
5
6
7
8
9
KMeans
0
1
2
3
4
5
6
7
8
9
HAC, max linkage
8000
7000
6000
5000
4000
3000
2000
1000

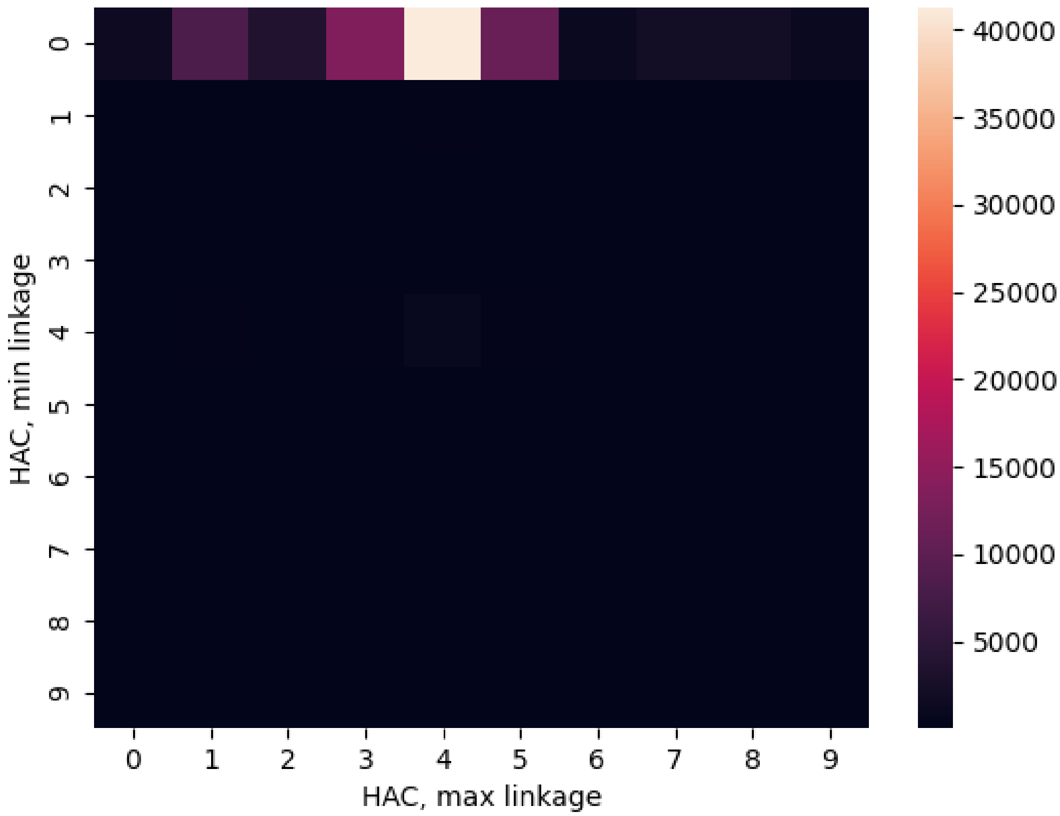


KMeans
HAC, min linkage
0
1
2
3
4
5
6
7
8
9
2000
4000
6000
8000
10000
12000
14000
16000



HAC, max linkage
0
1
2
3
4
5
6
7
8
9
HAC, centroid linkage
40000
35000
30000
25000
20000
15000
10000
5000


HAC, min linkage
HAC, centroid linkage
0
1
2
3
4
5
6
7
8
9
80000
70000
60000
50000
40000
30000
20000
10000



Max-linkage HAC is closest to k-means. This is because max linkage HAC and k-means both produce clusters that are large–they are both motivated to create clusters that are big.

7. The main challenge with this approach is that the resulting clusters don't line up exactly in 0-9 order such that the transition matrix would be correctly aligned. Assuming this is corrected by hand, I still don't think that this would be a reasonable evaluation. This is because we aren't asking these methods to correctly classify digits; we are only asking them cluster together those the methods think are similar to each other. Therefore, it would be incorrect to negatively evaluate a method that classifies fives as threes if they happen to be more similar to threes than they are to other fives.

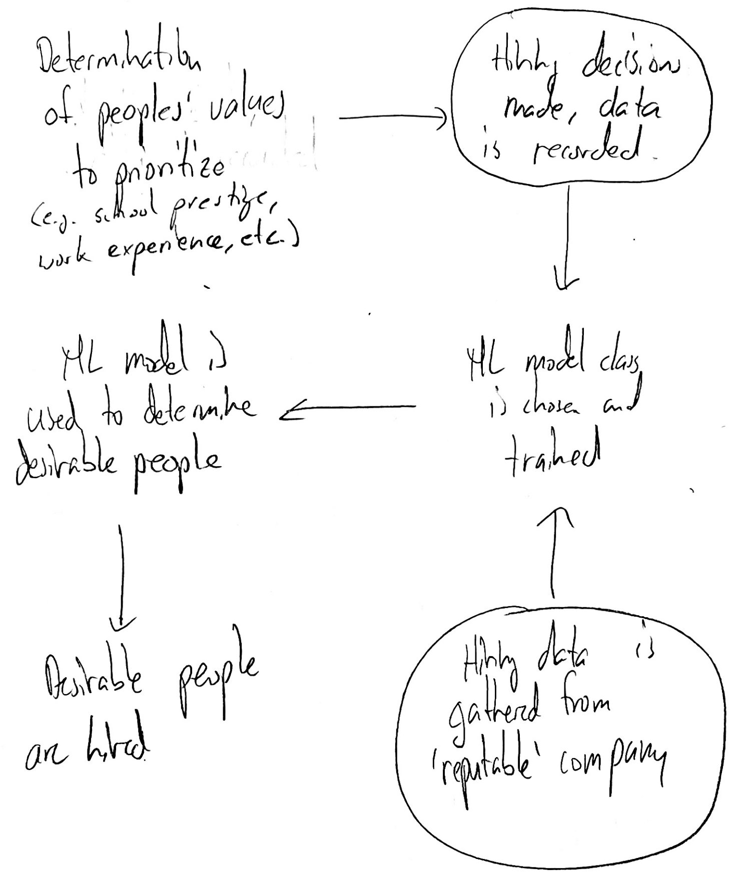
Problem 3 (Ethics Assignment, 5pts)

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



I did this exercise based on http://www.survivalofthebestfit.com/game: a game that exemplifies algorithmic bias in resume screening.

In this picture, the two circled points demonstrate two ways in which data is gathered for inputting into a machine learning model

At the first circled point, two alternate decisions would be to,

1. Make decisions about who should be hired, and then have a different person analyze those decisions and explicitly point out those that may be regarded as containing bias. Create two models, one that classifies people to be hired and those not to be hired, and one that classifies people into those that may have been picked with underlying bias and those that were picked without underlying bias. The resulting classifications could be used in further steps.

2. Not make decisions based off of gut-intuition to begin with, and rather train models to determine subjective features such as how ambitious a person is on a scale of 1 to 10, and how hard working a person is on a scale of 1 to 10. Feed data from people into a non-ML algorithm that simply picks the best N people based on this criteria

In the second circled point, alternative decisions would be to:

1. Not go off of existing data and use only data of hiring decisions made within the organization, by people who were trained not to express subconscious bias

2. Train a model, based off of humans picking apart people who were fairly and unfairly hired, to determine when a person from the data set was fairly or unfairly hired. Use this model to remove data points where such people were unfairly hired/not hired, and use this new model. Or, correct data points instead of removing them altogether.

Name

Rodney Lafuente Mercado

Collaborators and Resources

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

Did you attend office hours for help with this homework?

I did not work with anyone on this problem set

Calibration

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

more than 15