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:

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

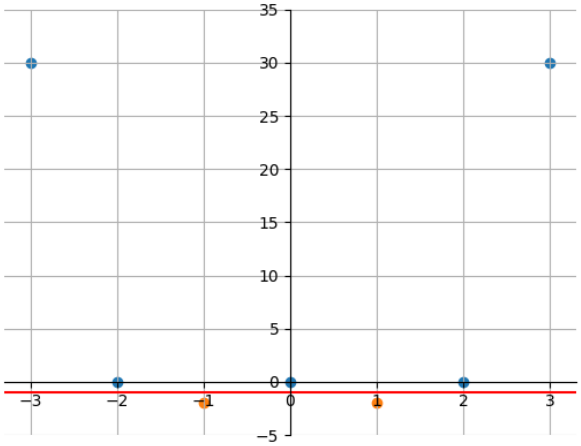
Make sure to follow the logical structure of the questions below when composing your answers, and to justify each step.

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 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

Solution 1.1:
Blue coordinates represent the transformed data with class $y = +1$; orange coordinates represent the transformed data with class $y = -1$. We can determine by inspection that the optimal decision boundary of the max margin classifier is $\boldsymbol{\phi}(x)_2 = -1$.



Solution 1.2:
The optimal decision boundary in 1.1 achieves a margin of 1.

Solution 1.3:
An unit vector orthogonal to the decision boundary is: $\mathbf{w} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$.

Solution 1.4:
The optimal decision boundary exists where the discriminant equals 0:

$$\mathbf{w}^\top \boldsymbol{\phi}(x) + w_0 = 0$$

Vectors on the decision boundary have the form, $\boldsymbol{\phi}(x) = \begin{bmatrix} 0 \\ \lambda \end{bmatrix}$ (where $\lambda \in \mathbb{R}$). Therefore, on the decision boundary, the discriminant can be written as:

$$\begin{aligned} \mathbf{w}^\top \begin{bmatrix} 0 \\ \lambda \end{bmatrix} + w_0 &= 0 \\ \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}^\top \begin{bmatrix} 0 \\ \lambda \end{bmatrix} + w_0 &= 0 \\ w_2 \lambda &= -w_0 \\ \lambda &= -\frac{w_0}{w_2} \end{aligned}$$

In 1.1, we found that the decision boundary was given by $\boldsymbol{\phi}(x)_2 = -1$. Comparing this to the above expression, we get:

$$\begin{aligned} -\frac{w_0}{w_2} &= -1 \text{ on the decision boundary} \\ \implies w_2 &= w_0 \end{aligned}$$

Therefore, $(\mathbf{w}, w_0) = (\begin{bmatrix} 0 \\ w_0 \end{bmatrix}, w_0)$ (for all $w_0 \in \mathbb{R}$) defines the set which makes the discriminant equal to 0.

Solution 1.5:
By inspection, the width between the support vectors (the vectors with the min margin) is 2. Given the

expression for width between support vectors $\frac{2}{\|\mathbf{w}\|}$, and the $\mathbf{w}$ we found in 1.4, we can find the optimal $\mathbf{w}$:

$$\begin{aligned}
\frac{2}{\|\mathbf{w}\|} &= 2 \\
\frac{2}{\left\|\begin{bmatrix} 0 \\ w_0 \end{bmatrix}\right\|} &= 2 \\
\frac{2}{\sqrt{0^2 + w_0^2}} &= 2 \\
w_0 &= 1
\end{aligned}$$

$$\therefore \mathbf{w}^* = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Solution 1.6:
From 1.4: $w_0^* = 1$.

Therefore, the discriminant function is $h(\boldsymbol{\phi}(x); \mathbf{w}^*, w_0^*) = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \cdot \boldsymbol{\phi}(x) + 1 = 0$.

Solution 1.7:
Support vectors exist anywhere on the lines $\boldsymbol{\phi}(x)_2 = 0$ and $\boldsymbol{\phi}(x)_2 = -2$, so they have the form $\begin{bmatrix} x_1 \\ 0 \end{bmatrix}$ and $\begin{bmatrix} x_1 \\ -2 \end{bmatrix}$ respectively. The constraint is tight when $y_i(\mathbf{w}^\top \boldsymbol{\phi}(x_i) + w_0) = 1$.

On $\boldsymbol{\phi}(x)_2 = 0$, $y_i = 1$:

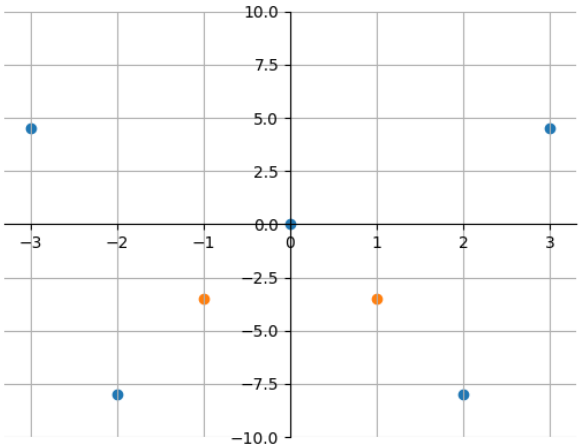
$$1\left(\begin{bmatrix} 0 \\ 1 \end{bmatrix}^\top \begin{bmatrix} x_1 \\ 0 \end{bmatrix} + 1\right) = 1(0 + 1) = 1$$

On $\boldsymbol{\phi}(x)_2 = 0$, $y_i = -1$:

$$1\left(\begin{bmatrix} 0 \\ 1 \end{bmatrix}^\top \begin{bmatrix} x_1 \\ -2 \end{bmatrix} + 1\right) = -1(-2 + 1) = 1$$

Thus, the constraints are tight for all support vectors of the classifier. □

Solution 1.8:
As before, blue coordinates represent the transformed data with class $y = +1$; orange coordinates represent the transformed data with class $y = -1$. However, this feature map is clearly not linearly separable - there is no one straight line that separates the blue and orange points. Thus, for this SVM classification problem, I would choose to use the feature map plotted in part 1.1, as calculation of the margin is much simpler than calculating the margin for a much more complex decision boundary (e.g. an oval / rectangle / pair of parallel lines) in this example.


10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
−3
−2
−1
0
1
2
3

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?

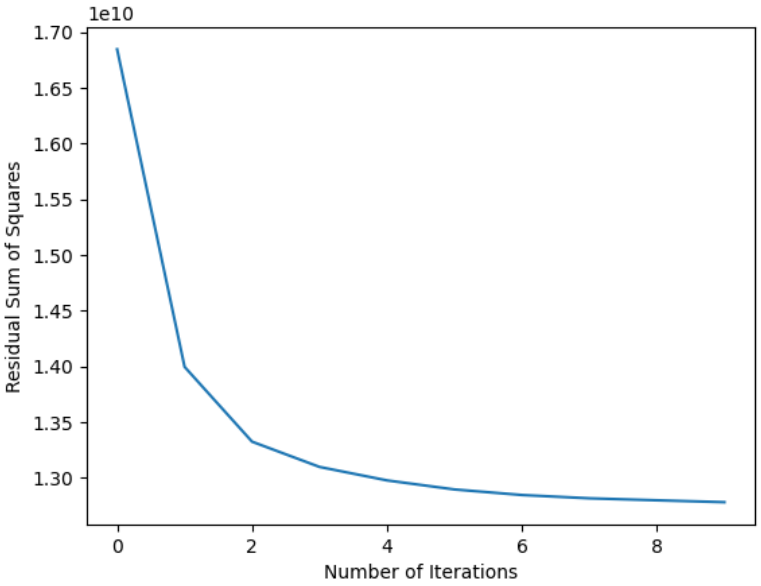
   Going back to the previous part: How does this help explain the crispness and blurriness of some of the clusters?

Problem 2 (cont.)

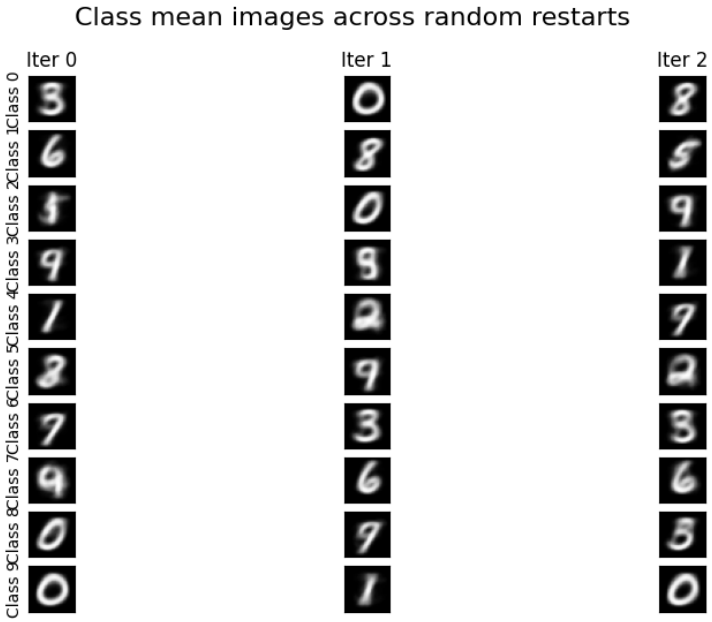
6. For your K-means with $K = 10$ model and HAC min/max/centroid models using 10 clusters on the `small_dataset` images, use the `seaborn` module's `heatmap` function to plot a confusion matrix between each pair of clustering methods. This will produce 6 matrices, one per pair of methods. The cell at 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

Solution 2.1:
The plot verifies that the K-means objective function is always decreasing as the number of iterations increases:



Solution 2.2:



Solution 2.3:

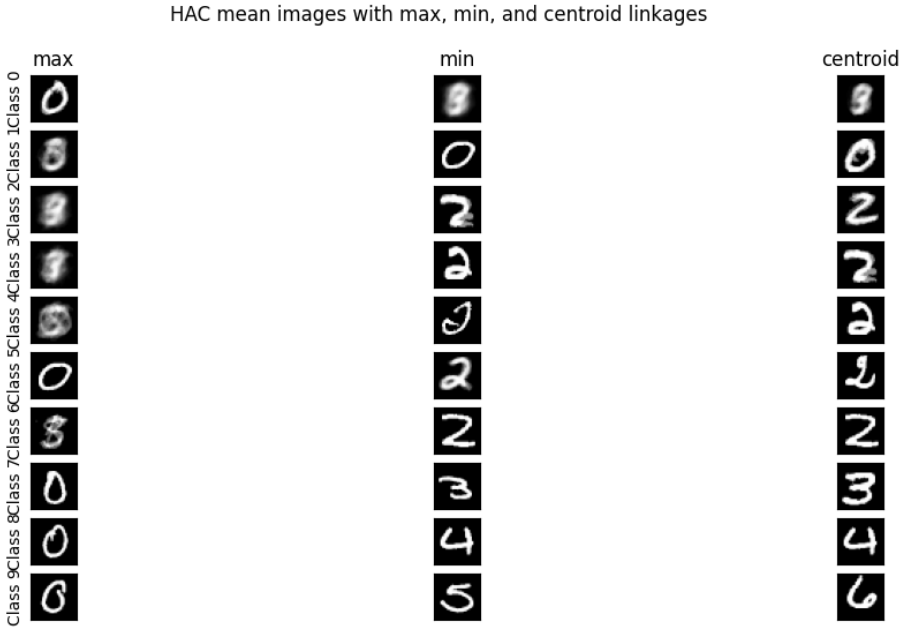
All of the pixels in the mean images (the centroids) appear to be greyer, rather than a crisp black background

with white colored text overlayed as seen in 2.2. As a result, the images created from standardized data appear blurrier.

This occurs because pixel values range from 0 (white) to 255 (black). If you consider an edge pixel in the image, which is usually black-ish, it will have a pixel value >200. Thus, the mean of an edge pixel is ∼227.5. Therefore, where we standardize an edge pixel that is pure black, the pixel value becomes $\frac{(255-227.5)}{\sigma_i} = \frac{(27.5)}{\sigma_i}$. This standardized pixel value is significantly lower than the original, so the edge pixel appears lighter/greyer. Analogously, if you consider a central pixel in the image, which is usually white-ish, it will have a pixel value <50. Thus, the mean of an edge pixel is ∼22.5. Therefore, where we standardize a central pixel that is pure black, the pixel value becomes $\frac{(50-22.5)}{\sigma_i} = \frac{(22.5)}{\sigma_i}$. This standardized pixel value is again lower than the original, so the central pixel appears lighter/greyer.

Ultimately, therefore, the pixels are all lighter and end up being more similar in color. This makes sense, given the pixels in our standardized images have a lower variance than in the originals (which have high contrast between black and white pixels).

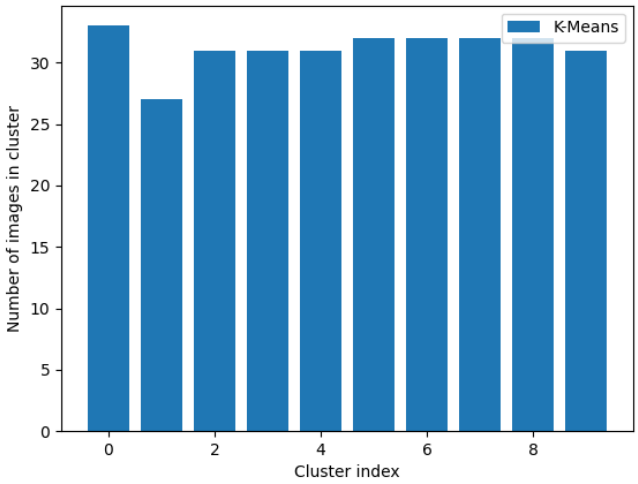
Solution 2.4:

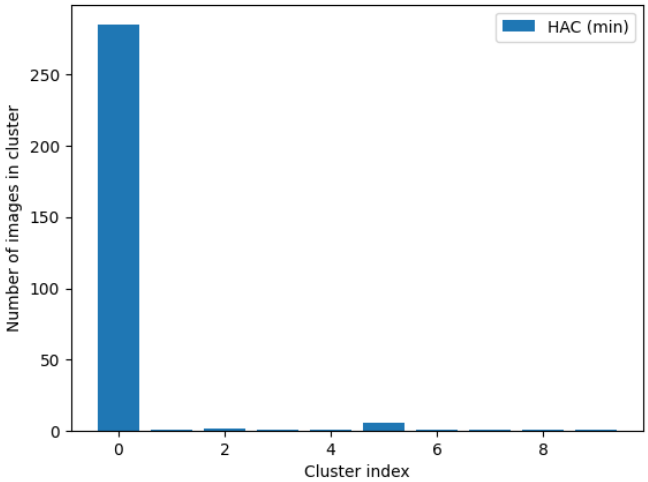


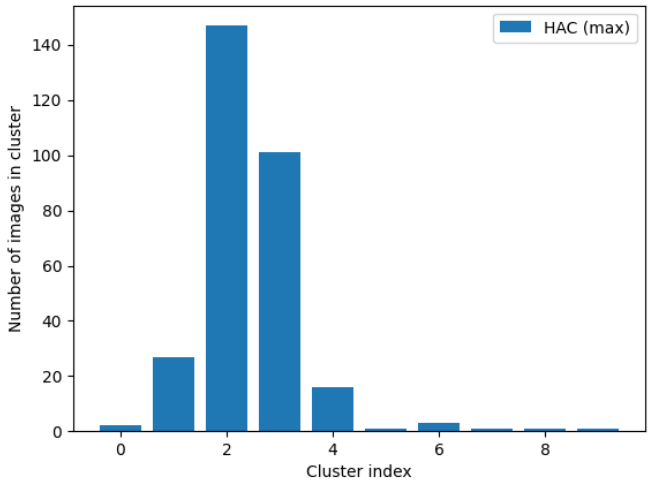
The crispness of the k-means images was quite high (for the non-standardized data). The HAC max cluster means appear to be the blurriest, whilst the HAC min and HAC centroid cluster means appear quite crisp in all but the first (0th) class.

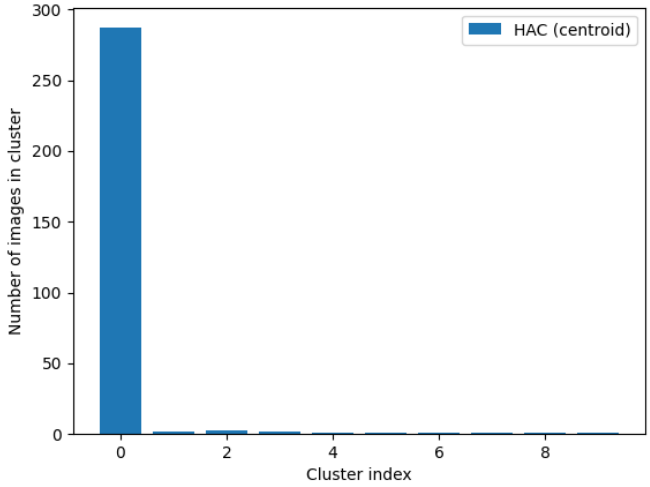
Unlike k-means, which needs to be run several times in order to avoid "getting stuck" at a local minimum of the objective function, HAC only needs to be run once, after having selected a linkage criterion, because it calculates the clusters in a deterministic way (without using calculus).

Solution 2.5:


K-Means
Number of images in cluster
0
5
10
15
20
25
30
0
2
4
6
8
Cluster index


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

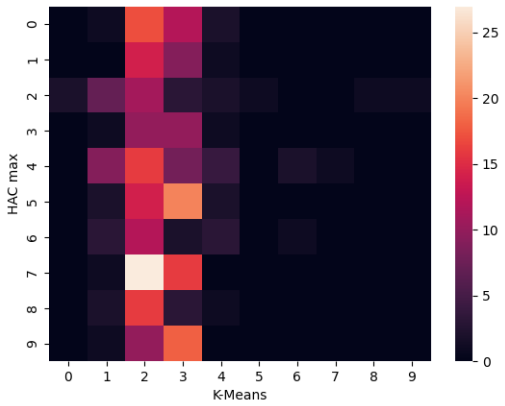


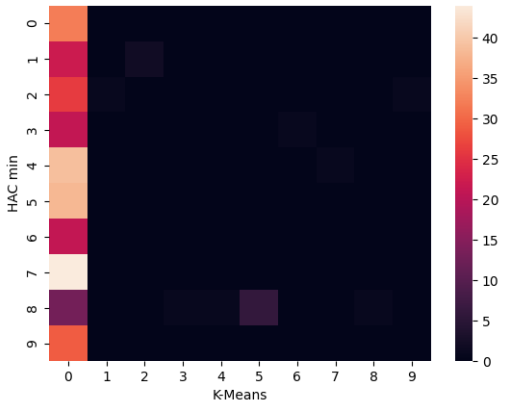


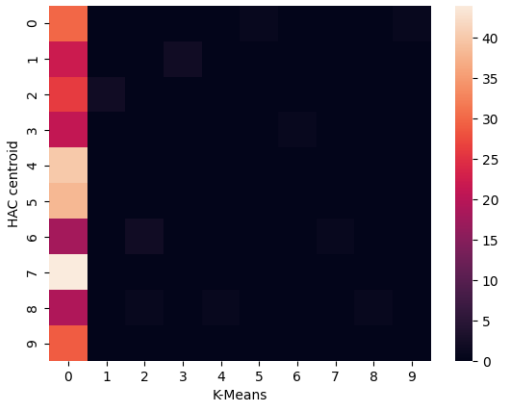
HAC min and HAC centroid seem to cluster most images into a single cluster, whilst HAC max clusters images across a broader range of clusters. Intuitively this makes sense: for min, when a cluster gets larger, the probability that this cluster is close to other clusters increases, increasing the likelihood that this large cluster increases in size again; for max, when a cluster gets larger, it gets closer to other larger sized clusters, and so the next max distance is unlikely to link this cluster to another already-large cluster, so the likelihood of the large cluster increasing in size again decreases. This explains why max-linkage produces compact clusters, while min-linkage latches onto any neighboring closest cluster and produces stringy clusters.

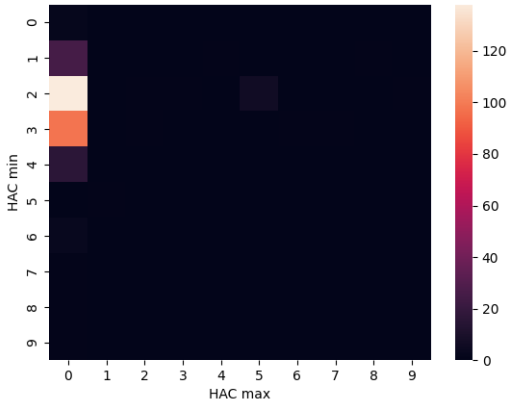
The blurriness of the outputs can be explained when a cluster mean is the average of many images, e.g. HAC max is quite blurry in many classes because there are lots of images in many of the classes; HAC min and HAC centroid have blurry outputs for class 0, but relatively crisp outputs for the other classes, which are averaging relatively few images; k-means is very crisp because there are only c. 30 (very few) images per class (an almost perfectly uniform distribution across clusters).

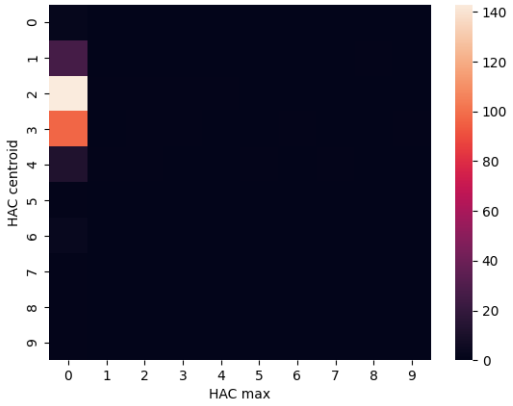
Solution 2.6:

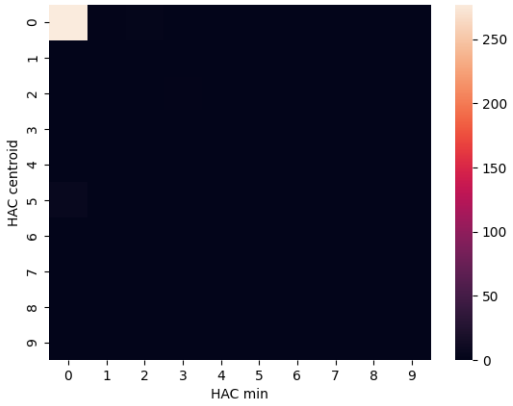

HAC max
K-Means
0 1 2 3 4 5 6 7 8 9
0 5 10 15 20 25


HAC min
K-Means
0 1 2 3 4 5 6 7 8 9
0 5 10 15 20 25 30 35 40


HAC centroid
K-Means
0 1 2 3 4 5 6 7 8 9
0 5 10 15 20 25 30 35 40







Based on the heatmaps , HAC max appears to be closest to k-means. This seems to be because the max-linkage distributes images across classes more evenly than HAC min and HAC centroid, which is similar to the way k-means distributes images across classes, as also seen in the "num. images in cluster plots" in 2.5. Thus, we see "hotter" points distributed across more of the HAC max/k-means heatmap than other heatmaps.

Solution 2.7:
Yes - how well the clustering matches the true digits is reasonable evaluation metric for the clustering. The more accurate the clusterings, the more we would see "hotter" cells distributed evenly across our heatmaps; by comparison, less correct clusterings would produce uneven heatmaps. Effectively, using the true digits to evaluate the clustering turns this problem into a supervised learning problem, rather than an unsupervised one - e.g. we could use something like 0/1 loss as a basic evaluation metric using such a method.

Problem 3 (Ethics Assignment, 5pts)

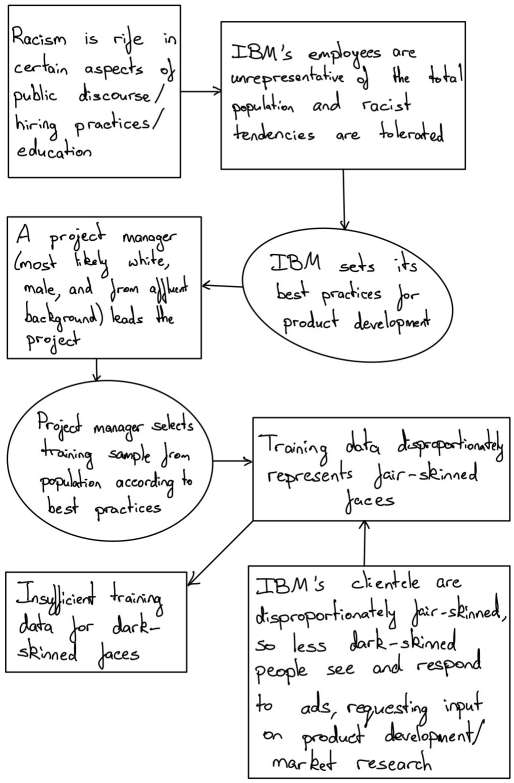
Select a real-life outcome in Artificial Intelligence or Machine Learning that you believe is morally wrong. You can select your own outcome from the news or select one of the outcomes in the options below:

- 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

IBM Diversity in faces: insufficient training data for darker-skinned faces



Alternative decisions (on next page):

"IBM sets its best practices for product development":

- IBM could have chosen to outsource or collaborate on setting their best practices, involving cultural, social, and legal experts and organizations in the process.
- IBM could have set quotas on diversity for its product managers (PMs), or given shared ownership of the project to PMs with different backgrounds, or required PMs' fieldwork plans to be signed off by a diversity expert.

"Project manager selects training sample from population according to best practices":

- Prior to selecting a training sample, the PM chooses to enrol in racial bias training.
- PM delegates choosing a training sample to a selection algorithm (such as that invented by Professor Ariel Procaccia at Harvard). Although there is a risk of introducing algorithmic bias, its use can be supplemented by the best practices written earlier in the causal chain.

Name

Collaborators and Resources

Whom did you work with, and did you use any resources beyond cs181-textbook and your notes?
https://www.cs.princeton.edu/courses/archive/fall18/cos324/files/kmeans.pdf
Josh Michels, Elias Nuwara, Ty Geri

Did you attend office hours for help with this homework?
Yes

Calibration

Approximately how long did this homework take you to complete (in hours)?
21