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:

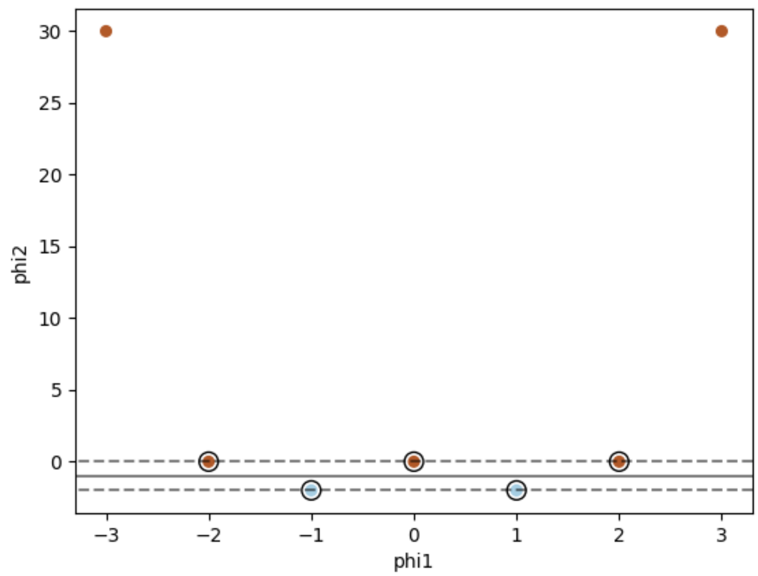
$$\begin{aligned} &\min_{\mathbf{w}, w_0} \frac{1}{2}\|\mathbf{w}\|_2^2 \\ \text{s.t.} \quad & y_i(\mathbf{w}^\top \boldsymbol{\phi}(x_i) + w_0) \geq 1, \ \forall i \in \{1, \dots, n\} \end{aligned}$$

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.

Solution

1. 

   The solid line is the optimal decision boundary and the dotted lines are the margin boundaries. Phi1 and Phi2, the axes labels, correspond to the two components of the ϕ basis function.

2. The value of the margin achieved by the optimal decision boundary found in Part 1 is 1, the orthogonal distance between the margin boundary and the optimal decision boundary.

3. A unit vector that is orthogonal to the decision boundary is the vector $[0\ 1]^T$, a vector with first component 0, second component 1. $\boldsymbol{w}$ corresponding to the optimal decision boundary points in the same direction as this unit vector. More on this shortly.

4. From Part 3, we know that $\boldsymbol{w} = k[0\ 1]^T$, for some positive constant k. Since the boundary is where the discriminant is equal to 0, we can say in this setting that the data inputs x for which the data point sits on the decision boundary is the following:

$$H = \{x|\boldsymbol{w}^T\phi(x) + w_0\}$$

   We want to know all possible $(\boldsymbol{w}, w_0)$ that can describe this set. We can scale $\boldsymbol{w}$ by some constant, as stated previously, but to ensure the condition for this set is met (discriminant equalling zero), we must appropriately scale w_0 for each scaling of $\boldsymbol{w}$. If we scale $\boldsymbol{w}$ by k, then we must also scale w_0 by k, since $\{x|k\boldsymbol{w}^T\phi(x) + kw_0\}$ is the same as our H, which is clear if we divide both sides of the condition by k. Therefore, we have that all possible $(\boldsymbol{w}, w_0)$ that define the optimal decision boundary from 1.1 is $([0\ k]^T, k)$, for some positive constant k. In other words, the second component of $\boldsymbol{w}$ must equal w_0, and the first component of $\boldsymbol{w}$ is 0.

5. The particular solution that will be optimal will yield a margin of size 1. We know that that

$$\text{margin}(x_n|\phi, \boldsymbol{w}, w_0) = \frac{y_n(\boldsymbol{w}^T\phi(x) + w_0)}{\|\boldsymbol{w}\|_2}$$

To find the optimal $\boldsymbol{w}$, we note that we only need to check margin points. This is because given an optimal $\boldsymbol{w}$, the denominator of the above expression is fixed, and for inputs x_n that are data points farther from the decision boundary than the margin points, the numerator will be larger. Thus, since the data is also linearly separable with a hard margin formulation, to find the optimal $\boldsymbol{w}$ and w_0, we only need to really check with one margin point to ensure that the size of the margin is indeed one and use that corresponding $\boldsymbol{w}$ to be the optimal $\boldsymbol{w}$. This is because if this decision boundary is optimal for one margin point, it will be optimal for all margin points, by the definition of margin points. Since we want the optimal margin of size 1, we can solve for $\boldsymbol{w}$ as follows:

$$\text{margin}(x_n|\phi, \boldsymbol{w}, w_0) = \frac{y_n(\boldsymbol{w}^T\phi(x) + w_0)}{\|\boldsymbol{w}\|_2} = \frac{1}{\|\boldsymbol{w}\|_2} = 1$$

$$\|\boldsymbol{w}\|_2 = 1$$

Then since we know the direction of $\boldsymbol{w}$, and we now know the optimal magnitude of $\boldsymbol{w}$, we know that the optimal $\boldsymbol{w}$ altogether is $[0\ 1]^T$.

6. Based on the Part 5 discussion, we can plug in one margin point with the optimal $\boldsymbol{w}$ already found to get the corresponding optimal w_0 for our decision boundary. Let's choose margin point $x = (-2, 1)$. We see that $\phi(x) = (-2, 0)$. For an optimal decision boundary, the discriminant at this margin point will be equal to 1. Then we see that

$$\boldsymbol{w}^T\phi(x) + w_0 = (1)(0) + w_0 = 1$$

From this, we see that $w_0 = 1$ works, as expected based on our result for Part 4, the optimal $\boldsymbol{w}$ found in Part 5, and the verification we just did in this part with a margin point.

7. Since the data is linearly separable and we have margin points lying on the margin boundary as well as a few points farther from the margin boundary, the only support vectors correspond to those margin points: $\{(-2, 1), (-1, -1), (0, 1), (1, -1), (2, 1)\}$. We can plug them each into our discriminant to very that the constraint we found is tight, meaning that the value of the discriminant at these points is exactly 1 for positively classified points and exactly -1 for negatively classified points. We see the following:

$h(x = -2|\phi, \boldsymbol{w}, w_0) = (0)(-2) + (1)(0) + 1 = 1$

$h(x = -1|\phi, \boldsymbol{w}, w_0) = (0)(-1) + (1)(-2) + 1 = -1$

$h(x = -0|\phi, \boldsymbol{w}, w_0) = (0)(0) + (1)(0) + 1 = 1$

$h(x = 1|\phi, \boldsymbol{w}, w_0) = (0)(1) + (1)(-2) + 1 = -1$

$h(x = 2|\phi, \boldsymbol{w}, w_0) = (0)(2) + (1)(0) + 1 = 1$

The value of the discriminant at each of those x's exactly equals the $y \in \{1, -1\}$ of that data point, as desired.

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. Run K-means for 5 different restarts for different values of $K = 5, 10, 20$. Make a plot of the final K-means objective value after your algorithm converges (y-axis) v. the values of K (x-axis), with each data point having an error bar. To compute these error bars, you will use the 5 final objective values from the restarts for each K to calculate a standard deviation for each K.

   How does the final value of the objective function and its standard deviation change with K? (Note: Our code takes 10 minutes to run for this part.)

3. For $K = 10$ and for 3 random restarts, show the mean image (aka the centroid) for each cluster. To render an image, use the pyplot `imshow` function. There should be 30 total images. Include all of these images as part of a single plot; your plot must fit on one page.

4. Repeat Part 3, 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 3: How do the centroids visually differ? Why?

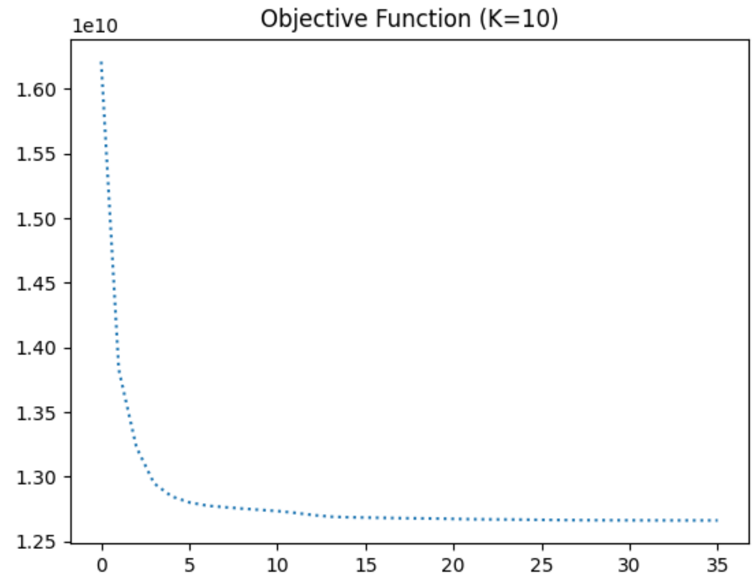
5. Implement HAC for min, max, and centroid-based linkages. Fit these models to the `small_dataset`. For each of these 3 linkage criteria, find the mean image for each cluster when using 10 clusters. Display these images (30 total) on a single plot. How do these centroids compare to those found with K-means? **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.

6. For each of the 3 HAC linkages (max/min/centroid), plot "Distance between most recently merged clusters" (y-axis) v. "Total number of merges completed" (x-axis). Does this plot suggest that there are any natural cut points?

7. For each of the max and min HAC linkages, 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?

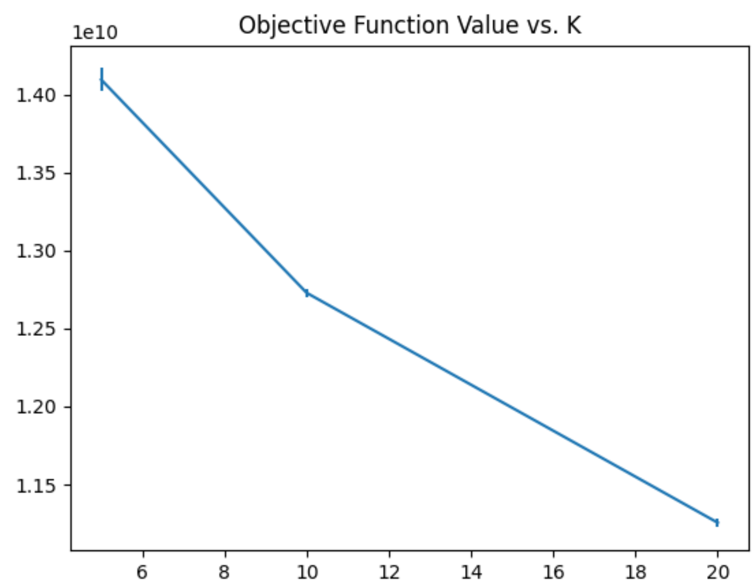
8. For your K-means with $K = 10$ model and HAC min/max/centroid models using 10 clusters on the `small_dataset` images, use the `seaborn` module's `heatmap` function to plot a confusion matrix of clusters v. actual digits. This is 4 matrices, one per method, each method vs. true labeling. The cell at the ith row, jth column of your confusion matrix is the number of times that an image with the true label of j appears in cluster i. How well do the different approaches match the digits? Is this matching a reasonable evaluation metric for the clustering? Explain why or why not.

Solution



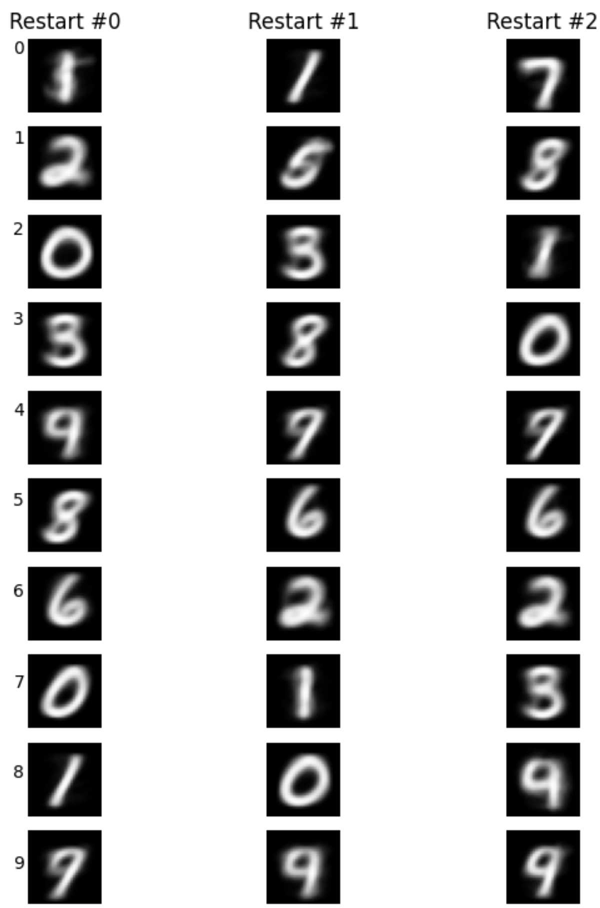
1.

Above is the graph for the K-means objective function (the residual sum of squares) as a function of iterations (x axis) for K=10. We can verify visually from the plot as well as in my code (if statement) that the objective function never increases.



2.

As K increases on the x-axis increases, the final value of the objective function (y-axis, on average) and the standard deviation of the final value of that function both decrease.

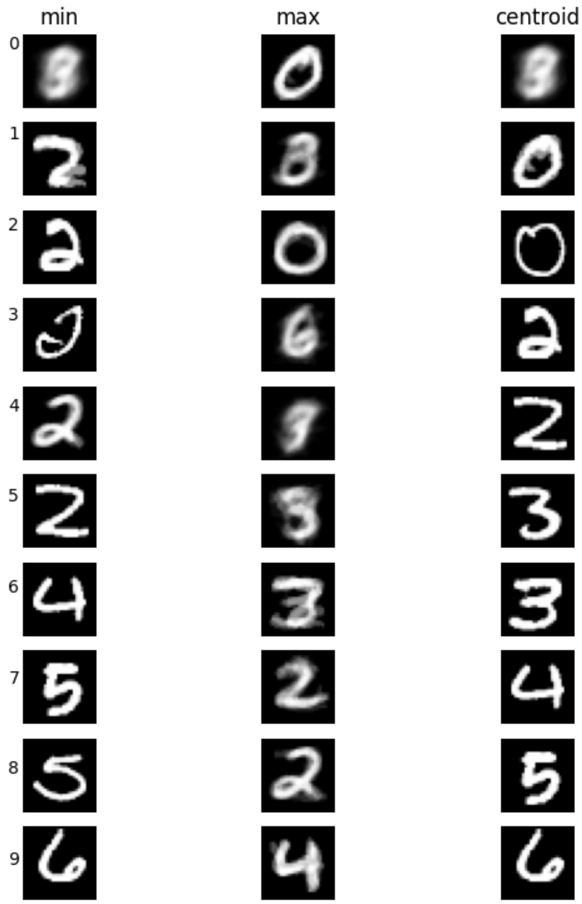


3.

On the y axis, the indices (0-9) printed for each row correspond to the cluster number. For example the first row is for cluster 0 for each restart.

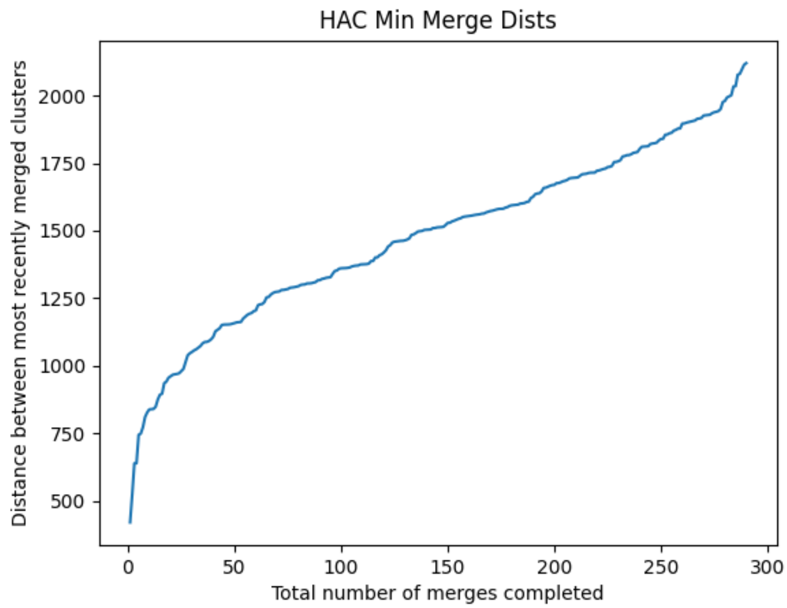
4.

The centroids appear "cloudier" around the edges. Our standardization has essentially decreased the variance of each pixel. If we have a dataset with high variance, perhaps corresponding to some pattern such as varying units, KMeans will more likely cluster together data with lower variance, which might end up clustering data by unit rather than by the features expressed in the units of the data. Therefore, standardization removes this complicating factor (nonmeaningful variation in the data). Here, since we may have some "units" variation regarding how "brightly" each digit was written, standardization has allowed "light" areas of images to seep into the standardization means and variances, resulting in lighter/cloudier mean images, but has also allowed for more efficient separation of the types of digits, whereas without standardization we might see "repeat" mean images. For example we might see a cluster of "brightly written" 9's along with another cluster of what looks like 9's, but less "brightly written" or perhaps slanted differently. This variation in where the image is bright is taken care of with standardization, which helps us "center" our data and ignores brightness variation.

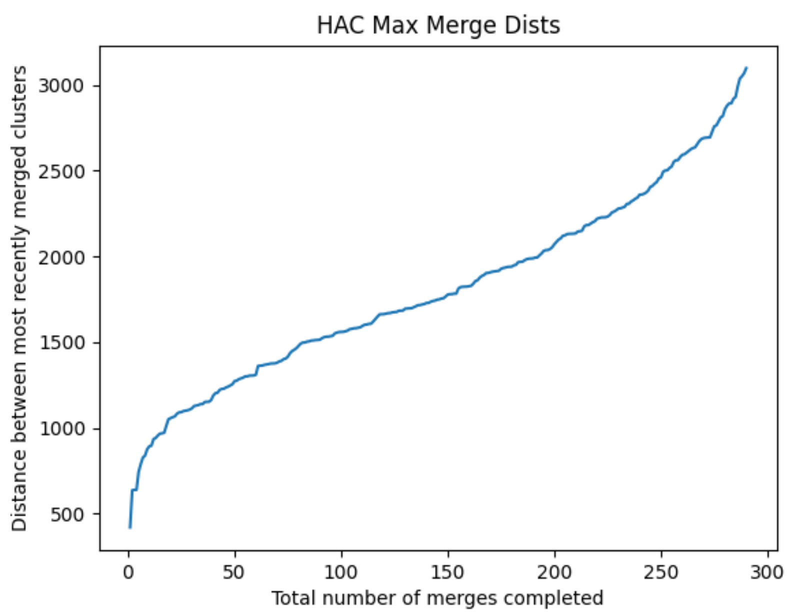


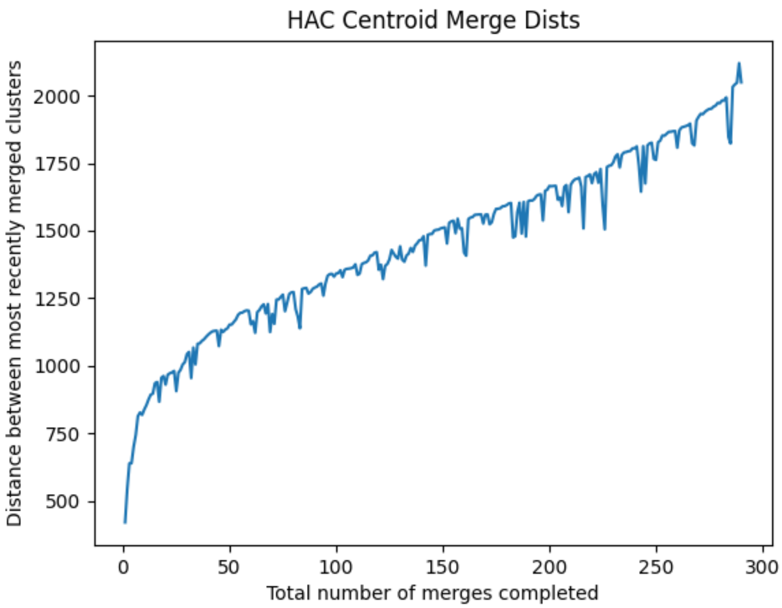
5.

The centroids for HAC show less distintly learned classes. We can see this since regardless of distance metric, HAC has learned some very distinct types of handwriting style for a few numbers, but everything else gets clustered into "clouds" that must be very unlike the popular, distinct numbers/handwriting styles for those numbers. KMeans was better able to find the distinct types of numbers using its "mean" functionality, whereas since HAC is looking to merge similar images, it has learned to distinguish unique handwriting patterns in distinct number shapes, but is rather confused for number shapes that are more similar, such as 9, 1, 8, 3, etc.

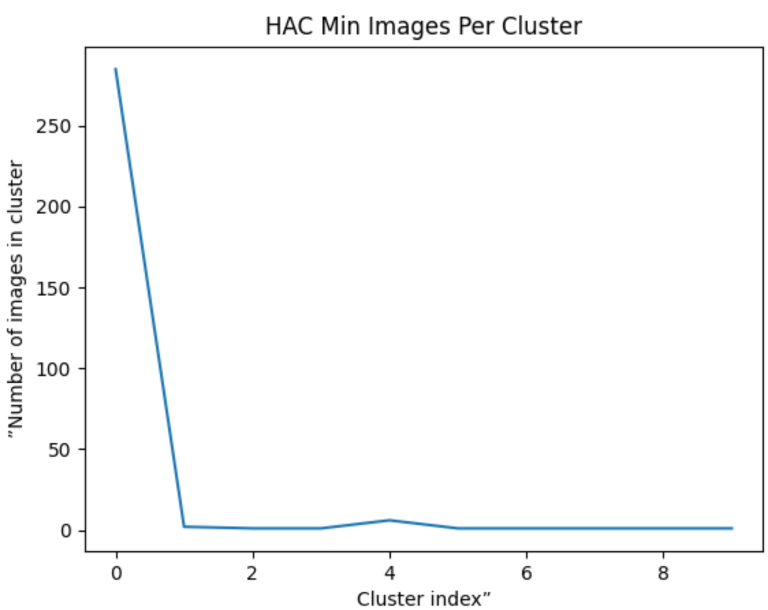

HAC Min Merge Dists
Distance between most recently merged clusters
500
750
1000
1250
1500
1750
2000
0
50
100
150
200
250
300
Total number of merges completed

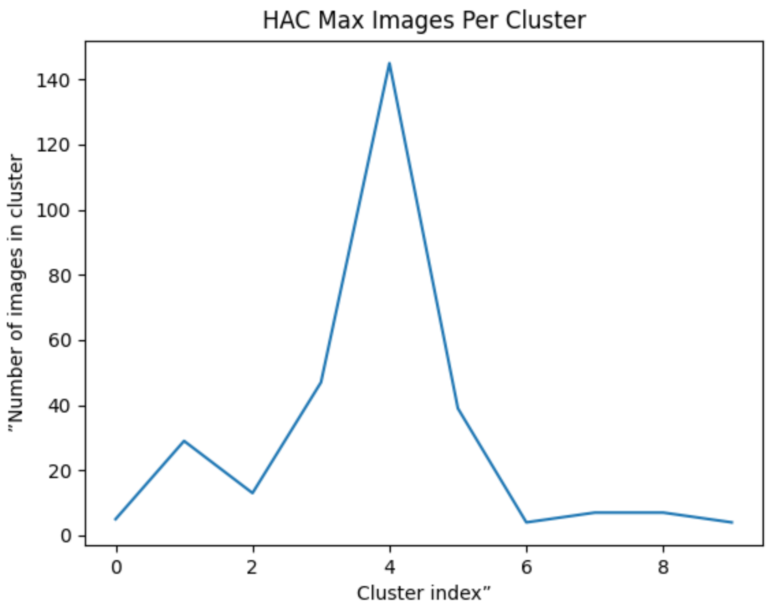
6.


HAC Max Merge Dists
Distance between most recently merged clusters
500
1000
1500
2000
2500
3000
0
50
100
150
200
250
300
Total number of merges completed

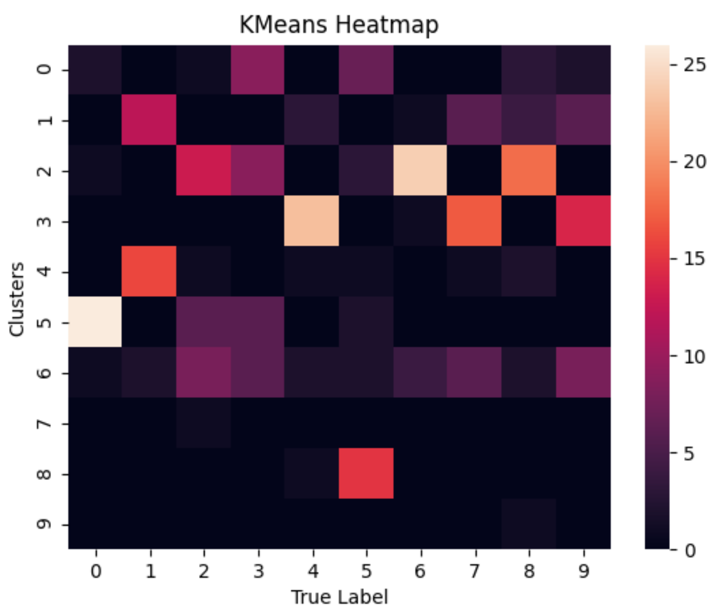


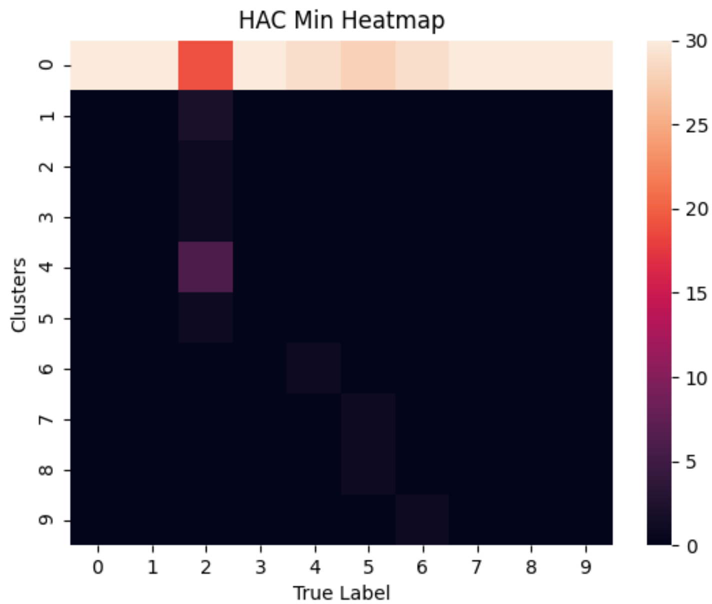
The graph shows that an interesting cut point occurs at around 100-150 merges, which is where the distance between most recently merged clusters starts increasing at a relatively constant rate rather than increasing slower. This signals that at around 100-150 merges, the clusters formed are similarly distanced from each other and more stable, snake-like merging will begin occurring.

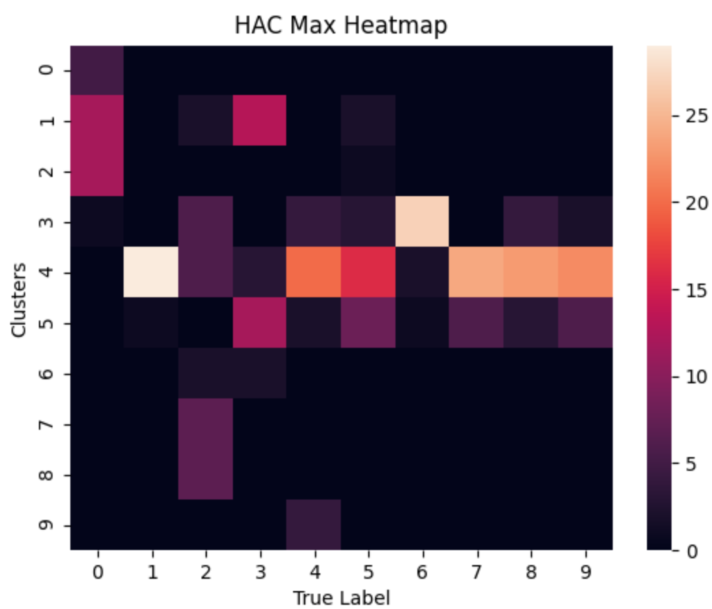
7. 

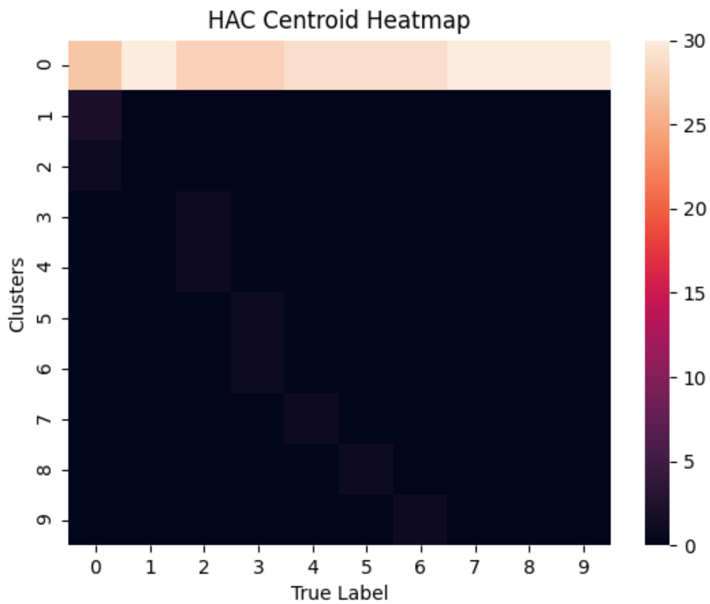


With the min-linkage, we see more images bucketed into one cluster of "unknowns". As mentioned previously, HAC in general seems to favor learning clusters of uniquely handwritten-style digit types, and since HAC-min is merging the closest clusters based on most similar images, this issue is more pronounced. HAC-max is merging the closest clusters based on most different images in those two clusters, which leads to more dispersion, allow HAC-max to roughly learn the most types of digits of all the HAC methods.

8. 


HAC Min Heatmap
Clusters
True Label
0
1
2
3
4
5
6
7
8
9
30
25
20
15
10
5
0


HAC Max Heatmap
Clusters
True Label
0
1
2
3
4
5
6
7
8
9
25
20
15
10
5
0



A heatmap with very "bright" values along the diagonals and "dark" values everywhere else (based on my legend) would typically signal good classfication. However, since we used unsupervised methods, its not necessarily the case, for example, that the first cluster was learned to correspond to "class 0", etc. Generally, we can see that, as we previously observed, HAC does a poor job of learning types of digits and has bucketed most images together in one cluster of "indistinguishables". HAC-max does the best job of all the HAC's, but overall is still not very strong. KMeans does a much better job, and if we were to "rearrange" the rows of the heatmap, we can see that for each column of "true label", we roughly see on row with a very bright value. If we were to refer to the mean images we found, we can see that we could almost roughly rearrange the rows to have a "bright" diagonal, signally that there has been decent separation of images into classes based on digit type and that the 10 different digit types have been decently learned, especially compared to HAC methods.

However, evaluation based on exact matching of the heatmap to having that "bright diagonal" won't be proper evaluation since, as already mentioned, cluster numbers don't map directly to class labels. Thus, we need to use both the heatmap and the mean images to see if we can "rearrange" the heatmap/notice distinguishable classes amongst the mean images to evaluate how well our unsupervised method has done.

Problem 3 (Ethics Assignment, 15pts)

Read the article "Amazon Doesn't Consider the Race of Its Customers. Should It?". Please write no more than 1 concise paragraph each in response to the below reflection questions. We do not expect you to do any outside research, though we encourage you to connect to lecture materials and the reading for the module where relevant.

1. Some people think that Amazon's process for determining which neighborhoods would receive same-day delivery was wrongfully discriminatory, but others disagree. Based on our definitions and discussions from lecture, do you believe that Amazon's same-day delivery process was wrongfully discriminatory? Explain your reasoning.

2. Basing decisions about how to treat others on social group membership often strikes us as being wrongfully discriminatory. For example, most people would say that refusing to hire someone because they are a woman is wrongful discrimination, at least under normal circumstances.

   However, there are some cases in which some people argue that social group membership *should* be taken into consideration when deciding how to treat others. The title of the article poses the question: Do you think that should Amazon consider the race of its customers in its same-day delivery processes? If so, then how?

3. There are many different technical definitions of fairness in machine learning. In this problem, we'll introduce you to the intuition behind two common definitions and invite you to reflect on their limitations.

   Say that Amazon decides to develop a new algorithm to decide its same-day delivery coverage areas. Given your machine learning expertise, Amazon hires you to help them.

   Assume for simplification that Amazon's same-day delivery coverage algorithm f takes as input x, features about an individual Amazon user's account, and outputs binary class label y for whether or not same-day delivery will be offered to user x. User x also has (often unobserved) sensitive attributes a. In this example, we will assume a is a binary label so that $a = 1$ if the user is not white, 0 otherwise.

   One technical notion of algorithmic fairness is called "group fairness"[a]. An algorithm satisfies group fairness with respect to protected attribute a if it assigns the same proportion of positive labels to the group of white and the group of non-white Amazon users. In other words, if 50% of white users have access to same-day shipping, then 50% of non-white users should have access to same-day shipping too.

   What are some limitations or potential issues that may arise with enforcing this definition of fairness in practice? Are there ways that a classifier that satisfies group fairness may still result in discriminatory outcomes?

4. Another technical notion of algorithmic fairness is called "individual fairness"[b]. An algorithm satisfies individual fairness if for all pairs of users x_1 and x_2 that are similar *without taking into consideration their race* a, the algorithm will assign similar probabilities for the attaining the positive label (roughly, $x_1 \sim x_2 \Rightarrow p(x_1) \sim p(x_2)$)[c]. In other words, if two individuals have almost-identical user profiles, then they should both be eligible for same-day shipping, even if one is white and the other is non-white.

   What are some limitations or potential issues that may arise with enforcing this definition of fairness in practice? Are there ways that a classifier that satisfies individual fairness may still result in discriminatory outcomes?

[a]Group fairness is also sometimes referred to as "independence" or "demographic parity". https://fairmlbook.org/classification.html

[b]https://arxiv.org/pdf/1104.3913.pdf

[c]This is an intuitive description of individual fairness (likewise with group fairness above) rather than a precise formalization.

Solution

1. Based on our discussions, I firstly think that this case is an instance of disparate impact, since the outcome of the policy had discriminatory aspects that did not arise from animus or internal biases stemming from Amazon's intentions or stereotypes. Identifying whether discrimination is indeed wrongful requires us to consider procedural and substantive fairness. I believe this case involves procedural unfairness because the policy led unequal treatment of customers based on race. More concretely, it led to Prime members living in certain districts, where the population in these districts had a high proportion of black residents, to pay the same for Amazon's service but to receive inferior benefits (no same-day shipping) as their counterparts living in predominantly-white districts. I also believe this case involves substantive unfairness because it reinforces lack of access to resources for a racial group that is historically disadvantaged socially, fiscally, etc. Therefore, I believe that Amazon's same-day delivery process was wrongfully discriminatory.

2. I think that this question gets at a key issue with how we define "consider" when applied to how an entity "considers" a certain factor that could or could not be discriminatory. For services like delivery, there does not seem to be any clear, inherent connection with any variable that should rationally be used to discriminate (i.e. no rational argument exists for why race should be the most prominent factor when determining whether a person should receive the delivery service). Though this could potentially be argued against, I think stronger arguments exist for considering factors such as distance to the nearest fulfillment center, percentage of Prime members in the area, etc. The issue then arises that we have correlated "proxy variables", which if not carefully considered, could potentially lead to discriminatory policies. I believe that any algorithm/process implemented by an entity should be strictly evaluated after design to see if any systematically discriminatory actions might be taken by a person using this algorithm to make decisions. Assuming that our datasets are already inherently biased to self-reinforce potentially problematic status-quos, and the fact that it is virtually impossible to remove any source of unfairness (i.e. controlling for equal access based on race might not be fair given another metric), Amazon should be held responsible for thorough validation of its algorithms, which might require "considering" race.

3. Some limitations might surround what 50% of each group we choose to give the service to. Racial group is just one factor, and there are still many other factors such as income, gender, address, etc. that must be considered to decide who gets service. One way that this classifier might still be discriminatory is that since 50% of each racial group isn't covered, there are potentially members of each group who might need the service more than those who currently receive it. For example, we might want to provide free same-day shipping to those who use Amazon's service more frequently or who have lower household income. Both of these policies might provide more access to those who can benefit the most from the service who don't currently have it.

4. One major limitation is to decide which variables, such as race, gender, etc. that we want to disregard in this individual fairness heuristic. We might also want to consider which of these factors is most important/should be weighted most heavily if we can only select for a finite number. This methodology also might be blind to overall discrimination of the group if we only consider outcomes on the individual level. Same probability of having access to a service doesn't correspond to true assignment of equal access. For example, race and household income are likely correlated in the data, since due to transfers of intergenerational wealth, white users are likely more wealthy that black users. It therefore is possible that the algorithm might never make comparisons between people of different races, because other correlating factors have made it so that comparisons are only made between people for whom discrimination is less of an issue since the individuals are unlikely to be strongly adversely impacted.

Problem 4 (Bonus Ethics Assignment, 0pts)

Estimated total time for completion: 45 minutes.

In our lecture from class, we discussed philosophical and legal frameworks to examine algorithmic discrimination. But these aren't the only frameworks! A growing body of work in the humanities and social sciences, particularly in feminist studies, critical race studies, sociology, anthropology, and the history of science has emerged to study the social consequences of machine learning.

In this bonus problem, you will be introduced to one framework inspired by scholarship in science and technology studies. To complete the below questions, first watch the 28-minute 2019 NeurIPS talk "The Values of Machine Learning" by Ria Kalluri. Please write no more than 1 paragraph in response to each of the below reflection questions:

1. In their talk, Ria discussed opportunities for shifting power to each of four possible stakeholders: an input source, data source, expert, and decision-maker. Choose one of these stakeholders, and discuss one possible way we as machine learning practitioners and model-builders could shift power to them.

2. What do you think will it take to achieve a world where AI can shift power towards historically marginalized communities? What obstacles or barriers stand in between our current world and your own AI dreams?

Solution

1. One key stakeholder is the data source, whose key concerns surround transparency and privacy. An aspect of data acquisition that I find very compelling is that despite how crucial it is to success of all the other stakeholders, the value placed on acquiring it from individuals is often low (think Mechanical Turk). Often people provide data for free, not knowing the pipelines that it goes through and the value that it generates, usually not transferred back to them. Therefore, the current practices serve to benefit those with the power and resources to think about models, seek out data, and ultimately make decisions are disproportionately benefiting on the margin from data sources. One way to shift power back to this stakeholder that has been suggested is to adequately compensate them. Another is to advocate for transparency in its usage and limits on how far the privacy of the data can be extended. There exist rules on these areas, but most are currently quite lax. As ML practitioners, I think its important to consider both how we think about reporting our models' impact to data sources but also how to incentivize data sources to provide information about potentially high social impact areas that aren't well-sourced right now. This type of intervention could power the types of models and studies that might create more change, but which existing power structures don't enable proper collection of currently.

2. I think that in our current world, especially with the funneling of wealth and power to a few elite at the expense of the majority, power structures and the status quo are deeply entrenched. Historically, change occurs when a key tipping point is reached, meaning that policy is very reactionary. This is an idea I think we can see in the fields of climate change, public health, etc., all the way down to the simple way individual people make decisions on, say, time management or food choices. The bottom up surveillance structure, or even just democratized access to resources, transparency, information, and more are needed to keep those in power accountable, but to do that requires CS as a field and all others seeking to subvert the status quo to accumulate a large pile of evidence. This can even be seen in the Amazon free-delivery example, where a discriminatory practice was not uncovered until substantial data and analysis was completed, time invested to change the policy, and consistent vocality on the issue. As AI researchers/students, it becomes very crucial that some of us go into the world not prioritizing the goals of those already in power (i.e. profit, opacity, ...) and work actively to audit the assumptions by which we accept the world today, and try to define and reach a global optimum.

The power to define that optimum and the resources to make it reality are what we are working hard to define in a fair, equal way, and perhaps more than anything we need clear, justifiable evidence of why one definition should prevail over another; otherwise, the status quo exploits ambiguity for its own gain and maintains itself through the lack of cohesion acting against it – an unreachable tipping point. Therefore, I think it is my and all our responsibilities to work hard to define concretely the issues, the definitions, the change, the reasoning in a way that harnesses the power of the minority and the marginalized to be able to overcome existing barriers and achieve "AI dreams".

Name

Kathryn Wantlin

Collaborators and Resources

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

Yash, Andrew, Rylan, Richard, Lucy's OH

Calibration

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

20 hours