Your Name
email@fas.harvard.edu
CS181-S16

Assignment #4
Due: 5:00pm Feb 1, 2016

Collaborators: John Doe, Fred Doe

Homework 4: Clustering

There is a mathematical component and a programming component to this homework. Please submit ONLY your PDF to Canvas, and push all of your work to your Github repository. If a question requires you to make any plots, please include those in the writeup.

Problem 1 (The Curse of Dimensionality, 4pts)
In d dimensions, consider a hypersphere of unit radius, centered at zero, which is inscribed in a hypercube, also centered at zero, with edges of length two. What fraction of the hypercube's volume is contained within the hypersphere? Write this as a function of d. What happens when d becomes large?

Solution

- Any point in the sphere is contained within the hypercube. Proof by contradiction: let x be a point of the sphere not within the hypercube. $\exists i$ s.t. $x_i > 2$. Therefore $\|x\|_2 \geq |x_i| > 2$. Contradiction.
- Therefore, the fraction of the hypercube's volume contained within the hypersphere is just the ratio of their volumes.
- The volume of a hypercube with edges of length 2 is $V_c(d) = 2^d$
- The volume of a hypersphere of unit radius is $V_s(d) = \frac{\pi^{\frac{d}{2}}}{\Gamma(\frac{d}{2}+1)}$ where Γ is Euler's Gamma function.

$$R = \frac{\pi^{\frac{d}{2}}}{2^d \Gamma(\frac{d}{2}+1)} \tag{1}$$

which goes to 0 when $d \to \infty$, because $\frac{\sqrt{\pi}}{2} < 1$ and Γ is an increasing function for $d \geq 1$.

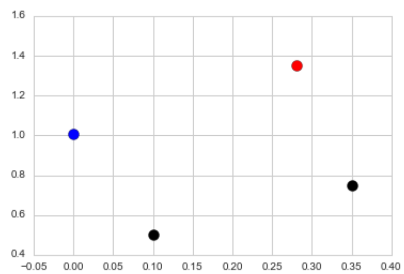
Check 1.1: You justified that the answer was the ratio of volumes of each object.

Check 1.2: You correctly recalled the volume contained within a hypersphere and hypercube in d dimensions.

Check 1.3: You gave the correct limit and gave a valid argument to support your claim.

Problem 2 (Norms, Distances, and Hierarchical Clustering, 5 pts)

Consider the following four data points, belonging to three clusters: the black cluster $((x_1, y_1) = (0.1, 0.5)$ and $(x_2, y_2) = (0.35, 0.75))$, the red cluster $(x_3, y_3) = (0.28, 1.35)$ cluster, and the blue cluster $(x_4, y_4) = (0, 1.01)$.



At each step of hierarchical clustering, the two most similar (or least dissimilar) clusters are merged together. This step is repeated until there is one single group. Different distances can be used to measure group dissimilarity. Recall the definition of the l_1, l_2, and l_∞ norm:

- For $\mathbf{x} \in \mathbb{R}^n$, $\|\mathbf{x}\|_1 = \sum_{i=1}^n |x_i|$
- For $\mathbf{x} \in \mathbb{R}^n$, $\|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^n x_i^2}$
- For $\mathbf{x} \in \mathbb{R}^n$, $\|\mathbf{x}\|_\infty = \max_{i=1}^n |x_i|$

Also recall the definition of single-link distance, complete-link distance, and average-link distance between two clusters:

- Single-link clustering: for clusters G and H, $d_S(G, H) = \min_{i \in G, j \in H} d(i, j)$
- Complete-link clustering: for clusters G and H, $d_C(G, H) = \max_{i \in G, j \in H} d(i, j)$
- Average-link clustering: for clusters G and H, $d_A(G, H) = \frac{1}{|G||H|} \sum_{i \in G} \sum_{j \in H} d(i, j)$

Warm up question. Draw the 2D unit sphere for each norm, defined as $\mathcal{S} = \{x \in \mathbb{R}^2 : \|x\| = 1\}$. Feel free to do it by hand, take a picture and include it in your pdf.

Main question. For each norm (l_1, l_2, l_∞) and each clustering method (single, complete, or average link clustering), specify which 2 clusters would be the first to merge.

Solution

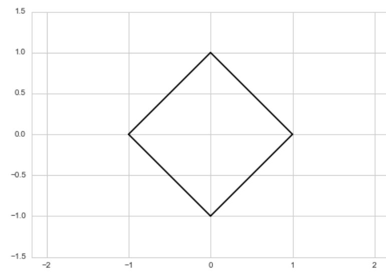


Figure 1: 2D sphere for the $l1$ norm

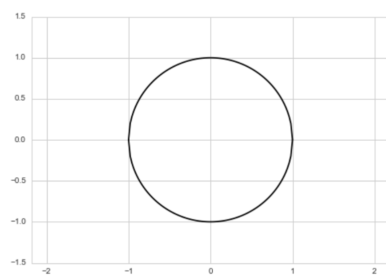


Figure 2: 2D sphere for the $l2$ norm

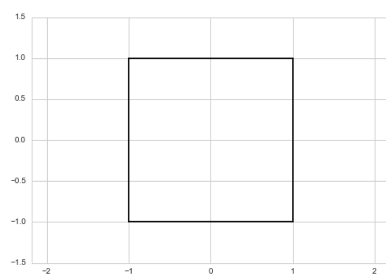


Figure 3: 2D sphere for the l_∞ norm

clustering type \ norm	l_1	l_2	l_∞
single-link clustering	Black and Blue	Black and Blue	Red and Blue
complete-link clustering	Black and Blue	Red and Blue	Red and Blue
average-link clustering	Black and Blue	Red and Blue	Red and Blue

Table 1: The clusters that will be merged first

Check 2.1: Your plots of the 2D sphere for each norm match the figures above.

Check 2.2: There are 9 answers that should have been given in total, 3 norms $\times$ 3 clustering mechanisms. Report the number of MISTAKES that you made (i.e number of times your answer does not match the one in Table 1.)

K-Means [15 pts]

Implement K-Means clustering from scratch.[1]. You have been provided with the MNIST dataset. You can learn more about it at `http://yann.lecun.com/exdb/mnist/`. The MNIST task is widely used in supervised learning, and modern algorithms with neural networks do very well on this task. We can also use MNIST for interesting unsupervised tasks. You are given representations of 6000 MNIST images, each of which are 28×28 handwritten digits. In this problem, you will implement K-means clustering on MNIST, to show how this relatively simple algorithm can cluster similar-looking images together quite well.

Problem 3 (K-means, 15pts)

The given code loads the images into your environment as a 6000x28x28 array. Implement K-means clustering on it for a few different values of K, and show results from the fit. Show the mean images for each class, and by selecting a few representative images for each class. You should explain how you selected these representative images. To render an image, use the numpy imshow function, which the distribution code gives an example of. Use squared norm as your distance metric. You should feel free to explore other metrics along with squared norm if you are interested in seeing the effects of using those. Also, your code should use the entire provided 6000-image dataset (which, by the way, is only 10% of the full MNIST set).

Are the results wildly different for different restarts and/or different K? Plot the K-means objective function as a function of iteration and verify that it never increases.

Finally, implement K-means++ and see if gives you more satisfying initializations (and final results) for K-means. Explain your findings.

As in past problem sets, please include your plots in this document. There may be tons of plots for this problem, so feel free to take up multiple pages, as long as it is organized.

Solution

You should find that with K = 10, that you get 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. However, the representative images for each cluster should match. The staff solution finds 5 representative images for each cluster, and they all are the same digit for each cluster. Most likely, you will have found that the final results vary slightly in the objective function value, but not too much in the actual representative images. 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

[1] That is, don't use a third-party machine learning implementation like `scikit-learn`; `numpy` is fine.

two different handwriting styles for it. Also, you should have found that Kmeans objective never increases, which you showed via a plot visualization.

Lastly, the staff solution found that KMeans++ usually gave an initialization with a lower objective function value, but did not help much in the final objective values, or the representative images. Please see the solution code.

> **Check 3.1**: You coded up a reasonable way of finding representative images for each cluster, and indicated what the method was. Examples are the 5 images nearest the mean, or 5 random images from the cluster, along with an explanation.

> **Check 3.2**: You should have found that for K=10, there is a cluster for each digit. You showed this by including a few representative images per cluster.

> **Check 3.3**: You plotted the objective function, and showed that it is never increasing.

> **Check 3.4**: You implemented K-Means++. You likely found that this didn't help much in the final answer, since Kmeans already does decently well. You should include some analysis of what the differences are. Good answers include noting that KMeans++ tended to take fewer iterations to converge, had lower objective function values at the beginning, or had around the same or lower objective values at the end. It was generally not much worse than KMeans.

Problem 4 (Calibration, 1pt)
Approximately how long did this homework take you to complete?