Machine Learning Project, Assignment 1, Report

Mathijs Afman

Feb 29, 2024

1 Introduction

For this assignment, the goal was to build a K-means clusterer that clusters images of written digits with the labels 0 through 9. The K-means clusterer works by picking random data-points as initial cluster centroids, and then iteratively assigning data-points and updating these centroids until convergence. After this each cluster is assigned a label, and all data-points in this cluster are predicted to have this label. This resulted in a list of predicted labels, which could then be checked against true labels to produce a confusion matrix. In this report I go in depth about the data itself, the methods used to perform clustering, and highlight the strong and weak points of the clusterer with the confusion matrix.

2 Data

The images are in gray-scale and have a resolution of 28x28 for a total of 784 pixels. An image is represented as an array of values ranging from 0 to 1, where a one represents a white pixel and a zero a black pixel. Therefore the array had the length of the total resolution of the image. An example of such an image can be seen in Figure 1.

The data was pretty well balanced, with the most common label occurring 117 times and the least common label occurring 87 times. In total there were 1000 images with 10 unique labels (0-9).

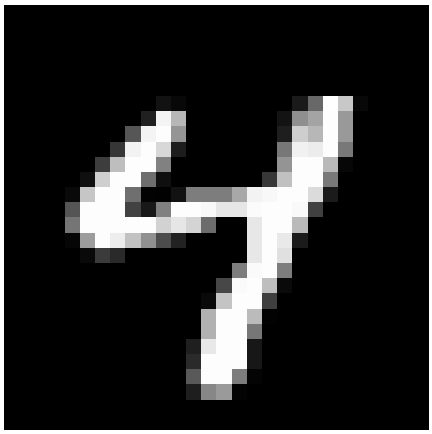

Figure 1: Written Digit

3 Method

Little pre-processing was needed because the images were already in array format. I just had to pick the right data-type in the program which fit my needs.

The K-means clusterer works assigning data-points to centroids, which then get labeled. Initially the centroids are randomly chosen from the data-points. Then the clusterer repeats the following two steps a specified number of times (In this case, 100 times).

1. Assign all data-points to the cluster with the closest centroid.
2. Update the centroids to be the mean of the data-points in the cluster.

In this system distance is measured with the euclidean method. After the specified number of iterations the clusters can be assigned a label. For every cluster the entire cluster gets the label that occurs the most often among the data-points in the cluster. This then results in a list of predicted labels, which can be scored with adjusted rand score and interpreted by use of a confusion matrix.

4 Results

4.1 Adjusted Rand Score

For the K-means clusterer an adjusted rand score between 0.25 and 0.40 was expected. This score can vary a lot between runs because of the random initialization of the centroids. Therefore we take the average of 5 runs, which resulted in a value of 0.311. This isn't particularly high but it was expected. To see why the score wasn't higher we can take a look at the confusion matrix.

4.2 Confusion Matrix

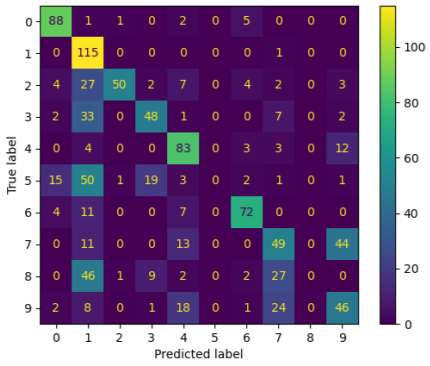

Figure 2: Confusion Matrix

Fig. 2 is a confusion matrix of a single run of the clusterer. All values across the diagonal from the top left to the bottom right are correct predictions (true positives) and everything not on this diagonal are incorrect predictions. The matrix shows that the labels 1 and 4 often get confused for other labels. This is due to their similarity to other labels in the way we chose to measure distance. The clusterer only predicted a 1 to be something else once, but this is probably because the clusterer predicted almost a third of all data-points to be labeled 1.

Something I also noticed immediately is that in this run the labels 5 and 8 were not predicted. This might be due to the random initialization of the cluster centroids. If two centroids are initialized close together they might converge to a similar point. This causes one of the two centroids to have no data-points assigned to its cluster.

It might also be due to the way that labels are assigned to clusters. As we choose the most common label in a cluster to be the label for the entire cluster, two clusters might be assigned the same label. This causes other labels to have no predictions at all.

I speculate that another reason that the clusterer performs poorly, is the amount of black pixels around the most images. The amount of pixels that carry information that we are interested in is pretty small, compared to the size of the total image. Of course the images need to have the same resolution for this method to work, but it might be interesting to look into this in the future.

5 Conclusion

In conclusion, the K-means clusterer lived up to the expectations, but didn't perform particularly well due to the method of initializing and assigning labels. A solution to this could be to develop a method that ensures the convergence of clusters for *every* unique label. In the future it might be interesting to test what effect different ways of measuring distance might have on this clusterer. It could also be worth it to go deeper into pre-processing the image and removing as much black space around the digit as possible before using it for the clusterer.