K-means Clustering Centroids as Representations of MNIST

Kevin Buhler
University of Colorado Boulder

Nate Weaver
University of Colorado Boulder

1 Introduction

In CSCI 4022 we have covered a multitude of different techniques, ranging from clustering to the PageRank algorithm. However, we haven't gone over a technique that humans naturally do every day: image classification.

Image classification is increasingly prevalent in our daily lives. In these areas, the ability to accurately classify images facilitates everything from automated product tagging to enhanced user interactions through personalized content in apps. Traditionally, when we consider the task of image classification, sophisticated models such as convolutional neural networks (CNNs) or vision transformer architectures come to mind. These models are known for their high accuracy in tasks like number recognition but come with significant drawbacks like computational intensity and the inability to understand their decision-making processes.

Given these challenges, exploring alternative, less conventional methods for image classification becomes appealing. Instead of these large scale models, could we try an unconventional technique from CSCI 4022 to classify images, namely K-mean clustering in an efficient and interpretable manner. K-means clustering can segment images based on pixel similarity which allows for a form of image classification that is not only efficient but also more interpretable. This interpretability is crucial in applications where understanding the model's decision making processes is as important as the decision itself, such as in medical diagnostics or security-related image analysis.

Moreover, using K-means as a clustering approach before applying more complex classification methods can enhance performance by simplifying the input features and reducing noise. This preprocessing step can make subsequent analyses both faster and more accurate, providing a scalable solution that is both simple and efficient.

We posit that with a large amount of image data then K-means clustering's centroids, when plotted, will resemble the most common item assigned to the cluster.

2 Data

In the computer vision field, one of the most famous image dataset is MNIST. The MNIST dataset (Modified National Institute of Standards and Technology) data is public and is easily accessible through Tensorflow or Keras in their dataset module, but is also available on general sites like Kaggle as well.

MNIST consists of 70,000 28x28 pixel images of hand written numbers ranging from the digits 0-9. In the dataset, each image is labeled with the digit it represents, providing a straightforward, single-feature dataset used for training various image processing systems. This dataset is typically used in machine learning validation testing as it is split into two sets, 60,000 digits in the training dataset and 10,000 in the testing. It has long been used to benchmark the performance of novel machine learning architectures and we can use it to benchmark results of our k-means image classification algorithm.

After obtaining good results with MNIST alone, we decided to do some preliminary experiments with the Fashion-MNIST dataset. Like MNIST, Fashion-MNIST consists of 70,000 28x28 images but of common clothing items instead of digits. It –like MNIST– also has 10 classes so we were able to use it without any modifications to our algorithm.

3 Real-world

In the real world, the problem of centroid initialization in K-means clustering is important in various industries with applications like customer segmentation, inventory categorization, and image recognition. Effective clustering helps companies optimize marketing strategies, improve customer service, and enhance decision-making processes. Companies like Amazon and Netflix, for example, use clustering to personalize recommendations, showing the commercial significance of clustering algorithms.

With a brief literature review on Google and Google Scholar we found a couple of papers who have employed our method.
Princeton paper: https://www.cs.princeton.edu/courses/archive/fall18/cos324/files/kmeans.pdf
Papers:https://scholar.google.com/scholar?hl=en&as_sdt=0%2C6&q=Using+k-means+clustering+to+compress+an+image+database&btnG=

However these methods primarily use MNIST, so we find new results by using Fashion-MNIST.

4 Exploratory

Preliminary work we did to establish our hypothesis was based on past understanding of K-means was mostly done in the classroom. We were introduced to three versions of initialization techniques including random, MaxMin, and Hierarchical. From this we were under the impression that MaxMin or Hierarchical would outperform random, and be able to establish a lower error rate overall. This is because each of these has a ‘method’ to their selection trying to optimize what goes where rather than just a random guess. From similar logic we also assumed that ten would be the right amount of clusters, each of which would be assigned to the values 0-9.

5 Methods

---

given $x_1, \ldots, x_N \in \mathbf{R}^n$ and $z_1, \ldots, z_k \in \mathbf{R}^n$

repeat

Update partition: assign i to G_j, $j = \operatorname{argmin}_{j'} \|x_i - z_{j'}\|^2$

Update centroids: $z_j = \frac{1}{|G_j|} \sum_{i \in G_j} x_i$

until $z_1, \ldots, z_k$ stop changing

Figure 1: K means pseudocode from February 1 2024 class presentation slide 22

In this project we used K-means clustering with different techniques of centroid initialization. For our centroid initialization we experimented with hierarchical, MaxMin, random images as vectors, digits from 1-10, and complete noise.

We experimented with different distance measurements such as L2, L1, And L-∞.

First we process data. We take the 28x28 images and then flatten the pixels into a one dimensional vector. Then we can find these flattened vectors and then feed them into a K-means algorithm. We will use $k=10$, since we have 10 different digits. After the algorithm converges, say after 20 iterations, we can capture the centroids. We can then change the shape of these 10 centroids from 1x728 vector → 28x28 images.

6 Results

We found that using L2 distance makes our k-means converge faster.

We find that normalizing the image values to range [0,1] has minimal impact on the time to converge.

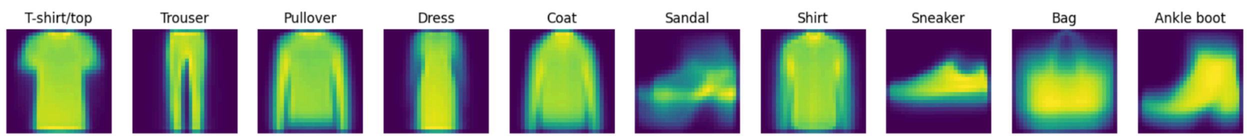

Figure 2: Average of Fashion-MNIST classes

Hierarchical clustering also works well, although compared to L2 distance with K-means it took approximately 2x the iterations to converge.

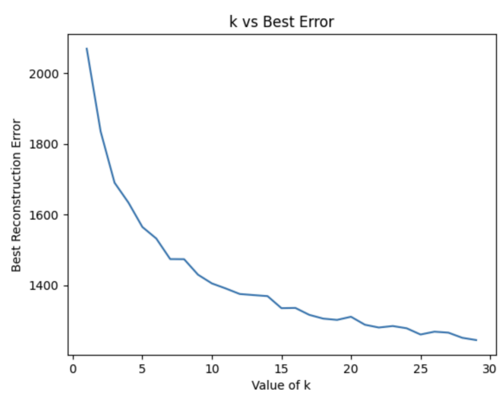

Figure 3: Elbow plot for Fashion-MNIST

We constructed an elbow plot and found that the optimal number of classes for Fashion-MNIST was 10 as it seems to be the elbow in our plot. This is interesting because Fashion-MNIST has 10 classes of images, so it would make sense that K-means naturally represents each class. However, we found that not all images get represented as a centroid. We tried to remedy this by hand selecting digits 0-9 as the initial centroids, shown in Figure 4 below.

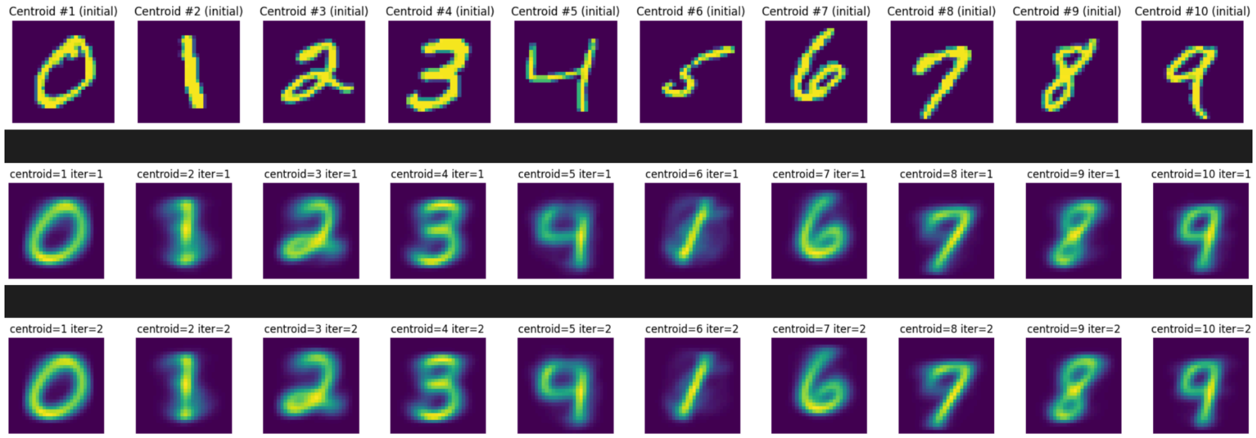

Figure 4: K-means when manually selecting the initial centroids to be numbers from 0-9

Even with hand selected centroids seen in Figure 4 we can see that some numbers tend to get overrepresented. For example, four and nine look very similar, but the centroid almost always turns out to look like a nine because it is easier for the centroid to adapt to the top of the nine than maintain the original four shape. In addition, fives are shown to react similarly to one as one dominates the centroid representation due to the diagonal central axis shared between them. It is still important to point out that these adaptations are subject to change, based on the random values 0-9 used to initialize the centroids.

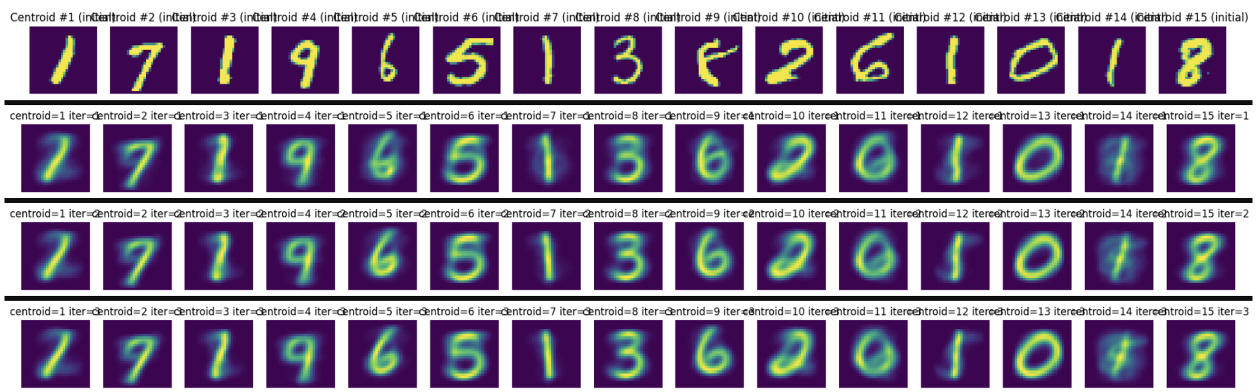

Figure 5: K-means with random selection of initial centroids and k = 15 (tolerance = 0.05)

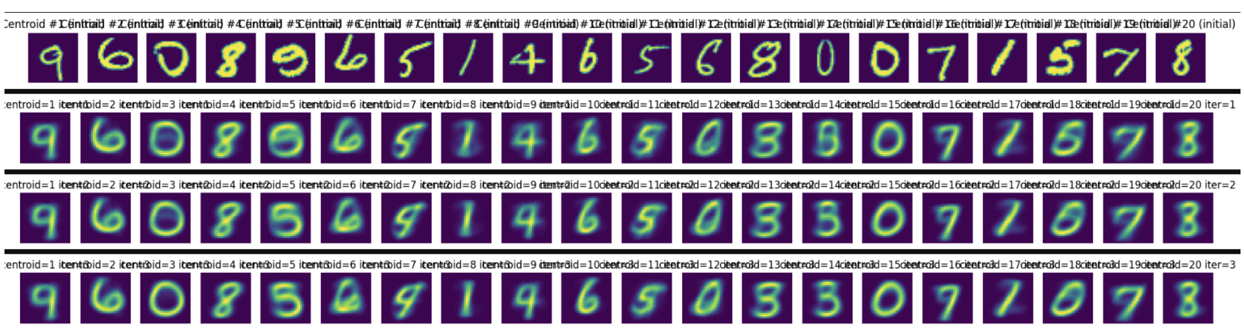

Figure 6: K-means with random selection of initial centroids and k = 20 (tolerance = 0.05)

We take this further by experimenting with visualizing different values of k, and find that with sufficiently high values of k (greater than 10) each digit is represented by a centroid. When using these sufficiently high values of k, we did note lower error values, but this is expected as with more clusters to be a part of, there should be a smaller distance between them. Though these values are handwritten and often inconsistent, the elbow plot associated with these higher k values showed insignificant improvement compared to a $k = 10$.

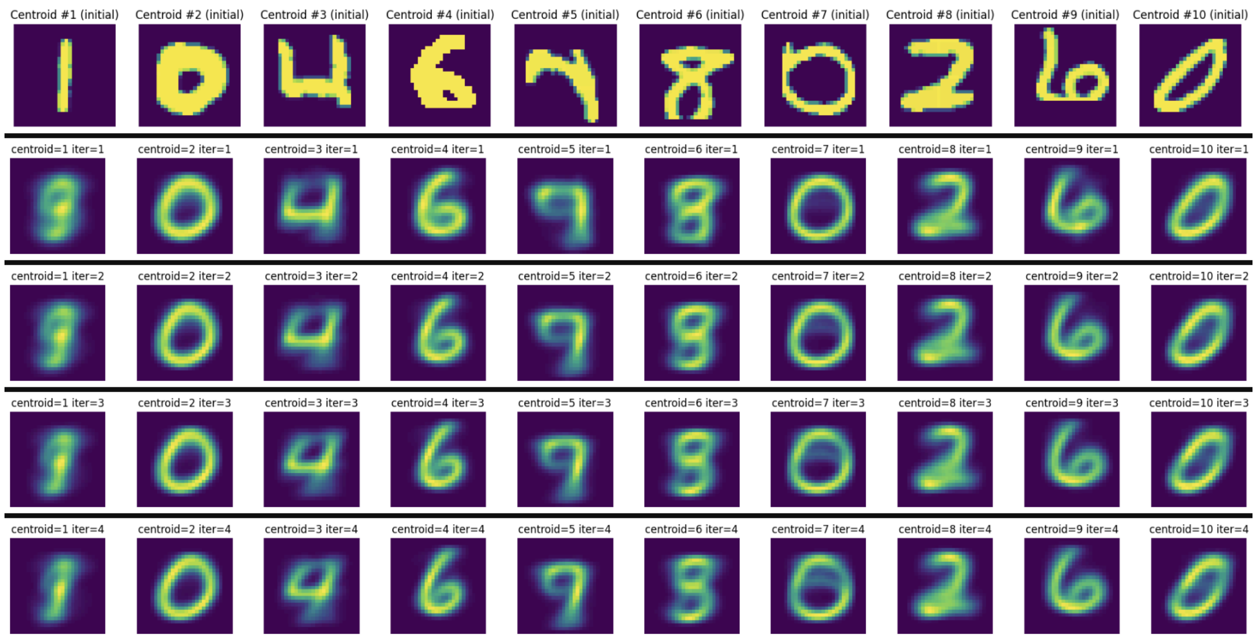

Figure 7: MaxMin initialization of centroids on $k = 10$

Using MaxMin as seen in Figure 7 demonstrates the adaptivity of these centroids, creating some of the most blurry or adaptive centroids we have seen thus far. It is also interesting to note that even when searching for the furthest distance point from the already existing ones, three zeroes were collected. This alone demonstrates the unpredictable nature of some of these handwritten values, as one digit can look so different from its counterparts.

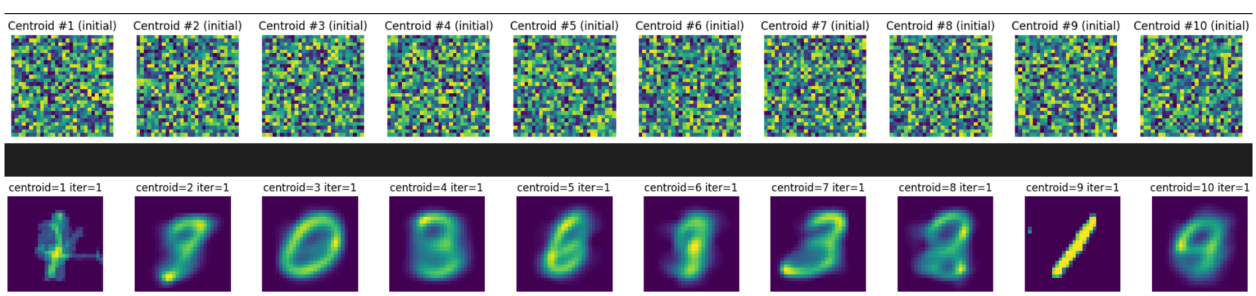

Figure 8: K-mean convergence from complete noise centroids, Top: before convergence and Bottom: after convergence

Using completely random vectors and complete noise as the initial centroids has a negative impact on distinguishing each value. Notice how in Figure 8 the centroid representations look like a mix of different digits. This represents a high ability to adapt to other values, but for our purposes of number classification this is a negative property.

7 Conclusion

Overall we found that centroids very closely resemble the cluster assignment, for both MNIST and Fashion-MNIST. We found the optimal number of clusters to be 10. However, some numbers like 5 and 4 tend to be too similar to 1 and 9 respectively so they don't get a clear centroid representation. The best distance method was using L2 distance as it converged most quickly. We find that Hierarchical initialized centroids result in good representations, but that K-means is more efficient as it converges quicker. From our data it is important to point out that blurry centroids may suggest that these conform better to all of its surrounding values, however it is important to note that this does not mean they are necessarily better for the value clustering we were hoping to achieve. While we worked on very simple 28x28 images, we think that this is a good method to see what the algorithm represents and how it classifies certain images. This method is much easier to analyze rather than reverse engineering what CNN's (Convolutional Neural Network) representations at each layer are, for example. Using parameters of K=10, L2 as the distance metric results in the fastest to converge, most recognizable cluster representations for most digits.