Image Clustering using K-means algorithm

Code: https://github.com/parthjindal/Linear-Algebra-AIML/tree/master/Assignment1
The code notebook has also been attached as a pdf to this document

In the MNIST dataset, each image is of the shape $(28, 28)$ which on flattening creates a feature vector of shape $(784,)$ The training dataset for clustering is of size $10 * 100 = 1000$. Thus N = 1000, n = 784

For convergence the following criteria has been taken:

$If\ mean(Norm_{l2}(centroids_t - centroids_{t-1})) < tolerance$:

    $converge()$

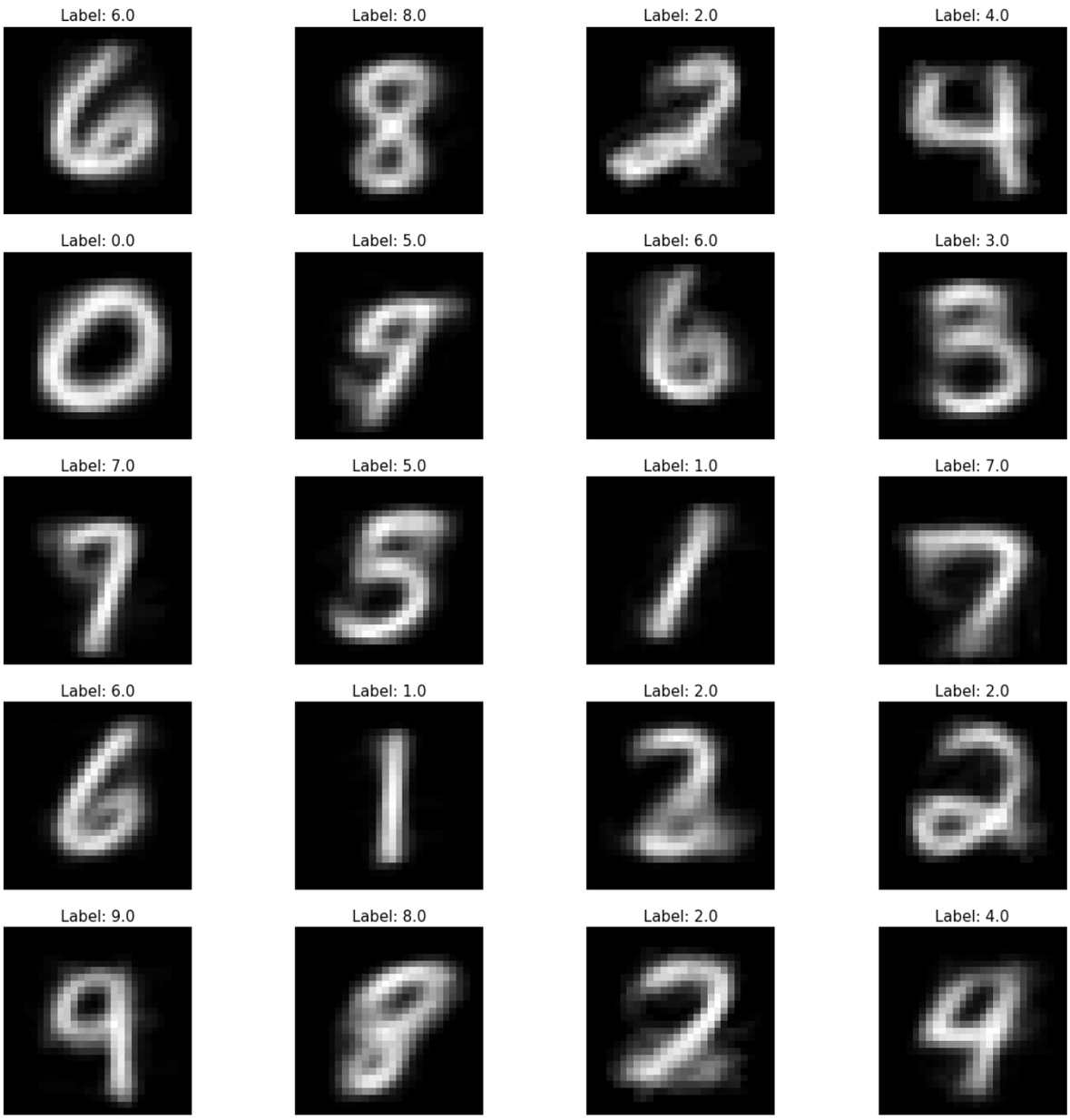
$else$: $continue$

where tolerance is a hyperparameter in $(1e^{-4}, 1e^{-6})$

For k = 20, The following cluster representatives were observed with their labels found by finding the maximum labelled images present inside this cluster

Initial Seed representative for cluster centroids:

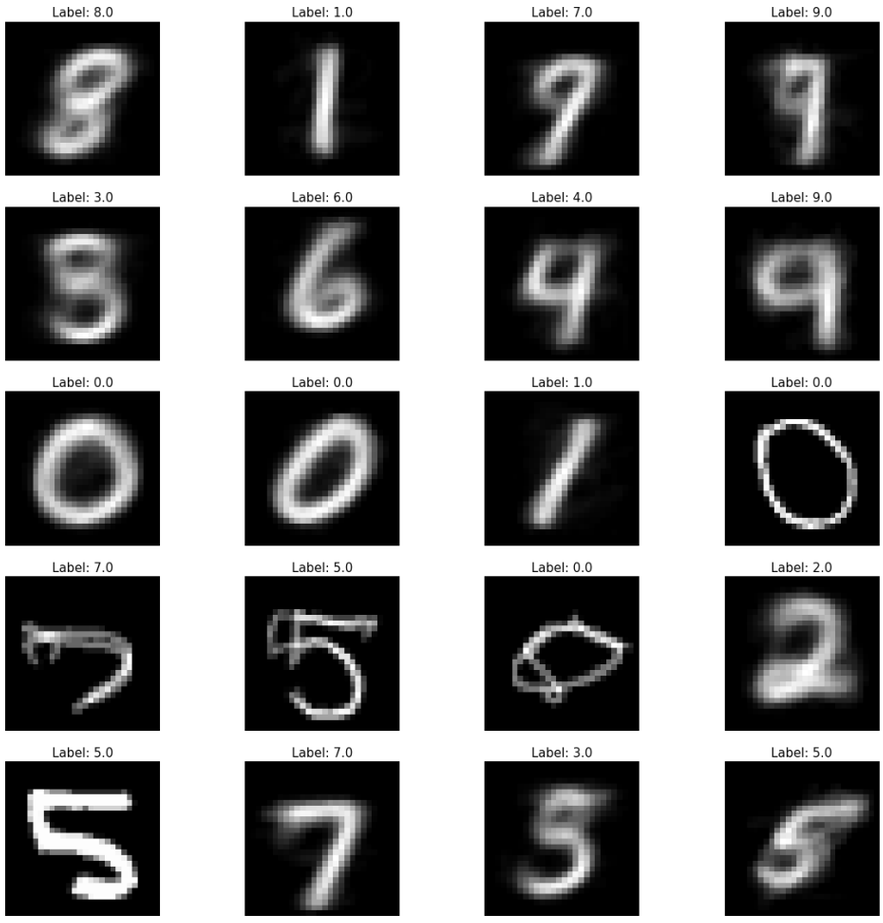
From Given training Dataset



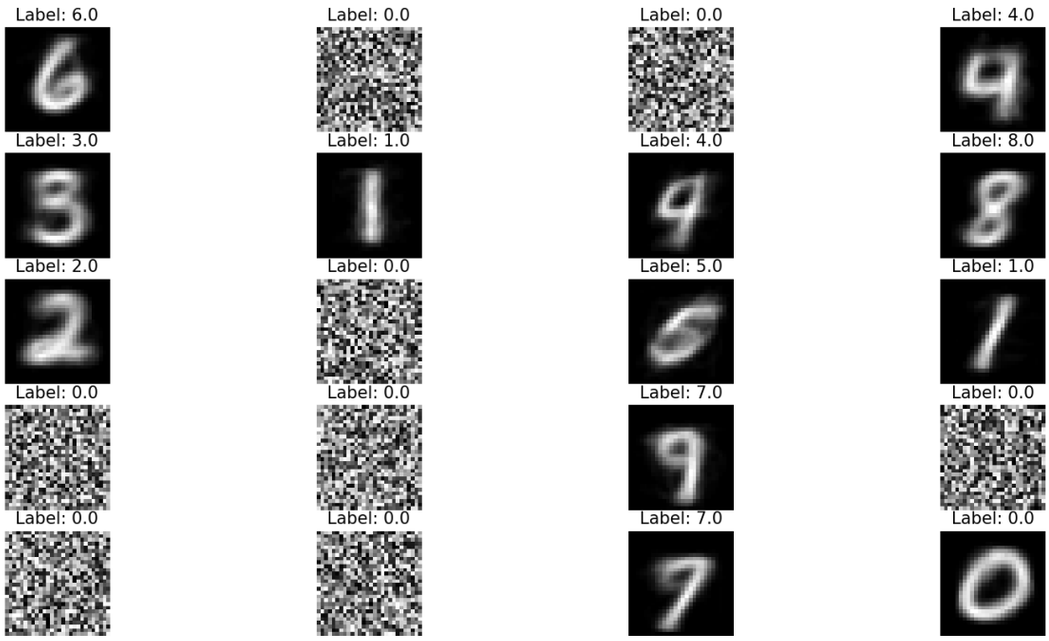
Random Initialization of Seed representatives:

Note: In random initialization, it might happen that certain clusters are empty during clustering assignment, Which might lead to poor convergence and high J_{clust}.

The following representatives are found by reassigning those cluster representatives as **zero feature vectors for convergence**

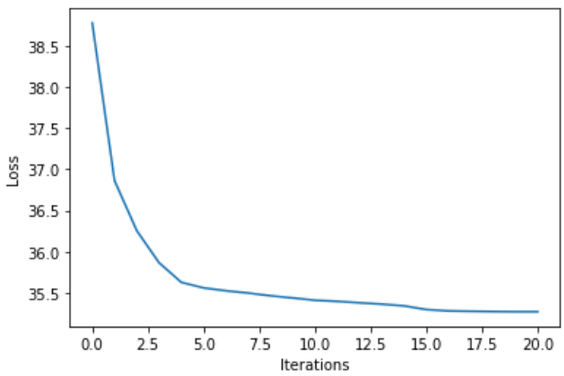


The following representatives are found by reassigning those cluster representatives as **random feature vectors for convergence.** Observe the random noise images representatives in the final cluster representatives

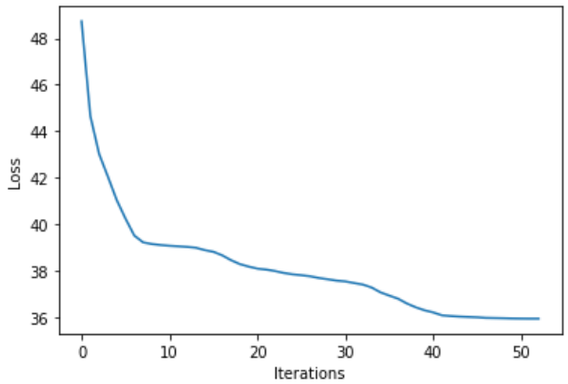


K	Accuracy	Initial Seed	Final J_{clust}	Total iterations
20	58%	Random	35.95	53
20	60%	Dataset	35.27	21

J_{clust} vs No. of iterations



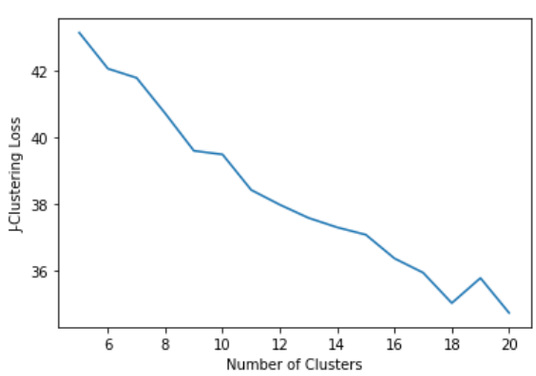
Training Data Seed

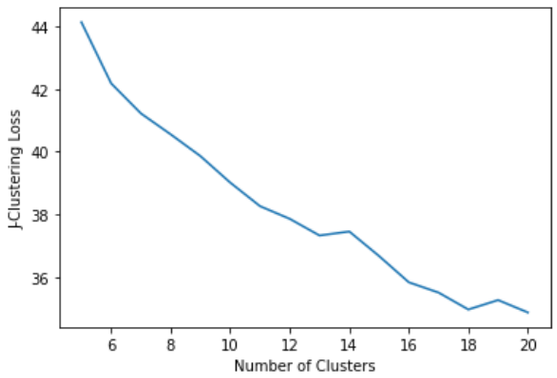


Random Seed

	Loss	
K	**Random Seed**	**Training Set seed**
5	43.143	44.128
6	42.063	**42.177**
7	41.787	**41.219**
8	40.718	**40.553**
9	39.597	**39.858**
10	39.487	39.015
11	38.421	**38.264**
12	37.978	37.859
13	37.582	37.328
14	37.298	37.456
15	37.077	36.674
16	36.366	**35.836**
17	35.937	**35.508**
18	35.025	**34.969**
19	35.778	**35.270**
20	34.730	**35.9747**

J_{clust} vs Number of Clusters





Random Seed

From training dataset

For the random initialization: The best representative is for k = 20 according to J_{clust}, For data-set initialization: The best representative is for k = 18 according to J_{clust}. However the loss will continue to decrease as we increase k(overfitting), We can use the elbow method to find the optimal no. of clusters.
For random: it is around 12, and for dataset initialization it is around 13.

Even though essentially the no. of classes are 10, k > 10 performs better due to their being various styles of writing digits and their inherent distributions.

Yes, The choice of initial condition has an effect on the k-means algorithm.
A random initialization is very much prone to finding a cluster representative not in the distribution of any of the training set feature vectors, This leads to empty cluster formation which needs to be handled by reinitialization.
With a data-set initialization, generally we find a small J_{clust} loss, higher average accuracy on test dataset and a lesser variance test accuracy in comparison to random initialization.
It is to be noted that we can certainly view initialization as a very important part of Kmeans since it converges to a local minima very easily. Newer' algorithms such as KMeans++ uses the same base algorithm but define a heuristic to find the seeds of cluster centroid representatives.