CS412 HW5

Wangfei Wang

1 kMeans

a) Plot a graph where the x axis is n clusters as the numbers between 2 and 20. Let the y axis be inertia which is a property of the kMeans object after the fit. Label this Figure 5.1.

See Figure 5.1.

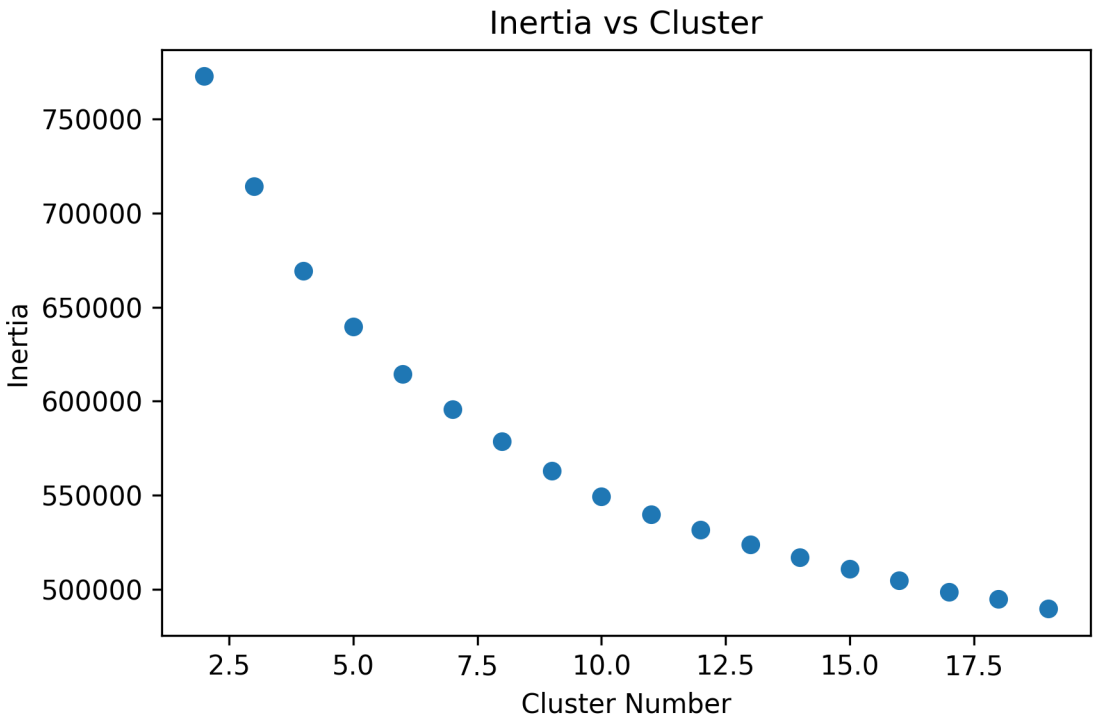

Figure 5.1 Inertia vs number of clusters.

b) Based on your experimental data, which value for n clusters is the best application for the data? Does this match your expectations or not?

n = 20 minimizes inertia and therefore n clusters = 20 is the best application for the data. It is sort of as what I've expected because the 256D data may not well classify the 10 groups very well and as the number of clusters increases, the inertia may be further decreased. In this case, the model has the tendency to get overfitted.

c) Now repeat the graph from a) where n init = 1 and max iter = 1. Label this Figure 5.2

See Figure 5.2.

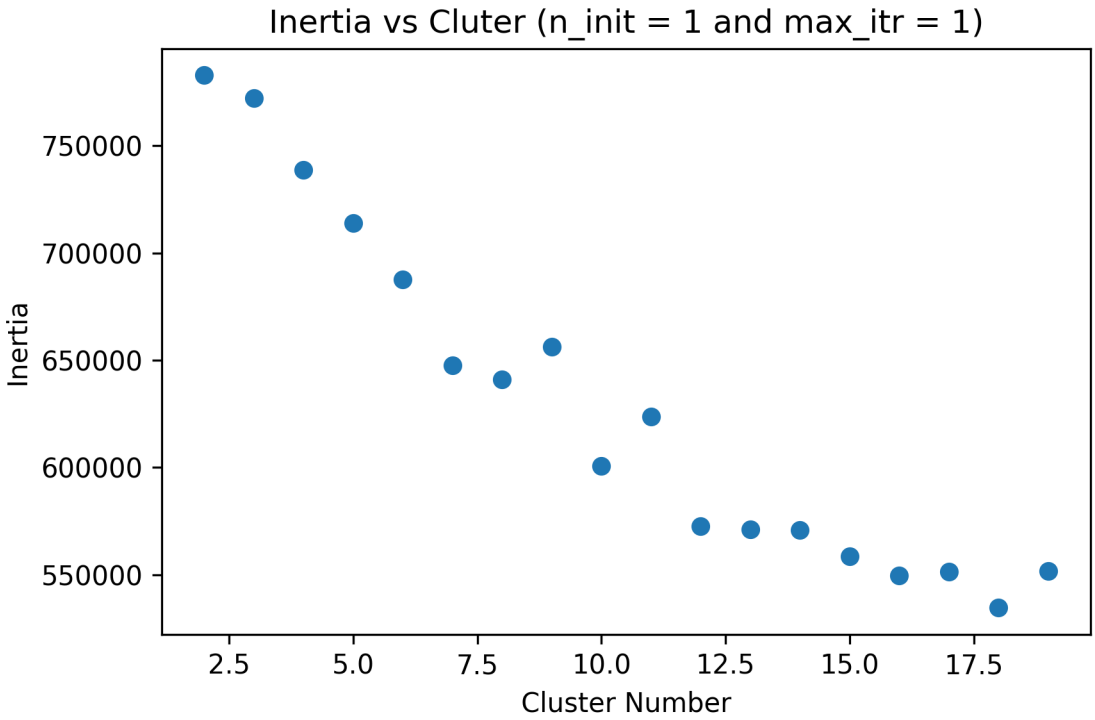

Figure 5.2 Inertia vs number of clusters (n_init = 1 and max_iter = 1).

d) Explain the differences, if any, between these two graphs.

The general pattern of the two figures are similar, except Figure 5.2 is not monotonically decreasing as number of clusters increases. Also in c), number of cluster = 19 minimizes the inertia. n_init = 1 and max_itr = 1 represent the number central seed used is 1 and the maximum number of iterations of k-means for a single run is 1. This means that the k-means in this parameter setup, the model tends to be less stable and the initial centroids chosen for the algorithm could influence the model a lot. Therefore, the inertia vs cluster number plot in this case fluttered.

e) Let n init and max iter be default again. Plot a graph where the x axis is n clusters from 2-20 and the y-axis is n iter. Label this figure 5.3

See Figure 5.3.

f) How do we expect the number of iterations to be affected by our number of clusters? Does the data match your expectations, why or why not?

In Figure 5.3, the number of iterations get bigger as the number of clusters goes up. It means that it usually takes more iterations for models with bigger cluster numbers to

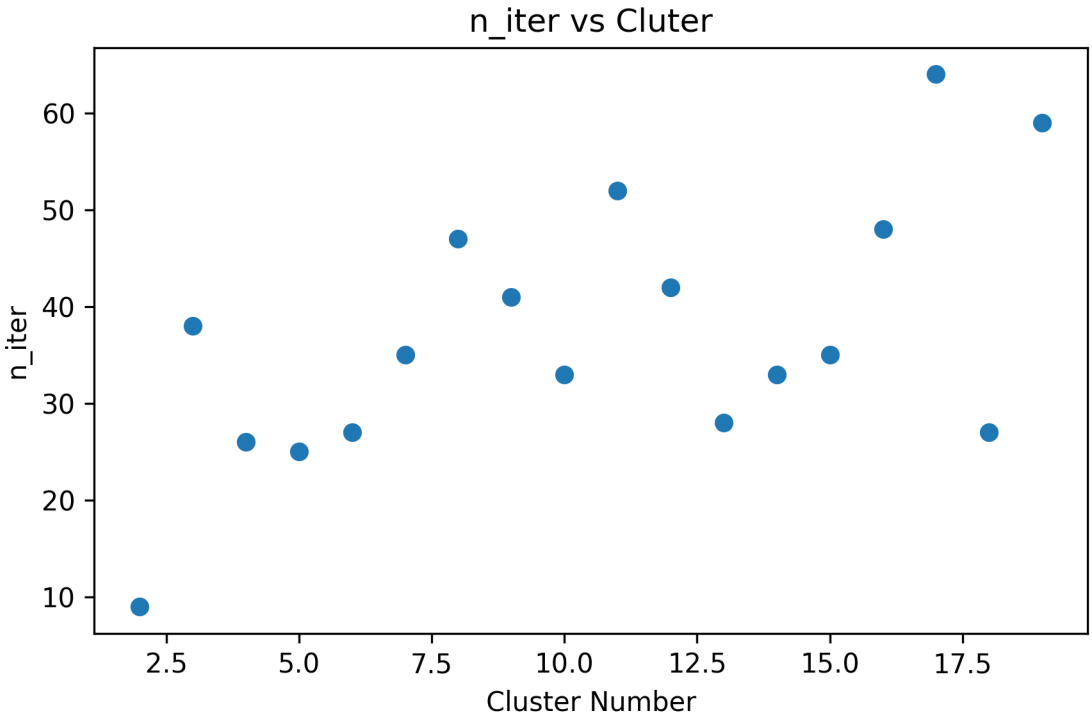

Figure 5.3 n_iter vs number of clusters plot.

minimize the objective function.

But since the model was established with random initialization, the figure would change as the difference choice of random init. But the general pattern should be similar.

g) Graduate Student Question Plot the data points as classified by the kMeans clusterer fit predict onto your 2D features from HW1. Now that you are considering data beyond only ones and fives, do the two features you initially selected in HW1 still seem to effectively separate the data? Explain why or why not.

The data set (training + testing) with all ten digits and labels removed was used for this part.

I used number of cluster = 20 for this part because it minimizes the inertia in Part a). The labels were predicted using the k-means model with k = 20 and default n_init and max_iter. Assign the 20 clusters with different colors and then plot them onto 2D space with features used in HW1 (x axis = mean intensity, y axis = intensity SD).

The features initially selected in HW1 cannot effectively separate the data anymore (see Figure 5.4).

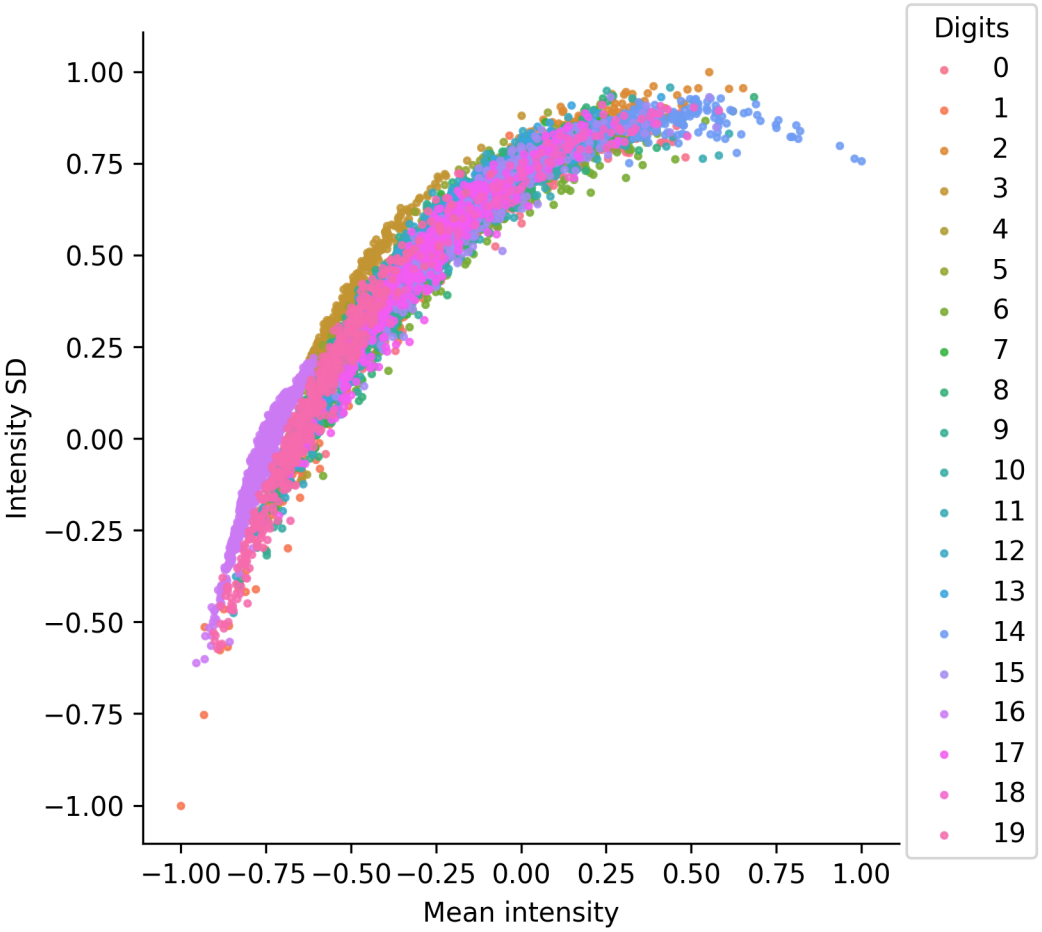

Figure 5.4. k-means clusterings (k = 20) on the data set (training + testing) with all ten digits and labels removed plotted on 2D feature space.

2 Extra Credit

Using the 10 kMeans clustering approach from the portion above, try to match the clusters to an appropriate digit. Compare the accuracy of this unsupervised learning to the accuracy of supervised approaches and report your results.

2.1 Unsupervised: 10 k-means

Now k = 10 was used for the model. The same data set: (training + testing) with all ten digits and labels removed was also used for this part. Adjusted Rand index was used to evalute 10 kMeans method here.

Adjusted Rand index is espressed as:

$$ARI = \frac{RI - E[RI]}{max(RI) - E[RI]}$$

$$RI = \frac{a+b}{C_2^{n_{samples}}}$$

where C is a ground truth class assignment and K the clustering, a is the number of pairs of elements that are in the same set in C and in the same set in K and b is the number of pairs of elements that are in different sets in C and in different sets in K.

Using metrics.adjusted_rand_score in sklearn, ARI was calculated as 0.5586.

2.2 Supervised: SVM

I used SVM on the training set (20% of the data set) for all 10 digits. The parameters for SVM I used are kernel = 'poly', degree = 2, C = 38.88 and gamma = 'auto'. The test accuracy is only 0.123 in this case, showing SVM with these parameters cannot separate 10 digits well.