Kobe Davis (901199860)

Prof. Doliotis

CS 445

3 March 2020

Assignment 4: K-Means Clustering

Experiment 1

Average Mean Squared Error: 10.06665342378373

Mean Squared Separation: 331.3535970684946

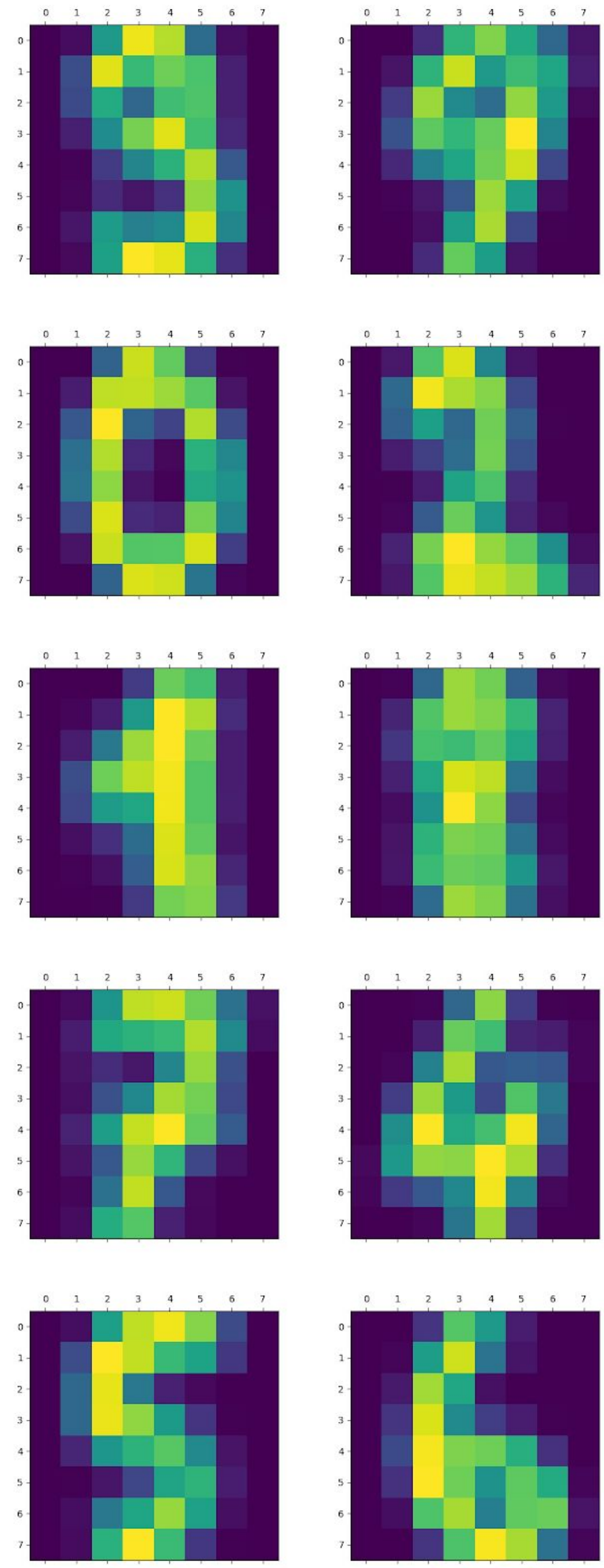
Mean Entropy: 0.9411981206441195

Success Rate: 72.73%

Confusion Matrix:

Actual Predicted	0.0	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0
0.0	176	0	1	0	0	0	1	0	0	0
1.0	0	61	2	0	4	0	1	0	7	3
2.0	0	21	150	0	0	0	0	0	1	0
3.0	0	1	8	165	0	31	0	0	31	145
4.0	2	0	0	0	144	1	1	0	0	0
5.0	0	0	0	1	0	147	0	0	2	3
6.0	0	4	0	0	0	1	176	0	2	0
7.0	0	0	3	7	1	0	0	140	1	0
8.0	0	95	13	8	7	0	2	3	121	2
9.0	0	0	0	2	25	2	0	36	9	27

Images:



Discussion

For Experiment 1 the K-Means Clustering algorithm performed as expected, though likely on the lower range of its potential accuracy. I’ve personally seen a few runs reach the low 80s. This is of course due to the random nature of the initial seeds, some runs will be better than others. AMSE is good but could be a bit lower compared to other runs I’ve had. ME is also good, the smaller the better here. It would have been interesting to put the MSS metric to use in determining the best clustering of data.

Moving on to the confusion matrix we can see from the diagonal that predictions went well except for on 2s and 9s. For this run’s initial seeds the clusters ended up having trouble creating distinct differences between 2s and 9s. Looking at the images this is fairly evident as even for human eyes the 2s and 9s look a bit obscured. While I could have taken the 15 seconds to run the program a few times to get a really good run, I think runs like this do a good job of demonstrating the weaknesses of random initial seeding.

Experiment 2

Average Mean Squared Error: 7.39622632272209

Mean Squared Separation: 397.83030393132213

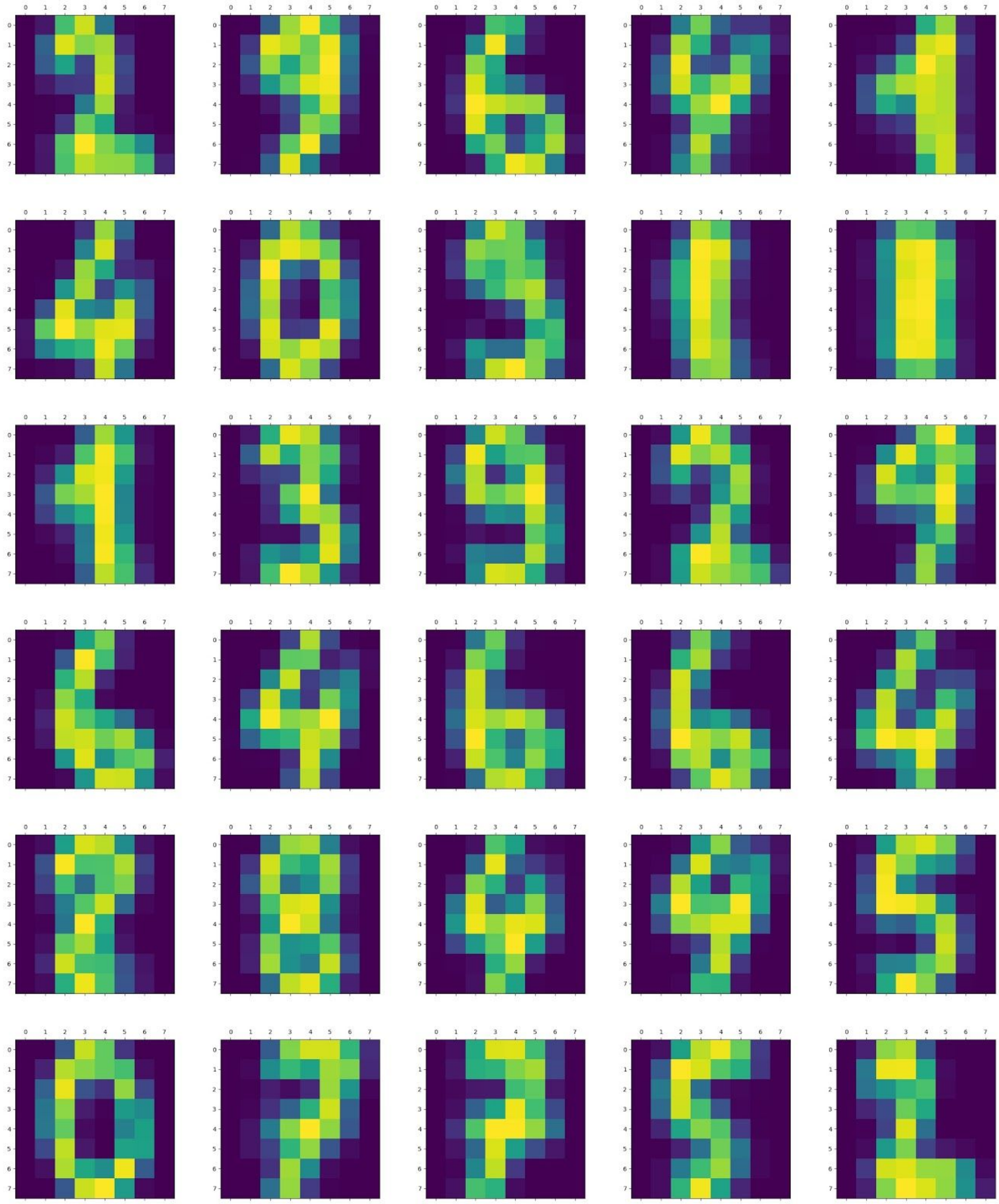
Mean Entropy: 0.4122538502982458

Success Rate: 89.71%

Confusion Matrix:

Actual	0.0	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0
Predicted										
0.0	177	0	0	0	0	0	0	0	0	1
1.0	0	152	4	0	4	0	1	0	21	1
2.0	0	21	154	2	0	0	0	0	1	0
3.0	0	2	0	135	0	0	0	0	1	4
4.0	1	0	0	0	177	1	0	1	1	4
5.0	0	0	0	3	0	170	2	0	2	1
6.0	0	2	0	0	0	1	177	0	1	0
7.0	0	0	2	3	0	0	0	166	1	0
8.0	0	1	17	13	0	0	1	3	138	3
9.0	0	4	0	27	0	10	0	9	8	166

Images:



Discussion

For Experiment 2, the K-Means Clustering algorithm performed well. No massive mistakes either as was the case in Experiment 1. While mistakes as in Experiment 1 seem to be the exception, it seems as though raising the number of clusters severely reduces the probability of such mistakes occurring. Though there is a limit to everything, no matter how many clusters we allow for, if someone has awful handwriting and their 7s appear to be 1s, they will inevitably be pooled into the 1s cluster rather than the 7s cluster. That isn't necessarily the algorithms problem though, especially if people even have trouble deciphering poorly written numbers from one another.

Experiment 2 demonstrates how raising the number of clusters can make substantial improvements to accuracy and distinction. All error metrics have improved over its Experiment 1 counterpart, the accuracy is soaring at roughly 90%, and the confusion matrix presents a clean and and contrasting diagonal. Considering its ability to distinguish unique properties, the higher cluster count seems to allow the algorithm to separate even nuanced variations within particular sets of numbers. This is clear in the images section where multiple 1s, 9s, 6s, etc. are observed. These separate categories within a single number are likely important for ensuring that slight variations do not end up in another numbers cluster.

While most, if not all aspects of this experiment trumped the previous experiment, it's important to state that since the initial seeds are random we do not know if the initial seeds in the first experiment were as awful as can get, while the initial seeds in this experiment were some of the best. This disparity between the two starting points may contribute greatly to the apparent improvement in the second experiment, and can not be discounted as a distortion of the impact of increasing the total number of clusters.