James Bao (PSU ID: 934097519)

Machine Learning Assign #4

Using K-means clustering to classify handwritten digits

Introduction

The purpose of this exercise was to design and implement a K-means clustering algorithm to classify handwritten digits pixel data stored in a 64-pixel numeric format. The datasets used were a 3,823-sample training dataset and a 1,747-sample validation dataset obtained from Bogazici University through the course website. The algorithm used K values of 10 and 30 (all clusters were randomly seeded using the training dataset samples) to generate the classification clusters. The test dataset samples were then classified into these clusters.

NOTE: The program is written in such a way that it requires the user to save the locally generated images from a training run to their local machine and manually identify them before moving onto the testing stage. The sets of clusters mentioned in this report are attached in the zip file for reference.

Part 1 : K = 10 run data

For Part I, the training algorithm was run 5 times to generate 5 sets of cluster coordinates. The mean squared error (MSE), mean squared separation (MSS), and mean cluster entropy of the clusters was calculated for each run. Theoretically, the 10 clusters should correspond to all 10 handwritten digits. This was the case for some of the runs, but it was not true for the run with the lowest MSE. For this reason, I decided to include the data from both the lowest MSE run and a run with all 10 clusters representing different digits.

Run #1 data: (all digits represented)

Iteration 36

Mean Squared Error: [603.43599529 383.29770783 568.90148547 680.6856742
624.35649765 643.03218326 824.89459289 812.1031237 706.80149863 638.69637016]

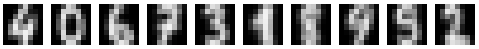
Average Mean Squared Error: 648.6205129071493

Mean Squared Separation: 1310.483379456965

Entropy per Cluster: [0.07300323 -0. 0.38315676 0.68512843 0.90090036 0.76922002
1.90050672 1.79667994 1.72083618 0.37803938]

Average Entropy: 0.8607471027507003

Run #1 cluster digits visualized: (all digits represented)



(4, 0, 6, 7, 3, 1, 8, 9, 5?, 2)

*NOTE: The 9th cluster (index 8) could either be a ‘5’ or a ‘9’, but was manually assigned to ‘5’ since the 8th cluster (index 7) is more obviously a ‘9’.

Run #5 data: (lowest average MSE)

Iteration 37

Mean Squared Error: [357.79417146 654.1496488 743.06748117 833.39407558
324.14787037 736.14256117 681.57678005 513.59881645 828.23657393 467.09883545]

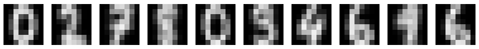
Average Mean Squared Error: 613.9206814418673

Mean Squared Separation: 1327.1174693638889

Entropy per Cluster: [-0. 0.38449879 1.05862636 1.92703145 0.09918337 1.9888155
0.09271439 0.68574989 1.82097176 0.35506192]

Average Entropy: 0.8412653421768255

Run #5 cluster digits visualized: (lowest average MSE)



(0, 2, 7, 8, 0, 3, 4, 6, 9, 6)

Part 1 : K = 10 run confusion matrices

Run #1 (all digits represented). **Accuracy = 0.6800222593210907 (68.0%)**

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	176	0	1	0	0	0	0	0	0	0
1	0	66	4	0	5	0	1	0	9	3
2	0	21	148	1	0	0	0	0	1	0
3	0	1	9	157	0	0	0	0	5	16
4	2	0	0	0	131	1	1	0	0	0
5	0	0	0	8	0	86	2	0	41	130
6	0	4	0	0	0	1	175	0	2	0
7	0	1	4	7	1	0	0	149	1	2
8	0	89	11	9	7	86	2	3	109	4
9	0	0	0	1	37	8	0	27	6	25

Run #5 (lowest average MSE). **Accuracy = 0.6327212020033389 (63.3%)**

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	177	0	1	0	0	0	1	0	0	0
1	0	0	0	0	0	0	0	0	0	0
2	0	18	153	0	0	0	0	0	1	0
3	0	1	6	165	0	74	0	0	40	145
4	1	0	0	0	161	2	0	1	0	0
5	0	0	0	0	0	0	0	0	0	0
6	0	7	0	0	0	4	179	0	3	0
7	0	1	4	9	6	0	0	166	1	5
8	0	95	10	9	8	99	1	4	113	7
9	0	60	3	0	6	3	0	8	16	23

Part 2 : K = 30 run data

For Part I, the training algorithm was run 5 times to generate 5 sets of cluster coordinates. The mean squared error (MSE), mean squared separation (MSS), and mean cluster entropy of the clusters was calculated for each run. Each run included 30 clusters, which means all 10 handwritten digits were represented at least once in each set of clusters. For Part 2, I decided to include the data from both the lowest MSE run and the 2nd lowest MSE run.

Run #3 data: (lowest average MSE)

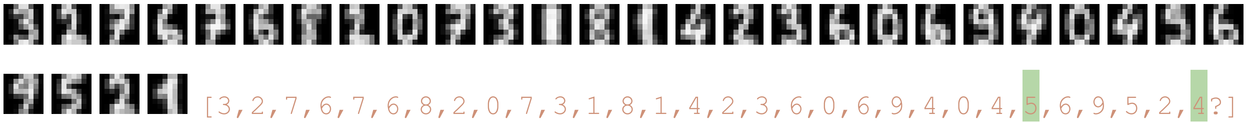
Iteration 23

Average Mean Squared Error: 470.5987665652052

Mean Squared Separation: 1544.7313580116595

Average Entropy: 0.4520679435799652

Run #3 cluster digits visualized: (lowest average MSE)



```
[3,2,7,6,7,6,8,2,0,7,3,1,8,1,4,2,3,6,0,6,9,4,0,4,5,6,9,5,2,4?]
```

* The ‘4’ at index 29 could actually be a ‘1’

Run #2 data: (2nd lowest average MSE)

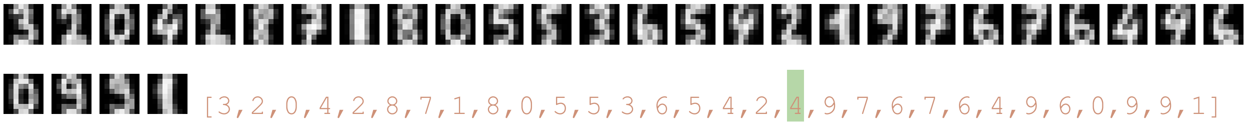
Iteration 26

Average Mean Squared Error: 473.8782466863699

Mean Squared Separation: 1544.779534881426

Average Entropy: 0.3716759517125476

Run #2 cluster digits visualized: (2nd lowest average MSE)



```
[3,2,0,4,2,8,7,1,8,0,5,5,3,6,5,4,2,4,9,7,6,7,6,4,9,6,0,9,9,1]
```

Part 2 : K = 30 run confusion matrices

Run #3 (lowest average MSE). **Accuracy = 0.8258208124652198 (82.6%)**

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	177	0	0	0	0	0	0	0	0	0
1	0	98	1	0	5	0	0	0	23	0
2	0	20	172	2	0	0	0	0	0	0
3	0	3	0	170	0	0	0	0	2	12
4	1	53	2	0	172	1	0	1	4	8
5	0	0	0	2	0	117	0	0	1	57
6	0	2	0	1	1	4	180	0	3	0
7	0	0	2	4	0	0	0	169	1	0
8	0	3	0	4	3	56	1	3	131	5
9	0	3	0	0	0	4	0	6	9	98

Run #2 (2nd lowest average MSE). **Accuracy = 0.8792431830829159 (87.9%)**

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	177	0	1	0	0	0	2	0	0	1
1	0	93	2	0	5	0	0	0	17	0
2	0	21	168	1	0	0	0	0	0	0
3	0	2	0	152	0	0	0	0	2	3
4	1	54	2	0	164	1	0	0	3	8
5	0	0	0	3	0	175	0	0	2	2
6	0	2	0	0	1	0	176	0	1	0
7	0	0	2	5	0	0	0	171	1	0
8	0	8	1	6	0	0	3	1	141	3
9	0	2	1	16	11	6	0	7	7	163

Discussion of Results

Part 1 Discussion-

When comparing the five runs with K=10, it became apparent that the trials with the lowest MSE values had slightly lower classification accuracy of the test data than trials with higher

MSE values. This is mostly because the run with the lowest error did not produce 10 distinct clusters corresponding to all 10 digits. This also suggests that clusters with lower MSE values do a better job of identifying variations of a particular digit, but given the limited number of possible clusters, the number of digits that can be correctly identified is smaller. Consequently, the accuracy rate for the lowest-error run's clusters on the test data is lower (63%) than for the run that was able to distinguish all 10 digits (68%).

Both K=10 runs showed similar misclassification patterns for certain digits, usually for digits that are very similar to each other. Notable misclassification pairs for these runs include: 1 as 8, 5 as 8, 8 as 5, and 9 as 5. The clusters generated from the lowest-error run completely failed to classify all of the 1's and the 5's in the test dataset, since neither digit was represented in the clusters generated for the lowest-error run.

Part 2 Discussion-

The accuracy rates for K=30 (30 clusters) were much higher than for the K=10 (10 clusters) runs, as the superfluous number of clusters per run ensured that each of the 10 digits would be represented at least once in the set of clusters (which was confirmed upon visual inspection of the cluster coordinates generated by Runs #3 and #2). While the accuracy values for the K=10 cluster sets were between 63% and 68%, the accuracy values for the K=30 cluster sets were all between 82%-90% [see Appendix].

The K=30 runs exhibited similar identification errors as in the K=10 clusters, with most errors involving digits that are similar to the target value- 1 vs 4, 5 vs 9, 3 vs 8, etc. A more probable source of error is misidentification of what digits the clusters represented during the visual inspection process. In both analyzed runs, one cluster that corresponded to target class values of '1' was mistakenly identified as a '4'. Likewise, for Run #3, there was one cluster that corresponded to '9' but was mistakenly identified as a '5'. [See Part 2: K=30 run data; the misidentified clusters are highlighted in green].

Appendix

Confusion Matrix for Run 1: (Accuracy for Run 1: 0.9031719532554258)

```
[[177   0   1   0   0   0   0   0   0   0]
 [  0 147   5   0   7   0   1   0  19   1]
 [  0  21 164   1   0   0   0   0   0   0]
 [  0   3   3 168   0   4   0   0   4  37]
 [  1   0   0   0 171   1   1   0   0   0]
 [  0   1   0   1   0 175   2   0   2   2]
 [  0   2   0   0   0   0 175   0   0   0]
 [  0   0   2   6   2   0   0 173   1   4]
 [  0   5   2   7   1   0   2   1 140   3]
 [  0   3   0   0   0   2   0   5   8 133]]
Accuracy for Run 1: 0.9031719532554258

Confusion Matrix for Run 2: (2nd lowest avg MSE)
[[177   0   1   0   0   0   2   0   0   1]
 [  0  93   2   0   5   0   0   0  17   0]
 [  0  21 168   1   0   0   0   0   0   0]
 [  0   2   0 152   0   0   0   0   2   3]
 [  1  54   2   0 164   1   0   0   3   8]
 [  0   0   0   3   0 175   0   0   2   2]
 [  0   2   0   0   1   0 176   0   1   0]
 [  0   0   2   5   0   0   0 171   1   0]
 [  0   8   1   6   0   0   3   1 141   3]
 [  0   2   1  16  11   6   0   7   7 163]]
Accuracy for Run 2: 0.8792431830829159

Confusion Matrix for Run 3: (lowest avg MSE)
[[177   0   0   0   0   0   0   0   0   0]
 [  0  98   1   0   5   0   0   0  23   0]
 [  0  20 172   2   0   0   0   0   0   0]
 [  0   3   0 170   0   0   0   0   2  12]
 [  1  53   2   0 172   1   0   1   4   8]
 [  0   0   0   2   0 117   0   0   1  57]
 [  0   2   0   1   1   4 180   0   3   0]
 [  0   0   2   4   0   0   0 169   1   0]
 [  0   3   0   4   3  56   1   3 131   5]
 [  0   3   0   0   0   4   0   6   9  98]]
Accuracy for Run 3: 0.8258208124652198

Confusion Matrix for Run 4:
[[177   0   1   0   0   0   2   0   0   1]
 [  0 152   6   0   8   0   2   0  22   1]
 [  0  20 158   2   0   0   0   0   0   0]
 [  0   2   1 164   0   6   0   0   2  50]
 [  1   0   0   0 157   1   1   0   0   5]
 [  0   0   0   2   0 172   1   0   4   2]
 [  0   3   0   0   0   0 173   0   1   0]
 [  0   0   2   6   0   0   0 172   1   0]
 [  0   2   9   6   2   0   2   2 138   3]
 [  0   3   0   3  14   3   0   5   6 118]]
Accuracy for Run 4: 0.8797996661101837

Confusion Matrix for Run 5
[[177   0   0   0   0   0   0   0   0   1]
 [  0  93   0   0   1   0   0   0  15   0]
 [  0  38 121   0   2   0   1   0   5   0]
 [  0   1  52 135   0   0   0   0   2   4]
 [  1  45   2   0 175   1   0   1   2   3]
 [  0   0   0   3   0 173   2   0   2   1]
 [  0   1   0   0   0   1 176   0   1   0]
 [  0   0   2   5   0   0   0 160   0   0]
 [  0   1   0   8   0   0   2   1 138   2]
 [  0   3   0  32   3   7   0  17   9 169]]
Accuracy for Run 5: 0.8441847523650529
-----------------------------------------------------------------------
[Run 1] Iteration 31
Average Mean Squared Error:  485.64558642656635
```

```
Mean Squared Separation:  1478.0279003359371
Average Entropy:  0.38400410760463727

[Run 2] Iteration 26
Average Mean Squared Error:  473.8782466863699
Mean Squared Separation:  1544.779534881426
Average Entropy:  0.3716759517125476

[Run 3] Iteration 23
Average Mean Squared Error:  470.5987665652052
Mean Squared Separation:  1544.7313580116595
Average Entropy:  0.4520679435799652

[Run 4] Iteration 32
Average Mean Squared Error:  478.18681958880853
Mean Squared Separation:  1535.2975481224657
Average Entropy:  0.43333220147189555

[Run 5] Iteration 34
Average Mean Squared Error:  475.5970038630779
Mean Squared Separation:  1558.2027499706821
Average Entropy:  0.41257207976379556
```