Juan Diego Gonzalez German

1001401837

11/13/2019

CSE 4309 - Assignments - Assignment 6

Task 1 (70 points, programming)

In this task you will implement EM clustering.

Output on the Screen

After each iteration, you should print the weight and mean of each Gaussian. After all iterations are done, you should print out the mean and covariance matrix for each Gaussian.

Output for answers.pdf

In your answers.pdf document, you need to provide the complete output for the following invocations of your program:

```
em_cluster('point_set1.txt', 2, 5)
em_cluster('point_set1.txt', 3, 5)
```

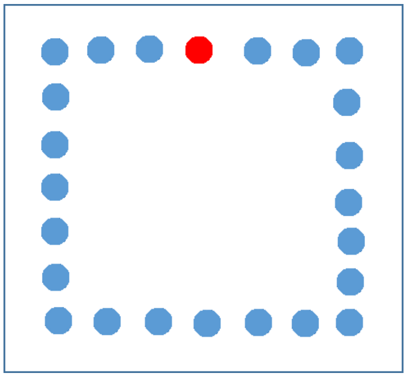
```
Output for: python em_cluster.py point_set1.txt 2 5

After iteration 1
weight 1 = 0.4333, mean 1 = (341.4052, 354.9801)
weight 2 = 0.5667, mean 2 = (290.5882, 265.0700)
After iteration 2
weight 1 = 0.4326, mean 1 = (333.5072, 356.9847)
weight 2 = 0.5674, mean 2 = (296.6775, 263.6629)
After iteration 3
weight 1 = 0.4347, mean 1 = (333.9176, 355.5479)
weight 2 = 0.5653, mean 2 = (296.2261, 264.4233)
After iteration 4
weight 1 = 0.4372, mean 1 = (337.5030, 352.2921)
weight 2 = 0.5628, mean 2 = (293.2688, 266.5374)
After final iteration
weight 1 = 0.4403, mean 1 = (343.1999, 347.3539)
Sigma 0 row 1 = 26146.4332, -9820.6101
Sigma 0 row 2 = -9820.6101, 31129.9746
weight 2 = 0.5597, mean 2 = (288.5455, 269.9527)
Sigma 1 row 1 = 22380.3039, 2214.3467
Sigma 1 row 2 = 2214.3467, 29861.8285
```

Output for: python em_cluster.py point_set1.txt 3 5

```
After iteration 1
weight 1 = 0.3000, mean 1 = (323.2244, 279.7012)
weight 2 = 0.5667, mean 2 = (330.9389, 307.9190)
weight 3 = 0.1333, mean 3 = (210.8213, 342.2493)
After iteration 2
weight 1 = 0.2821, mean 1 = (365.4902, 264.4754)
weight 2 = 0.5126, mean 2 = (320.1002, 309.3303)
weight 3 = 0.2053, mean 3 = (221.2500, 345.1441)
After iteration 3
weight 1 = 0.2495, mean 1 = (394.8045, 250.3074)
weight 2 = 0.4412, mean 2 = (323.3145, 302.2860)
weight 3 = 0.3092, mean 3 = (231.0057, 349.8736)
After iteration 4
weight 1 = 0.2162, mean 1 = (421.0936, 239.7723)
weight 2 = 0.3894, mean 2 = (328.3520, 289.5219)
weight 3 = 0.3944, mean 3 = (237.5955, 353.5824)
After final iteration
weight 1 = 0.1976, mean 1 = (438.0624, 244.9315)
Sigma 0 row 1 = 29951.7247, -21172.5788
Sigma 0 row 2 = -21172.5788, 42611.3033
weight 2 = 0.3782, mean 2 = (327.9940, 278.8420)
Sigma 1 row 1 = 34626.8291, 13083.4717
Sigma 1 row 2 = 13083.4717, 53707.3653
weight 3 = 0.4242, mean 3 = (240.4675, 354.0105)
Sigma 2 row 1 = 835.1933, 787.3228
Sigma 2 row 2 = 787.3228, 2772.0714
```

Task 2 (10 points)



Consider the set of points above. Each dot corresponds to a point. Consider a clustering consisting of two clusters, where the first cluster is the set of all the blue dots, and the second cluster has the red dot as its only element. Can this clustering be the final result of the k-means algorithm? Justify your answer.

Your answer should be based on your visual estimations of Euclidean distances between dots. For this particular question, no greater precision is needed.

No, not all dots are colored into the cluster whose mean is closer to them. For example, the red dot belongs to a cluster whose mean is itself, and thus the blue dots next to it are closer to the red cluster mean than they are to the blue cluster mean, which is somewhere near the middle of the square, therefore the algorithm still has to loop some more.

Task 3 (10 points)

For both parts of this task, assume that the algorithm in question **never needs to break a tie** (i.e., it never needs to choose between two distances that are equal).

Part a: Will the k-means algorithm always give the same results when applied to the same dataset with the same k? To phrase the question in an alternative way, is there any dataset where the algorithm can produce different results if run multiple times, with the value of k kept the same? If your answer is that k-means will always give the same results, justify why. If your answer is that k-means can produce different results if run multiple times, provide an example.

No, considering a point that for the last iteration of the algorithm is equidistant to the mean of two cluster, the algorithm may assign to one cluster in one run, and to another in the other, meaning that the final result may differ with respect to what cluster that point is assigned to

Part b: Same question as in part a, but for agglomerative clustering with the d_{min} distance. Here, as "result" we consider all intermediate clusterings, between the first step (with each object being its own cluster) and the last step (where all objects belong to a single cluster). Will this agglomerative algorithm always give the same results when applied to the same dataset? If your answer is "yes", justify why. If your answer is "no", provide an example.

No. Given a set were one point is, at the beginning, equidistant to two other points, the algorithm may merge two single-element cluster together in one run, but the other pair in another. So, while the final cluster will be the same in any case, the intermediate clusterings may vary a little

Task 4 (10 points)

Consider a dataset consisting of these eight points: 2, 4, 7, 11, 16, 22, 29, 37.

Part a: Show the results (all intermediate clusterings) obtained by applying agglomerative clustering to this dataset, using the d_{min} distance.

1. {2},{4},{7},{11},{16},{22},{29},{37}
2. {2,4},{7},{11},{16},{22},{29},{37}
3. {2,4,7},{11},{16},{22},{29},{37}
4. {2,4,7,11},{16},{22},{29},{37}
5. {2,4,7,11,16},{22},{29},{37}
6. {2,4,7,11,16,22},{29},{37}
7. {2,4,7,11,16,22,29},{37}
8. {2,4,7,11,16,22,29,37}

Part b: Show the results (all intermediate clusterings) obtained by applying agglomerative clustering to this dataset, using the d_{max} distance.

1. {2},{4},{7},{11},{16},{22},{29},{37}
2. {2,37},{4},{7},{11},{16},{22},{29}
3. {2,37,4}{7},{11},{16},{22},{29}
4. {2,37,4,7},{11},{16},{22},{29}
5. {2,37,4,7,29},{11},{16},{22}
6. {2,37,4,7,29,11},{16},{22}
7. {2,37,4,7,29,11,16},{22}
8. {2,37,4,7,29,11,16,22}