Assignment 4 for Large Scale Data Mining

Name: Zijian Zhang
Matrikelnr.: 3184680

May 31, 2016

1

1.1

Centroid after each iteration as such:

$$\begin{aligned}
[1.0, 4.0, 9.0, 16.0, 25.0, 36.0, 49.0, 64.0, 81.0] & \\
&([2.5])[9.0, 16.0, 25.0, 36.0, 49.0, 64.0, 81.0] \\
&([4.67])[16.0, 25.0, 36.0, 49.0, 64.0, 81.0] \\
[4.67]&([20.5])[36.0, 49.0, 64.0, 81.0] \\
[4.67, 20.5]&([42.5])[64.0, 81.0] \\
&([11.0])[42.5, 64.0, 81.0] \\
[11.0, 42.5]&([72.5]) \\
[11.0]&([57.5]) \\
&([31.67])
\end{aligned}$$

where in parentheses are new merged centroids. Because the numbers only occur on 1-dimension, only the adjacent points (centroids) are assigned to the brother branch of the hierarchy tree.

1.2

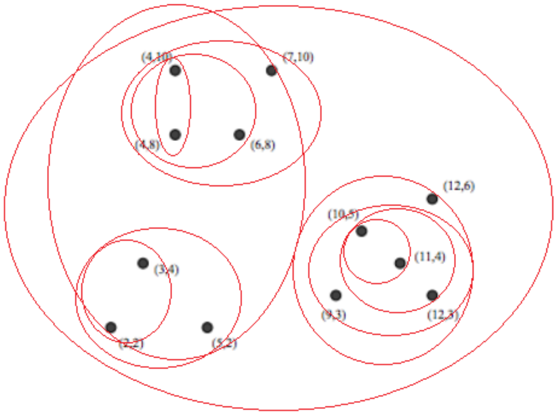
For easier expression I assigned each point with a index from left to right, from bottom to top. Index increases from 0. Thus the whole data set is $[(2, 2), (3, 4), (4, 8), (4, 10), (5, 2), (6, 8), (7, 10), (9, 3), (10, 5), (11, 4), (12, 3), (12, 6)]$.

1.2.1 Using minimum of the distances between any two points

combined data point pair and distance during each iteration, and grouping situation after each iteration (note: "combine point x with y" means combine cluster that contains x with cluster that contains y):

```
Iteration #1
combine point 8 with 9, distance between which is 1.41
[0, 1, 2, 3, 4, 5, 6, 7, 9, 9, 10, 11]

Iteration #2
combine point 9 with 10, distance between which is 1.41
[0, 1, 2, 3, 4, 5, 6, 7, 10, 10, 10, 11]

Iteration #3
combine point 2 with 3, distance between which is 2.0
[0, 1, 3, 3, 4, 5, 6, 7, 10, 10, 10, 11]

Iteration #4
combine point 2 with 5, distance between which is 2.0
[0, 1, 5, 5, 4, 5, 6, 7, 10, 10, 10, 11]

Iteration #5
combine point 0 with 1, distance between which is 2.24
[1, 1, 5, 5, 4, 5, 6, 7, 10, 10, 10, 11]

Iteration #6
combine point 5 with 6, distance between which is 2.24
[1, 1, 6, 6, 4, 6, 6, 7, 10, 10, 10, 11]

Iteration #7
combine point 7 with 8, distance between which is 2.24
[1, 1, 6, 6, 4, 6, 6, 10, 10, 10, 10, 11]

Iteration #8
combine point 8 with 11, distance between which is 2.24
[1, 1, 6, 6, 4, 6, 6, 11, 11, 11, 11, 11]

Iteration #9
combine point 1 with 4, distance between which is 2.83
[4, 4, 6, 6, 4, 6, 6, 11, 11, 11, 11, 11]

Iteration #10
combine point 1 with 2, distance between which is 4.12
[6, 6, 6, 6, 6, 6, 6, 11, 11, 11, 11, 11]

Iteration #11
combine point 4 with 7, distance between which is 4.12
[11, 11, 11, 11, 11, 11, 11, 11, 11, 11, 11, 11]
```

The result is:



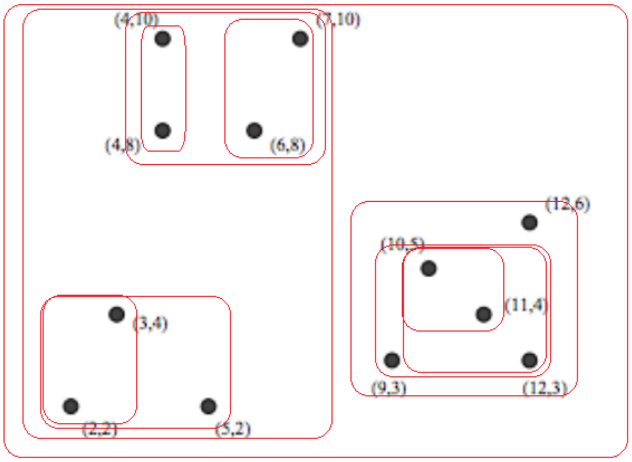
1.2.2 Using average of distance between pairs of points

combined data point pair and distance during each iteration, and grouping situation after each iteration (note: ”combine point x with y” means combine cluster that contains x with cluster that contains y):

```
Iteration #1
combine point 8 with 9, distance between which is 0.71
[0, 1, 2, 3, 4, 5, 6, 7, 9, 9, 10, 11]

Iteration #2
combine point 2 with 3, distance between which is 1.0
[0, 1, 3, 3, 4, 5, 6, 7, 9, 9, 10, 11]

Iteration #3
combine point 0 with 1, distance between which is 1.12
[1, 1, 3, 3, 4, 5, 6, 7, 9, 9, 10, 11]

Iteration #4
combine point 5 with 6, distance between which is 1.12
[1, 1, 3, 3, 4, 6, 6, 7, 9, 9, 10, 11]

Iteration #5
combine point 8 with 10, distance between which is 1.41
[1, 1, 3, 3, 4, 6, 6, 7, 10, 10, 10, 11]

Iteration #6
combine point 7 with 8, distance between which is 1.87
[1, 1, 3, 3, 4, 6, 6, 10, 10, 10, 10, 11]

Iteration #7
combine point 0 with 4, distance between which is 1.94
```

```
[4, 4, 3, 3, 4, 6, 6, 10, 10, 10, 10, 11]

Iteration #8
combine point 2 with 5, distance between which is 2.29
[4, 4, 6, 6, 4, 6, 6, 10, 10, 10, 10, 11]

Iteration #9
combine point 7 with 11, distance between which is 2.34
[4, 4, 6, 6, 4, 6, 6, 11, 11, 11, 11, 11]

Iteration #10
combine point 2 with 4, distance between which is 6.32
[4, 4, 4, 4, 4, 4, 4, 11, 11, 11, 11, 11]

Iteration #11
combine point 0 with 7, distance between which is 7.44
[11, 11, 11, 11, 11, 11, 11, 11, 11, 11, 11, 11]
```

The result is:



Conclusion: Using minimum of the distance between two points in different clusters as metric makes the merge procedure look like clusters ”expansion”. For example point 3, 4, 6 and 7 illustrate this trend. However when using the average distance, all of those independent points at first merge, if they lie really near to each other. Same example of points 3, 4, 6 and 7.

2

The medium of a sequence of numbers could be of two situations:
1.the "central" number, if the length of the sequence is odd and
2.the average of middle two numbers if the length is even.

For the first case, denote the medium number as m. If we use $m + v$ as v with in a single cluster, instead of m, where $v \in \mathbb{R}$ and $m + v$ doesn't outstrip the neighbour numbers of either side. Let $v > 0$, compare that using medium number m, using $m+v$ lead to that $\sum_{i=1}^{index\_of\_m} |v - x_i|$ increase $index\_of\_m * v$ in total and $\sum_{i=index\_of\_m}^{length\_of\_seq} |v - x_i|$ decrease $(index\_of\_m - 1) * v$. So it's bigger than using the medium number. If $m + v$ across the neighbour number, the incretion is even larger.

For the second case. Denotes are the same as the first case. if $m + v$ varies between the middle two numbers, the sum of absolute error doesn't change at all. Because every v it increase to the one side, it could be decreased at the other side. However if $m+v$ crosses the either middle number, the sum increase two times of v at the one side more than the other side. Thus using the medium also leads to the minimum of the absolute error.

Concluded that using the medium of each cluster is the best choice of v_i

3

3.1

The first choice is $(3, 4)$, then the second choice should be as far from the first as possible thus point $(12, 6)$, and the third should be $(4, 10)$

3.2

3.2.1

For the cluster on the top:

$$\begin{aligned} N &= 4 \\ SUM &= (21, 36) \\ SUMSQ &= (117, 328) \end{aligned}$$

For the cluster bottom left:

$$\begin{aligned} N &= 3 \\ SUM &= (10, 8) \\ SUMSQ &= (40, 24) \end{aligned}$$

For the cluster bottom right:

$$\begin{aligned} N &= 5 \\ SUM &= (54, 21) \\ SUMSQ &= (590, 95) \end{aligned}$$

3.2.2

For the cluster on the top:

$$\begin{aligned} Va &= (1.69, 1) \\ SD &= (1.29, 1) \end{aligned}$$

For the cluster bottom left:

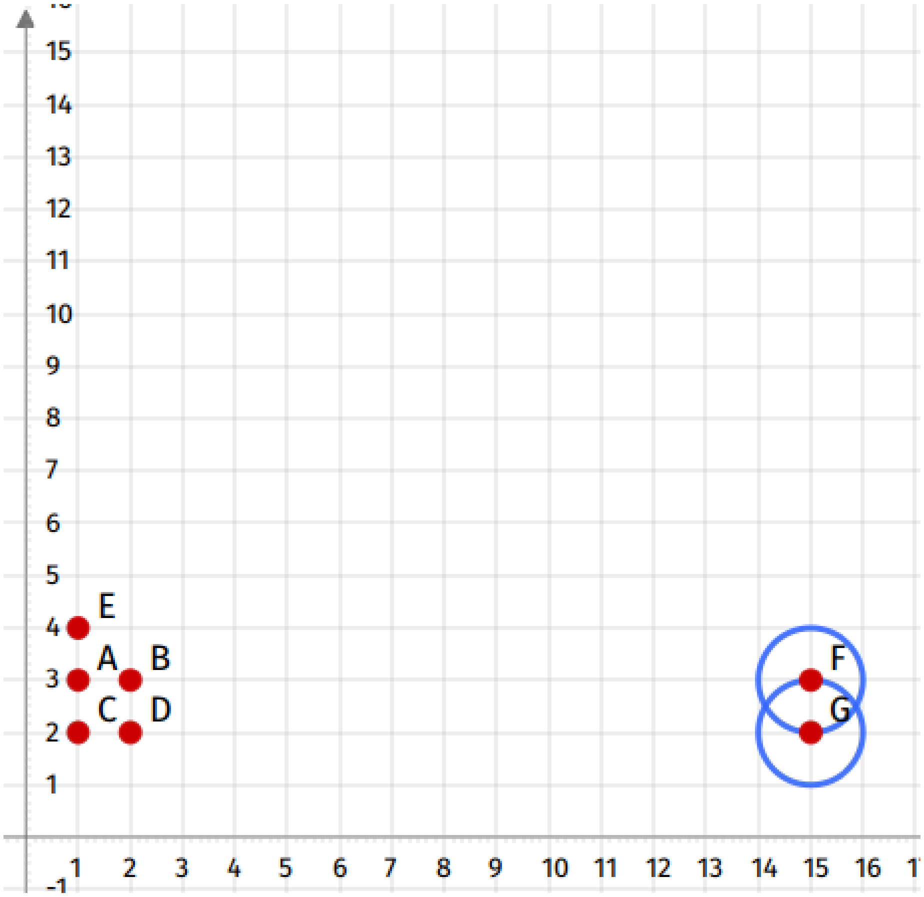
$$\begin{aligned} Va &= (2.22, 0.89) \\ SD &= (1.49, 0.94) \end{aligned}$$

For the cluster bottom right:

$$\begin{aligned} Va &= (1.36, 1.36) \\ SD &= (1.17, 1.17) \end{aligned}$$

4

For example graph below. Obviously $k = 2$. We choose point F and G as the 2 initial points, one of which could be assigned into different cluster at the end of algorithm.


15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
-1
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
E
A
B
C
D
F
G