Neural Network and Fuzzy Systems (EECE 5860)

Project-1 Report

Submitted by – Mohammad Saber
(MUID: 005852837)

CONTENTS

PROBLEM ONE: DEVELOP A FUZZY INFERENCE SYSTEM TO SIMULATE THE CAMERA TRACKING SYSTEM

A) USING FUZZY TOOLBOX TO GENERATE FUZZY INFERENCE SYSTEM:

Two fuzzy inference systems are required to solve the given problem. We can open the fuzzy system toolbox from MATLAB by typing 'fuzzy' in MATLAB command line. For first fuzzy inference system we need only one input named *Range* and one output named *Scale* as depicted in Figure 1.

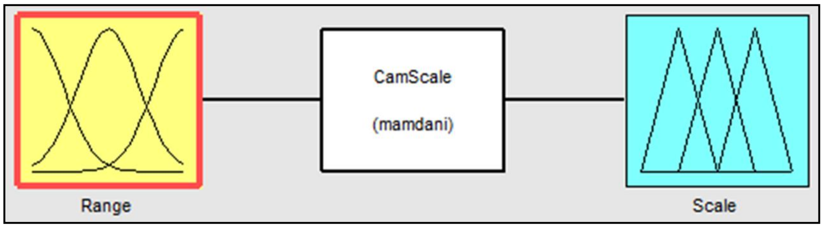


Figure 1: Fuzzy inference system to generate scale from input range

Now we have to create membership functions for Range and Scale. We can just click on the input box and output box (Figure 1) to set all membership functions according to the given problem. Figure 2 and 3 shows all membership functions used for this fuzzy inference system.

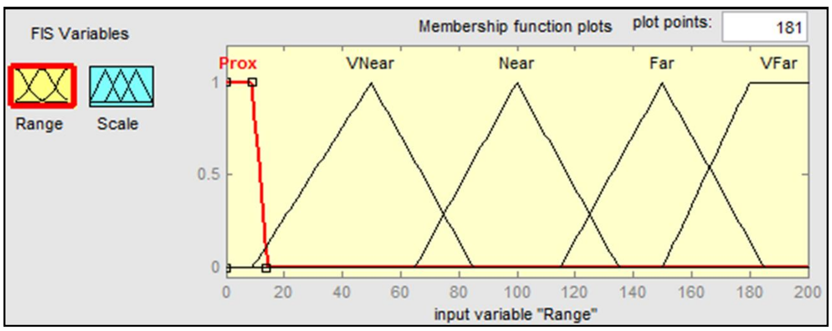


Figure 2: Membership functions for input variable "Range"

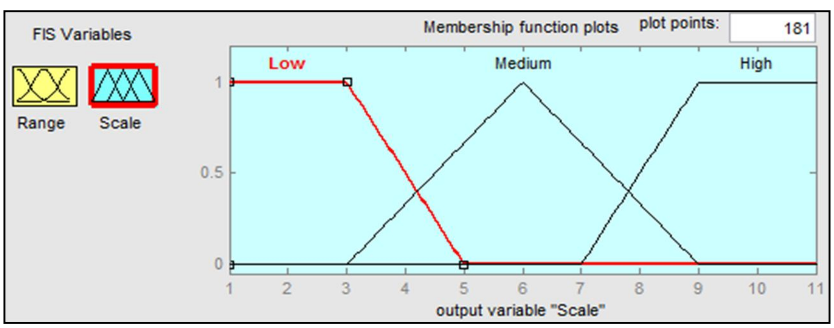


Figure 3: Membership functions for output variable "Scale"

Then I have set the following rules by clicking on the middle white box of Figure 1.

1. If (Range is VFar) then (Scale is Low) (1)
2. If (Range is Far) then (Scale is Low) (1)
3. If (Range is Near) then (Scale is Medium) (1)
4. If (Range is VNear) then (Scale is High) (1)
5. If (Range is Prox) then (Scale is High) (1)

Figure 4: Set of rules for first fuzzy inference system

For second fuzzy system we have three inputs and two outputs (Figure 5). One of the inputs of second fuzzy system would be the output first fuzzy system. Thus the input variable "Range" will have same membership functions as output variable of first fuzzy inference system (Figure 3). Two output variables "Pan" and "Tilt" will also have the same membership functions according to the problem specification. Figure 6, 7 and 8 show membership functions for input and output variables required to implement second fuzzy inferences system.

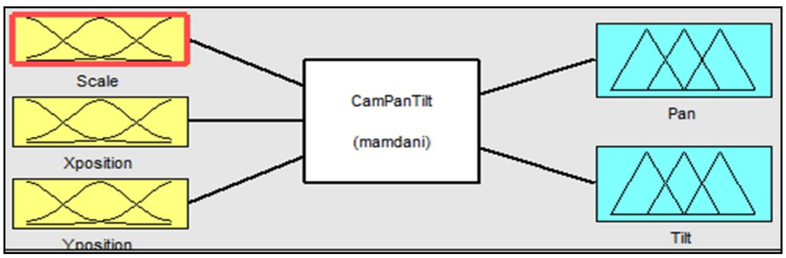


Figure 5: Second fuzzy inference system

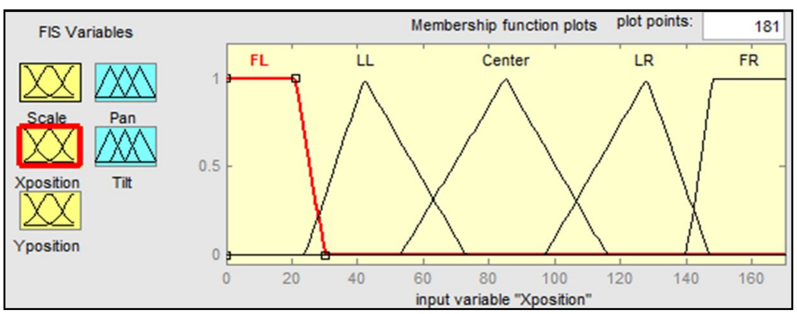


Figure 6: Membership functions for input variable "Xposition"

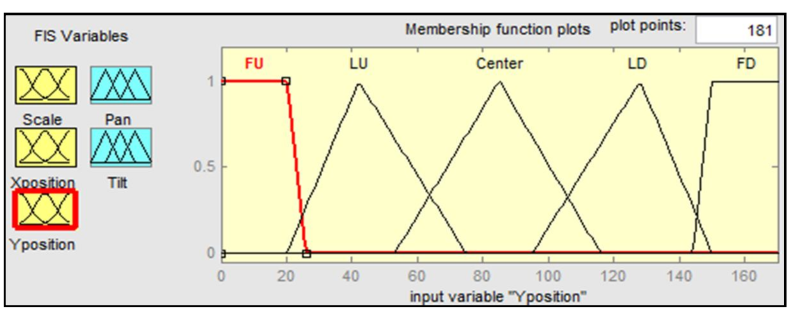


Figure 7: Membership functions for input variable "Yposition"

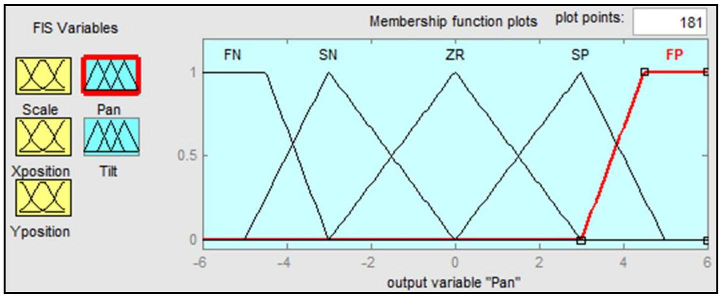


Figure 8: Membership functions for output variable "Pan" and "Tilt"

Similarly like first fuzzy inference system, we have to set 30 rules for second fuzzy inference system which are shown in Figure 9.

1. If (Scale is Low) and (Xposition is FL) then (Pan is FN) (1)
2. If (Scale is Low) and (Xposition is LL) then (Pan is SN) (1)
3. If (Scale is Low) and (Xposition is Center) then (Pan is ZR) (1)
4. If (Scale is Low) and (Xposition is LR) then (Pan is SP) (1)
5. If (Scale is Low) and (Xposition is FR) then (Pan is FP) (1)
6. If (Scale is Medium) and (Xposition is FL) then (Pan is SN) (1)
7. If (Scale is Medium) and (Xposition is LL) then (Pan is SN) (1)
8. If (Scale is Medium) and (Xposition is Center) then (Pan is ZR) (1)
9. If (Scale is Medium) and (Xposition is LR) then (Pan is SP) (1)
10. If (Scale is Medium) and (Xposition is FR) then (Pan is SP) (1)
11. If (Scale is High) and (Xposition is FL) then (Pan is SN) (1)
12. If (Scale is High) and (Xposition is LL) then (Pan is ZR) (1)
13. If (Scale is High) and (Xposition is Center) then (Pan is ZR) (1)
14. If (Scale is High) and (Xposition is LR) then (Pan is ZR) (1)
15. If (Scale is High) and (Xposition is FR) then (Pan is SP) (1)
16. If (Scale is Low) and (Yposition is FD) then (Tilt is FP) (1)
17. If (Scale is Low) and (Yposition is LD) then (Tilt is SP) (1)
18. If (Scale is Low) and (Yposition is Center) then (Tilt is ZR) (1)
19. If (Scale is Low) and (Yposition is LU) then (Tilt is SN) (1)
20. If (Scale is Low) and (Yposition is FU) then (Tilt is FN) (1)
21. If (Scale is Medium) and (Yposition is FD) then (Tilt is SP) (1)
22. If (Scale is Medium) and (Yposition is LD) then (Tilt is SP) (1)
23. If (Scale is Medium) and (Yposition is Center) then (Tilt is ZR) (1)
24. If (Scale is Medium) and (Yposition is LU) then (Tilt is SN) (1)
25. If (Scale is Medium) and (Yposition is FU) then (Tilt is SN) (1)
26. If (Scale is High) and (Yposition is FD) then (Tilt is SP) (1)
27. If (Scale is High) and (Yposition is LD) then (Tilt is ZR) (1)
28. If (Scale is High) and (Yposition is Center) then (Tilt is ZR) (1)
29. If (Scale is High) and (Yposition is LU) then (Tilt is ZR) (1)
30. If (Scale is High) and (Yposition is FU) then (Tilt is SN) (1)

Figure 9: Set of rules for the second fuzzy inference system

Now, we save these two fuzzy inference systems as "CamScale.fis" and "CamPanTilt.fis" respectively for later use.

B) USING MATLAB COMMAND LINE TO SIMULATE THE CAMERA TRACKING SYSTEM:

In part A, we have created two fuzzy inference system using MATLAB GUI. Now, we will simulate this two systems in MATLAB command line. Table 1 shows the given input for the simulation.

Range	Xposition	Yposition
100	85	100
200	150	20
50	30	53
101	150	20
200	0	0
0	170	170
100	85	85

Table 1: Given input to run simulation for the Camera Tracking System

After giving the first column of Table 1 as input for first fuzzy inference system we will get the following output for the scale variable.

Scale
6.0000
2.5295
9.4705
6.0000
2.5295
9.4705
6.0000

Table 2: Output of first fuzzy inference system

Now we will use the result of Table 2 along with second and third column of Table 1 as input of second fuzzy inference system to generate the final output of “Pan” and “Tilt” variables. In MATLAB command line we have used functions *readfis()* and *evalfis()*for reading previously generated *.fis file and running fuzzy inference system simulator respectively. Final outputs of *Pan* and *Tilt* generated by our simulation are shown in Table 3 and 4 respectively.

```
function [Pan Tilt] = CameraTracking()
Range = [100; 200; 50; 101; 200; 0; 100];
Horizontal_Pos = [85; 150; 30; 150; 0; 170; 85];
Vertical_Pos = [100; 20; 53; 20; 0; 170; 85];
fis1 = readfis('CamScale');
Scale = evalfis(Range,fis1);
fis2 = readfis('CamPanTilt');
[Pan Tilt] = evalfis([Scale Horizontal_PosVertical_Pos],fis2);
End
```

Figure 10: MATLAB command line codes to run the simulation of camera tracking system

Pan	
0.0000	0.4596
4.8649	-4.8649
0.0000	0.0000
2.6674	-2.6674
-4.8649	-4.8649
2.6674	2.6674
0.0000	0.0000

Table 3: Final result of output variable "Pan"

Tilt		
0	0	1
0	0	1
0	1	1
0	0	1
0	0	1
1	0	1
1	0	1
1	1	1
1	0	1
1	0	1
0	0	1
0	0	1
0	1	1
0	0	1
0	0	1
0	1	0
0	1	0
0	1	1
0	1	0
0	1	0
1	1	0
1	1	0
1	1	1
1	1	0
1	1	0
0	1	0
0	1	0
0	1	1
0	1	0
0	1	0

Table 4: Final result of output variable "Tilt"

PROBLEM TWO: FCM FOR CLUSTERING ANALYSIS OF THE WINE DATA

A) DATA ANALYSIS STATEGY TO SOLVE CLUSTERING PROBLEM:

A 13-dimentional data set is given for clustering (Appendix –A). I have tried to analyze 13-dimension data by observing and plotted it using MATLAB. But it's very difficult or almost impossible to make a decision from that 13-dimensional graph. It's didn't make any sense though clusters overlap each other and some point of one cluster was inside the range of another cluster. Later I figure out that we are seeing 2-dimnensionally but our plot is 13-dimention. Thus some points which seems to be inside a given clusteroriginal from another cluster of different dimension. Figure 11 shows the simulation of fuzzy C-means (FCM) algorithm on 13-dimensional data.

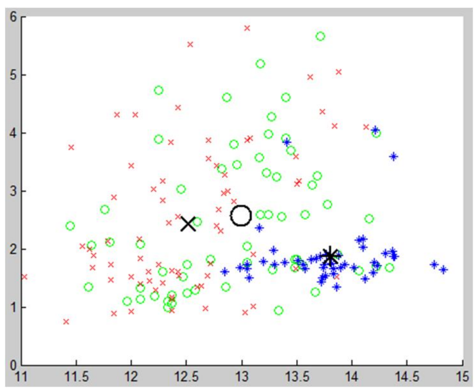


Figure 11: Result of running FCM on 13 dimensional data (Cluster size =3)

Then I decided to go for 2-dimensional data at a time. If we try to plot or analyze 2-dimensional data from given 13-dimensional data, then we have 78 combinations (12+11+10+9+8+7+6+5+4+3+2+1 =78) of 2-dimensional data. I have partitioned the original data in 78 pairs using a MATLAB code (Appendix-B). Then try to find out center, radius and standard deviation for all 78 combinations. Then take the average of 156 (78combinations*2clusters) radius and standard deviation for comparison with other results. Similarly I have run the simulation for cluster 3, 4 and 5 and take the average of all cluster radius and standard deviations for future analysis. I have written a MATLAB functions to simulate fuzzy C-means algorithm (Appendix-C) for different cluster size and store the all cluster centers, radius and standard deviation into *.txt files for analysis. It also shows a 2-dimensional graph of clustering operation on given data set. I wrote another MATLAB function to call the simulation of FCM algorithm for all 78 combinations (Appendix-D).

B) RESULTS OF ANALYSIS OF WINE DATA CLUSTERING:

After running the simulation and collecting the results I found that average cluster radius for cluster 2, 3, 4 and 5 are 70.15, 49.11, 35.66 and 28.02 unit respectively. When we increase the number of cluster then radius of cluster become smaller which makes sense but we can't find out any conclusion about how many cluster is good for this data using this information. Again we have standard deviations 208.87, 121.79, 78.83 and 59.25 for cluster number 2, 3, 4 and 5 respectively. This information clearly says that if I go for bigger number of cluster then standard deviation and radius will be further smaller. Thus we can't make any decision from only standard deviation or radius. We need to find out any second order calculation to make a decision. Table 5 contains the result summary of my findings. Full results are included in Appendix-E.

No of Clusters	Average radius	Average STD	Radius*Cluster	STD improve	Real STD improve
2	70.15	208.87	140.31	-	-
3	49.11	121.79	147.33	71.5 %	21.5 %
4	35.66	78.83	142.63	54.5 %	21.17 %
5	28.02	59.25	140.08	33.04 %	8.04 %

Table 5: Summary of wine data clustering result

C) FINAL DECISION OF SELECTING NUMBER OF CLUSTER AND DICISION

If we observe in the Table 5 carefully, especially second order calculations (last 3 columns), then we can see radius*cluster is bigger (147.33 which is clearly bigger than other chosen number of clusters) for cluster 3. From my observation if we have close cluster centers then radius of clusters will be smaller. Those close clustering are unnecessary clustering and could be merged, and really useful cluster centers are far away from each other. Figure 11 illustrate the above discussion for same data pair.

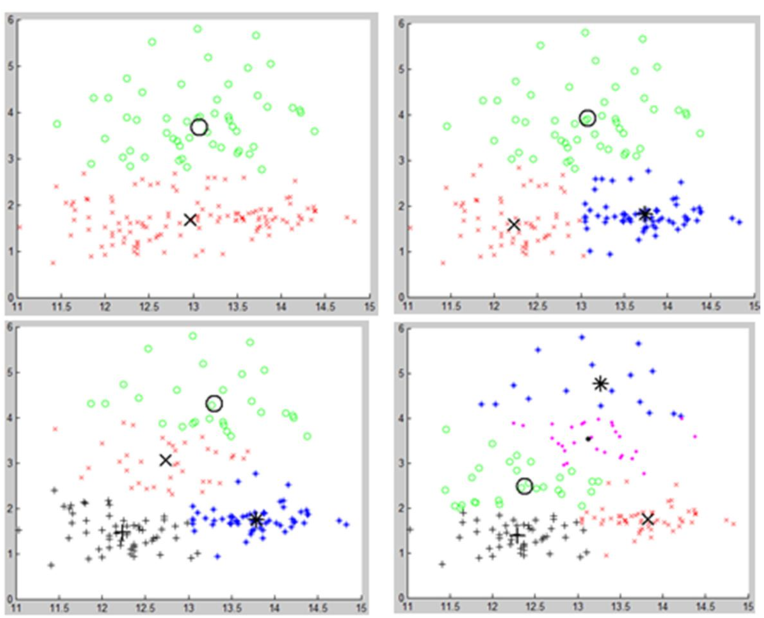


Figure 11: Cluster same pair of data using cluster number 2, 3, 4 and 5.

Again, if we consider the standard deviation for every cluster number, then we can see 71.5% improvement from cluster 2 to cluster 3 which is the biggest improvement from comparing other cluster number. But it's normal if we increase the number of cluster then standard deviation will decrease i.e. improvement may be bias with choosing greater number of cluster. Thus I deduced the ratio of the number of clusters to find real improvement in last column of Table 5. Formula I have used to find real improvement of standard deviation for choosing new number of clusters over pervious number of clusters is given below:

((STD of previous cluster)/(STD of chosen cluster) – (# of chosen cluster)/(# of previous cluster))*100%

For example, if we choose cluster size 3 instead of 2 then our real standard deviation improvement will be ((208.87/121.79)-(3/2))*100% = 21.5 %. In this way, I have calculated all real improvement of standard deviation and found that cluster number 3 is the best among other chosen cluster number. Considering above discussion I think ***three clusters*** would be appropriate for given wind data set. Figure 12 shows sample simulation of cluster number three for some pair of data from 78 pairs. Centers, radii, standard deviations and graphical representation of cluster number 2, 3, 4 and 5 for all 78 pairs are given in Appendix-E and F.

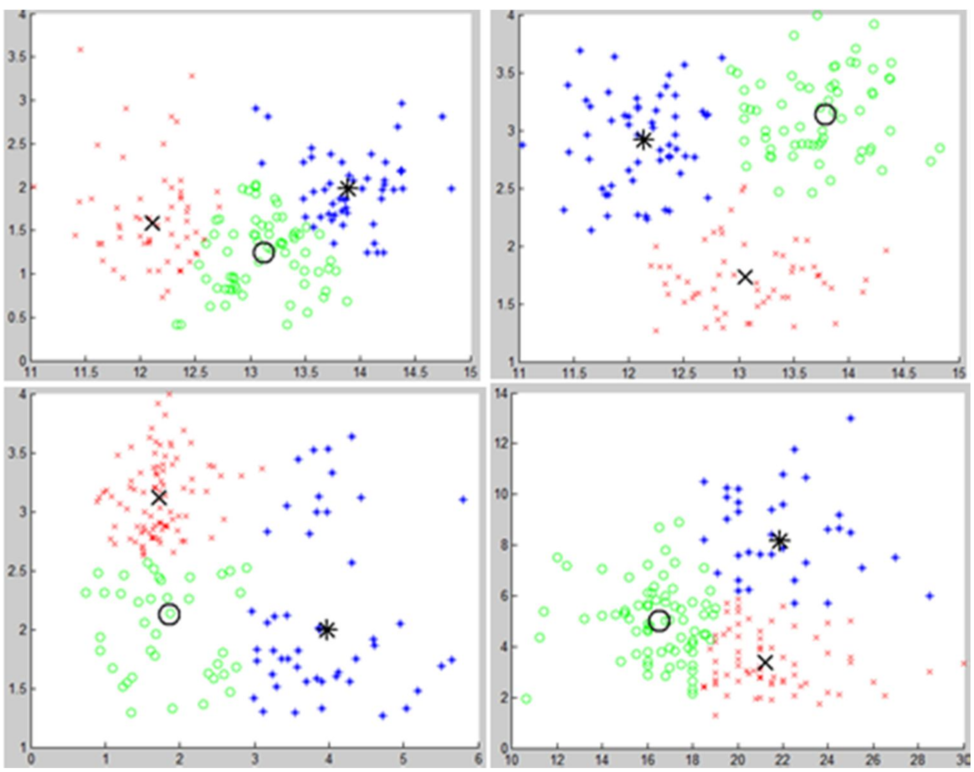


Figure 12: Sample simulation on 4 pair among 78 pairs of data set for cluster number 3.

To find the cluster members of each cluster, I have written a MATLAB function which is attached in Appendix – G. The first cluster contains 46 members, second cluster contains 71 members and third cluster contains 61 members. All members from each cluster are given below:

Cluster 1members:

1,2,3,4,6,7,8,9,10,11,12,13,14,15,16,17,18,19,23,24,27,28,30,31,32,33,
34,35,38,39,42,43,46,47,48,49,50,51,52,53,54,55,56,58,59,74.

Total members = 46.

Cluster 2 members:

60,62,64,65,67,68,72,73,76,77,78,80,81,84,85,86,87,88,91,92,93,94,95,9
8,100,102,103,104,106,107,108,109,111,112,114,115,116,117,118,119,120,
122,123,124,125,126,127,128,129,130,132,133,134,138,139,140,141,143,14
4,147,150,151,152,153,157,161,163,166,171,172,178.

Total members = 71.

Cluster 3 members:

5,20,21,22,25,26,29,36,37,40,41,44,45,57,61,63,66,69,70,71,75,79,82,83
,89,90,96,97,99,101,105,110,113,121,131,135,136,137,142,145,146,148,14
9,154,155,156,158,159,160,162,164,165,167,168,169,170,173,174,175,176,
177.

Total members = 61.

APPENDIX–A : GIVEN WINE DATA SET

Alcohol	Malic acid	Ash	Alcalinity of ash	Magnesium	Total phenols	Flavanoids	Nonflavanoid phenols	Proanthocyanins	Color intensity	Hue	OD280/OD315 of diluted wines	Proline
14.23	1.71	2.43	15.6	127	2.8	3.06	0.28	2.29	5.64	1.04	3.92	1065
13.2	1.78	2.14	11.2	100	2.65	2.76	0.26	1.28	4.38	1.05	3.4	1050
13.16	2.36	2.67	18.6	101	2.8	3.24	0.3	2.81	5.68	1.03	3.17	1185
14.37	1.95	2.5	16.8	113	3.85	3.49	0.24	2.18	7.8	0.86	3.45	1480
13.24	2.59	2.87	21	118	2.8	2.69	0.39	1.82	4.32	1.04	2.93	735
14.2	1.76	2.45	15.2	112	3.27	3.39	0.34	1.97	6.75	1.05	2.85	1450
14.39	1.87	2.45	14.6	96	2.5	2.52	0.3	1.98	5.25	1.02	3.58	1290
14.06	2.15	2.61	17.6	121	2.6	2.51	0.31	1.25	5.05	1.06	3.58	1295
14.83	1.64	2.17	14	97	2.8	2.98	0.29	1.98	5.2	1.08	2.85	1045
13.86	1.35	2.27	16	98	2.98	3.15	0.22	1.85	7.22	1.01	3.55	1045
14.1	2.16	2.3	18	105	2.95	3.32	0.22	2.38	5.75	1.25	3.17	1510
14.12	1.48	2.32	16.8	95	2.2	2.43	0.26	1.57	5	1.17	2.82	1280
13.75	1.73	2.41	16	89	2.6	2.76	0.29	1.81	5.6	1.15	2.9	1320
14.75	1.73	2.39	11.4	91	3.1	3.69	0.43	2.81	5.4	1.25	2.73	1150
14.38	1.87	2.38	12	102	3.3	3.64	0.29	2.96	7.5	1.2	3	1547
13.63	1.81	2.7	17.2	112	2.85	2.91	0.3	1.46	7.3	1.28	2.88	1310
14.3	1.92	2.72	20	120	2.8	3.14	0.33	1.97	6.2	1.07	2.65	1280
13.83	1.57	2.62	20	115	2.95	3.4	0.4	1.72	6.6	1.13	2.57	1130
14.19	1.59	2.48	16.5	108	3.3	3.93	0.32	1.86	8.7	1.23	2.82	1680
13.64	3.1	2.56	15.2	116	2.7	3.03	0.17	1.66	5.1	0.96	3.36	845
14.06	1.63	2.28	16	126	3	3.17	0.24	2.1	5.65	1.09	3.71	780
12.93	3.8	2.65	18.6	102	2.41	2.41	0.25	1.98	4.5	1.03	3.52	770
13.71	1.86	2.36	16.6	101	2.61	2.88	0.27	1.69	3.8	1.11	4	1035
12.85	1.6	2.52	17.8	95	2.48	2.37	0.26	1.46	3.93	1.09	3.63	1015
13.5	1.81	2.61	20	96	2.53	2.61	0.28	1.66	3.52	1.12	3.82	845
13.05	2.05	3.22	25	124	2.63	2.68	0.47	1.92	3.58	1.13	3.2	830
13.39	1.77	2.62	16.1	93	2.85	2.94	0.34	1.45	4.8	0.92	3.22	1195
13.3	1.72	2.14	17	94	2.4	2.19	0.27	1.35	3.95	1.02	2.77	1285
13.87	1.9	2.8	19.4	107	2.95	2.97	0.37	1.76	4.5	1.25	3.4	915
14.02	1.68	2.21	16	96	2.65	2.33	0.26	1.98	4.7	1.04	3.59	1035
13.73	1.5	2.7	22.5	101	3	3.25	0.29	2.38	5.7	1.19	2.71	1285
13.58	1.66	2.36	19.1	106	2.86	3.19	0.22	1.95	6.9	1.09	2.88	1515
13.68	1.83	2.36	17.2	104	2.42	2.69	0.42	1.97	3.84	1.23	2.87	990
13.76	1.53	2.7	19.5	132	2.95	2.74	0.5	1.35	5.4	1.25	3	1235
13.51	1.8	2.65	19	110	2.35	2.53	0.29	1.54	4.2	1.1	2.87	1095
13.48	1.81	2.41	20.5	100	2.7	2.98	0.26	1.86	5.1	1.04	3.47	920
13.28	1.64	2.84	15.5	110	2.6	2.68	0.34	1.36	4.6	1.09	2.78	880
13.05	1.65	2.55	18	98	2.45	2.43	0.29	1.44	4.25	1.12	2.51	1105

13.07	1.5	2.1	15.5	98	2.4	2.64	0.28	1.37	3.7	1.18	2.69	1020
14.22	3.99	2.51	13.2	128	3	3.04	0.2	2.08	5.1	0.89	3.53	760
13.56	1.71	2.31	16.2	117	3.15	3.29	0.34	2.34	6.13	0.95	3.38	795
13.41	3.84	2.12	18.8	90	2.45	2.68	0.27	1.48	4.28	0.91	3	1035
13.88	1.89	2.59	15	101	3.25	3.56	0.17	1.7	5.43	0.88	3.56	1095
13.24	3.98	2.29	17.5	103	2.64	2.63	0.32	1.66	4.36	0.82	3	680
13.05	1.77	2.1	17	107	3	3	0.28	2.03	5.04	0.88	3.35	885
14.21	4.04	2.44	18.9	111	2.85	2.65	0.3	1.25	5.24	0.87	3.33	1080
14.38	3.59	2.28	16	102	3.25	3.17	0.27	2.19	4.9	1.04	3.44	1065
13.9	1.68	2.12	16	101	3.1	3.39	0.21	2.14	6.1	0.91	3.33	985
14.1	2.02	2.4	18.8	103	2.75	2.92	0.32	2.38	6.2	1.07	2.75	1060
13.94	1.73	2.27	17.4	108	2.88	3.54	0.32	2.08	8.9	1.12	3.1	1260
13.05	1.73	2.04	12.4	92	2.72	3.27	0.17	2.91	7.2	1.12	2.91	1150
13.83	1.65	2.6	17.2	94	2.45	2.99	0.22	2.29	5.6	1.24	3.37	1265
13.82	1.75	2.42	14	111	3.88	3.74	0.32	1.87	7.05	1.01	3.26	1190
13.77	1.9	2.68	17.1	115	3	2.79	0.39	1.68	6.3	1.13	2.93	1375
13.74	1.67	2.25	16.4	118	2.6	2.9	0.21	1.62	5.85	0.92	3.2	1060
13.56	1.73	2.46	20.5	116	2.96	2.78	0.2	2.45	6.25	0.98	3.03	1120
14.22	1.7	2.3	16.3	118	3.2	3	0.26	2.03	6.38	0.94	3.31	970
13.29	1.97	2.68	16.8	102	3	3.23	0.31	1.66	6	1.07	2.84	1270
13.72	1.43	2.5	16.7	108	3.4	3.67	0.19	2.04	6.8	0.89	2.87	1285
12.37	0.94	1.36	10.6	88	1.98	0.57	0.28	0.42	1.95	1.05	1.82	520
12.33	1.1	2.28	16	101	2.05	1.09	0.63	0.41	3.27	1.25	1.67	680
12.64	1.36	2.02	16.8	100	2.02	1.41	0.53	0.62	5.75	0.98	1.59	450
13.67	1.25	1.92	18	94	2.1	1.79	0.32	0.73	3.8	1.23	2.46	630
12.37	1.13	2.16	19	87	3.5	3.1	0.19	1.87	4.45	1.22	2.87	420
12.17	1.45	2.53	19	104	1.89	1.75	0.45	1.03	2.95	1.45	2.23	355
12.37	1.21	2.56	18.1	98	2.42	2.65	0.37	2.08	4.6	1.19	2.3	678
13.11	1.01	1.7	15	78	2.98	3.18	0.26	2.28	5.3	1.12	3.18	502
12.37	1.17	1.92	19.6	78	2.11	2	0.27	1.04	4.68	1.12	3.48	510
13.34	0.94	2.36	17	110	2.53	1.3	0.55	0.42	3.17	1.02	1.93	750
12.21	1.19	1.75	16.8	151	1.85	1.28	0.14	2.5	2.85	1.28	3.07	718
12.29	1.61	2.21	20.4	103	1.1	1.02	0.37	1.46	3.05	0.906	1.82	870
13.86	1.51	2.67	25	86	2.95	2.86	0.21	1.87	3.38	1.36	3.16	410
13.49	1.66	2.24	24	87	1.88	1.84	0.27	1.03	3.74	0.98	2.78	472
12.99	1.67	2.6	30	139	3.3	2.89	0.21	1.96	3.35	1.31	3.5	985
11.96	1.09	2.3	21	101	3.38	2.14	0.13	1.65	3.21	0.99	3.13	886
11.66	1.88	1.92	16	97	1.61	1.57	0.34	1.15	3.8	1.23	2.14	428
13.03	0.9	1.71	16	86	1.95	2.03	0.24	1.46	4.6	1.19	2.48	392
11.84	2.89	2.23	18	112	1.72	1.32	0.43	0.95	2.65	0.96	2.52	500
12.33	0.99	1.95	14.8	136	1.9	1.85	0.35	2.76	3.4	1.06	2.31	750

12.7	3.87	2.4	23	101	2.83	2.55	0.43	1.95	2.57	1.19	3.13	463
12	0.92	2	19	86	2.42	2.26	0.3	1.43	2.5	1.38	3.12	278
12.72	1.81	2.2	18.8	86	2.2	2.53	0.26	1.77	3.9	1.16	3.14	714
12.08	1.13	2.51	24	78	2	1.58	0.4	1.4	2.2	1.31	2.72	630
13.05	3.86	2.32	22.5	85	1.65	1.59	0.61	1.62	4.8	0.84	2.01	515
11.84	0.89	2.58	18	94	2.2	2.21	0.22	2.35	3.05	0.79	3.08	520
12.67	0.98	2.24	18	99	2.2	1.94	0.3	1.46	2.62	1.23	3.16	450
12.16	1.61	2.31	22.8	90	1.78	1.69	0.43	1.56	2.45	1.33	2.26	495
11.65	1.67	2.62	26	88	1.92	1.61	0.4	1.34	2.6	1.36	3.21	562
11.64	2.06	2.46	21.6	84	1.95	1.69	0.48	1.35	2.8	1	2.75	680
12.08	1.33	2.3	23.6	70	2.2	1.59	0.42	1.38	1.74	1.07	3.21	625
12.08	1.83	2.32	18.5	81	1.6	1.5	0.52	1.64	2.4	1.08	2.27	480
12	1.51	2.42	22	86	1.45	1.25	0.5	1.63	3.6	1.05	2.65	450
12.69	1.53	2.26	20.7	80	1.38	1.46	0.58	1.62	3.05	0.96	2.06	495
12.29	2.83	2.22	18	88	2.45	2.25	0.25	1.99	2.15	1.15	3.3	290
11.62	1.99	2.28	18	98	3.02	2.26	0.17	1.35	3.25	1.16	2.96	345
12.47	1.52	2.2	19	162	2.5	2.27	0.32	3.28	2.6	1.16	2.63	937
11.81	2.12	2.74	21.5	134	1.6	0.99	0.14	1.56	2.5	0.95	2.26	625
12.29	1.41	1.98	16	85	2.55	2.5	0.29	1.77	2.9	1.23	2.74	428
12.37	1.07	2.1	18.5	88	3.52	3.75	0.24	1.95	4.5	1.04	2.77	660
12.29	3.17	2.21	18	88	2.85	2.99	0.45	2.81	2.3	1.42	2.83	406
12.08	2.08	1.7	17.5	97	2.23	2.17	0.26	1.4	3.3	1.27	2.96	710
12.6	1.34	1.9	18.5	88	1.45	1.36	0.29	1.35	2.45	1.04	2.77	562
12.34	2.45	2.46	21	98	2.56	2.11	0.34	1.31	2.8	0.8	3.38	438
11.82	1.72	1.88	19.5	86	2.5	1.64	0.37	1.42	2.06	0.94	2.44	415
12.51	1.73	1.98	20.5	85	2.2	1.92	0.32	1.48	2.94	1.04	3.57	672
12.42	2.55	2.27	22	90	1.68	1.84	0.66	1.42	2.7	0.86	3.3	315
12.25	1.73	2.12	19	80	1.65	2.03	0.37	1.63	3.4	1	3.17	510
12.72	1.75	2.28	22.5	84	1.38	1.76	0.48	1.63	3.3	0.88	2.42	488
12.22	1.29	1.94	19	92	2.36	2.04	0.39	2.08	2.7	0.86	3.02	312
11.61	1.35	2.7	20	94	2.74	2.92	0.29	2.49	2.65	0.96	3.26	680
11.46	3.74	1.82	19.5	107	3.18	2.58	0.24	3.58	2.9	0.75	2.81	562
12.52	2.43	2.17	21	88	2.55	2.27	0.26	1.22	2	0.9	2.78	325
11.76	2.68	2.92	20	103	1.75	2.03	0.6	1.05	3.8	1.23	2.5	607
11.41	0.74	2.5	21	88	2.48	2.01	0.42	1.44	3.08	1.1	2.31	434
12.08	1.39	2.5	22.5	84	2.56	2.29	0.43	1.04	2.9	0.93	3.19	385
11.03	1.51	2.2	21.5	85	2.46	2.17	0.52	2.01	1.9	1.71	2.87	407
11.82	1.47	1.99	20.8	86	1.98	1.6	0.3	1.53	1.95	0.95	3.33	495
12.42	1.61	2.19	22.5	108	2	2.09	0.34	1.61	2.06	1.06	2.96	345
12.77	3.43	1.98	16	80	1.63	1.25	0.43	0.83	3.4	0.7	2.12	372
12	3.43	2	19	87	2	1.64	0.37	1.87	1.28	0.93	3.05	564

11.45	2.4	2.42	20	96	2.9	2.79	0.32	1.83	3.25	0.8	3.39	625
11.56	2.05	3.23	28.5	119	3.18	5.08	0.47	1.87	6	0.93	3.69	465
12.42	4.43	2.73	26.5	102	2.2	2.13	0.43	1.71	2.08	0.92	3.12	365
13.05	5.8	2.13	21.5	86	2.62	2.65	0.3	2.01	2.6	0.73	3.1	380
11.87	4.31	2.39	21	82	2.86	3.03	0.21	2.91	2.8	0.75	3.64	380
12.07	2.16	2.17	21	85	2.6	2.65	0.37	1.35	2.76	0.86	3.28	378
12.43	1.53	2.29	21.5	86	2.74	3.15	0.39	1.77	3.94	0.69	2.84	352
11.79	2.13	2.78	28.5	92	2.13	2.24	0.58	1.76	3	0.97	2.44	466
12.37	1.63	2.3	24.5	88	2.22	2.45	0.4	1.9	2.12	0.89	2.78	342
12.04	4.3	2.38	22	80	2.1	1.75	0.42	1.35	2.6	0.79	2.57	580
12.86	1.35	2.32	18	122	1.51	1.25	0.21	0.94	4.1	0.76	1.29	630
12.88	2.99	2.4	20	104	1.3	1.22	0.24	0.83	5.4	0.74	1.42	530
12.81	2.31	2.4	24	98	1.15	1.09	0.27	0.83	5.7	0.66	1.36	560
12.7	3.55	2.36	21.5	106	1.7	1.2	0.17	0.84	5	0.78	1.29	600
12.51	1.24	2.25	17.5	85	2	0.58	0.6	1.25	5.45	0.75	1.51	650
12.6	2.46	2.2	18.5	94	1.62	0.66	0.63	0.94	7.1	0.73	1.58	695
12.25	4.72	2.54	21	89	1.38	0.47	0.53	0.8	3.85	0.75	1.27	720
12.53	5.51	2.64	25	96	1.79	0.6	0.63	1.1	5	0.82	1.69	515
13.49	3.59	2.19	19.5	88	1.62	0.48	0.58	0.88	5.7	0.81	1.82	580
12.84	2.96	2.61	24	101	2.32	0.6	0.53	0.81	4.92	0.89	2.15	590
12.93	2.81	2.7	21	96	1.54	0.5	0.53	0.75	4.6	0.77	2.31	600
13.36	2.56	2.35	20	89	1.4	0.5	0.37	0.64	5.6	0.7	2.47	780
13.52	3.17	2.72	23.5	97	1.55	0.52	0.5	0.55	4.35	0.89	2.06	520
13.62	4.95	2.35	20	92	2	0.8	0.47	1.02	4.4	0.91	2.05	550
12.25	3.88	2.2	18.5	112	1.38	0.78	0.29	1.14	8.21	0.65	2	855
13.16	3.57	2.15	21	102	1.5	0.55	0.43	1.3	4	0.6	1.68	830
13.88	5.04	2.23	20	80	0.98	0.34	0.4	0.68	4.9	0.58	1.33	415
12.87	4.61	2.48	21.5	86	1.7	0.65	0.47	0.86	7.65	0.54	1.86	625
13.32	3.24	2.38	21.5	92	1.93	0.76	0.45	1.25	8.42	0.55	1.62	650
13.08	3.9	2.36	21.5	113	1.41	1.39	0.34	1.14	9.4	0.57	1.33	550
13.5	3.12	2.62	24	123	1.4	1.57	0.22	1.25	8.6	0.59	1.3	500
12.79	2.67	2.48	22	112	1.48	1.36	0.24	1.26	10.8	0.48	1.47	480
13.11	1.9	2.75	25.5	116	2.2	1.28	0.26	1.56	7.1	0.61	1.33	425
13.23	3.3	2.28	18.5	98	1.8	0.83	0.61	1.87	10.52	0.56	1.51	675
12.58	1.29	2.1	20	103	1.48	0.58	0.53	1.4	7.6	0.58	1.55	640
13.17	5.19	2.32	22	93	1.74	0.63	0.61	1.55	7.9	0.6	1.48	725
13.84	4.12	2.38	19.5	89	1.8	0.83	0.48	1.56	9.01	0.57	1.64	480
12.45	3.03	2.64	27	97	1.9	0.58	0.63	1.14	7.5	0.67	1.73	880
14.34	1.68	2.7	25	98	2.8	1.31	0.53	2.7	13	0.57	1.96	660
13.48	1.67	2.64	22.5	89	2.6	1.1	0.52	2.29	11.75	0.57	1.78	620
12.36	3.83	2.38	21	88	2.3	0.92	0.5	1.04	7.65	0.56	1.58	520

13.69	3.26	2.54	20	107	1.83	0.56	0.5	0.8	5.88	0.96	1.82	680
12.85	3.27	2.58	22	106	1.65	0.6	0.6	0.96	5.58	0.87	2.11	570
12.96	3.45	2.35	18.5	106	1.39	0.7	0.4	0.94	5.28	0.68	1.75	675
13.78	2.76	2.3	22	90	1.35	0.68	0.41	1.03	9.58	0.7	1.68	615
13.73	4.36	2.26	22.5	88	1.28	0.47	0.52	1.15	6.62	0.78	1.75	520
13.45	3.7	2.6	23	111	1.7	0.92	0.43	1.46	10.68	0.85	1.56	695
12.82	3.37	2.3	19.5	88	1.48	0.66	0.4	0.97	10.26	0.72	1.75	685
13.58	2.58	2.69	24.5	105	1.55	0.84	0.39	1.54	8.66	0.74	1.8	750
13.4	4.6	2.86	25	112	1.98	0.96	0.27	1.11	8.5	0.67	1.92	630
12.2	3.03	2.32	19	96	1.25	0.49	0.4	0.73	5.5	0.66	1.83	510
12.77	2.39	2.28	19.5	86	1.39	0.51	0.48	0.64	9.899999	0.57	1.63	470
14.16	2.51	2.48	20	91	1.68	0.7	0.44	1.24	9.7	0.62	1.71	660
13.71	5.65	2.45	20.5	95	1.68	0.61	0.52	1.06	7.7	0.64	1.74	740
13.4	3.91	2.48	23	102	1.8	0.75	0.43	1.41	7.3	0.7	1.56	750
13.27	4.28	2.26	20	120	1.59	0.69	0.43	1.35	10.2	0.59	1.56	835
13.17	2.59	2.37	20	120	1.65	0.68	0.53	1.46	9.3	0.6	1.62	840
14.13	4.1	2.74	24.5	96	2.05	0.76	0.56	1.35	9.2	0.61	1.6	560

APPENDIX–B: PATITIONING 13-DIMENTIONAL DATA INTO 78 2-DIMENTIONAL DATA USING MATLAB CODE

```
functionWine_Data( )
      clearall;
      fcmdata = xlsread('Wine_Data.xls');          % Read Wine_Data.xls file
      data1_2 = [fcmdata(:, 1) fcmdata(:, 2)];
      data1_3 = [fcmdata(:, 1) fcmdata(:, 3)];
      data1_4 = [fcmdata(:, 1) fcmdata(:, 4)];
      data1_5 = [fcmdata(:, 1) fcmdata(:, 5)];
      data1_6 = [fcmdata(:, 1) fcmdata(:, 6)];
      data1_7 = [fcmdata(:, 1) fcmdata(:, 7)];
      data1_8 = [fcmdata(:, 1) fcmdata(:, 8)];
      data1_9 = [fcmdata(:, 1) fcmdata(:, 9)];
      data1_10 = [fcmdata(:, 1) fcmdata(:, 10)];
      data1_11 = [fcmdata(:, 1) fcmdata(:, 11)];
      data1_12 = [fcmdata(:, 1) fcmdata(:, 12)];
      data1_13 = [fcmdata(:, 1) fcmdata(:, 13)];

      data2_3 = [fcmdata(:, 2) fcmdata(:, 3)];
      data2_4 = [fcmdata(:, 2) fcmdata(:, 4)];
      data2_5 = [fcmdata(:, 2) fcmdata(:, 5)];
      data2_6 = [fcmdata(:, 2) fcmdata(:, 6)];
      data2_7 = [fcmdata(:, 2) fcmdata(:, 7)];
      data2_8 = [fcmdata(:, 2) fcmdata(:, 8)];
      data2_9 = [fcmdata(:, 2) fcmdata(:, 9)];
      data2_10 = [fcmdata(:, 2) fcmdata(:, 10)];
      data2_11 = [fcmdata(:, 2) fcmdata(:, 11)];
      data2_12 = [fcmdata(:, 2) fcmdata(:, 12)];
      data2_13 = [fcmdata(:, 2) fcmdata(:, 13)];

      data3_4 = [fcmdata(:, 3) fcmdata(:, 4)];
      data3_5 = [fcmdata(:, 3) fcmdata(:, 5)];
      data3_6 = [fcmdata(:, 3) fcmdata(:, 6)];
      data3_7 = [fcmdata(:, 3) fcmdata(:, 7)];
      data3_8 = [fcmdata(:, 3) fcmdata(:, 8)];
      data3_9 = [fcmdata(:, 3) fcmdata(:, 9)];
      data3_10 = [fcmdata(:, 3) fcmdata(:, 10)];
      data3_11 = [fcmdata(:, 3) fcmdata(:, 11)];
      data3_12 = [fcmdata(:, 3) fcmdata(:, 12)];
      data3_13 = [fcmdata(:, 3) fcmdata(:, 13)];

      data4_5 = [fcmdata(:, 4) fcmdata(:, 5)];
      data4_6 = [fcmdata(:, 4) fcmdata(:, 6)];
      data4_7 = [fcmdata(:, 4) fcmdata(:, 7)];
      data4_8 = [fcmdata(:, 4) fcmdata(:, 8)];
      data4_9 = [fcmdata(:, 4) fcmdata(:, 9)];
      data4_10 = [fcmdata(:, 4) fcmdata(:, 10)];
      data4_11 = [fcmdata(:, 4) fcmdata(:, 11)];
      data4_12 = [fcmdata(:, 4) fcmdata(:, 12)];
      data4_13 = [fcmdata(:, 4) fcmdata(:, 13)];

      data5_6 = [fcmdata(:, 5) fcmdata(:, 6)];
      data5_7 = [fcmdata(:, 5) fcmdata(:, 7)];
      data5_8 = [fcmdata(:, 5) fcmdata(:, 8)];
```

```
    data5_9 = [fcmdata(:, 5) fcmdata(:, 9)];
    data5_10 = [fcmdata(:, 5) fcmdata(:, 10)];
    data5_11 = [fcmdata(:, 5) fcmdata(:, 11)];
    data5_12 = [fcmdata(:, 5) fcmdata(:, 12)];
    data5_13 = [fcmdata(:, 5) fcmdata(:, 13)];

    data6_7 = [fcmdata(:, 6) fcmdata(:, 7)];
    data6_8 = [fcmdata(:, 6) fcmdata(:, 8)];
    data6_9 = [fcmdata(:, 6) fcmdata(:, 9)];
    data6_10 = [fcmdata(:, 6) fcmdata(:, 10)];
    data6_11 = [fcmdata(:, 6) fcmdata(:, 11)];
    data6_12 = [fcmdata(:, 6) fcmdata(:, 12)];
    data6_13 = [fcmdata(:, 6) fcmdata(:, 13)];

    data7_8 = [fcmdata(:, 7) fcmdata(:, 8)];
    data7_9 = [fcmdata(:, 7) fcmdata(:, 9)];
    data7_10 = [fcmdata(:, 7) fcmdata(:, 10)];
    data7_11 = [fcmdata(:, 7) fcmdata(:, 11)];
    data7_12 = [fcmdata(:, 7) fcmdata(:, 12)];
    data7_13 = [fcmdata(:, 7) fcmdata(:, 13)];

    data8_9 = [fcmdata(:, 8) fcmdata(:, 9)];
    data8_10 = [fcmdata(:, 8) fcmdata(:, 10)];
    data8_11 = [fcmdata(:, 8) fcmdata(:, 11)];
    data8_12 = [fcmdata(:, 8) fcmdata(:, 12)];
    data8_13 = [fcmdata(:, 8) fcmdata(:, 13)];

    data9_10 = [fcmdata(:, 9) fcmdata(:, 10)];
    data9_11 = [fcmdata(:, 9) fcmdata(:, 11)];
    data9_12 = [fcmdata(:, 9) fcmdata(:, 12)];
    data9_13 = [fcmdata(:, 9) fcmdata(:, 13)];

    data10_11 = [fcmdata(:, 10) fcmdata(:, 11)];
    data10_12 = [fcmdata(:, 10) fcmdata(:, 12)];
    data10_13 = [fcmdata(:, 10) fcmdata(:, 13)];

    data11_12 = [fcmdata(:, 11) fcmdata(:, 12)];
    data11_13 = [fcmdata(:, 11) fcmdata(:, 13)];

    data12_13 = [fcmdata(:, 12) fcmdata(:, 13)];
end
```

APPENDIX-C: IMPLEMENTATOIN OF FUZZY C-MEANS (FCM) ALGORITHM IN MATLAB FOR WINE CLUSTERING

```
function [result] = Wine_Clustering(data,nOfc )
      [center,U,objFcn] = fcm(data,nOfc);
      maxU = max(U);
      index1 = find(U(1, :) == maxU);
      index2 = find(U(2, :) == maxU);
      ifnOfc>=3
      index3 = find(U(3, :) == maxU);
      end
      ifnOfc>=4
      index4 = find(U(4, :) == maxU);
      end
      ifnOfc>=5
      index5 = find(U(5, :) == maxU);
      end

      figure
      line(data(index1,1),data(index1,2),'linestyle','none','marker',
      'o','color','g');
      line(data(index2,1),data(index2,2),'linestyle','none','marker',
      'x','color','r');
      ifnOfc>=3
      line(data(index3,1),data(index3,2),'linestyle','none','marker',
      '*','color','b');
      end
      ifnOfc>=4
      line(data(index4,1),data(index4,2),'linestyle','none','marker',
      '+','color','k');
      end
      ifnOfc>=5
      line(data(index5,1),data(index5,2),'linestyle','none','marker',
      '.','color','m');
      end

      holdon
      plot(center(1,1),center(1,2),'ko','markersize',15,'LineWidth',2)
      plot(center(2,1),center(2,2),'kx','markersize',15,'LineWidth',2)
      ifnOfc>=3
      plot(center(3,1),center(3,2),'k*','markersize',15,'LineWidth',2)
      end
      ifnOfc>=4
      plot(center(4,1),center(4,2),'k+','markersize',15,'LineWidth',2)
      end
      ifnOfc>=5
      plot(center(5,1),center(5,2),'k.','markersize',15,'LineWidth',2)
      end

      [m1 n1]=size(index1);
      [m2 n2]=size(index2);
      ifnOfc>=3
      [m3 n3]=size(index3);
      end
      ifnOfc>=4
```

```
[m4 n4]=size(index4);
end
ifnOfc>=5
[m5 n5]=size(index5);
end

std1=0;
r1=0;
std2=0;
r2=0;
ifnOfc>=3
std3=0;
r3=0;
end
ifnOfc>=4
std4=0;
r4=0;
end
ifnOfc>=5
std5=0;
r5=0;
end

fori=1:n1
temp=sqrt((data(index1(1,  i),  1)-center(1,  1))^2+(data(index1(1,  i),
2)-center(1, 2))^2);
if(r1<temp)
r1=temp;
end
std1=std1+temp;
end
fori=1:n2
temp=sqrt((data(index2(1,  i),  1)-center(2,  1))^2+(data(index2(1,  i),
2)-center(2, 2))^2);
if(r2<temp)
r2=temp;
end
std2=std2+temp;

end
ifnOfc>=3
fori=1:n3
temp=sqrt((data(index3(1,  i),  1)-center(3,  1))^2+(data(index3(1,  i),
2)-center(3, 2))^2);
if(r3<temp)
r3=temp;
end
std3=std3+temp;
end
end
ifnOfc>=4
fori=1:n4
temp=sqrt((data(index4(1,  i),  1)-center(4,  1))^2+(data(index4(1,  i),
2)-center(4, 2))^2);
if(r4<temp)
r4=temp;
```

```
end
std4=std4+temp;
end
end
ifnOfc>=5
fori=1:n5
temp=sqrt((data(index5(1,  i),  1)-center(5,  1))^2+(data(index5(1,  i),
2)-center(5, 2))^2);
if(r5<temp)
r5=temp;
end
std5=std5+temp;
end
end

x1=center(1,1);
y1=center(1,2);
std1=std1/sqrt(n1);
x2=center(2,1);
y2=center(2,2);
std2=std2/sqrt(n2);
ifnOfc>=3
x3=center(3,1);
y3=center(3,2);
std3=std3/sqrt(n3);
end
ifnOfc>=4
x4=center(4,1);
y4=center(4,2);
std4=std4/sqrt(n4);
end
ifnOfc>=5
x5=center(5,1);
y5=center(5,2);
std5=std5/sqrt(n5);
end

result = [x1 y1 r1 std1; x2 y2 r2 std2];
ifnOfc>=3
result = [x1 y1 r1 std1; x2 y2 r2 std2; x3 y3 r3 std3];
end
ifnOfc>=4
result = [x1 y1 r1 std1; x2 y2 r2 std2; x3 y3 r3 std3; x4 y4 r4 std4];
end
ifnOfc>=5
result = [x1 y1 r1 std1; x2 y2 r2 std2; x3 y3 r3 std3; x4 y4 r4 std4;
x5 y5 r5 std5];
end

ifnOfc==2
fileID=fopen('clus2_c1.txt','a');
fprintf(fileID,'[%4.2f,%4.2f]\n',x1,y1);
fclose(fileID);
fileID=fopen('clus2_c2.txt','a');
fprintf(fileID,'[%4.2f,%4.2f]\n',x2,y2);
fclose(fileID);
```

```
fileID=fopen('clus2_r1.txt','a');
fprintf(fileID,'%4.4f\n',r1);
fclose(fileID);
fileID=fopen('clus2_r2.txt','a');
fprintf(fileID,'%4.4f\n',r2);
fclose(fileID);
fileID=fopen('clus2_std1.txt','a');
fprintf(fileID,'%4.4f\n',std1);
fclose(fileID);
fileID=fopen('clus2_std2.txt','a');
fprintf(fileID,'%4.4f\n',std2);
fclose(fileID);
end
ifnOfc==3
fileID=fopen('clus3_c1.txt','a');
fprintf(fileID,'[%4.2f,%4.2f]\n',x1,y1);
fclose(fileID);
fileID=fopen('clus3_c2.txt','a');
fprintf(fileID,'[%4.2f,%4.2f]\n',x2,y2);
fclose(fileID);
fileID=fopen('clus3_c3.txt','a');
fprintf(fileID,'[%4.2f,%4.2f]\n',x3,y3);
fclose(fileID);
fileID=fopen('clus3_r1.txt','a');
fprintf(fileID,'%4.4f\n',r1);
fclose(fileID);
fileID=fopen('clus3_r2.txt','a');
fprintf(fileID,'%4.4f\n',r2);
fclose(fileID);
fileID=fopen('clus3_r3.txt','a');
fprintf(fileID,'%4.4f\n',r3);
fclose(fileID);
fileID=fopen('clus3_std1.txt','a');
fprintf(fileID,'%4.4f\n',std1);
fclose(fileID);
fileID=fopen('clus3_std2.txt','a');
fprintf(fileID,'%4.4f\n',std2);
fclose(fileID);
fileID=fopen('clus3_std3.txt','a');
fprintf(fileID,'%4.4f\n',std3);
fclose(fileID);
end
ifnOfc==4
fileID=fopen('clus4_c1.txt','a');
fprintf(fileID,'[%4.2f,%4.2f]\n',x1,y1);
fclose(fileID);
fileID=fopen('clus4_c2.txt','a');
fprintf(fileID,'[%4.2f,%4.2f]\n',x2,y2);
fclose(fileID);
fileID=fopen('clus4_c3.txt','a');
fprintf(fileID,'[%4.2f,%4.2f]\n',x3,y3);
fclose(fileID);
fileID=fopen('clus4_c4.txt','a');
fprintf(fileID,'[%4.2f,%4.2f]\n',x4,y4);
fclose(fileID);
fileID=fopen('clus4_r1.txt','a');
fprintf(fileID,'%4.4f\n',r1);
```

```
fclose(fileID);
fileID=fopen('clus4_r2.txt','a');
fprintf(fileID,'%4.4f\n',r2);
fclose(fileID);
fileID=fopen('clus4_r3.txt','a');
fprintf(fileID,'%4.4f\n',r3);
fclose(fileID);
fileID=fopen('clus4_r4.txt','a');
fprintf(fileID,'%4.4f\n',r4);
fclose(fileID);
fileID=fopen('clus4_std1.txt','a');
fprintf(fileID,'%4.4f\n',std1);
fclose(fileID);
fileID=fopen('clus4_std2.txt','a');
fprintf(fileID,'%4.4f\n',std2);
fclose(fileID);
fileID=fopen('clus4_std3.txt','a');
fprintf(fileID,'%4.4f\n',std3);
fclose(fileID);
fileID=fopen('clus4_std4.txt','a');
fprintf(fileID,'%4.4f\n',std4);
fclose(fileID);
end
ifnOfc==5
fileID=fopen('clus5_c1.txt','a');
fprintf(fileID,'[%4.2f,%4.2f]\n',x1,y1);
fclose(fileID);
fileID=fopen('clus5_c2.txt','a');
fprintf(fileID,'[%4.2f,%4.2f]\n',x2,y2);
fclose(fileID);
fileID=fopen('clus5_c3.txt','a');
fprintf(fileID,'[%4.2f,%4.2f]\n',x3,y3);
fclose(fileID);
fileID=fopen('clus5_c4.txt','a');
fprintf(fileID,'[%4.2f,%4.2f]\n',x4,y4);
fclose(fileID);
fileID=fopen('clus5_c5.txt','a');
fprintf(fileID,'[%4.2f,%4.2f]\n',x5,y5);
fclose(fileID);
fileID=fopen('clus5_r1.txt','a');
fprintf(fileID,'%4.4f\n',r1);
fclose(fileID);
fileID=fopen('clus5_r2.txt','a');
fprintf(fileID,'%4.4f\n',r2);
fclose(fileID);
fileID=fopen('clus5_r3.txt','a');
fprintf(fileID,'%4.4f\n',r3);
fclose(fileID);
fileID=fopen('clus5_r4.txt','a');
fprintf(fileID,'%4.4f\n',r4);
fclose(fileID);
fileID=fopen('clus5_r5.txt','a');
fprintf(fileID,'%4.4f\n',r5);
fclose(fileID);
fileID=fopen('clus5_std1.txt','a');
fprintf(fileID,'%4.4f\n',std1);
fclose(fileID);
```

```
        fileID=fopen('clus5_std2.txt','a');
        fprintf(fileID,'%4.4f\n',std2);
        fclose(fileID);
        fileID=fopen('clus5_std3.txt','a');
        fprintf(fileID,'%4.4f\n',std3);
        fclose(fileID);
        fileID=fopen('clus5_std4.txt','a');
        fprintf(fileID,'%4.4f\n',std4);
        fclose(fileID);
        fileID=fopen('clus5_std5.txt','a');
        fprintf(fileID,'%4.4f\n',std5);
        fclose(fileID);
        end
end
```

APPENDIX-D: MATLAB CODE FOR RUNNING FCM SIMULATION FOR 78 COMBINATIONS WITHIN A GIVEN RANGE OF CLUSTER NUMBER

```
functionRunWineCluster( )
%     Wine_Data
      fori=2:5
            Wine_Clustering(data1_2,i)
            Wine_Clustering(data1_3,i)
            Wine_Clustering(data1_4,i)
            Wine_Clustering(data1_5,i)
            Wine_Clustering(data1_6,i)
            Wine_Clustering(data1_7,i)
            Wine_Clustering(data1_8,i)
            Wine_Clustering(data1_9,i)
            Wine_Clustering(data1_10,i)
            Wine_Clustering(data1_11,i)
            Wine_Clustering(data1_12,i)
            Wine_Clustering(data1_13,i)
            Wine_Clustering(data2_3,i)
            Wine_Clustering(data2_4,i)
            Wine_Clustering(data2_5,i)
            Wine_Clustering(data2_6,i)
            Wine_Clustering(data2_7,i)
            Wine_Clustering(data2_8,i)
            Wine_Clustering(data2_9,i)
            Wine_Clustering(data2_10,i)
            Wine_Clustering(data2_11,i)
            Wine_Clustering(data2_12,i)
            Wine_Clustering(data2_13,i)
            Wine_Clustering(data3_4,i)
            Wine_Clustering(data3_5,i)
            Wine_Clustering(data3_6,i)
            Wine_Clustering(data3_7,i)
            Wine_Clustering(data3_8,i)
            Wine_Clustering(data3_9,i)
            Wine_Clustering(data3_10,i)
            Wine_Clustering(data3_11,i)
            Wine_Clustering(data3_12,i)
            Wine_Clustering(data3_13,i)
            Wine_Clustering(data4_5,i)
            Wine_Clustering(data4_6,i)
            Wine_Clustering(data4_7,i)
            Wine_Clustering(data4_8,i)
            Wine_Clustering(data4_9,i)
            Wine_Clustering(data4_10,i)
            Wine_Clustering(data4_11,i)
            Wine_Clustering(data4_12,i)
            Wine_Clustering(data4_13,i)
            Wine_Clustering(data5_6,i)
            Wine_Clustering(data5_7,i)
            Wine_Clustering(data5_8,i)
            Wine_Clustering(data5_9,i)
            Wine_Clustering(data5_10,i)
            Wine_Clustering(data5_11,i)
            Wine_Clustering(data5_12,i)
```

```
            Wine_Clustering(data5_13,i)
            Wine_Clustering(data6_7,i)
            Wine_Clustering(data6_8,i)
            Wine_Clustering(data6_9,i)
            Wine_Clustering(data6_10,i)
            Wine_Clustering(data6_11,i)
            Wine_Clustering(data6_12,i)
            Wine_Clustering(data6_13,i)
            Wine_Clustering(data7_8,i)
            Wine_Clustering(data7_9,i)
            Wine_Clustering(data7_10,i)
            Wine_Clustering(data7_11,i)
            Wine_Clustering(data7_12,i)
            Wine_Clustering(data7_13,i)
            Wine_Clustering(data8_9,i)
            Wine_Clustering(data8_10,i)
            Wine_Clustering(data8_11,i)
            Wine_Clustering(data8_12,i)
            Wine_Clustering(data8_13,i)
            Wine_Clustering(data9_10,i)
            Wine_Clustering(data9_11,i)
            Wine_Clustering(data9_12,i)
            Wine_Clustering(data9_13,i)
            Wine_Clustering(data10_11,i)
            Wine_Clustering(data10_12,i)
            Wine_Clustering(data10_13,i)
            Wine_Clustering(data11_12,i)
            Wine_Clustering(data11_13,i)
            Wine_Clustering(data12_13,i)
      end
end
```

APPENDIX-E: COLLECTED DATA FROM FCM SIMULATION FOR CLUSTER NUMBER 2, 3, 4 AND 5

Number of Cluster = 2

No.	Pair	center 1	radius 1	Std 1	center 2	radius 2	std 2	avg radius	avgstd
1	[1 2]	[12.97,1.67]	1.9482	9.7999	[13.07,3.67]	2.1293	6.4399	2.03875	8.1199
2	[1 3]	[12.78,2.28]	1.2509	4.1712	[13.7,2.45]	1.1637	4.0834	1.2073	4.1273
3	[1 4]	[13.22,17.04]	6.4995	17.5045	[12.79,7.72]	7.7223	17	7.1109	17.4349
4	[1 5]	[13.33,115.04]	21.0907	68.7442	[12.83,91.08]	46.9713	61.5752	34.031	65.1597
5	[1 6]	[13.7,2.68]	1.525	6.3753	[12.4,2]	1.7134	5.629	1.6192	6.00215
6	[1 7]	[13.53,2.91]	1.9115	8.8451	[12.64,1.3]	2.9321	5.9392	2.4218	7.39215
7	[1 8]	[12.26,0.38]	1.1346	3.6831	[13.7,0.34]	1.2412	3.4699	1.1879	3.5765
8	[1 9]	[12.32,1.46]	2.2862	5.7704	[13.67,1.73]	1.5299	5.7787	1.90805	5.77455
9	[1 10]	[13.51,7.59]	2.468	12.6321	[12.62,3.53]	5.4772	13.2429	3.9726	12.9375
10	[1 11]	[12.27,0.98]	1.435	3.9456	[13.69,0.95]	1.1425	4.1259	1.28875	4.03575
11	[1 12]	[12.39,2.43]	1.5851	6.1772	[13.66,2.9]	1.5777	7.7413	1.5814	6.95925
12	[1 13]	[13.7,1150.9]	293.779	1329.7	[12.7,551.2]	529.0703	1187.18	411.42465	1258.442
13	[2 3]	[1.68,2.35]	1.2339	5	[3.86,2.39]	1.9561	4.5173	1.595	4.722
14	[2 4]	[1.93,17.06]	7.7793	20.4451	[2.73,22.29]	6.5376	16.6546	7.15845	18.54985
15	[2 5]	[2.33,115.04]	21.1068	69.5646	[2.35,91.08]	46.9695	61.9081	34.03815	65.73635
16	[2 6]	[1.66,2.51]	1.5053	6.5659	[3.74,1.9]	2.1814	6.5878	1.84335	6.57685
17	[2 7]	[13.53,2.91]	2.7394	7.6814	[12.64,1.3]	2.5682	8.7039	2.6538	8.19265
18	[2 8]	[1.68,0.34]	1.0836	3.9641	[3.87,0.41]	1.9298	4.3441	1.5067	4.1541
19	[2 9]	[3.81,1.33]	2.2465	6.0074	[1.68,1.71]	1.5778	6.332	1.91215	6.1697
20	[2 10]	[2.74,7.71]	5.4004	13.9221	[2.04,3.63]	3.9011	15.5499	4.65075	14.736
21	[2 11]	[3.87,0.77]	1.9355	4.5748	[1.68,1.04]	1.1432	4.2817	1.53935	4.42825
22	[2 12]	[1.68,2.9]	2.3368	7.4153	[3.72,2.03]	1.6399	6.3237	1.98835	6.8695
23	[2 13]	[2.5,551.2]	529.0709	1187.1	[1.9,1150.9]	293.7784	1329.93	411.42465	1258.516
24	[3 4]	[2.46, 22.3]	7.6949	15.799	[2.29,17.06]	6.5263	15.5077	7.1106	15.65335
25	[3 5]	[2.49,115.05]	46.9511	61.2006	[2.3,91.09]	21.0853	67.7932	34.0182	64.4969
26	[3 6]	[2.41,2.81]	1.0655	3.5945	[2.33,1.72]	1.0083	3.3668	1.0369	3.48065
27	[3 7]	[2.35,1.04]	1.0989	4.378	[2.4,2.85]	2.3845	5.0005	1.7417	4.68925
28	[3 8]	[2.58,0.37]	0.6598	1.703	[2.17,0.34]	0.8086	1.8199	0.7342	1.76145
29	[3 9]	[2.37,2.07]	1.2629	3.6868	[2.36,1.19]	1.6045	3.5859	1.4337	3.63635
30	[3 10]	[2.32,3.66]	5.1044	9.8574	[2.44,7.9]	2.4039	12.2806	3.75415	11.069
31	[3 11]	[2.55,0.92]	0.7037	2.482	[2.18,1]	0.8239	2.2524	0.7638	2.3672
32	[3 12]	[2.39,1.79]	1.045	4.2364	[2.36,3.12]	1.0263	2.9214	1.03565	3.5789
33	[3 13]	[2.4,1150.9]	293.7774	1329.63	[2.3,551.2]	529.0702	1187.09	411.4238	1258.36
34	[4 5]	[19.72,90.92]	21.2759	73.7124	[19.15,114,61]	47.3879	72.1415	34.3319	72.92695
35	[4 6]	[17.04,2.52]	7.8152	16.7776	[22.29,2.05]	6.4618	16.5011	7.1385	16.63935
36	[4 7]	[22.24,1.57]	7.8725	18.692	[17.02,2.47]	6.6986	16.98	7.28555	17.836
37	[4 8]	[22.31,0.41]	7.6962	15.6908	[17.06,0.32]	6.4576	15.2063	7.0769	15.44855
38	[4 9]	[22.29,1.46]	7.731	16.3626	[17.04,1.71]	6.5707	16.5085	7.15085	16.43555
39	[4 10]	[17.21,5.09]	8.6816	28.006	[22.14,4.8]	7.3137	23.2185	7.99765	25.61225
40	[4 11]	[22.3,0.87]	7.7095	15.8526	[17.05,1.04]	6.4542	15.318	7.08185	15.5853
41	[4 12]	[17.06,2.85]	6.5417	16.5805	[22.31,2.35]	7.779	17.3583	7.16035	16.9694
42	[4 13]	[17.3,1150.9]	293.8239	1330.5	[20.5,551.2]	529.0654	1188.56	411.44465	1259.534
43	[5 6]	[115.05,2.48]	21.086	68.1803	[91.09,2.19]	46.9475	61.6326	34.01675	64.90645
44	[5 7]	[115.05,2.34]	21.0835	69.0635	[91.08,1.85]	46.9528	62.1752	34.01815	65.61935
45	[5 8]	[91.09,0.38]	46.9489	61.14	[115.05,0.33]	21.0858	67.7155	34.01735	64.42775
46	[5 9]	[91.09,1.53]	21.0856	68.2196	[115.05,1.67]	46.9762	61.3446	34.0309	64.7821
47	[5 10]	[115.01,5.97]	47.11	66.2093	[91.08,4.6]	21.2729	72.2372	34.19145	69.22325
48	[5 11]	[91.09,0.95]	46.9494	61.196	[115.05,0.95]	21.0862	67.8081	34.0178	64.50205
49	[5 12]	[91.08,2.58]	46.9608	61.6479	[115.04,2.65]	21.0893	68.3745	34.02505	65.0112
50	[5 13]	[95.7,551.1]	529.2345	1195.14	[106.2,1150.8]	294.6086	1342.44	411.92155	1268.79
51	[6 7]	[1.73,1.07]	1.0926	5.2502	[2.79,2.86]	2.251	4.9679	1.6718	5.10905
52	[6 8]	[2.82,0.31]	0.733	2.7214	[1.7126,0.43]	1.0607	2.781	0.89685	2.7512
53	[6 9]	[1.77,1.2]	1.6244	4.8423	[2.83,1.99]	1.0887	4.2303	1.35655	4.5363
54	[6 10]	[2.29,3.66]	2.3972	13.5267	[2.23,7.9]	5.1324	11.4589	3.7648	12.4928

55	[6 11]	[2.81,1.05]	1.0739	3.187	[1.7,0.84]	0.7659	3.1782	0.9199	3.1826
56	[6 12]	[1.7,1.86]	1.1756	5.0748	[2.72,3.12]	1.1087	3.9383	1.14215	4.50655
57	[6 13]	[2.1,551.2]	293.7779	1329.66	[2.8,1150.9]	529.0703	1187.09	411.4241	1258.377
58	[7 8]	[1.03,0.43]	2.2456	4.2992	[2.84,0.3]	0.8968	3.8904	1.5712	4.0948
59	[7 9]	[2.85,1.93]	2.2342	5.9274	[1.06,1.18]	1.5379	5.3124	1.88605	5.6199
60	[7 10]	[1.62,7.98]	3.7357	15.2216	[2.14,3.7]	5.0286	13.3882	4.38215	14.3049
61	[7 11]	[2.83,1.06]	2.2562	4.6534	[1.01,0.8]	0.9828	4.1082	1.6195	4.3808
62	[7 12]	[2.76,3.11]	1.2739	4.4202	[0.95,1.84]	2.3954	6.1128	1.83465	5.2665
63	[7 13]	[2.9,1150.9]	293.7805	1329.7	[1.6,551.2]	529.071	1187.14	411.42575	1258.42
64	[8 9]	[0.4,1.19]	0.8156	3.0462	[0.31,2.09]	1.4873	2.8445	1.15145	2.94535
65	[8 10]	[0.35,3.67]	5.0885	9.7972	[0.39,7.91]	2.3854	11.7787	3.73695	10.78795
66	[8 11]	[0.32,1.12]	0.3119	1.4205	[0.42,0.73]	0.6278	1.4984	0.46985	1.45945
67	[8 12]	[0.45,1.78]	0.8856	3.2473	[0.31,3.12]	0.6753	2.4238	0.78045	2.83555
68	[8 13]	[0.4,551.2]	529.0701	1187.08	[0.3,1150.9]	293.7773	1329.62	411.4237	1258.352
69	[9 10]	[1.58,7.87]	5.2463	10.6121	[1.57,3.66]	2.3978	13.4077	3.82205	12.0099
70	[9 11]	[2.06,1.04]	0.8529	3.4771	[1.17,0.87]	1.5483	3.3096	1.2006	3.39335
71	[9 12]	[1.86,3.11]	1.1514	3.7191	[1.14,1.82]	1.7418	5.5218	1.4466	4.62045
72	[9 13]	[1.9,1150.9]	529.0702	1187.2	[1.4,551.2]	293.7774	1329.67	411.4238	1258.433
73	[10 11]	[3.67,1.03]	2.3903	11.9537	[7.92,0.8]	5.0863	9.9388	3.7383	10.94625
74	[10 12]	[7.93,2.15]	2.4048	13.6113	[3.67,2.8]	5.07	11.6265	3.7374	12.6189
75	[10 13]	[5.6,1150.9]	529.0831	1187.25	[4.8,551.2]	293.7829	1330.5	411.433	1258.878
76	[11 12]	[1.06,3.1]	0.8977	3.6547	[0.76,1.76]	0.8392	2.6759	0.86845	3.1653
77	[11 13]	[1.1,1150.9]	293.7773	1329.63	[0.9,551.2]	529.0701	1187.08	411.4237	1258.356
78	[12 13]	[2.4,551.2]	293.7808	1329.7	[3.1,1150.9]	529.0703	1187.1	411.42555	1258.399

Number of Cluster = 3

No.	Pair	center 1	radius 1	Std 1	center 2	radius 2	std 2	center 3	radius 3	std 3	avg radius	avgstd
1	[1 2]	[13.08,3.92]	1.1042	3.8054	[12.23,1.59]	1.3544	4.815	[13.08,3.92]	1.8799	6.1251	1.446167	4.915167
2	[1 3]	[12.97,2.36]	1.0757	3.0775	[12.97,2.36]	0.8686	2.4648	[13.90,2.45]	0.9717	2.6925	0.972	2.744933
3	[1 4]	[13.51,15.94]	2.4705	11.7407	[13.51,15.94]	5.4627	10	[12.92,24.10]	6	11	4.610467	11.06003
4	[1 5]	[13.14,102.16]	39.5325	41.8616	[13.14,102.16]	9.9236	39.1304	[12.67,86.96]	16.9712	30.1243	22.14243	37.03877
5	[1 6]	[12.14,2.25]	1.1291	3.909	[12.14,2.25]	1.286	4.3342	[13.08,1.68]	1.0793	3.5011	1.1648	3.914767
6	[1 7]	[12.17,2.08]	1.3259	4.5381	[12.17,2.08]	1.1449	4.2079	[13.75,3.02]	3.0083	4.6297	1.826367	4.458567
7	[1 8]	[12.07,0.37]	0.5063	2.0437	[12.07,0.37]	1.0491	2.4209	[13.94,0.33]	0.8932	2.1901	0.8162	2.218233
8	[1 9]	[13.12,1.25]	1.2559	3.4133	[12.11,1.58]	2.1013	4.3253	[13.12,1.25]	1.1531	4.0558	1.503433	3.931467
9	[1 10]	[13.43,9.06]	2.04	8.7787	[12.26,2.89]	1.6756	5.864	[13.43,9.06]	4.0437	7.1971	2.586433	7.279933
10	[1 11]	[12.06,1.03]	0.9054	2.536	[12.06,1.03]	1.2369	2.7416	[13.02,0.85]	0.5504	2.5692	0.897567	2.6156
11	[1 12]	[13.79,3.14]	1.2977	4.296	[13.06,1.73]	1.0974	3.9596	[12.14,2.92]	1.112	3.7809	1.169033	4.012167
12	[1 13]	[13.8,1220.89]	227.7952	647.0609	[13.80,1220.89]	459.1122	966.8322	[12.52,459.37]	181.3754	582.7196	289.4276	732.2042
13	[2 3]	[1.57,2.32]	1.5739	3	[2.76,2.45]	0.8181	2.4984	[1.57,2.32]	1.1491	3.8945	1.180367	3.159267
14	[2 4]	[1.76,16.1]	6.0179	12.3596	[1.76,16.10]	5.5579	10.7728	[2.46,19.85]	3.7262	12.9692	5.100667	12.03387
15	[2 5]	[2.35,102.15]	10.1009	39.9837	[2.34,86.97]	16.9969	31.8473	[2.21,122.47]	39.5386	42.0468	22.21213	37.95927
16	[2 6]	[3.95,1.9]	1.152	3.6583	[1.71,1.90]	1.181	4.1795	[3.95,1.90]	1.9856	5.768	1.439533	4.535267
17	[2 7]	[3.83,1.01]	2.564	6.9864	[1.80,2.98]	2.1197	4.3316	[1.65,1.60]	1.3221	4.7445	2.001933	5.354167
18	[2 8]	[4.32,0.41]	1.4883	2.6565	[1.58,0.33]	0.8466	2.9964	[2.89,0.42]	0.7199	2.4218	1.018267	2.691567
19	[2 9]	[1.6,1.79]	1.5173	5.3278	[2.61,1.22]	1.6828	3.2374	[4.17,1.40]	2.2252	4.2016	1.808433	4.2556
20	[2 10]	[3.19,9.1]	4.1828	8.0497	[2.22,5.56]	3.338	10.3869	[1.89,3.06]	3.9406	8.648	3.820467	9.0282
21	[2 11]	[4.31,0.75]	1.4947	2.7834	[2.86,0.81]	0.7615	2.5237	[1.58,1.07]	0.8412	3.3561	1.032467	2.887733
22	[2 12]	[1.87,2.13]	1.1407	4.2731	[3.97,1.99]	2.1348	6.3223	[1.72,3.12]	1.4027	4.2417	1.5594	4.9457
23	[2 13]	[1.87,1220.89]	181.3803	582.8323	[1.87,1220.89]	459.1121	966.8331	[2.56,742.21]	227.7942	647.3596	289.4289	732.3417
24	[3 4]	[2.28,15.97]	5.7345	8.3472	[2.34,19.87]	2.1474	10.5984	[2.28,15.97]	5.452	8.8696	4.444633	9.271733
25	[3 5]	[2.53,122.49]	9.8494	38.1604	[2.53,122.49]	39.5118	41.6356	[2.25,86.97]	16.9654	29.1691	22.10887	36.3217
26	[3 6]	[2.32,2.41]	0.726	2.3829	[2.32,2.41]	1.0506	2.4622	[2.45,3.01]	0.8656	2.3909	0.880733	2.412
27	[3 7]	[2.43,3.08]	1.0584	2.5826	[2.43,3.08]	2.1507	3.2557	[2.26,2.03]	0.7022	2.9385	1.303767	2.9256
28	[3 8]	[2.66,0.37]	0.5781	1.2294	[2.34,0.38]	0.2898	1.3154	[2.00,0.31]	0.6441	0.9269	0.504	1.157233
29	[3 9]	[2.37,2.37]	1.1396	2.1435	[2.37,2.37]	1.3341	2.3872	[2.39,1.63]	0.8816	2.9912	1.118433	2.5073
30	[3 10]	[2.44,9.12]	1.7683	6.2777	[2.28,2.94]	1.6848	5.3132	[2.44,9.12]	3.8871	6.4226	2.446733	6.0045
31	[3 11]	[2.6,1.07]	0.3969	1.2694	[2.60,1.07]	0.6442	1.5714	[2.12,1.08]	0.7642	1.7385	0.601767	1.526433
32	[3 12]	[2.4,3.35]	0.6693	2.6268	[2.40,3.35]	0.8966	2.2716	[2.39,1.68]	1.0367	2.2337	0.867533	2.377367
33	[3 13]	[2.46,1220.89]	459.1123	966.8266	[2.30,459.37]	181.3747	582.6831	[2.39,742.21]	227.7929	647.015	289.4266	732.1749
34	[4 5]	[19.31,122.37]	11.4255	48.6884	[19.31,122.37]	39.6342	47.8703	[20.17,87.04]	17.3848	38.3807	22.81483	44.9798

35	[4 6]	[15.98,2.69]	5.4286	9.8577	[19.85,2.14]	2.2916	12.0475	[24.21,2.06]	5.9168	8.8715	4.545667	10.2589
36	[4 7]	[24.04,1.54]	6.1115	11.1455	[19.78,1.77]	2.5738	12.7302	[15.99,2.74]	5.8115	10.6476	4.832267	11.50777
37	[4 8]	[24.26,0.42]	5.7453	8.2206	[15.97,0.30]	5.3721	8.5145	[19.86,0.38]	2.1594	10.2919	4.4256	9.009
38	[4 9]	[19.87,1.48]	5.5664	9.8493	[24.24,1.47]	5.7846	8.8028	[19.87,1.48]	2.1899	11.65	4.513633	10.1007
39	[4 10]	[16.53,5.01]	6.6689	17.4791	[21.88,8.18]	6.9648	15.7186	[21.20,3.39]	8.7951	17.7294	7.476267	16.9757
40	[4 11]	[24.26,0.9]	5.3781	8.6027	[24.26,0.90]	5.7558	8.3784	[19.87,0.91]	2.1767	10.595	4.436867	9.192033
41	[4 12]	[24.13,2.3]	5.994	10.9758	[19.79,2.51]	2.1644	11.3887	[15.96,2.99]	5.4906	9.9611	4.549667	10.7752
42	[4 13]	[20.78,459.37]	227.8202	647.9809	[16.97,1220.89]	459.1098	967.107	[20.78,459.37]	181.3814	584.3848	289.4371	733.1576
43	[5 6]	[86.97,2.12]	9.891	38.9304	[122.49,2.50]	39.5115	41.975	[86.97,2.12]	16.9678	29.7511	22.12343	36.8855
44	[5 7]	[102.16,2.12]	39.4981	42.3592	[102.16,2.12]	9.932	39.967	[86.96,1.79]	16.9644	30.6494	22.1315	37.65853
45	[5 8]	[102.16,0.35]	16.9657	29.0539	[102.16,0.35]	9.8391	38.0446	[122.49,0.31]	39.5094	41.5759	22.10473	36.2248
46	[5 9]	[102.15,1.16]	39.5518	41.7587	[102.15,1.60]	9.8678	38.7356	[86.96,1.51]	16.9628	29.6483	22.12747	36.7142
47	[5 10]	[86.85,4.17]	10.5712	39.2493	[122.15,5.89]	39.9842	48.1849	[86.85,4.17]	17.029	36.3337	22.52813	41.25597
48	[5 11]	[102.16,0.97]	9.8507	38.133	[122.49,0.94]	39.5101	41.6126	[86.97,0.95]	16.9662	29.1684	22.109	36.30467
49	[5 12]	[122.51,2.75]	9.8882	39.156	[122.51,2.75]	39.489	42.3046	[86.97,2.56]	16.9848	29.8463	22.12067	37.1023
50	[5 13]	[92.42,459.58]	181.6942	594.3945	[104.03,742.69]	227.7341	667.7653	[105.35,1221.03]	458.9766	972.8622	289.4683	745.0073
51	[6 7]	[2.27,2.12]	1.9919	3.2465	[1.65,0.84]	0.9897	3.255	[2.27,2.12]	0.9695	3.1559	1.317033	3.219133
52	[6 8]	[2.4,0.34]	0.4583	1.5593	[3.05,0.29]	0.8354	1.653	[1.60,0.44]	0.6166	1.9962	0.636767	1.736167
53	[6 9]	[2.97,2.17]	1.4226	3.4647	[1.63,1.11]	0.8211	3.0687	[2.43,1.55]	1.1333	2.7967	1.125667	3.110033
54	[6 10]	[2.56,5.5]	2.1308	8.4264	[2.22,2.97]	1.7019	6.45	[1.88,9.15]	3.9624	6.7615	2.598367	7.212633
55	[6 11]	[2.36,1.05]	0.8596	1.9828	[2.35,1.05]	0.6656	2.2639	[1.58,0.77]	0.6246	1.9997	0.7166	2.082133
56	[6 12]	[1.65,1.73]	0.9555	3.0464	[2.93,3.21]	0.9492	3.3186	[2.26,2.87]	1.0622	2.8428	0.988967	3.069267
57	[6 13]	[2.08,459.37]	181.3741	582.7155	[2.14,742.21]	227.7974	647.0374	[2.87,1220.89]	459.1141	966.8292	289.4285	732.194
58	[7 8]	[0.77,0.46]	2.0016	2.5564	[0.77,0.46]	0.6292	2.149	[2.02,0.35]	0.6362	2.3579	1.089	2.354433
59	[7 9]	[3.1,2.08]	1.9896	3.7938	[0.80,1.08]	1.2452	3.1796	[2.10,1.53]	1.41	3.8288	1.548267	3.600733
60	[7 10]	[2.75,5.55]	4.0137	7.7291	[1.95,3.04]	2.457	7.8006	[2.75,5.55]	2.3714	9.5362	2.947367	8.3553
61	[7 11]	[2.04,1.06]	1.9914	2.8352	[2.04,1.06]	0.6672	2.5808	[0.78,0.73]	0.7459	2.3129	1.134833	2.5763
62	[7 12]	[0.83,1.73]	0.8565	2.9183	[3.07,3.17]	2.0733	3.6377	[2.08,2.90]	0.8688	3.6146	1.2662	3.3902
63	[7 13]	[1.63,742.21]	181.3739	582.8325	[1.63,742.20]	227.8001	647.1095	[3.03,1220.89]	459.1158	966.8296	289.4299	732.2572
64	[8 9]	[0.35,1.64]	0.5775	1.7223	[0.29,2.46]	1.1259	1.7895	[0.35,1.64]	0.4068	2.2618	0.7034	1.924533
65	[8 10]	[0.36,2.94]	3.8737	5.8875	[0.36,2.94]	1.6625	4.8684	[0.34,5.48]	1.8244	6.2077	2.453533	5.654533
66	[8 11]	[0.45,0.67]	0.547	0.9456	[0.45,0.67]	0.2839	0.9673	[0.32,0.97]	0.261	1.0795	0.363967	0.997467
67	[8 12]	[0.34,2.74]	0.4834	1.737	[0.34,2.74]	0.5247	1.7317	[0.29,3.37]	0.6282	1.5157	0.545433	1.661467
68	[8 13]	[0.29,1220.89]	459.1123	966.8254	[0.39,742.21]	227.7927	647.0083	[0.39,459.37]	181.3746	582.6675	289.4265	732.1671
69	[9 10]	[1.44,9.08]	2.1723	7.9157	[1.44,9.08]	4.1187	6.8161	[1.60,2.95]	1.984	6.5268	2.758333	7.0862
70	[9 11]	[2.42,1.04]	0.774	2.5328	[0.98,0.79]	0.7313	2.1727	[2.42,1.04]	1.1929	2.0229	0.8994	2.2428
71	[9 12]	[1.09,1.72]	0.821	3.0528	[2.15,3.21]	1.482	3.5052	[1.09,1.72]	1.1979	3.1374	1.166967	3.2318
72	[9 13]	[1.53,742.21]	227.7947	647.0512	[1.92,1220.89]	459.1132	966.8336	[1.45,459.37]	181.3738	582.6993	289.4272	732.1947
73	[10 11]	[2.94,1.05]	1.8468	6.1951	[9.13,0.69]	3.8692	6.2197	[2.94,1.05]	1.6685	5.1143	2.4615	5.843033
74	[10 12]	[5.5,2.8]	1.9042	6.6138	[5.50,2.80]	2.0175	8.4568	[9.11,1.80]	3.8967	7.0453	2.606133	7.371967
75	[10 13]	[4.14,459.38]	227.7757	647.6956	[5.82,1220.89]	459.1164	966.8852	[4.14,459.38]	181.3849	583.3462	289.4257	732.6423
76	[11 12]	[1.09,2.73]	0.633	2.0924	[1.01,3.36]	0.6449	1.8211	[0.71,1.67]	0.5414	1.8475	0.606433	1.920333
77	[11 13]	[0.89,742.21]	181.3751	582.6729	[1.08,1220.89]	459.1123	966.8256	[0.89,742.21]	227.7926	647.0122	289.4267	732.1702
78	[12 13]	[3.07,1220.89]	227.7955	647.0694	[3.07,1220.89]	459.1131	966.8281	[2.49,459.37]	181.3752	582.7317	289.4279	732.2097

Number of Cluster = 4

No.	Pair	center 1	radius 1	Std 1	center 2	radius 2	std 2
1	[1 2]	[13.79,1.75]	1.0445	3.3289	[13.30,4.31]	1.5119	4.2511
2	[1 3]	[12.51,2.22]	0.8725	2.2221	[11.87,2.38]	0.9055	1.9697
3	[1 4]	[12.96,25.05]	4.9515	7.1105	[12.68,21.31]	1.8578	8
4	[1 5]	[12.62,86.27]	16.2795	25.1821	[12.88,137.04]	24.9588	26.9338
5	[1 6]	[13.88,2.91]	1.0615	3.7742	[12.36,1.80]	0.7928	2.927
6	[1 7]	[13.01,2.67]	2.8173	3.1184	[12.09,1.89]	1.1641	3.7667
7	[1 8]	[14.05,0.32]	0.7808	1.7449	[11.79,0.36]	0.7792	1.3432
8	[1 9]	[12.62,1.06]	0.9622	2.4321	[13.39,1.44]	0.9068	3.0843
9	[1 10]	[13.66,6.36]	2.1305	6.3329	[12.15,2.72]	1.445	4.664
10	[1 11]	[13.36,0.88]	0.5653	2.0647	[14.07,1.03]	0.7661	1.988
11	[1 12]	[13.81,3.19]	1.0724	3.9824	[13.50,1.79]	0.8626	2.0767
12	[1 13]	[13.88,1334.24]	345.7636	486.912	[12.88,680.03]	164.9686	496.7567
13	[2 3]	[1.34,2.16]	0.8974	2	[4.51,2.43]	1.3209	2.547
14	[2 4]	[2.90,21.26]	2.9097	10.8088	[1.73,15.67]	5.1271	8.9525

15	[2 5]	[2.32,99.48]	7.0908	27.8558	[2.32,86.29]	16.3153	26.9612
16	[2 6]	[1.75,2.88]	0.9957	3.145	[1.45,2.03]	0.9387	3.0924
17	[2 7]	[4.43,1.07]	2.092	5.4534	[1.52,1.73]	1.2948	3.8879
18	[2 8]	[4.97,0.47]	0.8438	1.4069	[2.41,0.39]	0.6617	1.8268
19	[2 9]	[1.68,2.11]	1.6443	3.0789	[1.60,1.44]	1.2109	3.0031
20	[2 10]	[1.83,5.88]	1.9233	6.0141	[1.68,3.00]	2.4577	6.5114
21	[2 11]	[3.27,0.80]	0.778	2.9121	[4.57,0.73]	1.2311	2.1185
22	[2 12]	[4.49,1.99]	1.7172	4.6313	[1.79,3.26]	1.3134	3.2786
23	[2 13]	[2.00,1025.10]	170.1096	409.4348	[1.78,1334.24]	345.765	486.9069
24	[3 4]	[2.30,18.55]	1.4981	4.8516	[2.26,15.48]	4.9648	8.2375
25	[3 5]	[2.23,86.28]	16.2783	24.24	[2.52,114.21]	9.8198	26.961
26	[3 6]	[2.35,1.53]	0.6112	1.7808	[2.44,2.60]	0.781	1.8434
27	[3 7]	[2.40,0.70]	1.0514	1.6973	[2.24,1.64]	0.7583	2.3262
28	[3 8]	[2.37,0.28]	0.2274	0.7668	[1.95,0.31]	0.5924	0.8213
29	[3 9]	[2.38,1.45]	0.6859	2.1295	[2.37,1.94]	0.864	1.961
30	[3 10]	[2.43,9.96]	3.055	3.8848	[2.40,5.01]	1.2869	4.6357
31	[3 11]	[2.67,1.04]	0.5726	1.2783	[2.33,1.08]	0.6447	1.1389
32	[3 12]	[2.24,2.90]	0.6829	2.0789	[2.47,2.34]	0.5905	1.7234
33	[3 13]	[2.36,1025.10]	170.1028	409.3806	[2.38,680.03]	164.9677	496.2743
34	[4 5]	[18.96,113.37]	12.2277	33.3096	[19.28,135.28]	26.7217	30.9196
35	[4 6]	[25.10,2.15]	5.0331	6.525	[18.51,2.29]	1.5758	6.5067
36	[4 7]	[18.50,2.18]	1.9509	8.8135	[21.21,1.47]	2.2007	8.796
37	[4 8]	[25.13,0.41]	4.8731	5.9511	[21.35,0.40]	1.6529	6.5887
38	[4 9]	[15.47,1.83]	5.0744	8.5545	[25.11,1.50]	4.914	6.5239
39	[4 10]	[19.02,3.90]	2.9983	11.9977	[15.84,5.40]	6.2711	12.8542
40	[4 11]	[25.12,0.89]	4.8968	6.1244	[21.33,0.85]	1.7089	6.782
41	[4 12]	[18.55,2.71]	1.8201	6.9377	[21.33,2.35]	1.8492	8.2912
42	[4 13]	[17.15,1334.23]	345.7682	487.0375	[17.55,1025.11]	170.1145	410.1978
43	[5 6]	[99.46,2.34]	6.6092	26.7551	[114.20,2.53]	9.7957	27.4363
44	[5 7]	[99.45,2.05]	6.7122	27.6888	[136.88,2.18]	25.1164	26.7498
45	[5 8]	[114.21,0.33]	9.7934	26.9159	[86.28,0.39]	16.2788	24.0774
46	[5 9]	[114.20,1.61]	9.8023	27.1357	[86.27,1.50]	16.2747	24.6299
47	[5 10]	[86.13,4.00]	16.2899	31.2041	[136.85,3.83]	25.1754	26.9107
48	[5 11]	[114.21,0.91]	9.7949	26.9571	[99.47,0.97]	6.5369	26.2822
49	[5 12]	[114.17,2.61]	9.8479	27.4159	[99.46,2.64]	6.6751	27.1863
50	[5 13]	[106.63,1025.37]	170.4532	421.6457	[100.27,679.88]	165.8637	517.2137
51	[6 7]	[1.63,0.70]	0.7458	2.062	[3.04,3.23]	1.8551	2.5262
52	[6 8]	[2.58,0.32]	0.3363	1.1633	[1.49,0.43]	0.5071	1.2894
53	[6 9]	[2.87,1.83]	1.0129	2.4614	[1.57,1.03]	0.7821	2.5584
54	[6 10]	[2.22,2.74]	1.4784	5.1731	[2.75,6.54]	1.6888	5.4693
55	[6 11]	[3.12,1.05]	0.7596	1.5964	[2.02,1.04]	0.5528	1.5329
56	[6 12]	[2.44,3.08]	0.7954	2.6433	[3.01,3.19]	0.9082	2.6829
57	[6 13]	[2.12,440.99]	162.9929	482.6148	[2.04,680.03]	164.969	496.4133
58	[7 8]	[0.68,0.48]	0.4624	1.2331	[1.60,0.39]	0.476	1.6794
59	[7 9]	[3.24,2.21]	1.8745	3.1117	[0.71,1.04]	0.8574	2.6113
60	[7 10]	[2.04,2.89]	1.7462	6.5407	[3.06,5.70]	2.1401	6.4513
61	[7 11]	[2.55,1.07]	0.7457	1.8306	[0.70,0.71]	0.4274	1.2848
62	[7 12]	[0.76,1.70]	0.8991	2.537	[3.21,3.17]	1.9401	2.9511
63	[7 13]	[1.88,440.99]	162.9916	482.6549	[3.05,1334.24]	345.765	486.9144
64	[8 9]	[0.31,1.96]	0.4002	1.2523	[0.31,2.79]	0.7923	1.1507
65	[8 10]	[0.36,2.81]	1.5339	4.2702	[0.44,9.97]	3.0319	3.6079
66	[8 11]	[0.44,0.90]	0.2476	0.7022	[0.28,1.03]	0.2778	0.6736
67	[8 12]	[0.32,2.88]	0.3086	1.1085	[0.45,1.60]	0.4233	1.3049
68	[8 13]	[0.28,1025.10]	170.103	409.3772	[0.41,680.03]	164.9675	496.2492
69	[9 10]	[1.41,9.66]	3.5852	4.7084	[1.62,2.75]	1.9659	5.332
70	[9 11]	[2.75,1.07]	0.8919	1.3573	[1.46,1.00]	0.6216	1.9153
71	[9 12]	[1.07,1.68]	0.8142	2.8361	[1.79,3.32]	0.7838	2.6712
72	[9 13]	[1.44,680.03]	164.968	496.5995	[1.90,1334.24]	345.7638	486.9131
73	[10 11]	[2.82,1.05]	1.5408	4.5453	[9.97,0.65]	3.0318	3.8866
74	[10 12]	[6.54,2.64]	1.5327	5.7868	[9.62,1.70]	3.3921	4.8544
75	[10 13]	[4.00,440.99]	162.9944	483.0574	[5.04,1025.10]	170.1339	409.5078
76	[11 12]	[1.00,3.42]	0.5917	1.4599	[0.67,1.60]	0.5831	1.4528

77	[11 13]	[1.11,1334.24]	345.7636	486.9083	[0.96,440.99]	162.9935	482.583
78	[12 13]	[2.54,440.99]	162.993	482.6377	[2.27,680.03]	164.9761	496.6422

No.	Pair	center 3	radius 3	std 3	center 4	radius 4	std 4	avg radius	avgstd
1	[1 2]	[12.74,3.07]	1.4441	3.8281	[12.24,1.46]	1.2262	3.7848	1.306675	3.798225
2	[1 3]	[13.34,2.44]	0.8299	2.3005	[14.06,2.44]	0.8189	1.9203	0.8567	2.10315
3	[1 4]	[13.62,15.56]	5	10	[12.80,18.56]	2	7	3.500075	8.010125
4	[1 5]	[13.42,114.22]	9.7861	27.363	[13.09,99.48]	6.5392	27.1601	14.3909	26.65975
5	[1 6]	[12.15,2.57]	1.1225	2.7809	[13.36,1.76]	0.939	2.7167	0.97895	3.0497
6	[1 7]	[13.14,0.79]	1.3053	4.0457	[13.95,3.11]	0.9899	2.8525	1.56915	3.445825
7	[1 8]	[13.30,0.36]	0.4352	1.6504	[12.47,0.39]	0.4297	1.7066	0.606225	1.611275
8	[1 9]	[13.99,2.07]	1.2574	2.7324	[12.05,1.71]	1.9583	3.6628	1.271175	2.9779
9	[1 10]	[13.21,4.53]	1.3977	5.1363	[13.44,9.57]	3.5473	4.9663	2.130125	5.274875
10	[1 11]	[12.55,0.92]	0.5655	2.0919	[11.86,1.03]	1.07	1.6883	0.741725	1.958225
11	[1 12]	[12.56,1.85]	0.942	2.6489	[12.15,3.02]	1.1268	3.3766	1.00095	3.02115
12	[1 13]	[12.48,440.99]	162.9936	482.6274	[13.60,1025.10]	170.109	409.4083	210.9587	468.9261
13	[2 3]	[1.83,2.48]	0.8168	2.6606	[3.22,2.39]	0.795	2.6101	0.957525	2.5008
14	[2 4]	[2.47,24.92]	5.14	8.2861	[2.04,18.70]	2.282	8.3152	3.8647	9.09065
15	[2 5]	[1.73,137.02]	24.9775	26.8766	[2.39,114.24]	9.7691	27.7898	14.53818	27.37085
16	[2 6]	[4.41,2.00]	1.5179	4.1427	[3.08,1.70]	1.1563	2.8722	1.15215	3.313075
17	[2 7]	[1.77,3.02]	2.0785	4.1879	[3.05,0.91]	1.1809	3.0293	1.66155	4.139625
18	[2 8]	[3.67,0.39]	0.6661	1.9845	[1.53,0.33]	0.7973	2.6538	0.742225	1.968
19	[2 9]	[4.30,1.39]	2.2569	3.8548	[2.99,1.10]	0.9063	2.969	1.5046	3.22645
20	[2 10]	[3.70,4.60]	2.9037	6.5045	[3.17,9.40]	3.8978	7.9241	2.795625	6.738525
21	[2 11]	[1.27,1.08]	0.6698	1.7465	[1.84,1.04]	0.7206	2.1531	0.849875	2.23255
22	[2 12]	[3.13,1.90]	1.6126	3.4465	[1.53,2.53]	1.2575	3.6914	1.475175	3.76195
23	[2 13]	[2.35,440.99]	162.997	482.736	[2.60,680.03]	164.9735	497.0243	210.9613	469.0255
24	[3 4]	[2.39,21.35]	1.6677	6.7073	[2.67,25.14]	4.8619	6.0495	3.248125	6.461475
25	[3 5]	[2.39,136.97]	25.0322	26.6863	[2.40,99.47]	6.5325	26.3044	14.4157	26.04793
26	[3 6]	[2.41,3.12]	0.827	1.9337	[2.24,2.08]	0.8829	2.0037	0.775525	1.8904
27	[3 7]	[2.41,3.26]	1.9976	2.336	[2.40,2.54]	0.8357	2.2888	1.16075	2.162075
28	[3 8]	[2.69,0.37]	0.551	1.1429	[2.32,0.47]	0.2279	0.7943	0.399675	0.881325
29	[3 9]	[2.35,0.91]	1.0993	1.8385	[2.33,2.74]	0.9835	1.6134	0.908175	1.8856
30	[3 10]	[2.27,2.81]	1.5565	4.853	[2.43,7.21]	1.3624	3.2418	1.8152	4.153825
31	[3 11]	[2.36,0.69]	0.3934	1.1583	[1.97,1.07]	0.6151	1.089	0.55645	1.166125
32	[3 12]	[2.37,1.61]	1.0333	1.6299	[2.44,3.41]	0.8401	1.9656	0.7867	1.84945
33	[3 13]	[2.50,1334.24]	345.7637	486.9087	[2.29,440.99]	162.9931	482.586	210.9568	468.7874
34	[4 5]	[20.30,86.26]	16.5946	32.4985	[18.87,99.18]	9.6647	38.0528	16.30218	33.69513
35	[4 6]	[21.31,2.01]	1.8796	8.0521	[15.46,2.75]	4.924	8.5887	3.353125	7.418125
36	[4 7]	[15.50,2.85]	5.4016	8.7179	[24.94,1.58]	5.2289	7.4435	3.695525	8.442725
37	[4 8]	[15.49,0.29]	4.886	7.9013	[18.56,0.35]	1.4611	4.5706	3.218275	6.252925
38	[4 9]	[18.50,1.59]	2.2241	6.5663	[21.29,1.43]	1.7885	7.4601	3.50025	7.2762
39	[4 10]	[21.55,8.76]	5.5944	13.221	[23.03,3.40]	6.9693	13.2637	5.458275	12.83415
40	[4 11]	[15.48,1.05]	4.8803	7.955	[18.54,1.00]	1.4487	4.8184	3.233675	6.41995
41	[4 12]	[15.48,3.04]	5.0293	8.8383	[25.08,2.34]	5.0528	7.1339	3.43785	7.800275
42	[4 13]	[20.74,440.98]	162.9883	483.7945	[20.04,680.02]	165.0507	497.8378	210.9804	469.7169
43	[5 6]	[86.28,2.12]	16.2754	24.8418	[136.96,2.43]	25.0442	26.7921	14.43113	26.45633
44	[5 7]	[114.16,2.40]	9.8436	27.9313	[86.26,1.79]	16.2657	25.7663	14.48448	27.03405
45	[5 8]	[99.47,0.36]	6.5344	26.2364	[136.98,0.27]	25.0249	26.6462	14.40788	25.96898
46	[5 9]	[99.46,1.58]	6.5806	26.7718	[136.96,2.11]	25.0629	26.7469	14.43013	26.32108
47	[5 10]	[114.22,6.57]	10.2272	29.3793	[99.38,5.02]	8.0952	31.4595	14.94693	29.7384
48	[5 11]	[136.97,1.12]	25.0261	26.6478	[86.28,0.96]	16.2786	24.2711	14.40913	26.03955
49	[5 12]	[86.28,2.56]	16.2909	24.9622	[136.90,2.92]	25.0984	26.717	14.47808	26.57035
50	[5 13]	[91.98,440.92]	163.0274	493.2241	[105.88,1334.45]	345.5569	492.2419	211.2253	481.0814
51	[6 7]	[2.54,2.51]	0.9167	2.129	[1.89,1.64]	0.9701	2.4229	1.121925	2.285025
52	[6 8]	[3.13,0.28]	0.7513	1.3225	[2.00,0.41]	0.3179	1.1858	0.47815	1.24025
53	[6 9]	[2.88,2.52]	1.1057	2.2961	[2.15,1.45]	1.0956	2.4215	0.999075	2.43435
54	[6 10]	[2.36,4.71]	1.5594	6.0926	[1.76,9.61]	3.5449	4.837	2.067875	5.393
55	[6 11]	[1.52,0.73]	0.5563	1.5225	[2.59,1.04]	0.6863	1.5112	0.63875	1.54075
56	[6 12]	[1.85,2.38]	1.0374	2.1635	[1.62,1.63]	0.7108	2.1717	0.86295	2.41535
57	[6 13]	[2.89,1334.24]	345.7638	486.913	[2.73,1025.10]	170.1085	409.3867	210.9586	468.832
58	[7 8]	[3.27,0.28]	1.8217	1.8423	[2.53,0.32]	0.44	1.5362	0.800025	1.57275
59	[7 9]	[1.70,1.42]	1.3341	2.4955	[2.62,1.72]	1.294	2.5337	1.34	2.68805

60	[7 10]	[0.96,5.26]	2.0466	4.6661	[0.95,9.40]	3.6154	6.836	2.387075	6.123525
61	[7 11]	[3.28,1.05]	1.8073	1.9692	[1.65,1.01]	0.6056	2.2616	0.8965	1.83655
62	[7 12]	[1.69,2.49]	0.8457	2.0414	[2.45,3.09]	0.8181	2.7115	1.12575	2.56025
63	[7 13]	[1.45,680.03]	164.9769	496.8145	[2.77,1025.10]	170.1142	409.3924	210.9619	468.9441
64	[8 9]	[0.36,1.43]	0.3233	1.3867	[0.44,0.86]	0.4836	1.3197	0.49985	1.27735
65	[8 10]	[0.37,7.23]	1.3794	3.3329	[0.34,5.02]	1.0912	4.2605	1.7591	3.867875
66	[8 11]	[0.44,0.64]	0.2582	0.8378	[0.33,1.24]	0.5082	0.7858	0.32295	0.74985
67	[8 12]	[0.44,2.26]	0.3146	1.0784	[0.28,3.43]	0.568	1.3118	0.403625	1.2009
68	[8 13]	[0.30,1334.24]	345.7635	486.908	[0.38,440.99]	162.993	482.577	210.9568	468.7779
69	[9 10]	[1.49,4.72]	1.4907	5.6248	[1.78,6.68]	1.4352	5.1204	2.11925	5.1964
70	[9 11]	[0.91,0.78]	0.6895	1.6837	[1.95,1.04]	0.6732	1.5269	0.71905	1.6208
71	[9 12]	[1.47,2.71]	0.8693	2.3035	[2.53,2.96]	1.2084	2.284	0.918925	2.5237
72	[9 13]	[1.48,440.99]	162.9927	482.6011	[1.85,1025.10]	170.104	409.3926	210.9571	468.8766
73	[10 11]	[7.22,0.86]	1.2946	3.3331	[5.01,0.98]	1.1122	4.3833	1.74485	4.037075
74	[10 12]	[4.70,2.83]	1.7779	6.5612	[2.75,2.84]	1.482	4.9609	2.046175	5.540825
75	[10 13]	[5.69,680.04]	164.974	498.6242	[6.20,1334.24]	345.7719	486.9563	210.9686	469.5364
76	[11 12]	[1.11,2.88]	0.6033	1.547	[1.00,2.26]	0.4509	1.2574	0.55725	1.429275
77	[11 13]	[1.04,1025.10]	170.1034	409.3781	[0.87,680.03]	164.9675	496.2741	210.957	468.7859
78	[12 13]	[3.15,1025.10]	170.106	409.3996	[3.02,1334.24]	345.764	486.9102	210.9598	468.8974

Number of Cluster = 5

No.	Pair	center 1	radius 1	Std 1	center 2	radius 2	std 2	center 3	radius 3	std 3
1	[1 2]	[13.13,3.53]	1.2534	2.8097	[13.82,1.74]	1.0133	3.1787	[12.29,1.38]	1.2679	3.1064
2	[1 3]	[12.93,2.37]	0.862	1.8747	[11.77,2.45]	0.8083	1.7484	[12.33,2.18]	0.8231	1.6438
3	[1 4]	[12.75,18.94]	1.8747	7.9154	[12.96,25.14]	4.8585	7	[12.66,21.53]	2	6
4	[1 5]	[13.06,95.86]	5.1507	15.4171	[12.65,140.24]	21.7631	21.504	[12.55,85.66]	15.665	21.9328
5	[1 6]	[14.05,2.97]	0.9404	2.3284	[12.29,1.81]	0.7089	2.5656	[13.28,2.64]	0.7236	1.943
6	[1 7]	[13.48,0.76]	1.0431	2.1876	[13.99,3.16]	0.9254	2.7352	[12.08,2.09]	1.0575	3.1564
7	[1 8]	[14.22,0.31]	0.6147	0.84	[11.71,0.36]	0.7015	1.0847	[13.62,0.34]	0.3509	1.3008
8	[1 9]	[12.39,2.03]	1.806	3.1271	[12.78,0.94]	0.8387	2.0921	[13.45,1.44]	0.8765	2.6053
9	[1 10]	[13.74,5.62]	2.2143	4.3439	[13.35,7.51]	1.4563	4.0348	[12.13,2.67]	1.391	4.2486
10	[1 11]	[11.77,1.01]	1.0185	1.3111	[12.97,0.79]	0.5157	1.6723	[13.58,0.96]	0.4869	1.6544
11	[1 12]	[12.05,2.99]	1.0314	3.1201	[13.24,3.00]	0.7362	2.3388	[13.50,1.73]	0.874	1.8582
12	[1 13]	[13.15,814.63]	122.3743	318.5531	[12.44,420.71]	142.7103	444.1256	[13.89,1342.12]	337.8835	435.9695
13	[2 3]	[3.81,2.35]	0.5867	2	[5.09,2.41]	0.7688	1.4388	[1.26,2.12]	0.8255	1.7647
14	[2 4]	[1.84,12.42]	2.2862	3.6327	[1.71,16.58]	2.4454	6.636	[2.12,19.41]	2.1069	8.6675
15	[2 5]	[2.29,118.04]	10.0992	21.819	[2.45,105.03]	6.1756	20.4296	[1.56,140.28]	21.7158	21.4911
16	[2 6]	[3.21,1.69]	1.1563	3.1921	[1.73,3.05]	0.8327	2.0162	[1.75,2.52]	0.8865	2.0252
17	[2 7]	[2.99,0.79]	0.9601	2.7672	[1.73,3.01]	2.0951	3.6144	[4.57,0.74]	1.0906	2.2278
18	[2 8]	[1.19,0.33]	0.4562	1.2187	[1.76,0.32]	0.4524	1.5273	[3.83,0.38]	0.5971	1.5766
19	[2 9]	[1.66,2.25]	1.6076	2.7042	[3.14,1.06]	0.9776	2.7591	[1.38,1.30]	0.9809	2.15
20	[2 10]	[1.79,5.99]	1.9508	5.7802	[3.44,2.82]	2.3673	3.9849	[3.47,5.17]	2.1701	4.4006
21	[2 11]	[1.76,1.06]	0.499	1.7865	[1.20,1.09]	0.6929	1.5416	[2.71,0.80]	0.7735	1.8078
22	[2 12]	[3.30,1.89]	1.7035	4.1391	[1.84,3.38]	1.2582	2.3222	[1.62,2.83]	0.8116	2.5225
23	[2 13]	[2.26,420.72]	142.7249	444.3159	[1.78,1342.12]	337.8825	435.9686	[2.62,814.65]	122.3504	305.9863
24	[3 4]	[2.29,16.33]	1.7397	4.7699	[2.68,25.26]	4.7441	6.2197	[2.31,19.05]	1.2693	5.3451
25	[3 5]	[2.42,104.86]	6.1405	19.1571	[2.22,85.68]	15.6815	21.0566	[2.33,140.18]	21.8246	21.4078
26	[3 6]	[2.30,1.46]	0.6777	1.3836	[2.58,2.76]	0.7726	1.7255	[2.18,2.46]	0.951	1.5213
27	[3 7]	[2.49,2.83]	0.7491	1.795	[2.23,2.21]	0.7176	1.7654	[2.40,0.67]	1.0446	1.545
28	[3 8]	[2.33,0.48]	0.2373	0.6826	[1.90,0.31]	0.5458	0.6272	[2.51,0.28]	0.2697	0.6955
29	[3 9]	[2.34,2.00]	0.6996	1.5994	[2.19,1.42]	0.4902	1.4287	[2.30,2.82]	0.9004	1.3351
30	[3 10]	[2.43,7.52]	1.185	2.8945	[2.42,10.10]	2.9155	3.2673	[2.42,5.51]	0.9484	2.9756
31	[3 11]	[2.32,0.68]	0.3428	0.8441	[2.61,0.85]	0.6285	0.8987	[1.94,1.09]	0.5782	0.9012
32	[3 12]	[2.36,1.58]	1.0334	1.5403	[2.52,2.86]	0.7752	1.3802	[2.45,2.14]	0.5287	1.385
33	[3 13]	[2.42,800.35]	119.6472	323.108	[2.50,1341.35]	338.6489	434.9401	[2.36,1063.32]	131.6851	280.5719
34	[4 5]	[19.71,139.54]	22.4723	26.1345	[20.41,85.70]	16.0227	29.8266	[19.22,104.61]	7.7335	25.1587
35	[4 6]	[12.24,2.86]	2.0308	3.2788	[16.38,2.68]	1.7933	5.6137	[19.08,2.19]	1.5233	6.8681
36	[4 7]	[18.97,2.05]	2.0804	7.7994	[25.05,1.58]	5.1215	7.4789	[12.53,3.04]	3.1374	3.7566
37	[4 8]	[19.07,0.36]	1.2738	4.9735	[21.65,0.41]	1.3495	4.5574	[12.14,0.28]	1.8635	2.9843
38	[4 9]	[21.63,1.43]	1.6035	5.358	[16.39,1.75]	1.8838	5.5271	[25.25,1.51]	4.7685	6.6562
39	[4 10]	[18.28,4.18]	2.9316	10.2644	[24.94,4.21]	5.1284	9.4897	[21.09,8.97]	5.6129	12.32
40	[4 11]	[21.64,0.84]	1.4057	4.7725	[19.06,0.96]	1.2679	5.3161	[12.14,1.10]	1.8644	2.9966

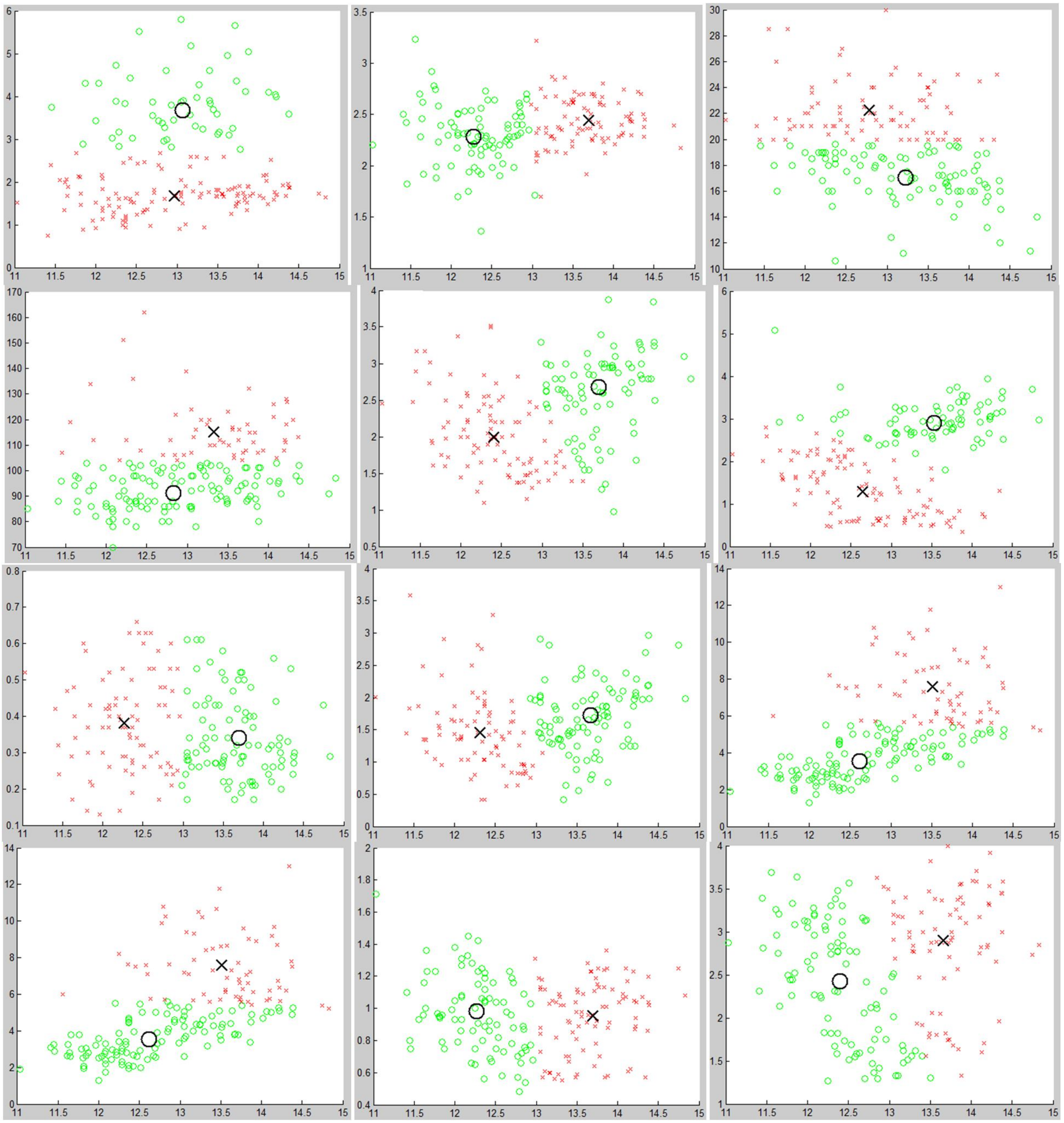
41	[4 12]	[19.07,2.63]	1.7098	6.9348	[25.21,2.35]	4.925	7.2461	[16.40,3.01]	1.887	5.9896
42	[4 13]	[17.17,1341.62]	338.385	435.4147	[19.31,805.38]	114.6303	313.3992	[20.82,614.84]	95.3904	342.2373
43	[5 6]	[117.97,2.55]	10.0368	21.4928	[95.91,2.24]	4.9825	14.3204	[85.67,2.12]	15.6747	21.6421
44	[5 7]	[117.92,2.50]	10.0957	22.1579	[95.91,1.87]	5.2335	15.6195	[104.87,2.18]	6.3273	20.7928
45	[5 8]	[85.68,0.39]	15.6847	20.9012	[95.91,0.38]	4.9075	13.6347	[117.95,0.32]	10.0511	21.0112
46	[5 9]	[140.20,2.21]	21.8309	21.4582	[104.86,1.63]	6.1503	19.7101	[117.94,1.69]	10.0686	21.1969
47	[5 10]	[95.17,5.11]	8.3826	19.1478	[103.87,5.21]	6.4595	20.7672	[139.26,3.50]	22.7617	20.7913
48	[5 11]	[104.98,0.98]	6.0212	19.2709	[140.31,1.16]	21.6904	21.4876	[95.95,0.94]	4.9642	13.7126
49	[5 12]	[95.92,2.60]	4.9968	15.0784	[85.68,2.59]	15.6909	21.8119	[104.91,2.55]	6.1668	19.9178
50	[5 13]	[105.20,1061.83]	137.1466	293.5173	[109.38,790.86]	129.4769	334.6036	[95.69,602.36]	94.0429	403.7972
51	[6 7]	[1.63,0.68]	0.7332	1.8407	[1.81,1.55]	1.0224	2.3681	[3.28,3.54]	1.5404	1.6208
52	[6 8]	[1.46,0.42]	0.4794	1.1699	[2.88,0.31]	0.2515	0.79	[3.34,0.25]	0.5484	0.727
53	[6 9]	[1.59,0.90]	0.6758	1.778	[2.99,1.92]	0.9006	1.982	[1.73,1.47]	0.6274	1.5691
54	[6 10]	[2.23,2.62]	1.3551	4.7441	[1.71,10.06]	3.1309	3.2591	[2.22,4.19]	1.4314	4.2535
55	[6 11]	[2.64,1.03]	0.4877	1.447	[1.74,0.80]	0.4456	1.0794	[1.41,0.70]	0.4459	0.8704
56	[6 12]	[2.85,3.43]	1.0007	1.9533	[1.62,1.60]	0.6976	1.9493	[2.25,2.97]	0.7041	2.097
57	[6 13]	[2.22,814.65]	122.348	305.9322	[2.79,1066.17]	128.835	262.8755	[2.20,420.72]	142.7175	444.1772
58	[7 8]	[3.49,0.28]	1.5973	1.2654	[1.46,0.39]	0.4632	1.3936	[2.20,0.32]	0.3653	1.1298
59	[7 9]	[2.85,1.84]	0.8128	1.8156	[3.28,2.31]	1.8552	2.8294	[0.68,1.02]	0.864	2.239
60	[7 10]	[3.17,6.38]	2.5512	4.5645	[0.76,5.45]	2.0551	4.3315	[2.62,4.44]	1.4139	4.1628
61	[7 11]	[2.80,1.06]	0.4034	1.5333	[1.45,0.94]	0.472	1.5186	[2.16,1.07]	0.6451	1.3949
62	[7 12]	[2.18,3.04]	0.8767	2.2799	[3.33,2.94]	1.9016	2.0522	[0.74,1.69]	0.9195	2.3389
63	[7 13]	[2.00,420.70]	142.7	444.2286	[1.32,621.33]	92.6796	289.6404	[2.91,1066.16]	128.8449	262.8584
64	[8 9]	[0.45,0.82]	0.4462	1.2333	[0.37,1.34]	0.3286	0.9606	[0.32,2.86]	0.7214	0.8625
65	[8 10]	[0.37,2.59]	1.3102	2.8186	[0.39,7.55]	1.1506	2.832	[0.34,5.55]	0.8319	2.5593
66	[8 11]	[0.50,0.92]	0.174	0.5564	[0.29,1.07]	0.1779	0.5262	[0.33,1.25]	0.4927	0.7409
67	[8 12]	[0.31,3.16]	0.3748	0.7986	[0.27,3.55]	0.4523	0.7375	[0.45,2.16]	0.3677	0.9206
68	[8 13]	[0.41,566.01]	83.9923	307.1318	[0.37,396.66]	118.6624	297.4696	[0.39,751.04]	134.9569	413.0292
69	[9 10]	[1.62,2.60]	1.9796	4.7678	[1.43,4.10]	1.3187	3.7117	[1.52,7.47]	1.4428	3.8208
70	[9 11]	[1.52,1.09]	0.6065	1.4039	[1.31,0.71]	0.3086	0.9928	[0.82,0.81]	0.6072	1.2947
71	[9 12]	[2.61,2.94]	0.9845	1.8719	[1.54,2.87]	0.7887	2.8535	[0.85,1.87]	0.6542	1.9953
72	[9 13]	[1.89,1342.12]	337.8816	435.9732	[1.88,1066.17]	128.8333	262.8932	[1.61,814.67]	122.3447	305.9159
73	[10 11]	[7.57,0.80]	1.2115	3.0547	[2.60,1.05]	1.3206	3.1442	[10.13,0.64]	2.8738	3.2438
74	[10 12]	[10.10,1.65]	2.9129	3.0306	[7.61,1.93]	1.739	3.9003	[4.20,2.63]	1.5644	4.8165
75	[10 13]	[3.77,420.46]	142.4668	444.8939	[5.17,1066.02]	128.9836	262.8283	[5.52,620.82]	93.1891	292.4178
76	[11 12]	[1.11,2.78]	0.6037	1.299	[1.02,3.16]	0.3615	1.0249	[0.95,2.14]	0.5572	1.3128
77	[11 13]	[1.05,1066.15]	128.8505	262.8393	[0.85,621.31]	92.6896	289.5742	[1.11,1342.11]	337.8866	435.9623
78	[12 13]	[2.61,420.71]	142.7131	444.1175	[3.01,1342.12]	337.8828	435.9696	[2.60,814.64]	122.3622	318.5791

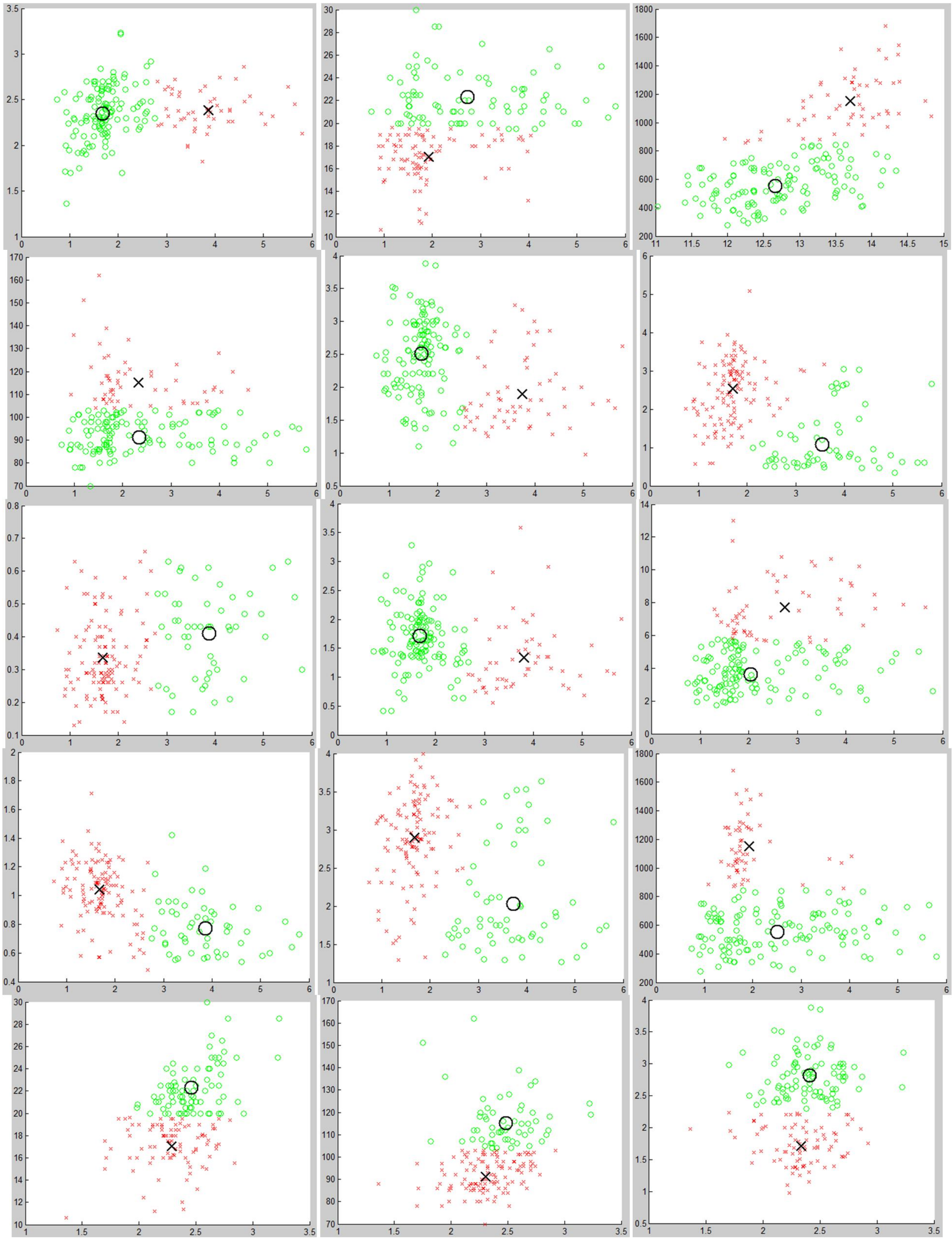
No.	Pair	center 4	radius 4	std 4	center 5	radius 5	std 5	avg radius	avgstd
1	[1 2]	[12.38,2.48]	1.555	3.3486	[13.27,4.76]	1.4729	3.4402	1.3125	3.17672
2	[1 3]	[13.59,2.49]	0.5768	1.717	[14.18,2.43]	0.6944	1.124	0.75292	1.62158
3	[1 4]	[13.76,12.38]	2	4	[13.55,16.39]	2	6	2.53348	6.21144
4	[1 5]	[13.47,117.93]	10.0988	21.5785	[13.20,104.84]	6.2402	19.8591	11.78356	20.0583
5	[1 6]	[12.02,2.57]	1.0166	2.5912	[13.29,1.64]	0.8792	2.4048	0.85374	2.3666
6	[1 7]	[13.16,2.74]	2.8358	2.9857	[12.54,0.97]	0.8076	2.6142	1.33388	2.73582
7	[1 8]	[12.33,0.39]	0.36	1.2059	[13.00,0.35]	0.3836	1.4184	0.48214	1.16996
8	[1 9]	[14.03,2.08]	1.0213	2.2863	[11.97,1.46]	1.0872	2.0001	1.12594	2.42218
9	[1 10]	[13.09,4.25]	1.5351	4.0402	[13.38,10.09]	3.0693	3.826	1.9332	4.0987
10	[1 11]	[14.18,1.02]	0.6543	1.3008	[12.37,1.02]	0.4786	1.6643	0.6308	1.52058
11	[1 12]	[14.01,3.28]	0.9259	2.4478	[12.55,1.76]	0.6959	2.2787	0.85268	2.40872
12	[1 13]	[13.74,1066.16]	128.84	262.9182	[12.76,621.35]	92.6534	289.6523	164.8923	350.2437
13	[2 3]	[2.69,2.46]	0.5769	1.8146	[1.74,2.44]	0.847	2.4228	0.72098	1.83934
14	[2 4]	[2.36,25.07]	4.9745	8.2956	[3.20,21.64]	2.7002	8.725	2.90264	7.19136
15	[2 5]	[2.27,96.01]	5.0425	15.9362	[2.27,85.68]	15.7123	23.6945	11.74908	20.67408
16	[2 6]	[4.49,1.97]	1.57	4.0011	[1.44,1.88]	0.9264	2.4582	1.07438	2.73856
17	[2 7]	[3.79,2.64]	2.0064	2.5085	[1.48,1.70]	1.2521	3.9703	1.48086	3.01764
18	[2 8]	[2.74,0.42]	0.5559	1.678	[5.12,0.48]	0.7022	1.1865	0.55276	1.43742
19	[2 9]	[4.35,1.37]	2.2884	3.9157	[1.86,1.64]	0.7547	2.0463	1.32184	2.71506
20	[2 10]	[1.55,3.11]	1.624	5.8803	[3.15,9.51]	3.7867	7.4504	2.37978	5.49928
21	[2 11]	[3.81,0.78]	0.633	1.7659	[5.10,0.71]	0.6994	1.2045	0.65956	1.62126
22	[2 12]	[1.56,1.94]	0.949	2.3751	[4.61,1.94]	1.7284	3.7507	1.29014	3.02192

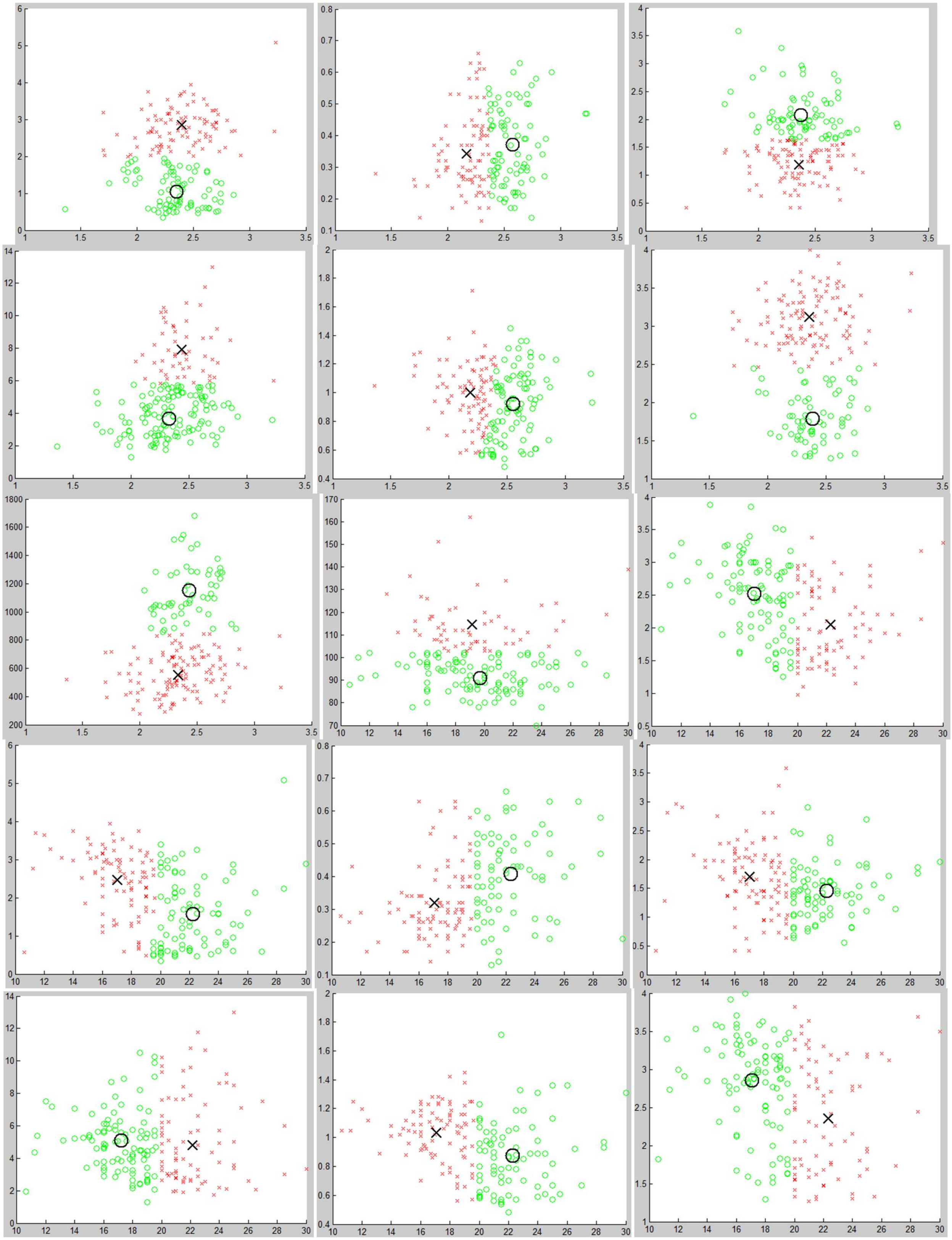
23	[2 13]	[2.60,621.36]	96.6467	300.7791	[2.01,1066.16]	128.8355	263.0609	165.688	350.0222
24	[3 4]	[2.17,12.17]	1.85	3.1233	[2.39,21.65]	1.3704	4.6243	2.1947	4.81646
25	[3 5]	[2.56,117.94]	10.0629	21.1793	[2.39,95.90]	4.9008	13.7879	11.72206	19.31774
26	[3 6]	[2.35,3.22]	0.6954	1.3395	[2.43,1.87]	0.5049	1.603	0.72032	1.51458
27	[3 7]	[2.31,1.47]	0.5871	1.7548	[2.39,3.44]	1.8447	1.8116	0.98862	1.73436
28	[3 8]	[2.22,0.28]	0.1801	0.5782	[2.71,0.42]	0.5173	0.9129	0.35004	0.69928
29	[3 9]	[2.62,1.52]	0.7227	1.5409	[2.36,0.87]	1.0932	1.4998	0.78122	1.48078
30	[3 10]	[2.26,2.59]	1.3318	3.3278	[2.35,4.03]	0.9764	3.0549	1.47142	3.10402
31	[3 11]	[2.64,1.16]	0.5789	0.8357	[2.29,1.07]	0.6452	1.137	0.55472	0.92334
32	[3 12]	[2.13,2.93]	0.6125	1.5776	[2.44,3.46]	0.8285	1.6542	0.75566	1.50746
33	[3 13]	[2.28,416.07]	138.0726	379.9689	[2.39,611.18]	96.1801	355.2772	164.8468	354.7732
34	[4 5]	[19.10,117.45]	12.0869	29.4865	[18.60,95.70]	8.6001	24.3821	13.3831	26.99768
35	[4 6]	[21.64,1.98]	1.5999	5.8663	[25.24,2.17]	4.8887	6.6591	2.3672	5.6572
36	[4 7]	[21.48,1.45]	2.066	8.0232	[16.40,2.76]	1.845	6.2383	2.85006	6.65928
37	[4 8]	[25.25,0.41]	4.7559	6.1255	[16.33,0.30]	1.7322	4.4571	2.19498	4.61956
38	[4 9]	[12.40,2.16]	2.5053	3.8423	[19.07,1.53]	2.0973	6.7445	2.57168	5.62562
39	[4 10]	[15.63,5.49]	6.1509	11.3199	[21.05,3.24]	2.8567	7.9116	4.5361	10.26112
40	[4 11]	[16.33,1.06]	1.7338	4.5396	[25.24,0.89]	4.7753	6.285	2.20942	4.78196
41	[4 12]	[12.29,2.95]	2.0345	3.2069	[21.64,2.33]	1.6889	6.5754	2.44904	5.99056
42	[4 13]	[16.93,1064.34]	130.6628	281.9129	[20.73,417.71]	139.7223	399.8607	163.7582	354.565
43	[5 6]	[140.23,2.39]	21.773	21.5819	[104.91,2.36]	6.2797	19.9053	11.74934	19.7885
44	[5 7]	[140.17,2.07]	21.8303	21.5378	[85.66,1.80]	15.6575	22.5021	11.82886	20.52202
45	[5 8]	[104.87,0.35]	6.1258	19.1019	[140.19,0.26]	21.8145	21.3835	11.71672	19.2065
46	[5 9]	[95.91,1.52]	5.075	14.3718	[85.68,1.49]	15.6792	21.3909	11.7608	19.62558
47	[5 10]	[117.08,6.54]	11.0101	25.7289	[85.34,3.68]	15.4659	26.8471	12.81596	22.65646
48	[5 11]	[118.04,0.90]	9.9608	21.0777	[85.69,0.96]	15.6919	21.0899	11.6657	19.32774
49	[5 12]	[140.15,2.87]	21.8536	21.4444	[117.94,2.69]	10.0918	21.7692	11.75998	20.00434
50	[5 13]	[91.40,411.89]	133.9956	364.6196	[105.92,1341.31]	338.6973	439.2881	166.6719	367.1652
51	[6 7]	[2.84,2.94]	0.9655	1.726	[2.42,2.31]	0.675	1.6248	0.9873	1.83608
52	[6 8]	[2.47,0.32]	0.2894	1.0214	[1.92,0.43]	0.3434	1.0704	0.38242	0.95574
53	[6 9]	[2.46,1.47]	1.0549	2.1549	[2.85,2.64]	1.0078	2.0186	0.8533	1.90052
54	[6 10]	[2.77,5.65]	1.7931	4.6321	[2.11,7.60]	1.7522	4.1738	1.89254	4.21252
55	[6 11]	[2.13,1.09]	0.699	1.5445	[3.15,1.05]	0.7307	1.4289	0.56178	1.27404
56	[6 12]	[2.96,2.90]	0.9913	1.7895	[1.71,2.22]	0.9908	1.8218	0.8769	1.92218
57	[6 13]	[2.89,1342.12]	337.8823	435.9721	[1.97,621.36]	96.6364	300.5458	165.6838	349.9006
58	[7 8]	[2.86,0.30]	0.3354	1.2845	[0.65,0.48]	0.4802	1.0592	0.64828	1.2265
59	[7 9]	[2.24,1.51]	1.3107	2.0362	[1.52,1.40]	1.3162	2.3528	1.23178	2.2546
60	[7 10]	[0.90,9.56]	3.4687	5.7689	[1.94,2.70]	1.568	5.0278	2.21138	4.7711
61	[7 11]	[3.42,1.06]	1.6684	1.5085	[0.66,0.70]	0.4586	1.2677	0.7295	1.4446
62	[7 12]	[2.89,3.33]	0.6664	1.9495	[1.60,2.33]	0.8062	1.7444	1.03408	2.07298
63	[7 13]	[1.82,814.60]	122.3959	318.616	[3.04,1342.12]	337.8856	435.9709	164.9012	350.2629
64	[8 9]	[0.35,1.69]	0.3173	1.0008	[0.29,2.07]	0.3937	1.0213	0.44144	1.0157
65	[8 10]	[0.44,10.12]	2.8813	3.213	[0.34,4.08]	0.7699	2.6319	1.38878	2.81096
66	[8 11]	[0.47,0.62]	0.2669	0.6105	[0.27,0.83]	0.2449	0.5472	0.27128	0.59624
67	[8 12]	[0.32,2.78]	0.3217	0.9415	[0.46,1.59]	0.4127	1.2395	0.38584	0.92754
68	[8 13]	[0.28,1052.62]	142.3799	321.8786	[0.29,1338.76]	341.2433	431.4585	164.247	354.1935
69	[9 10]	[1.77,5.52]	1.1682	4.5326	[1.39,10.01]	3.2668	3.6781	1.83522	4.1022
70	[9 11]	[2.80,1.08]	0.8422	1.1587	[2.00,1.04]	0.6723	1.5828	0.60736	1.28658
71	[9 12]	[1.88,3.37]	0.6849	1.8754	[1.33,1.63]	0.9714	1.3912	0.81674	1.99746
72	[9 13]	[1.35,621.37]	96.6333	300.5463	[1.52,420.72]	142.7225	444.0683	165.6831	349.8794
73	[10 11]	[5.56,0.97]	0.8206	2.7432	[4.08,1.03]	0.7865	2.8221	1.4026	3.0016
74	[10 12]	[5.66,3.03]	1.6691	5.3217	[2.63,2.89]	1.3616	4.3963	1.8494	4.29308
75	[10 13]	[5.56,813.89]	123.1465	318.9756	[6.20,1342.08]	337.9286	435.9597	165.1429	351.0151
76	[11 12]	[1.02,3.54]	0.4707	0.945	[0.66,1.58]	0.3901	1.2123	0.47664	1.1588
77	[11 13]	[0.99,420.69]	142.6938	444.0808	[0.89,814.58]	122.4253	318.5494	164.9092	350.2012
78	[12 13]	[3.20,1066.16]	128.8379	262.9061	[2.18,621.35]	92.6518	289.5949	164.8896	350.2334

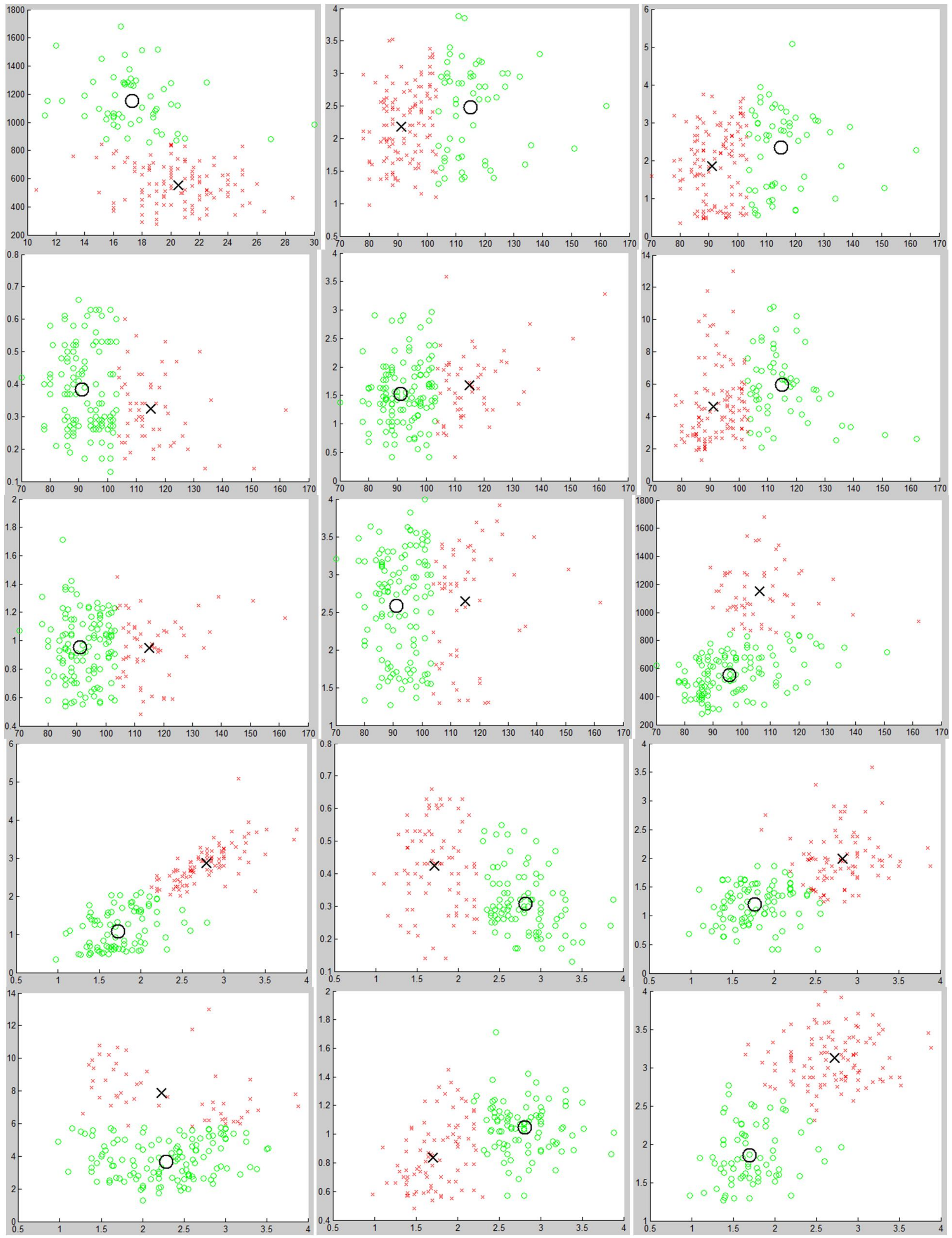
APPENDIX-F:GRAPHICAL REPRESENTATION OF FCM SIMULATION FOR CLUSTER NUMBER 2, 3, 4 AND 5

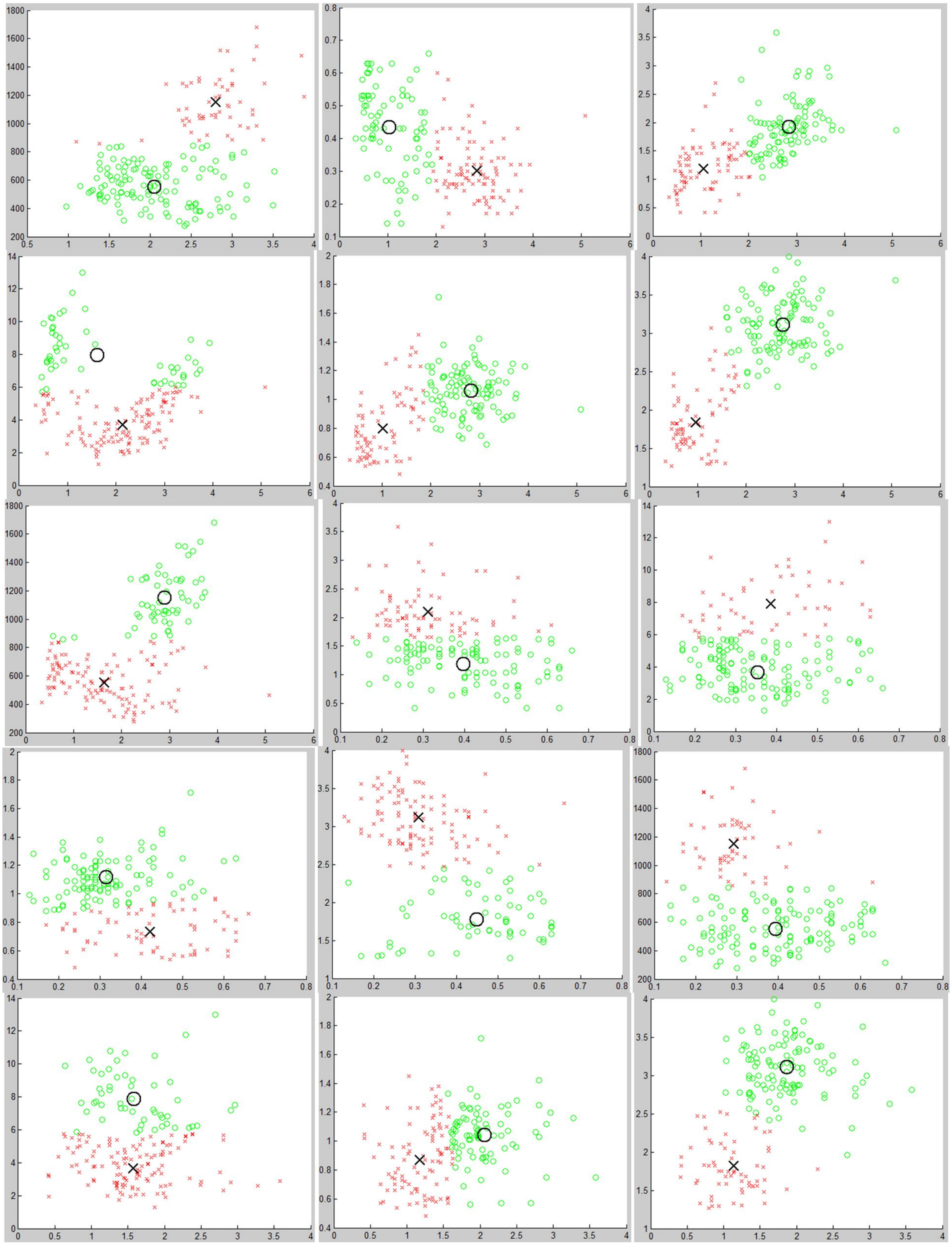
Number of Cluster = 2

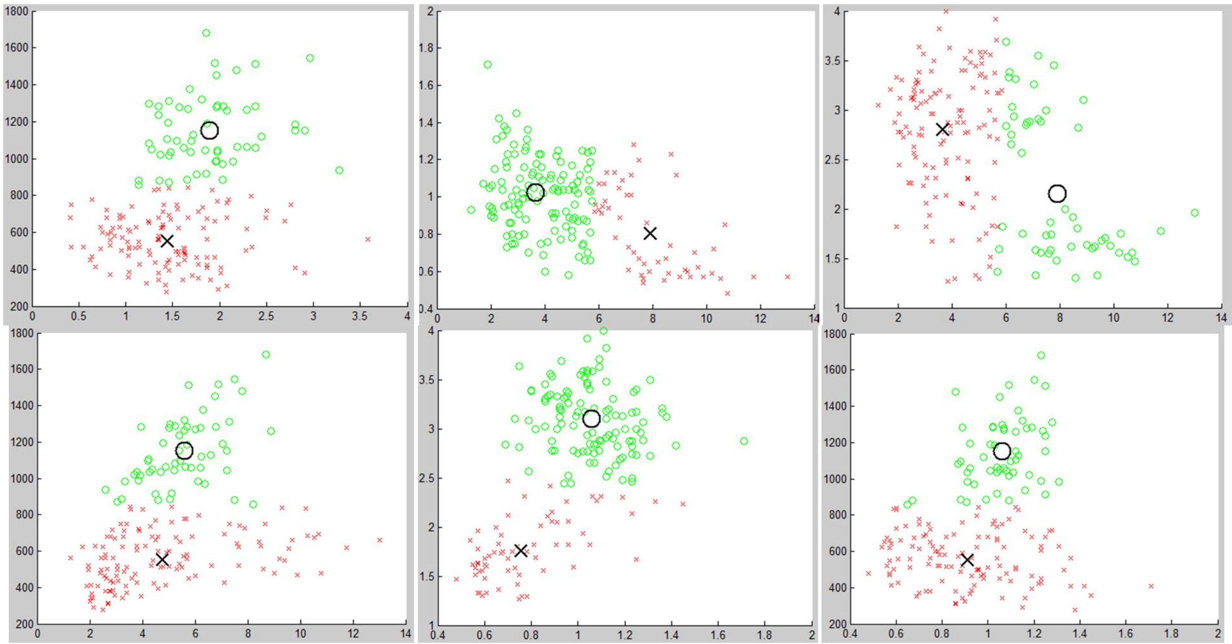




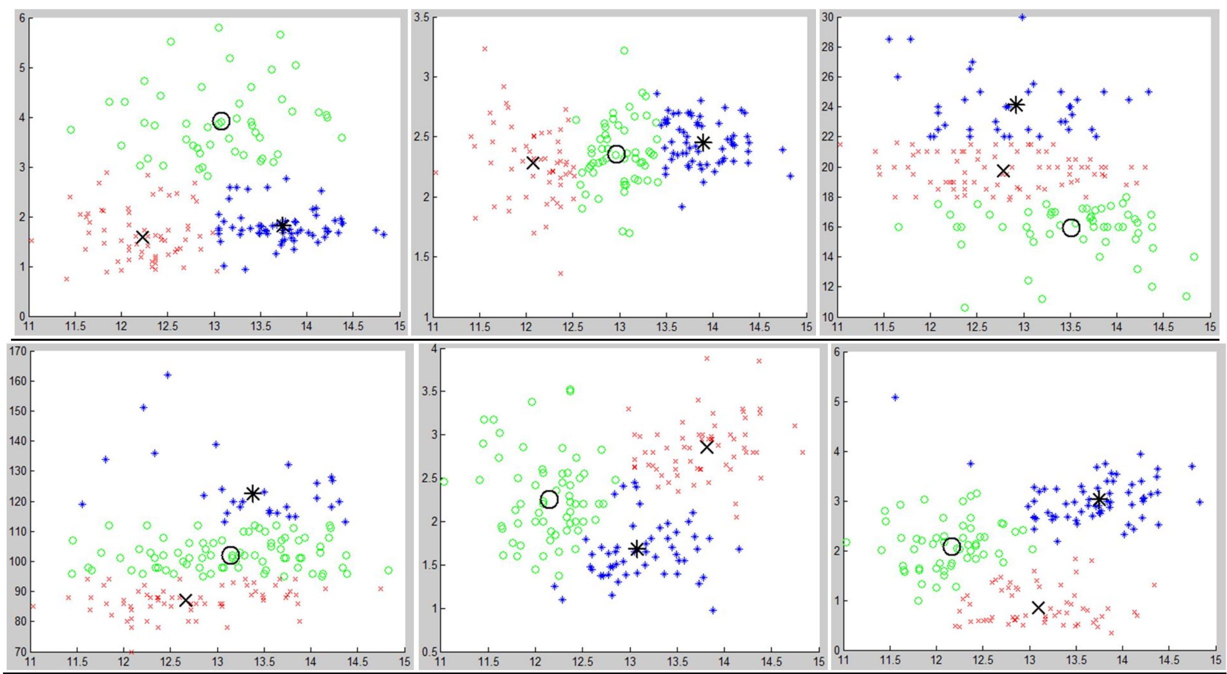


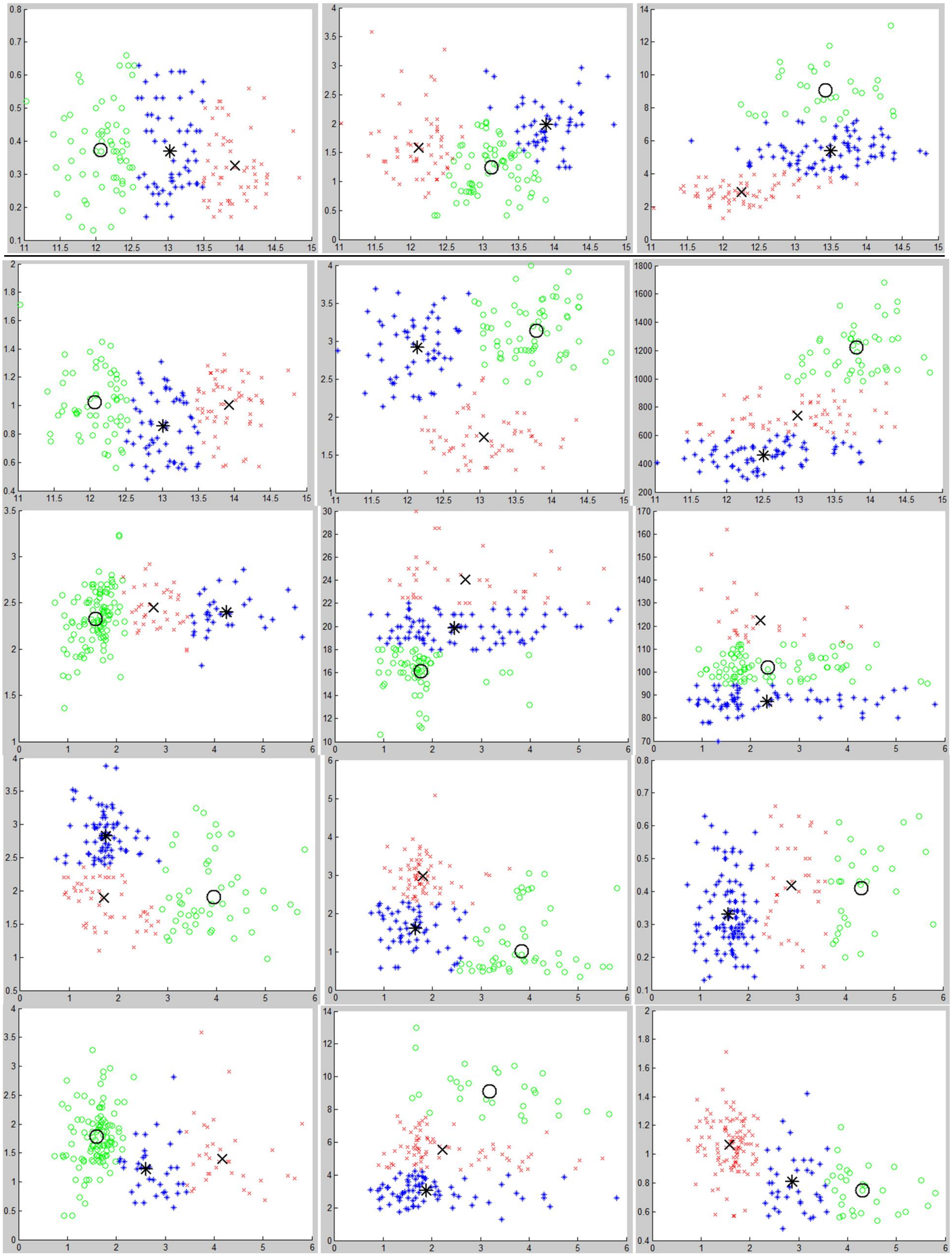


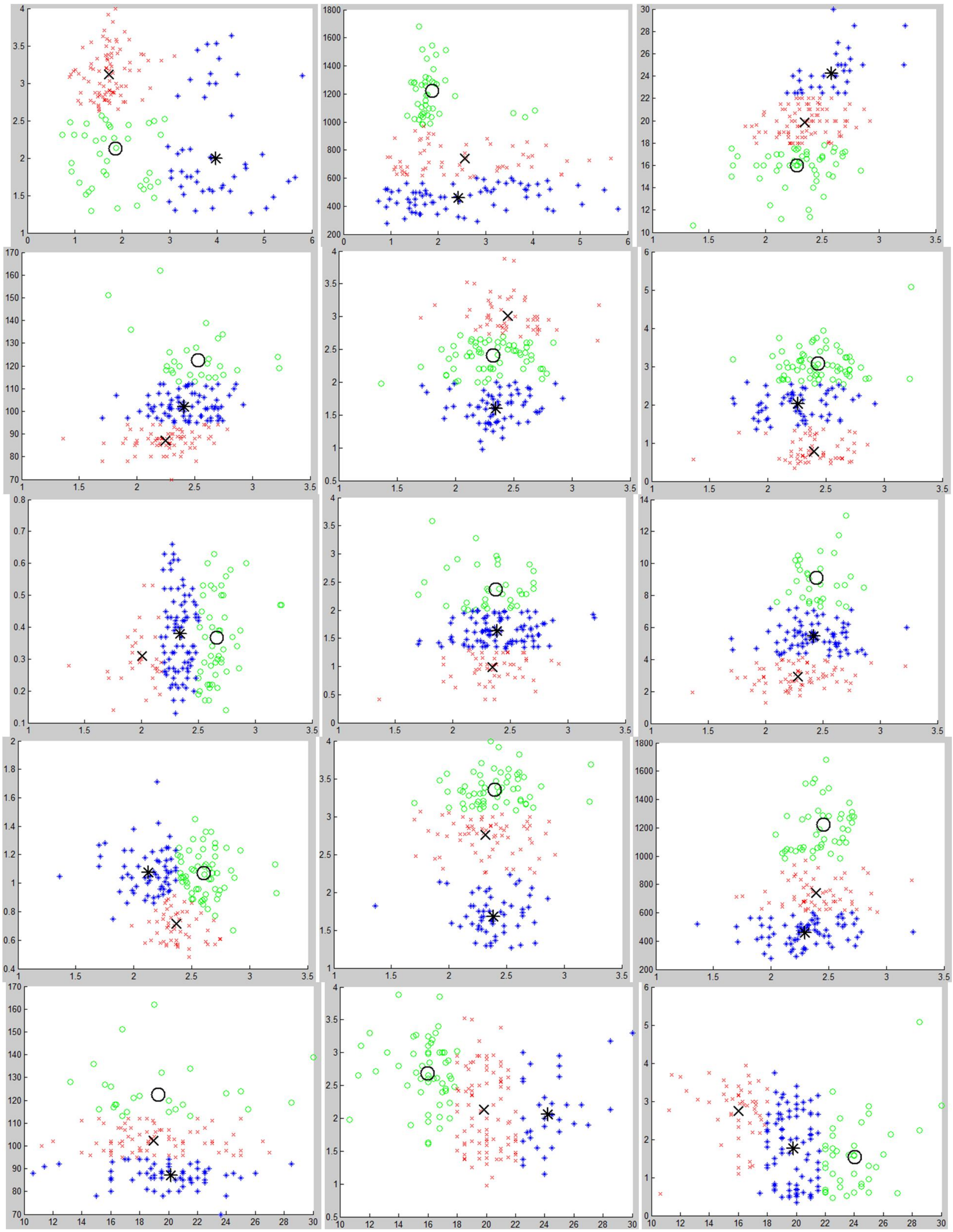


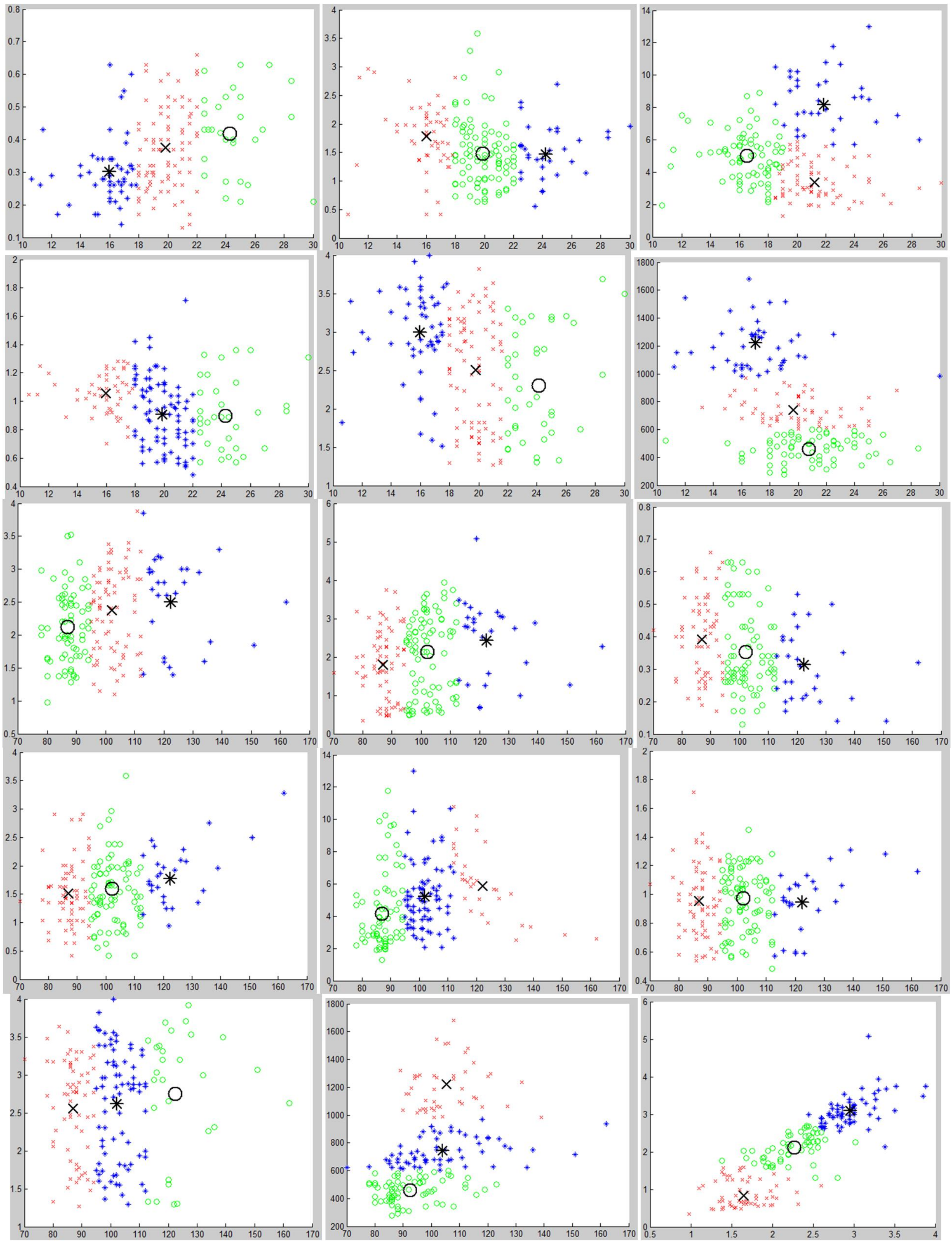


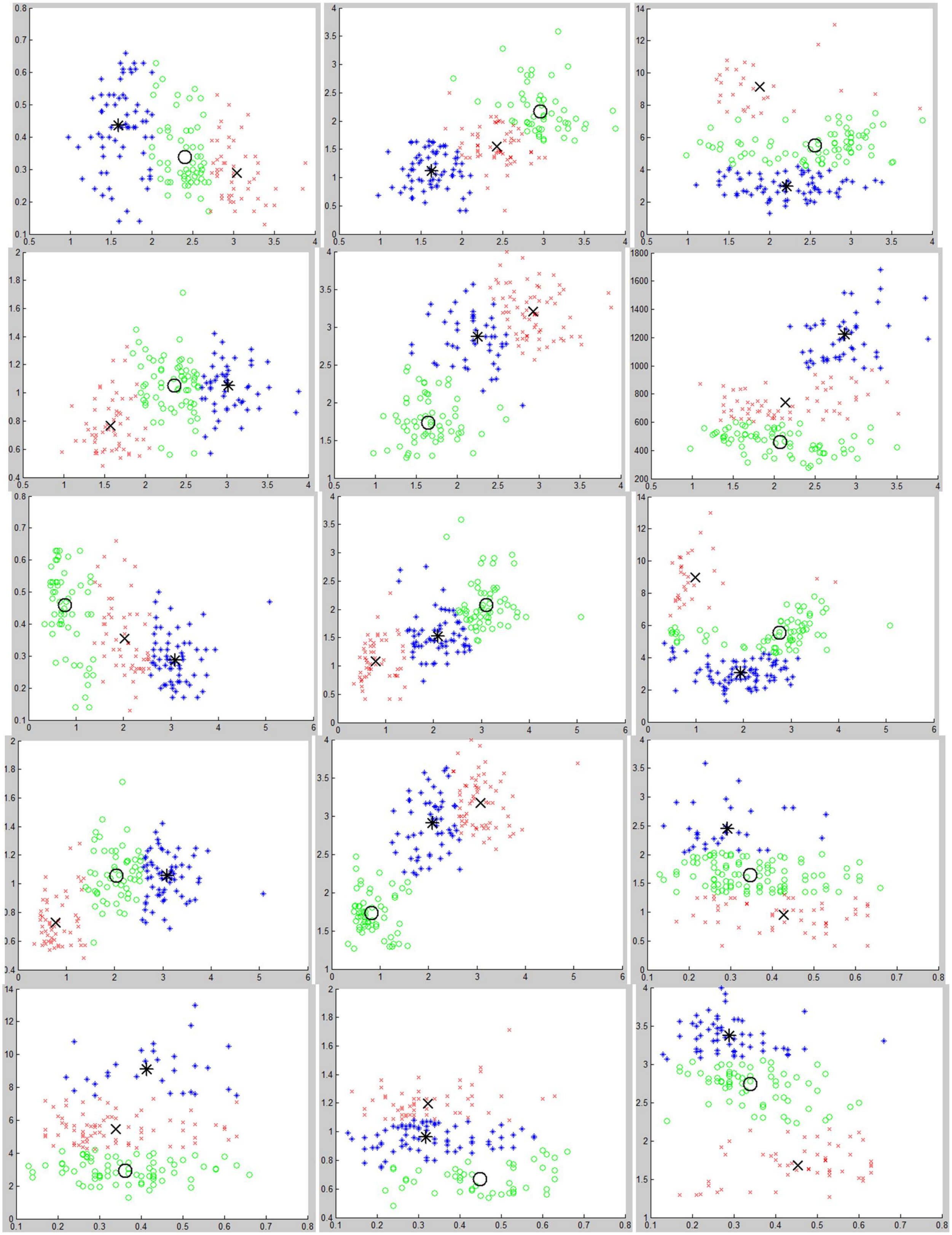
Number of Cluster = 3

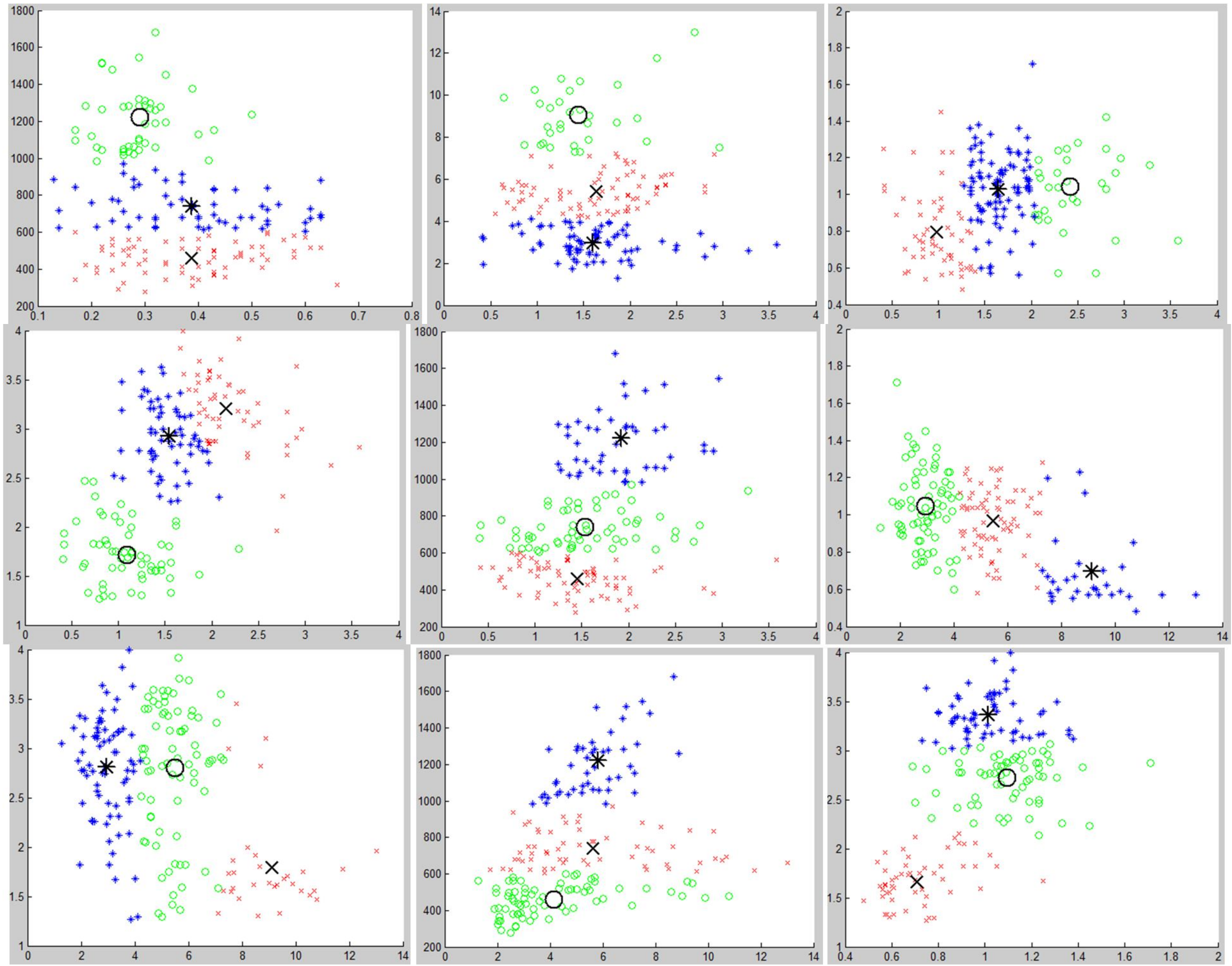




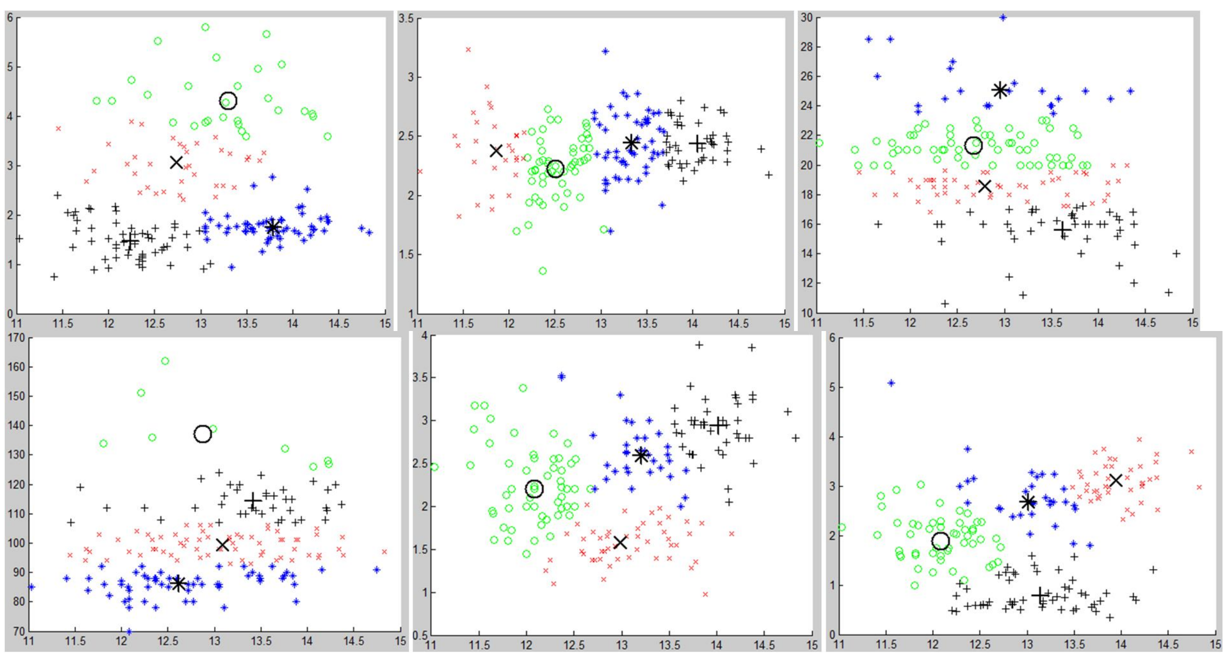


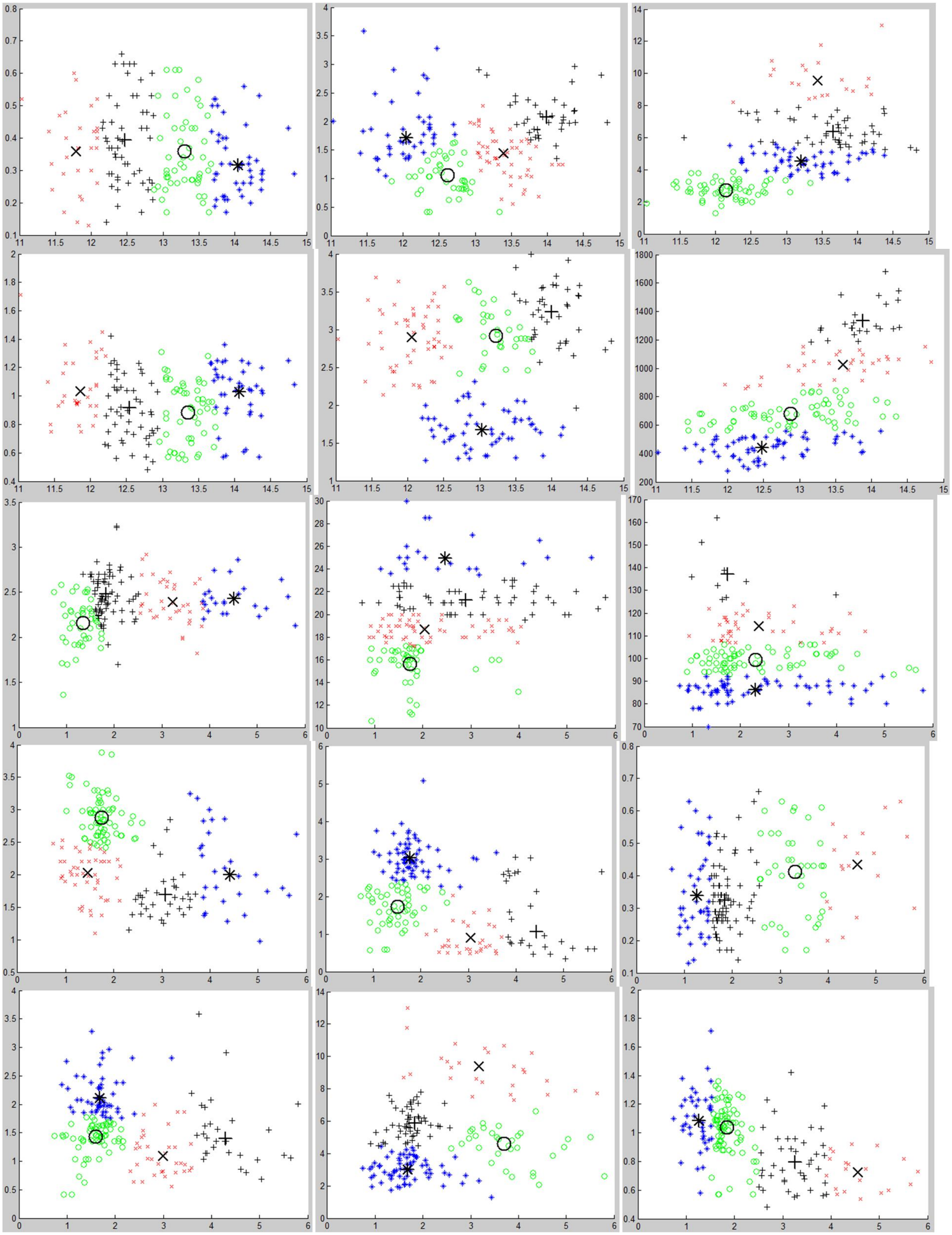


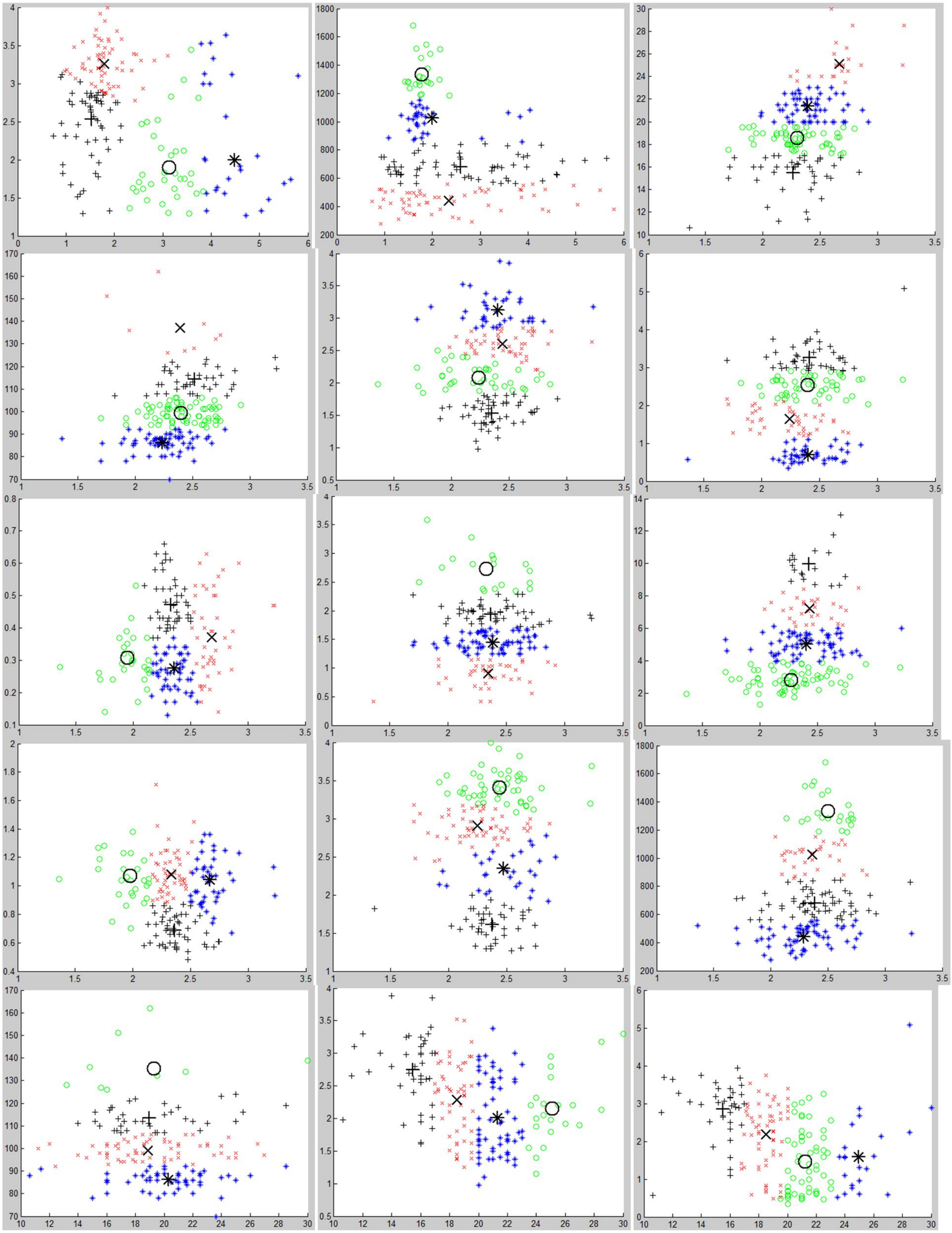


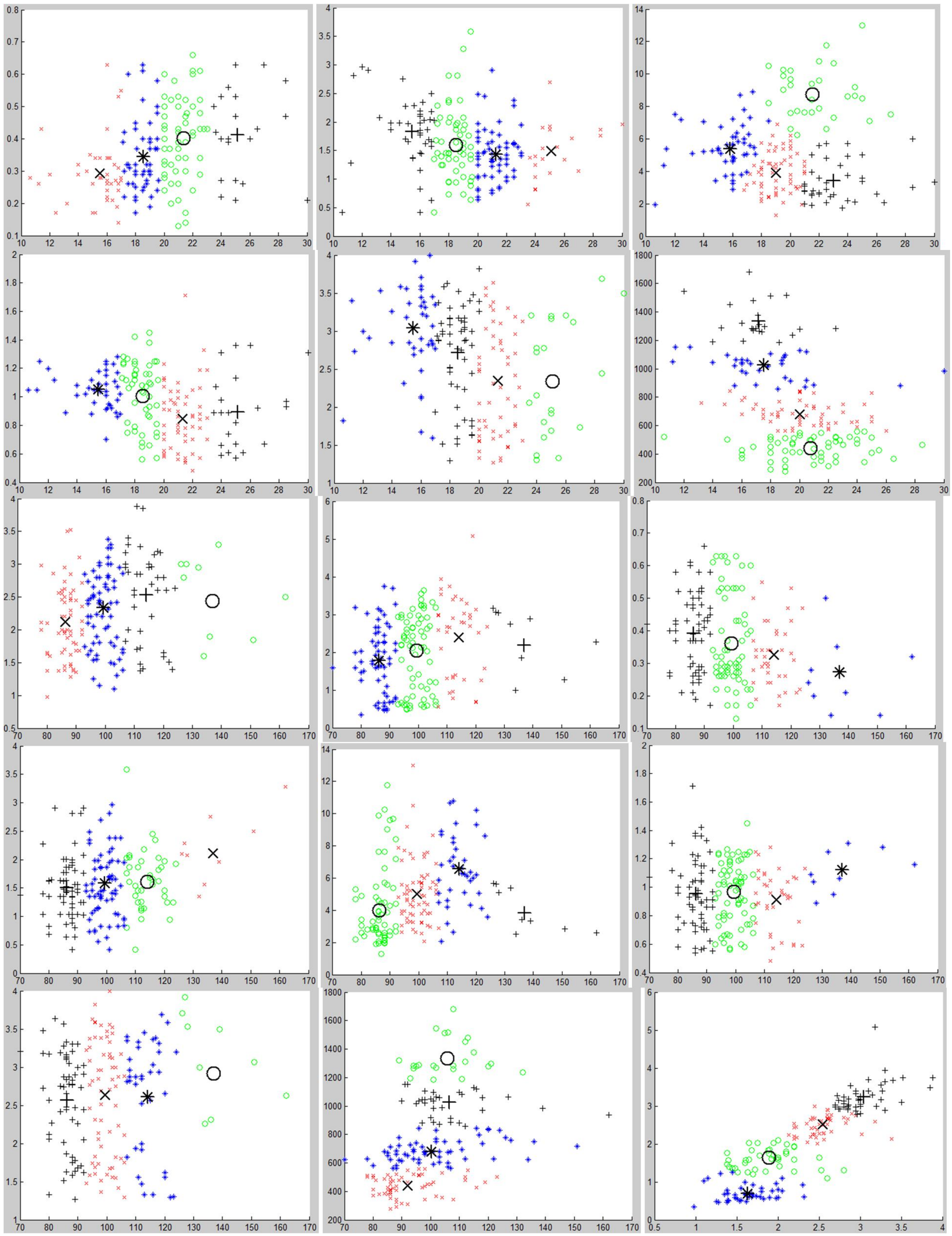


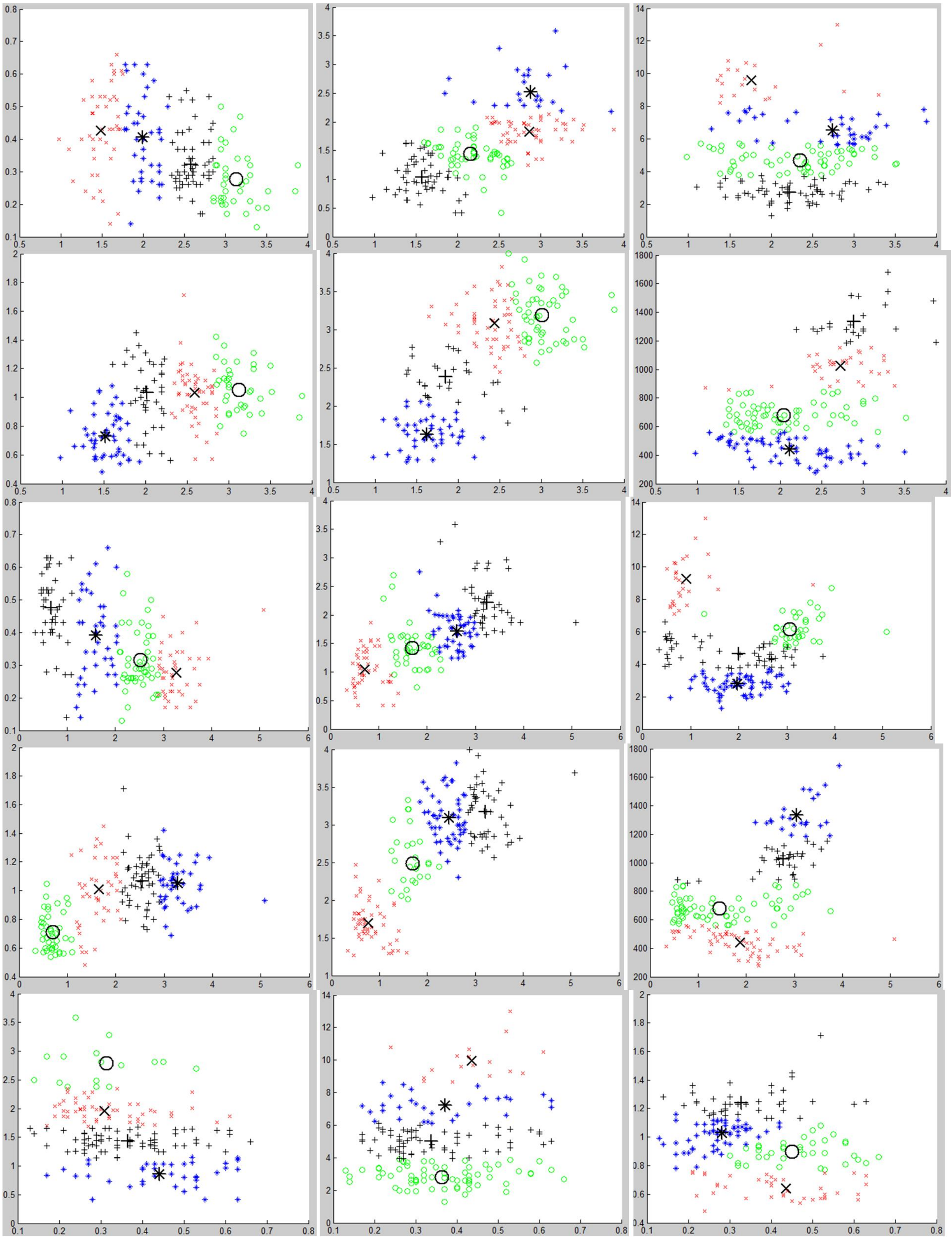
Number of Cluster = 4

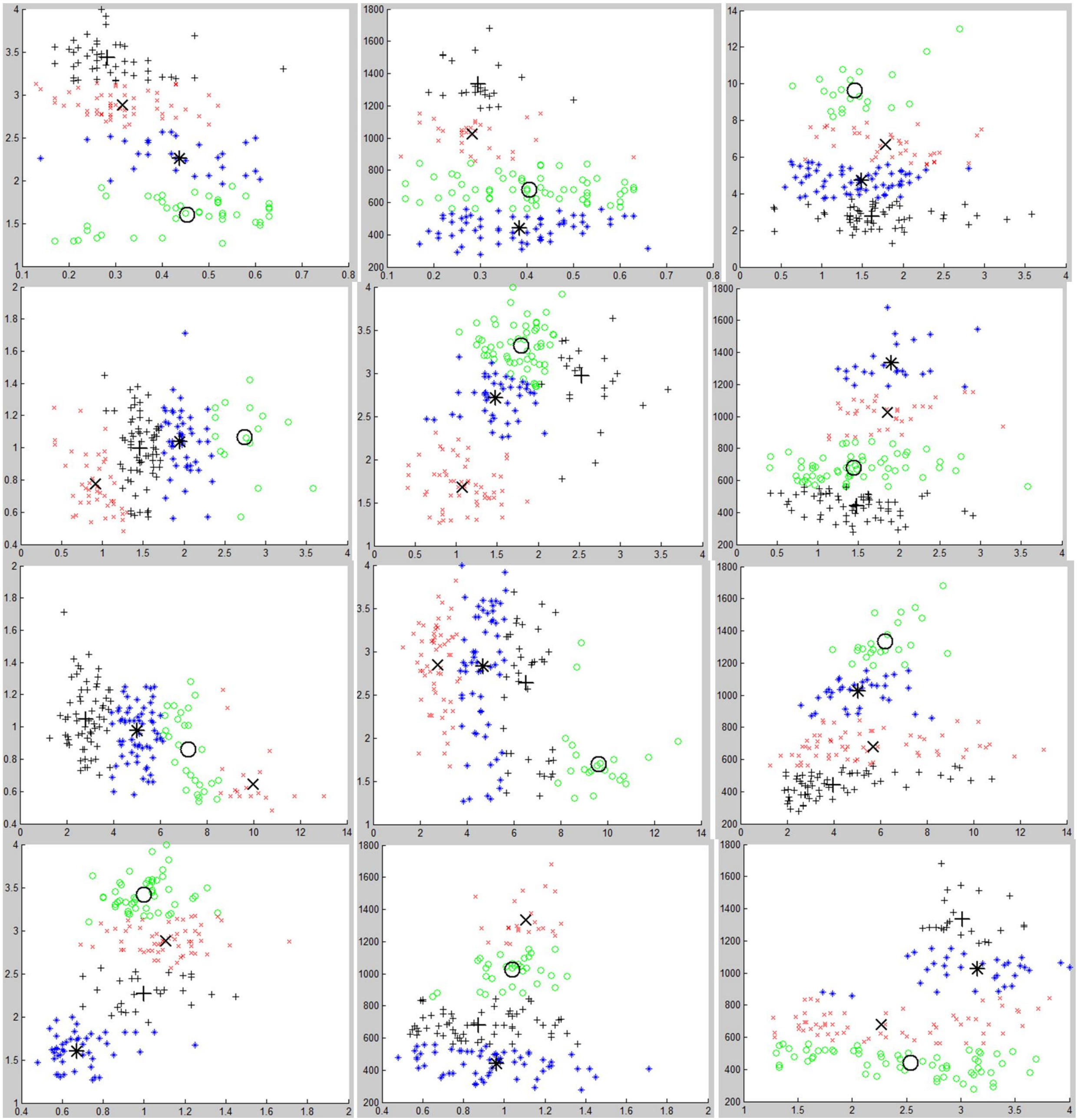




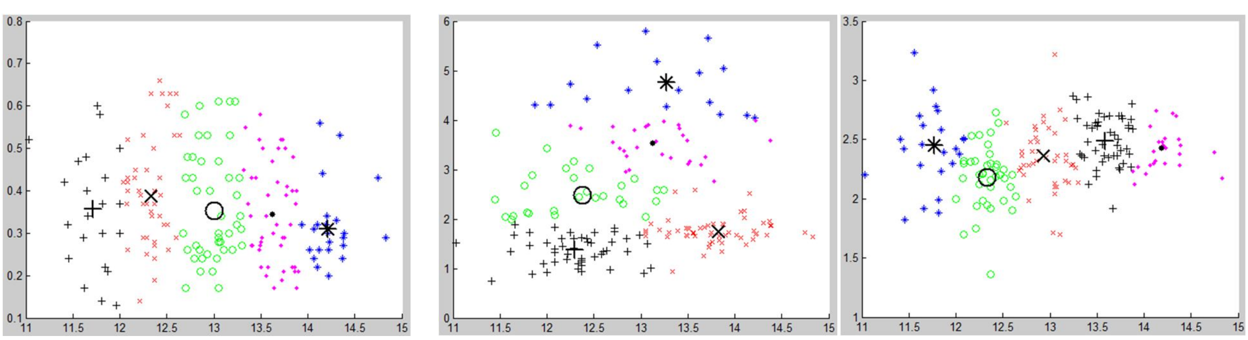


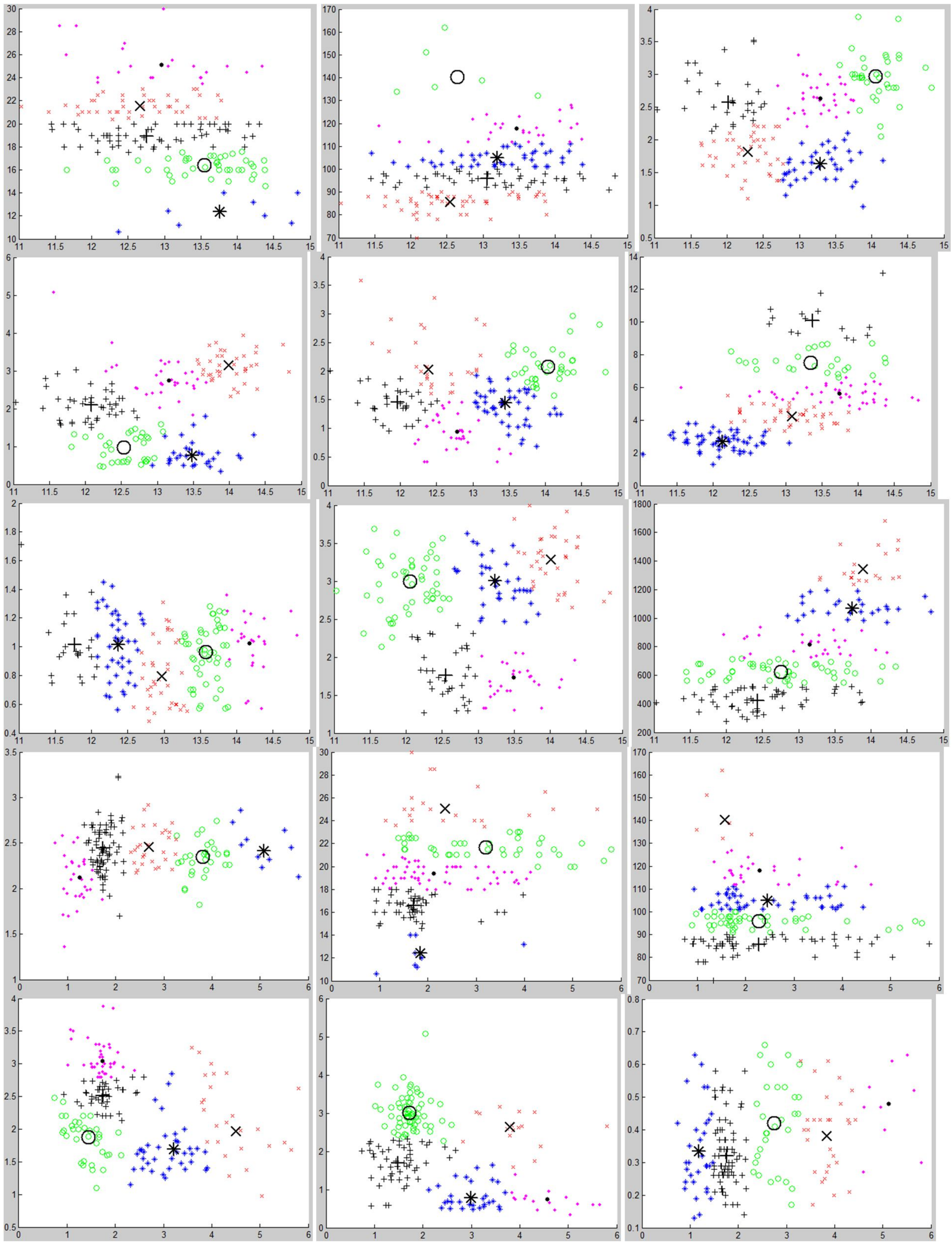


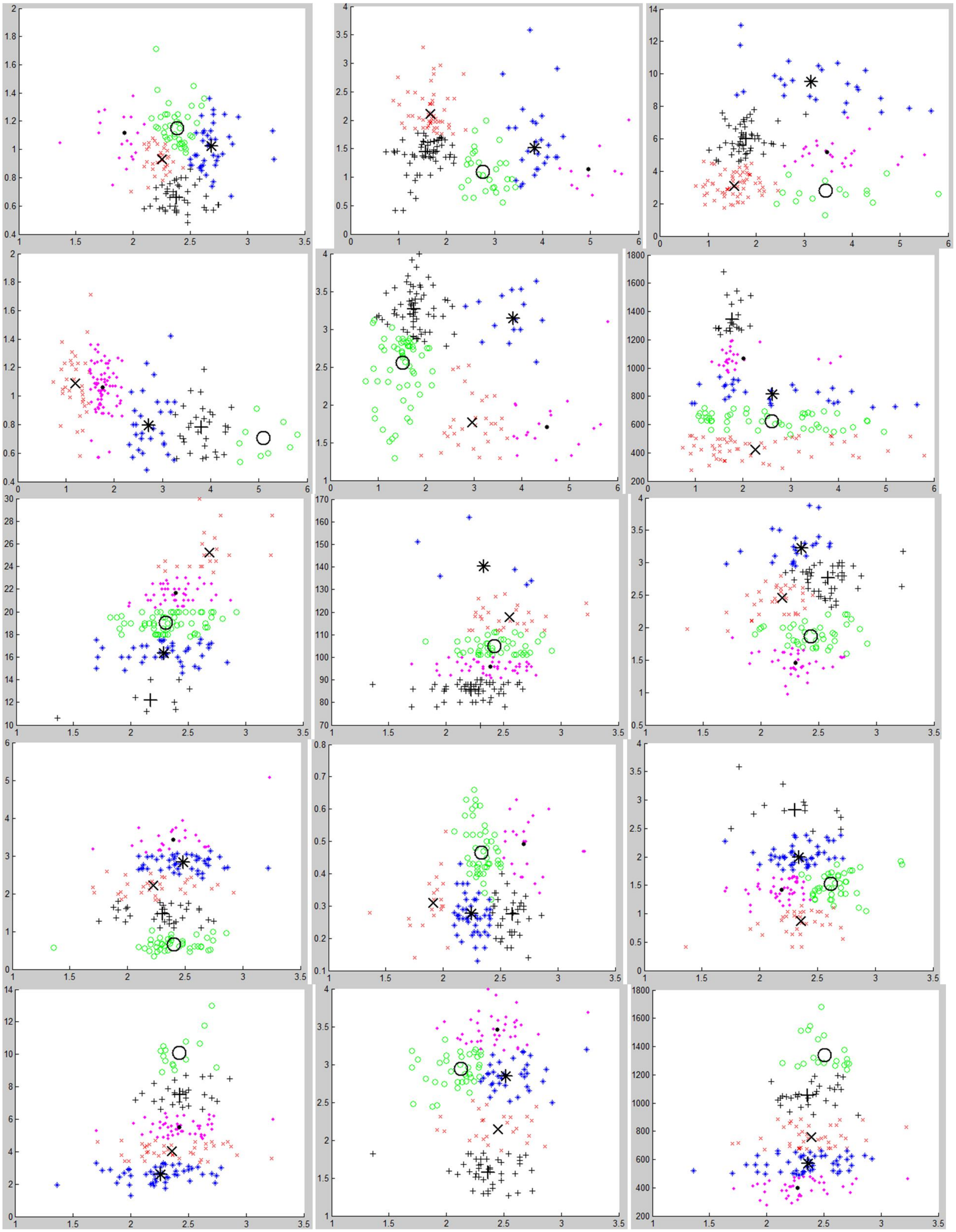


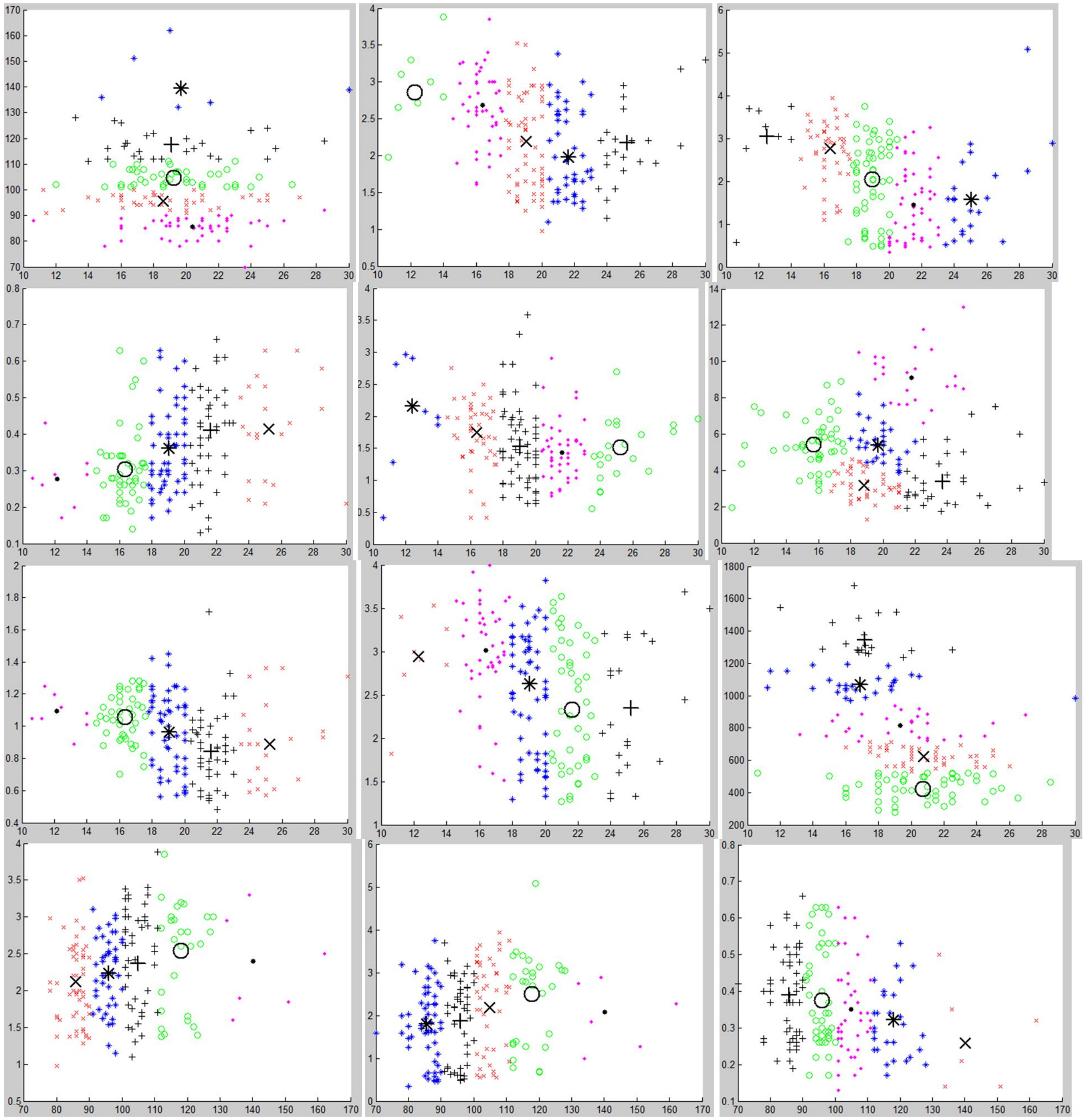


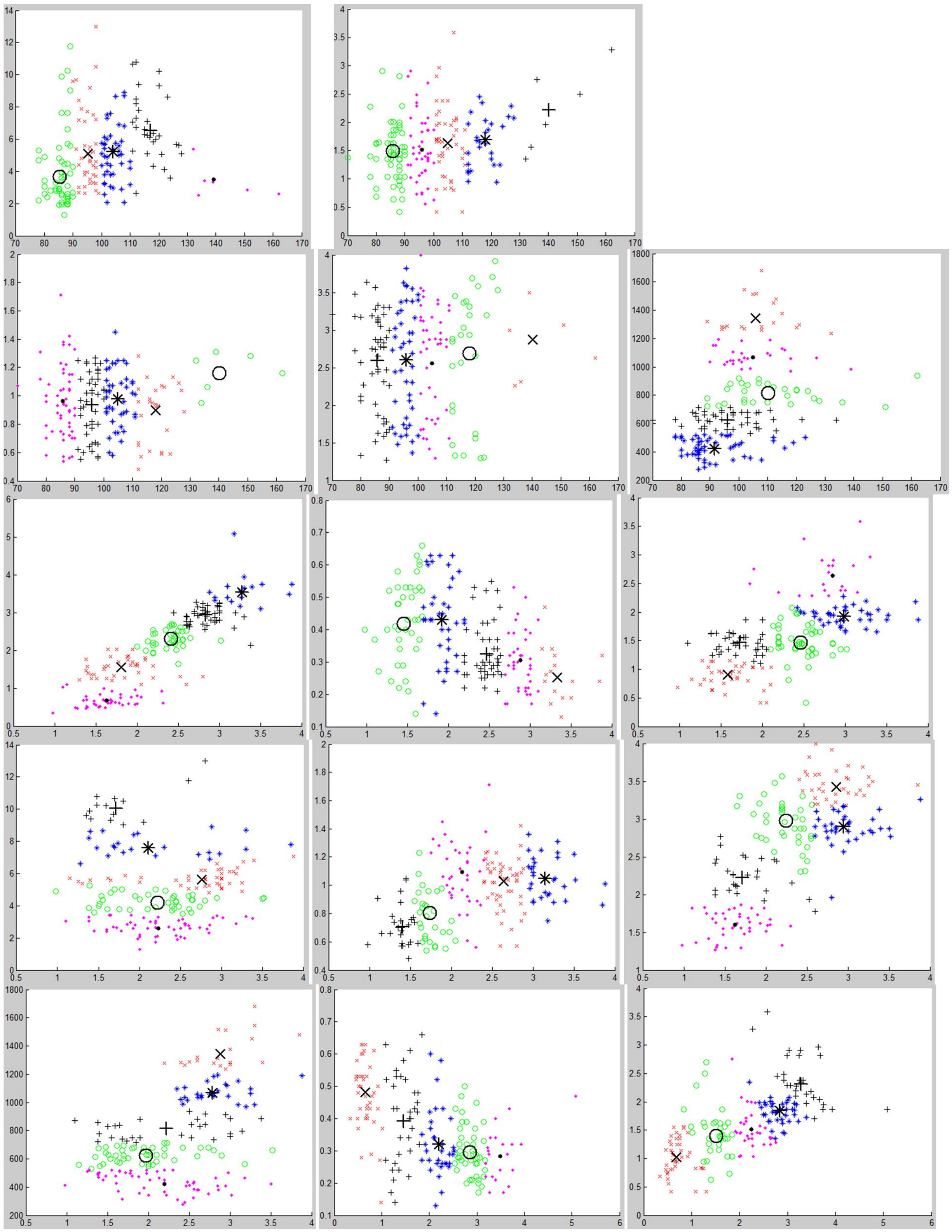
Number of Cluster = 5

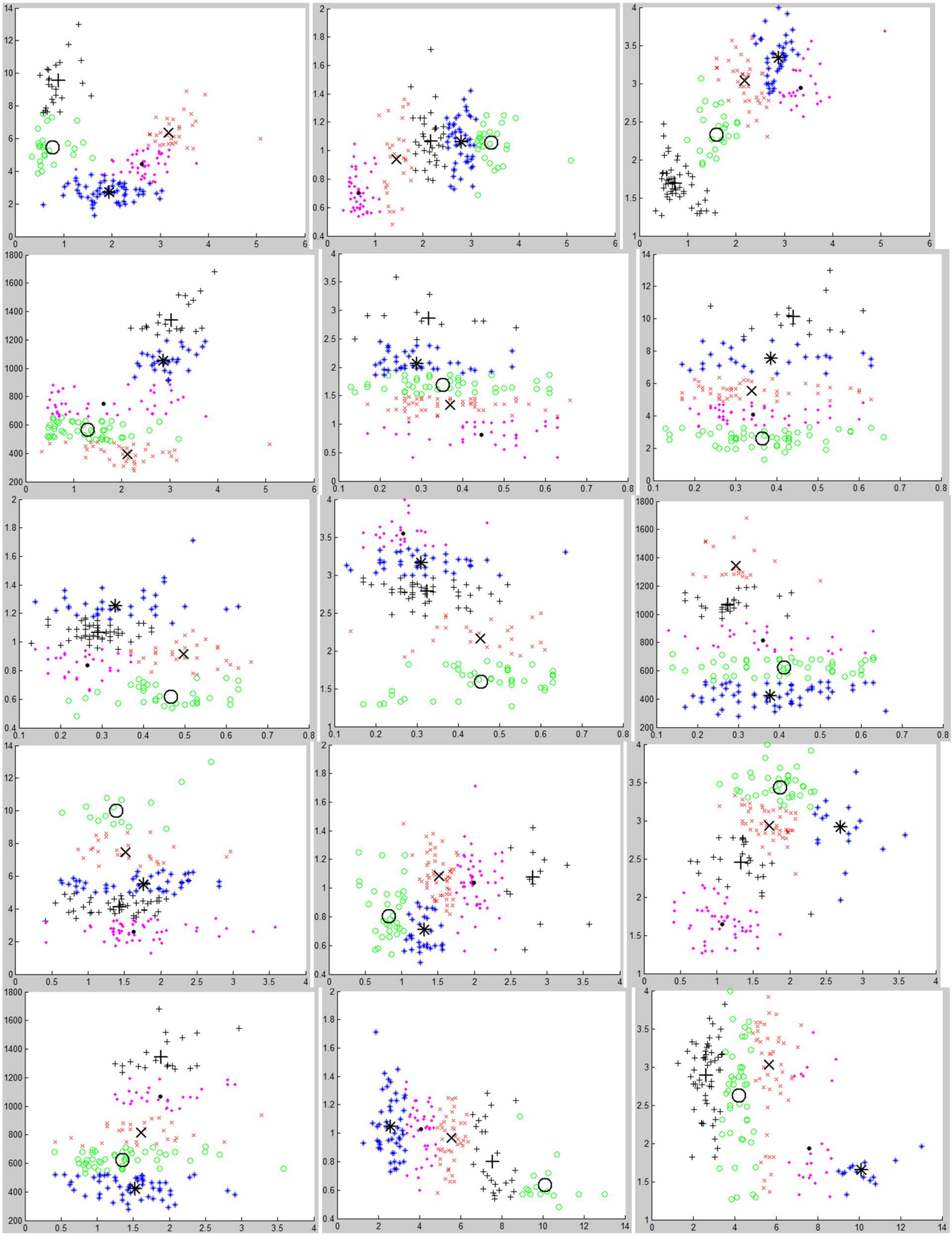


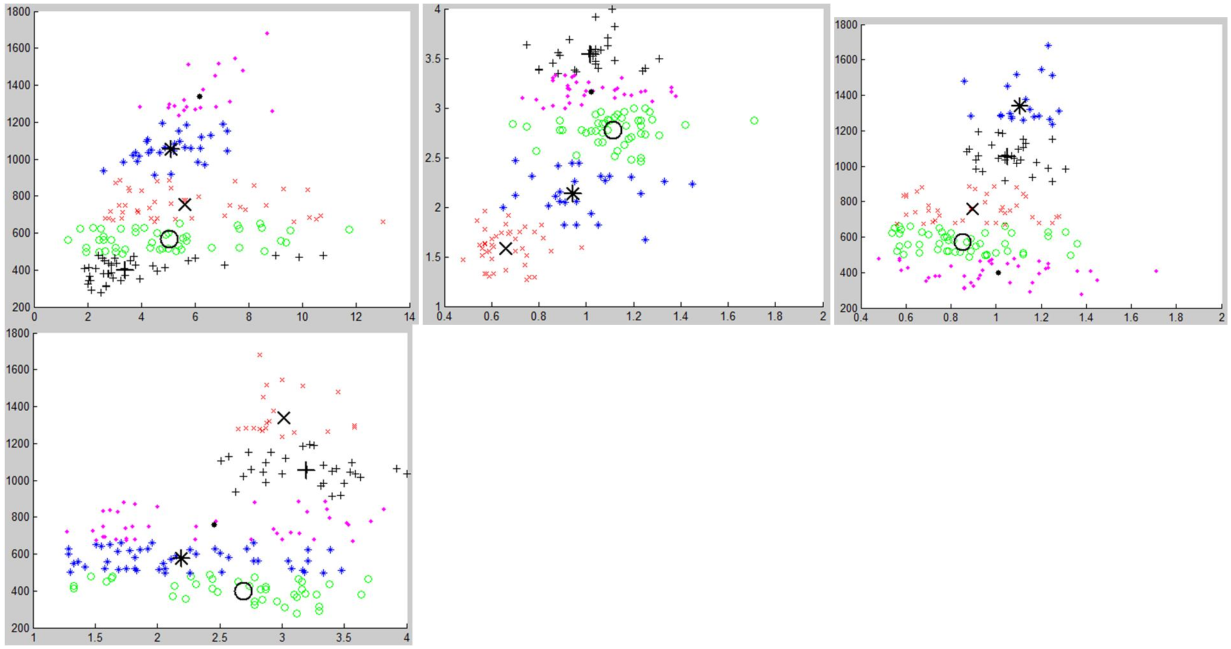












APPENDIX-G: MATLAB CODE FOR FINDING CLUSTER MEMBERS

```
function [result]=FindClusterMembers( data,nOfc )
    [center,U,objFcn] = fcm(data,nOfc);
maxU = max(U);
    index1 = find(U(1, :) == maxU)
    index2 = find(U(2, :) == maxU)
ifnOfc>=3
        index3 = find(U(3, :) == maxU)
end
ifnOfc>=4
        index4 = find(U(4, :) == maxU);
end
ifnOfc>=5
        index5 = find(U(5, :) == maxU);
end
fileID=fopen('cluster1.txt','a');
fprintf(fileID,'%i\n',index1);
fclose(fileID);
```

```
fileID=fopen('cluster2.txt','a');
fprintf(fileID,'%i\n',index2);
fclose(fileID);
ifnOfc>=3
fileID=fopen('cluster3.txt','a');
fprintf(fileID,'%i\n',index3);
fclose(fileID);
end
ifnOfc>=4
fileID=fopen('cluster4.txt','a');
fprintf(fileID,'%i\n',index4);
fclose(fileID);
end
ifnOfc>=5
fileID=fopen('cluster5.txt','a');
fprintf(fileID,'%i\n',index5);
fclose(fileID);
end
end
```

REFERENCES

[1] Fuzzy C-Means Clustering, http://www.mathworks.com/help/fuzzy/fuzzy-clustering.html#FP43419

[2] Fuzzy C-Means Clustering for Iris Data, http://www.mathworks.com/help/fuzzy/examples/fuzzy-c-means-clustering-for-iris-data.html

[3] Fuzzy C-Means (FCM), http://www.bindichen.co.uk/post/AI/fuzzy-c-means.html