*Nitheesha Singireddy*

*Suraj Kumar Reddy*

**Final Project Report**

**Clustering by passing messages between data points**

**Introduction**:

This paper discuss about clustering. Clustering was done based on the affinity between the data points to cluster the data. Objects within the cluster will have the Euclidean distance very low or closely related to each other whereas distance between the data points in separate clusters are high or not closely related.

In this paper we uses the word Exemplar, it is similar to the data points but additionally it acts like a role model to the data points usually we select center as an exemplar. After choosing exemplar, we need to be very careful as we need to iteratively refine it until it works fine better. Choosing bad exemplar will fiasco the output. Real valued messages are exchanged between the data points till a perfect exemplars and clusters are created.

**Initialization**:

To cluster the data points, Affinity propagation uses Euclidian metric −||xi − xj||.

AP is an extension to the k-Medions, in k- Medions initially it chooses k exemplars. Where as in AP all the data points were initially considered as the potential exemplars.

Consider every data point in the cluster as a nodes in the graph, such that messages are passed between the nodes.

Initially put this information aside,

1. Similarity between the data points from (i-j)

It means how good the data point j is suitable to the exemplar for the point i

1. Preference to be the data point as an exemplars (i-i).

It defines the total number of clusters.

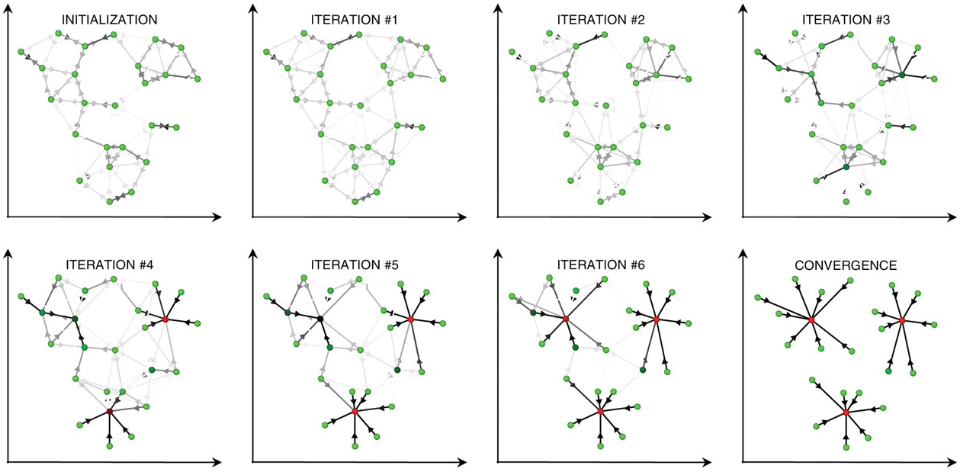
**Identification of clusters and exemplars:**

AP will transmits the real valued messages between the arrows (represented in below diagram) to the nodes in the graph till the good exemplars and clusters found.

1. Responsibility

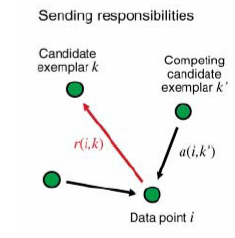
2. Availability

**Working of Affinity Propagation:**



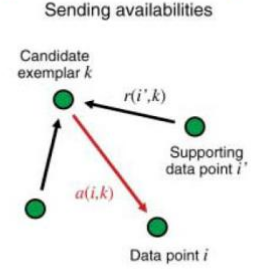
Let’s consider for the two dimensional data, in order to make the data points into a cluster, we used Euclidean distance to measure the similarity. The nodes with red are treated as exemplars and nodes with green are data points and darkness of arrow ranges from high to low where the tendency refers to the strength of the transmitted message between the red and green node respectively.

**Responsibility**:



R(I,k) is the responsibilities which has sent from the data point to the candidate exemplar it shows how good the data points favors the candidate exemplars when compared to other competing candidate exemplar.

**Availability**:



Point k will take the support from other neighboring points that point k should be an exemplar. It provide enough evidence of choosing k as exemplar.

**Algorithm**:

Availabilities are initialized to 0, a(i,k) =0, - initially every data point have no idea which exemplar it belongs to.

**Updating responsibilities:**



For 1st Iteration:

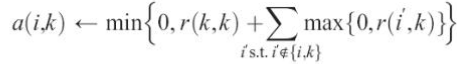


If r is larger, then node k is well suited to node I when compared to other exemplars k’

**Checking for Self-responsibility:** r(k,k) = s(k,k) – max {s(k,k)}

If r(k,k)<0 then exemplar would be belong to other exemplar.

**Updating Availabilities:**



If a(i,k) = 0 then node point k acts like exemplar and more suited to point i.

in updating responsibilities a(i,k’), if a(i,k’) <0 then responsibilities of the other points will increase. From data point i to exemplar k increases.

**Checking for Self-responsibility:**



**Cluster Identification:**

For the data point i, find



For k= i ; the data point i is an exemplar itself.

Else, the data point k is an exemplar of data point i.

**Iterate** the three steps till you get good exemplars and clusters.

1. Updating all responsibilities given the availabilities.
2. Updating all availabilities given the responsibilities.
3. Combining responsibilities and availabilities to monitor the exemplar decisions.

**Data Sets Used for Experiments:**

1. Randomly Generated 2-D Similarity Matrix

(Data1.csv, Data2.csv , Data3.csv)

1. Cartesian coordinates of 22 towns in West Germany Dataset

(<https://people.sc.fsu.edu/~jburkardt/datasets/spaeth2/spaeth2_03.txt>)

1. Synthetic 2-d data with N=5000 vectors and k=15 Gaussian clusters with different degree of cluster overlapping

(<https://cs.joensuu.fi/sipu/datasets/s4.txt>)

1. Unbalance Synthetic 2-D data from "Set-matching methods for external cluster validity", IEEE Trans. on Knowledge and Data Engineering, 28 (8), 2173-2186, August 2016

(<https://cs.joensuu.fi/sipu/datasets/unbalance.txt>)

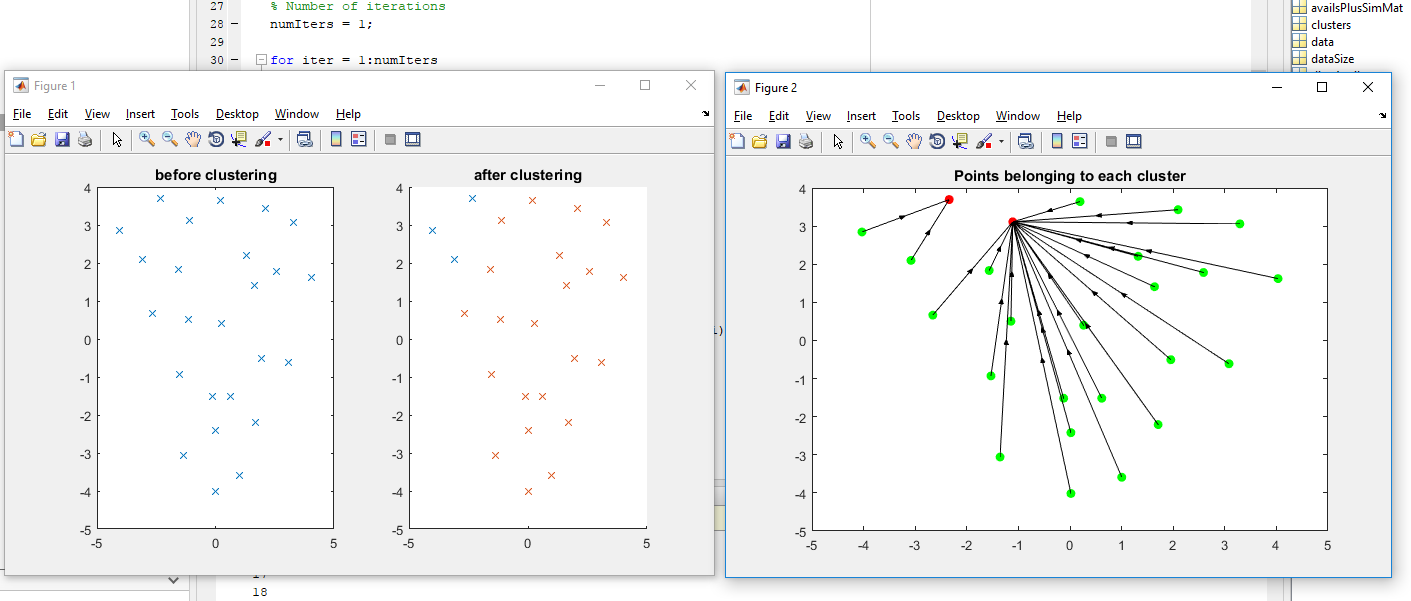
**Nodes with green color is a cluster point and nodes with red are exemplars**

**Experiments with Randomly Generated Data:**

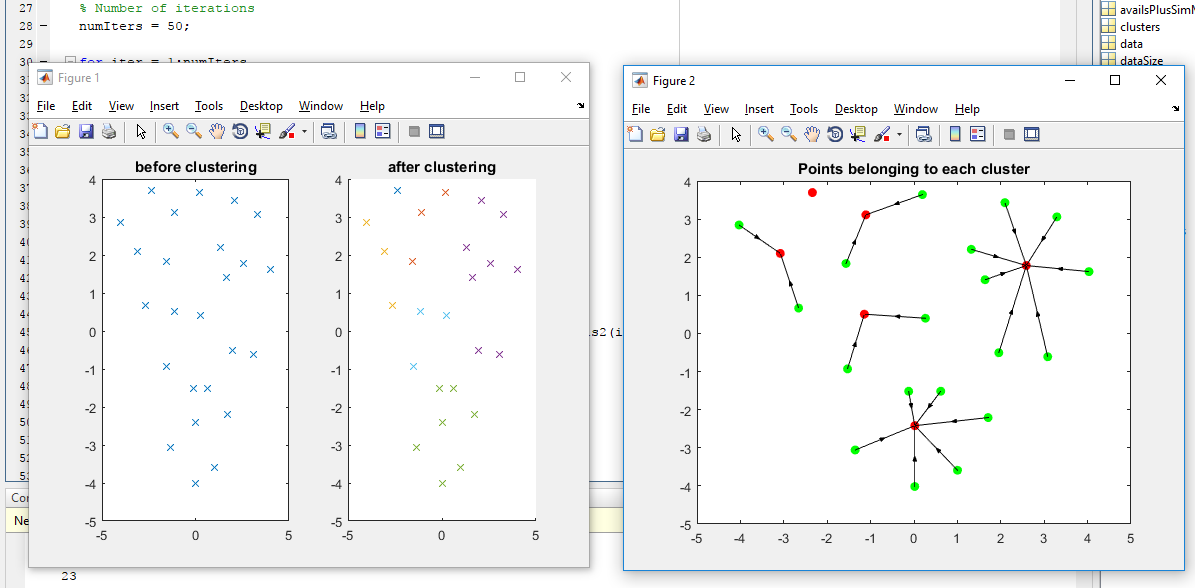
**Data1.csv**

This dataset consists of randomly generated two dimensional data with 25 rows forming 6 clusters.

For Iterations =1 - Only 2 clusters are formed



For iterations > 50 – it started forming constant 6 clusters

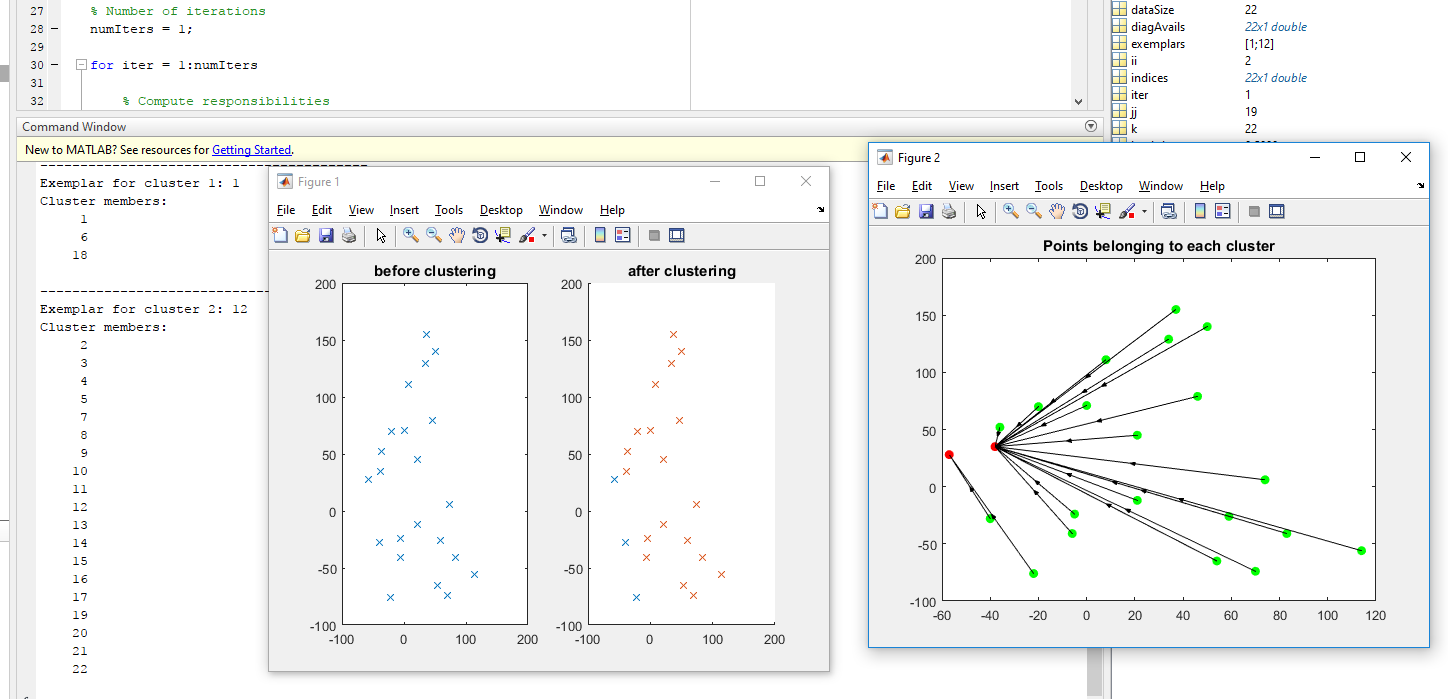


**Experiments with Cartesian coordinates of 22 towns in West Germany Dataset**

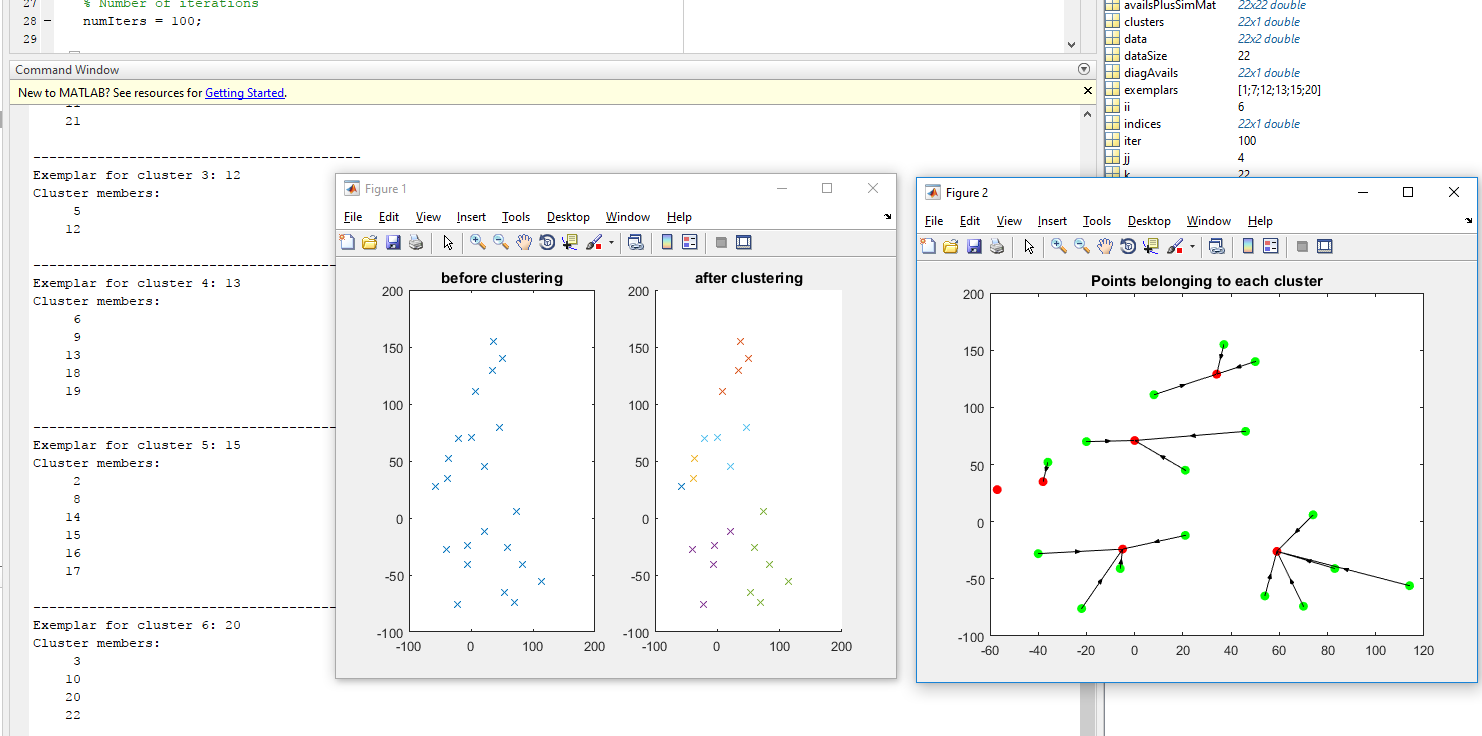
Coordinates.txt

This Real dataset consists of two coordinates of 22 towns in some part of Germany.

For Iterations =1 - Only 2 clusters are formed



For Iterations > 50 it starts forming 6 clusters.

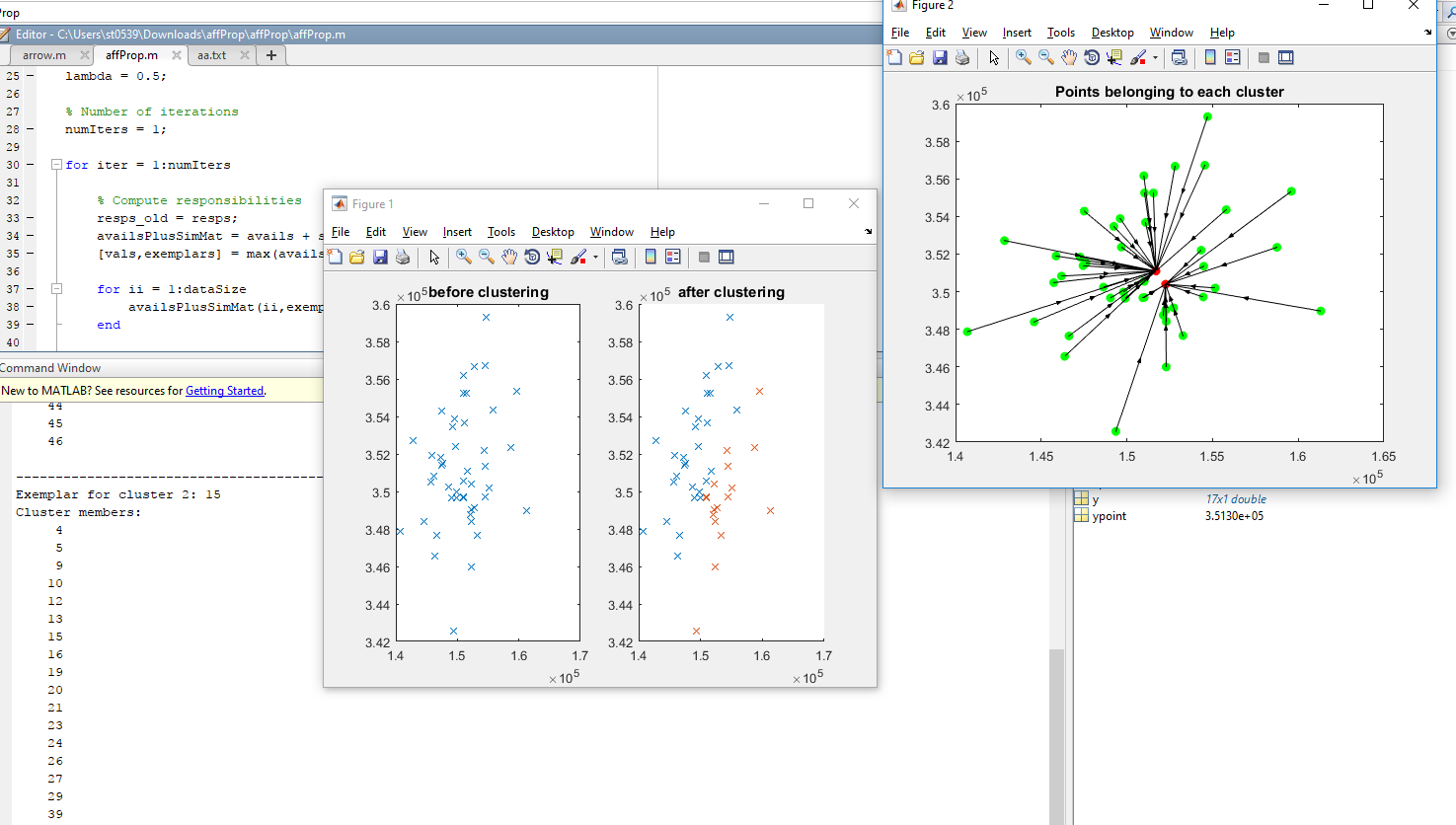


**Experiments with Unbalance Synthetic 2-D data**

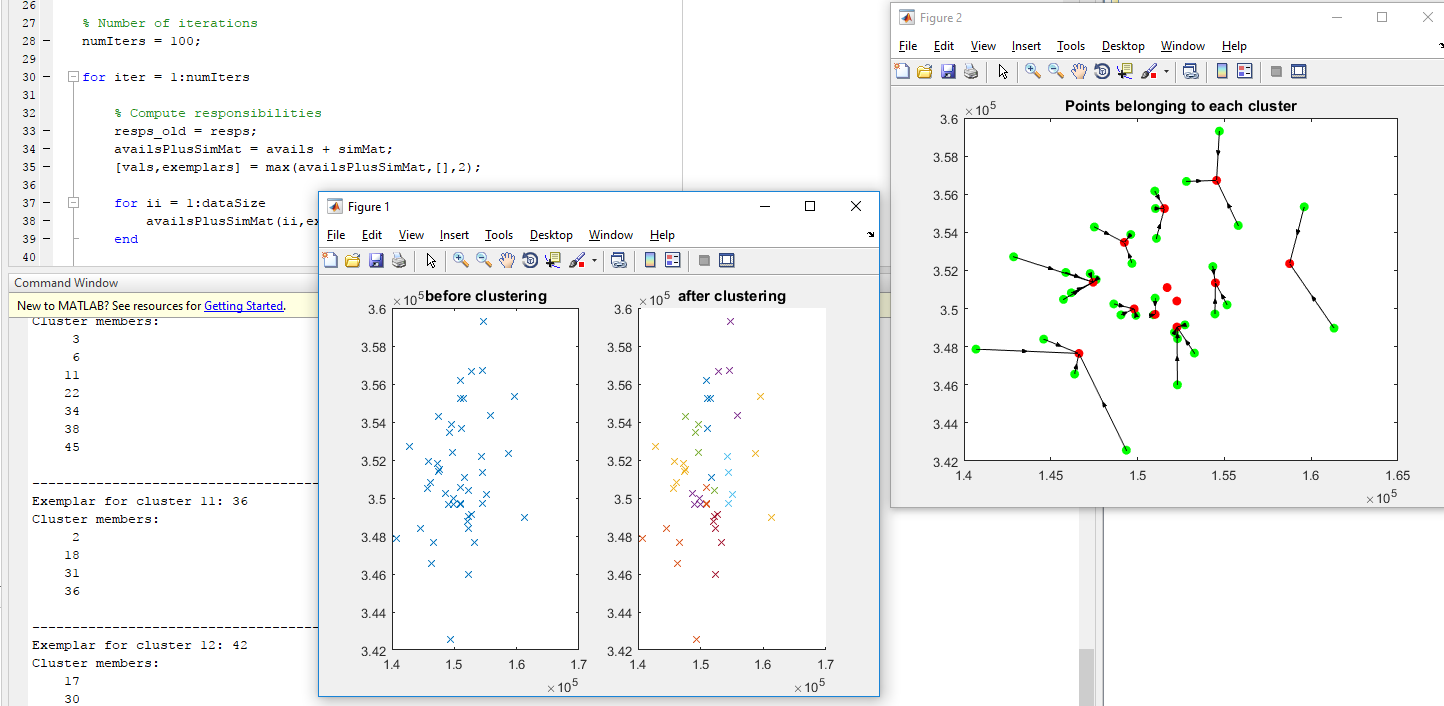
Unbalance.txt

This dataset consists of rows of 46 in 2-D

For iterations 1:



For iterations = 100



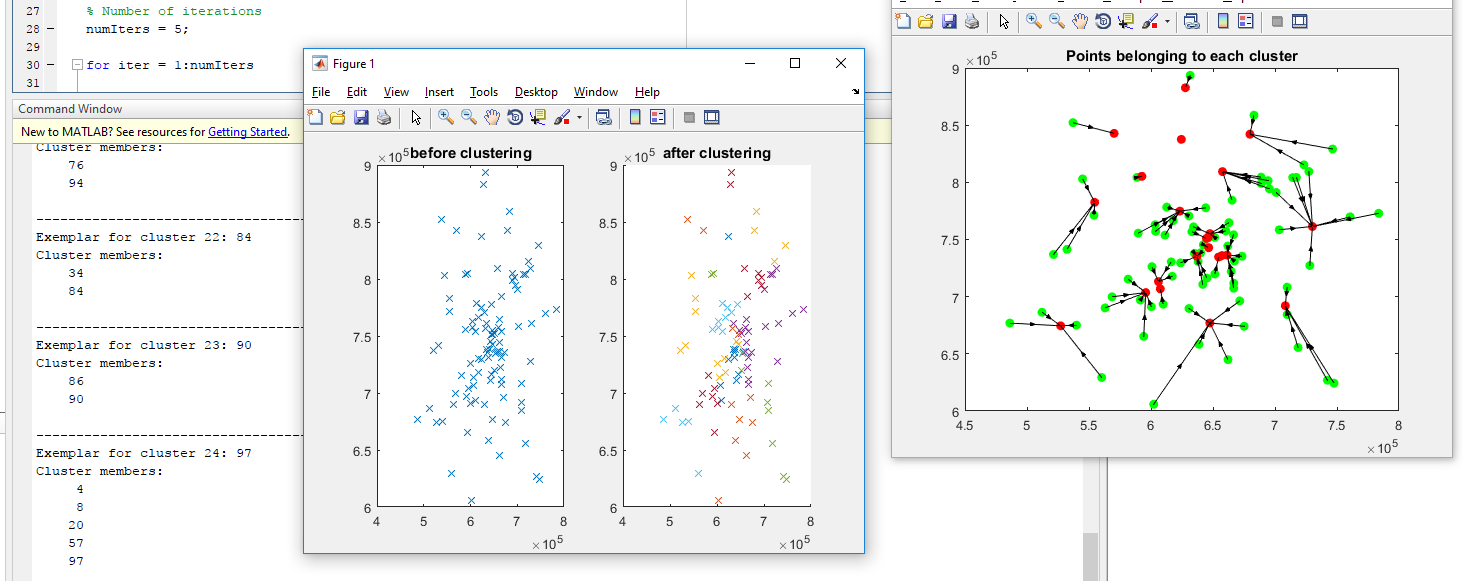
**Experiments with Gaussian clustering**

**(S-sets.txt)**

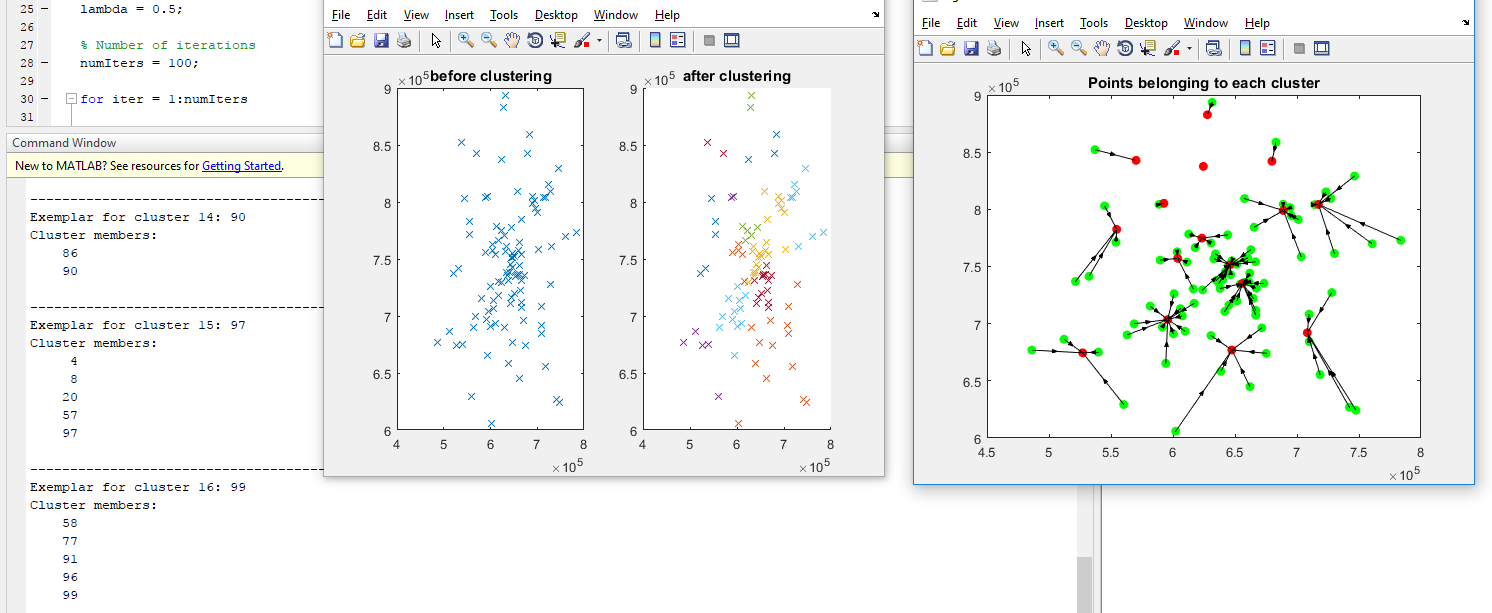
This dataset consists of 2-d data with N=5000 vectors

First we took the data of N = 100 Vectors

For iteration- 5 we got total 24 clusters



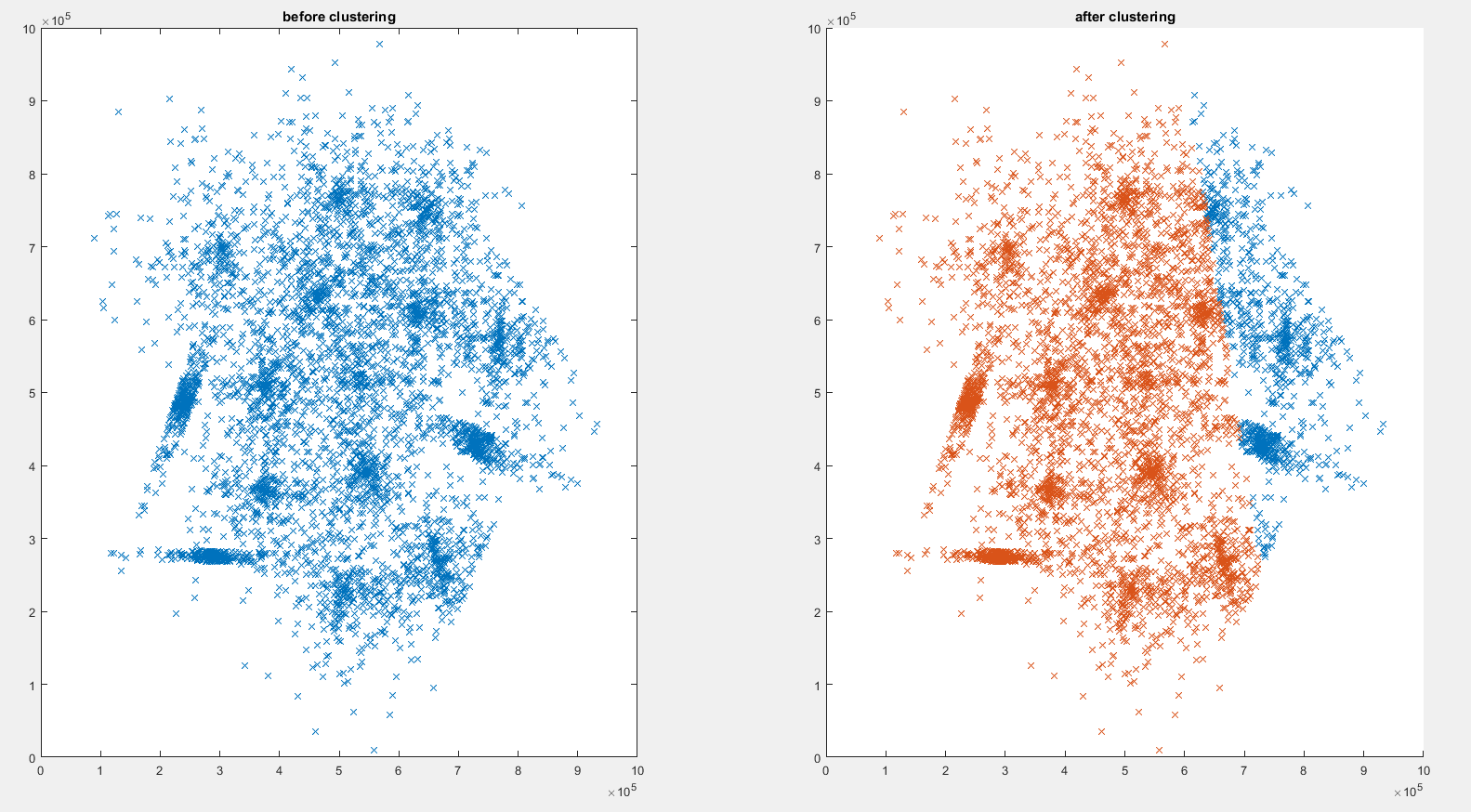
For iterations – 100 we got 16 clusters



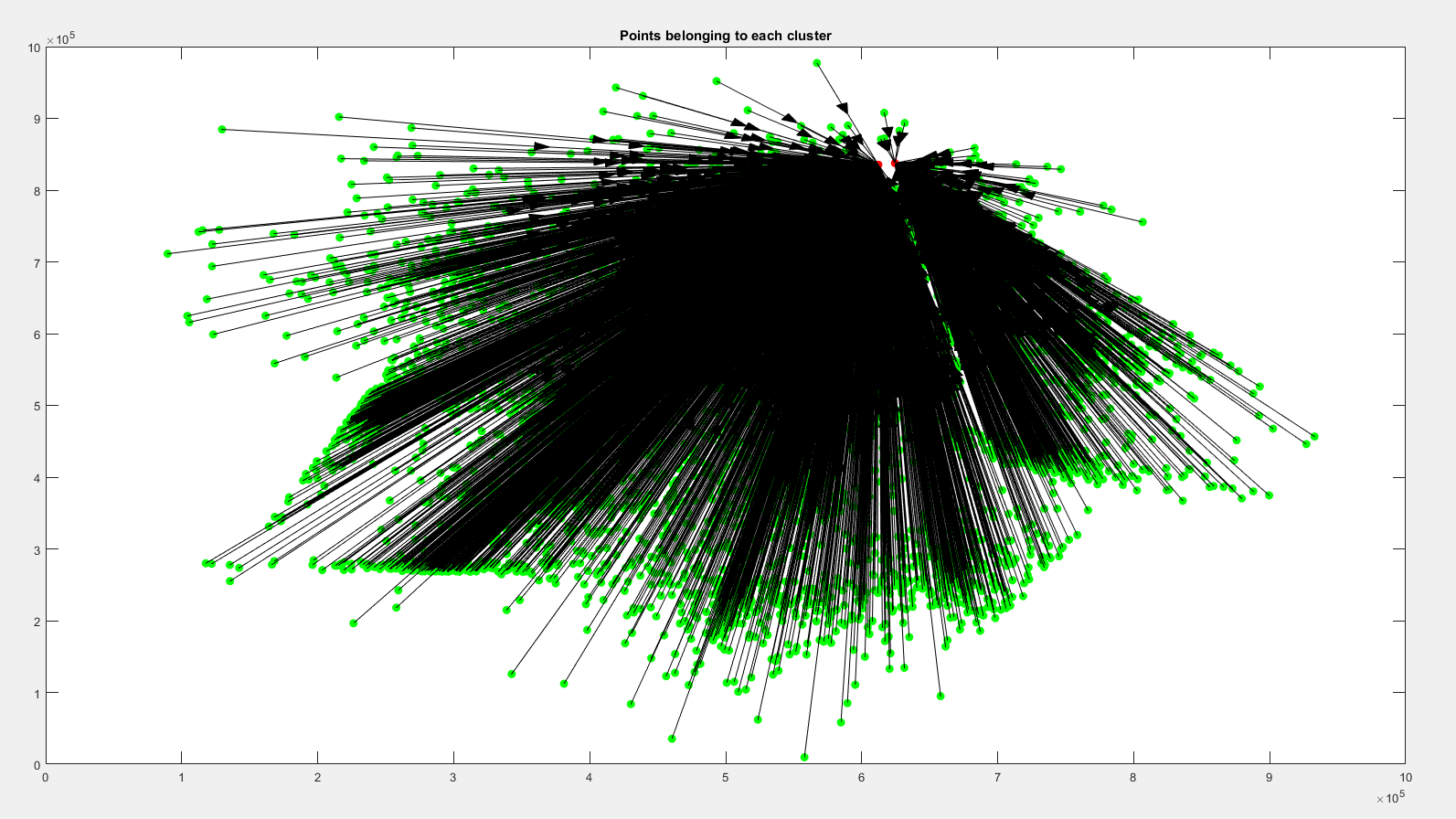
Now, consider the entire dataset of N= 5000 vectors.

(S-sets.txt)

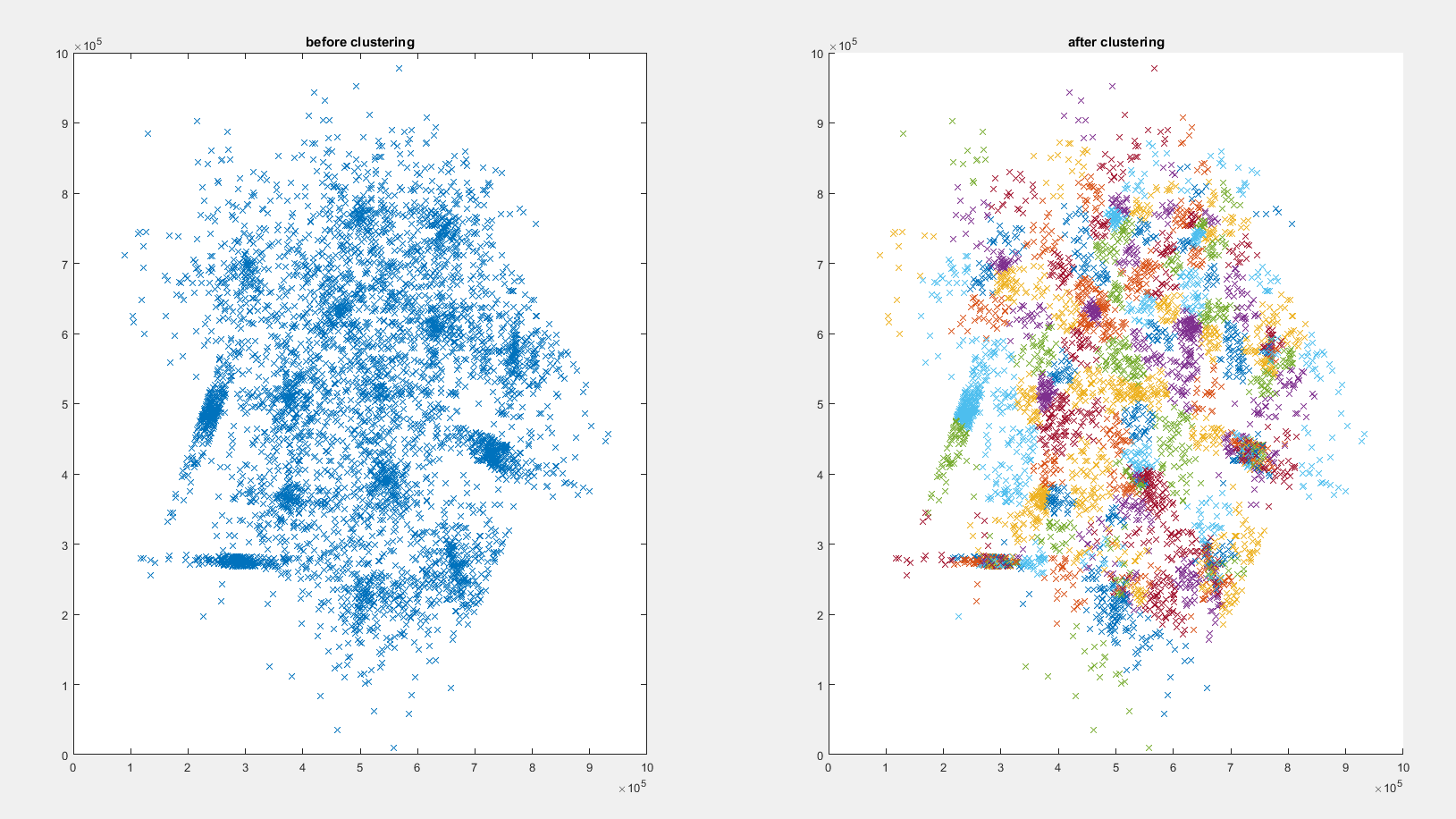
For iteration =1 – two clusters are formed.

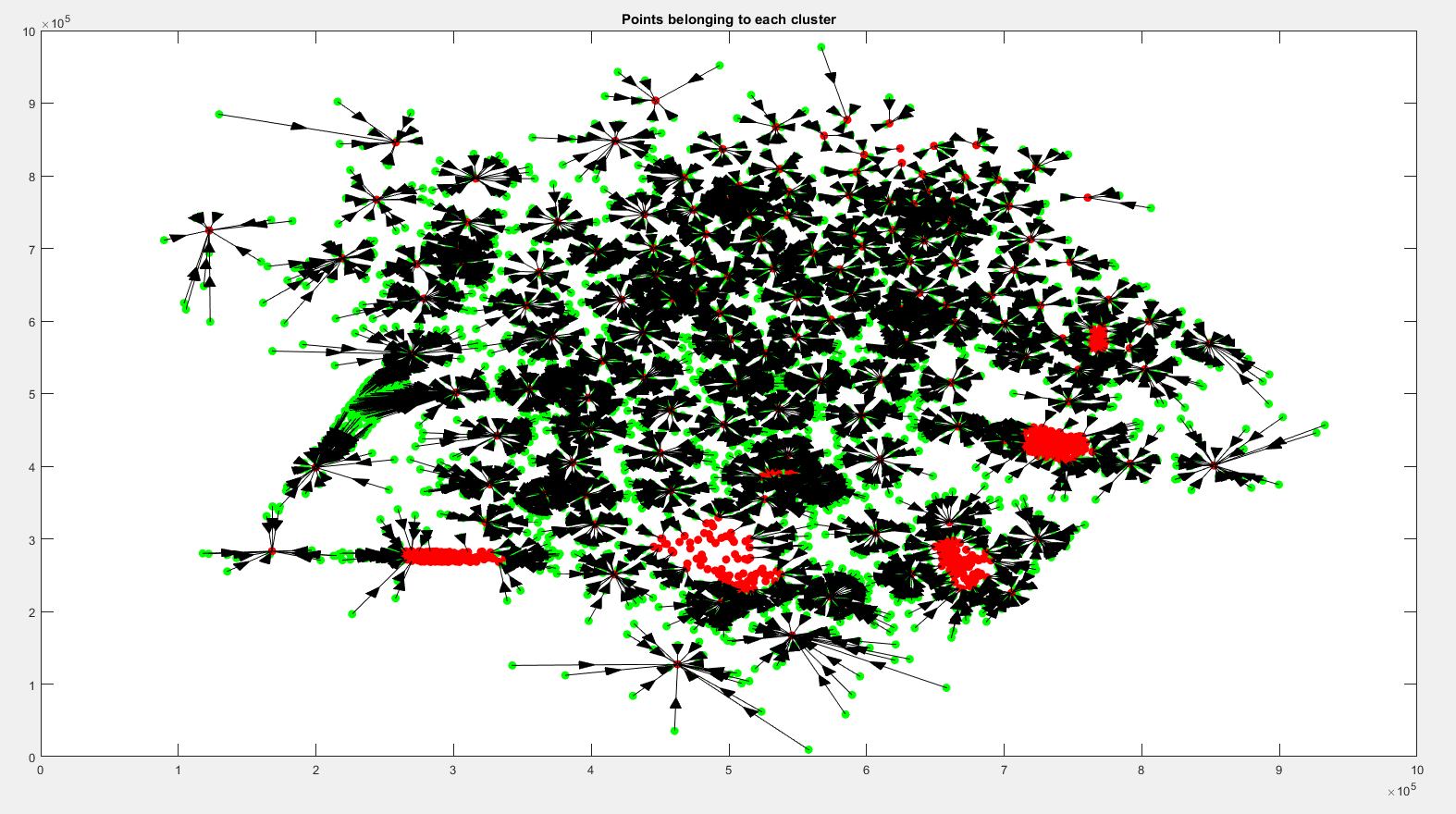


When points belongs to clusters



For iterations =100 total of 875 clusters are formed.





**Future Improvement/ Possible Improvements**

1. When we take a large dataset, the accuracy is very low. In the above dataset which has 5000 rows, it takes almost 2 minutes to execute. In the future enhancement we will work on the algorithm so that accuracy will be increased.
2. Our algorithm works exceptionally well for the small datasets but doesn’t works well for the extremely large datasets. When we tried to run the algorithm for the extremely large dataset and the cluster forming was not efficient.

**Thank you!**