CSE601: Data Mining
Homework 3

Group: 31

Anuj Rastogi – 5013 4324

Nalin Kumar – 5017 0479

Pranshu Pancholi – 5016 9864

What is Markov Clustering Algorithm?

Markov Clustering Algorithm is a graph based clustering algorithm which is based on the principle that highly connected nodes of the graph form one cluster while low connected nodes of the graph are in different cluster. Therefore, if a random walk is performed on the graph from 1 node to other there is a high probability that we stay in same cluster rather than move to a different cluster.

In Markov Clustering algorithm we perform a random walk from any node in the graph. At each step the system may change its state from the current state to another state, or may remain in the current state which is determined by certain probability distribution governed by transition probability matrix. The change of state is called transition and the probabilities associated with state changes are called transition probabilities. The algorithm is based on Markov property which states that given the present state, the future states are independent of the past states.

The algorithm leverages 2 operations on the adjacency matrix of the given graph. First is Expansion operation which promotes the denser region of the graph and second is Inflation operation which demotes the less favored region. The algorithm perform these operations until we reach a convergence. We used following formula to perform Inflation:

$$(\Gamma_r M)_{pq} = (M_{pq})^r / \sum_{i=1}^{k} (M_{iq})^r$$

Where, Γ_r is the inflation operator with power coefficient *r*.

In the end, to find the clusters generated we split the node into 2 types - attractor nodes and node that are being attracted (elements that have positive value within a row of attractor). Attractors and the elements they attract are clubbed together to form a single cluster.

Advantages:

1) It is simple and easy to understand and implement.
2) It is widely used in bioinformatics because of its noise tolerance and effectiveness.
3) It works on directed/undirected weighted and unweighted graphs.
4) It is scalable to large graph dataset.

Disadvantages:

1) It has higher running time of $O(N^3)$ due to matrix multiplication in Expansion step.
2) Overlapping clusters are found only in special cases and difficult to find in normal scenario.
3) Produces large number of clusters.

Algorithm Implementation:

We used Python programming to implement Markov clustering algorithm. The input to the algorithm is Expansion (E) and Inflation (R) hyper-parameters. Below is the pseudo code of the algorithm implementation-

Pseudo Code

1. Input the parameters i.e. dataset file name, expansion parameter E, Inflation parameter R.
2. Load the dataset.
3. Create a node index map. Assign index to each graph node by creating a map with key as node and value as its index. This
4. Find the total number of nodes (N) in the given graph by searching the maximum index value from the node index map.
5. Create an Adjacency Matrix of size N * N.
6. Load the Adjacency Matrix by setting 1 to the indexes corresponding to the node index in the indexing matrix for each edge.
7. Add a self-loop in the Adjacency Matrix by setting 1 in the diagonal.
8. Normalize the data by dividing each index by the sum of its column.
9. Iterate 100 times:
    1) Store a copy of Adjacency Matrix (transition matrix).
    2) **Expand** the transition matrix to the given Eth power.
    3) **Inflate** the transition matrix by raising the individual elements to given the Rth power.
    4) **Normalize** the data by dividing each index by the sum of its column.
    5) Check for **convergence**. If stored matrix is same as the calculated transition matrix then convergence has been reached and break out of the loop.
    6) Else **prune** the data. Perform pruning by setting the value which are close to 0 to zero. Assuming they will reach there eventually. Pruning is done to improve the performance.
10. Find the clusters generated by MCL algorithm. Iterate through each row of the resultant matrix. Create a list of indexes of the elements that are greater than 0. They belong to same cluster. Add the cluster to the cluster set.
11. Generate the cluster mapping of each node of the graph. Create a map with key as the node value and value as the cluster id it belongs to.
12. Write the cluster id mapping values to .clu file.

Experiment Results:

1) AT&T Web Network dataset.

Expansion Parameter (E)	Inflation Parameter (R)	# Clusters Formed
2	1.25	1
2	**1.35**	**5**
2	1.5	7
2	1.75	13
2	2	55
3	1.25	1
3	1.5	2
3	1.65	5
3	2	8

Pajeck Visualization for the 2 sets of parameters which generates 5 clusters are shown below –

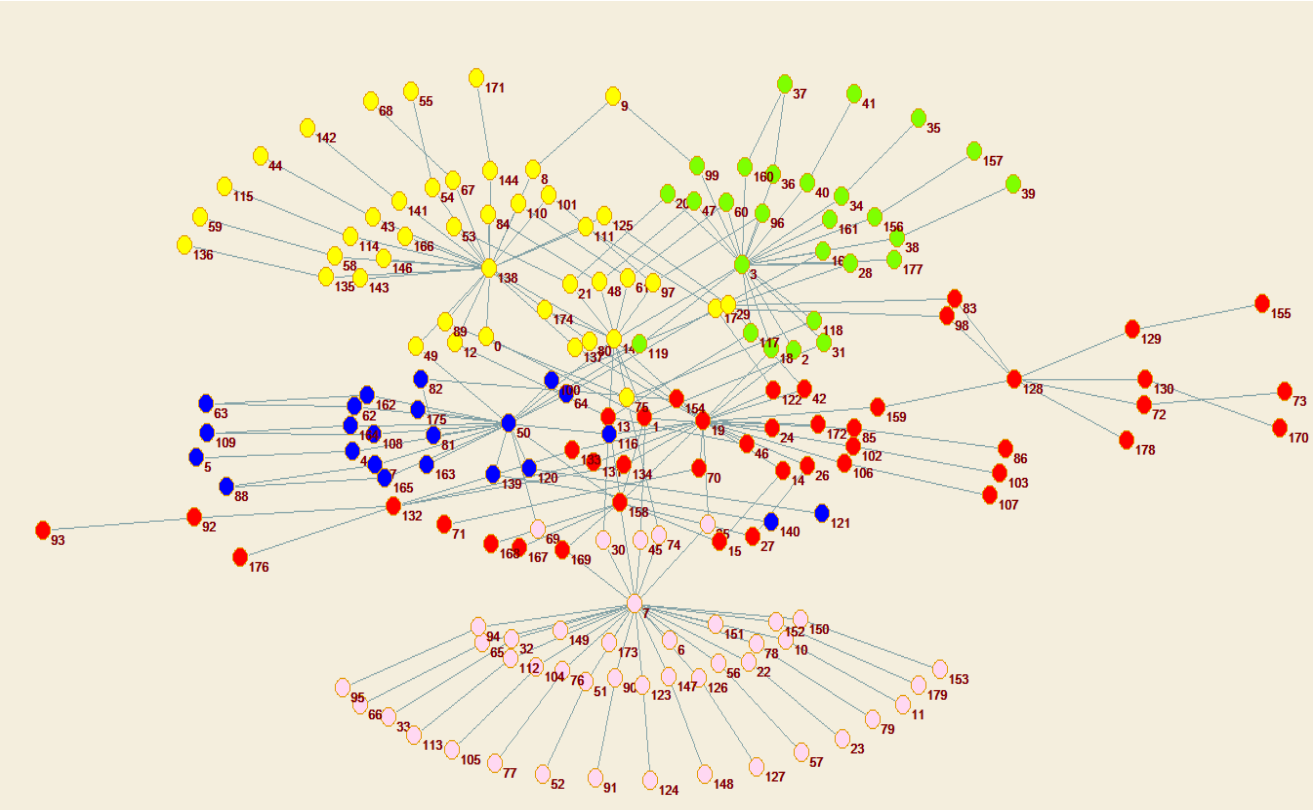

Fig 1. E = 2, M = 1.35

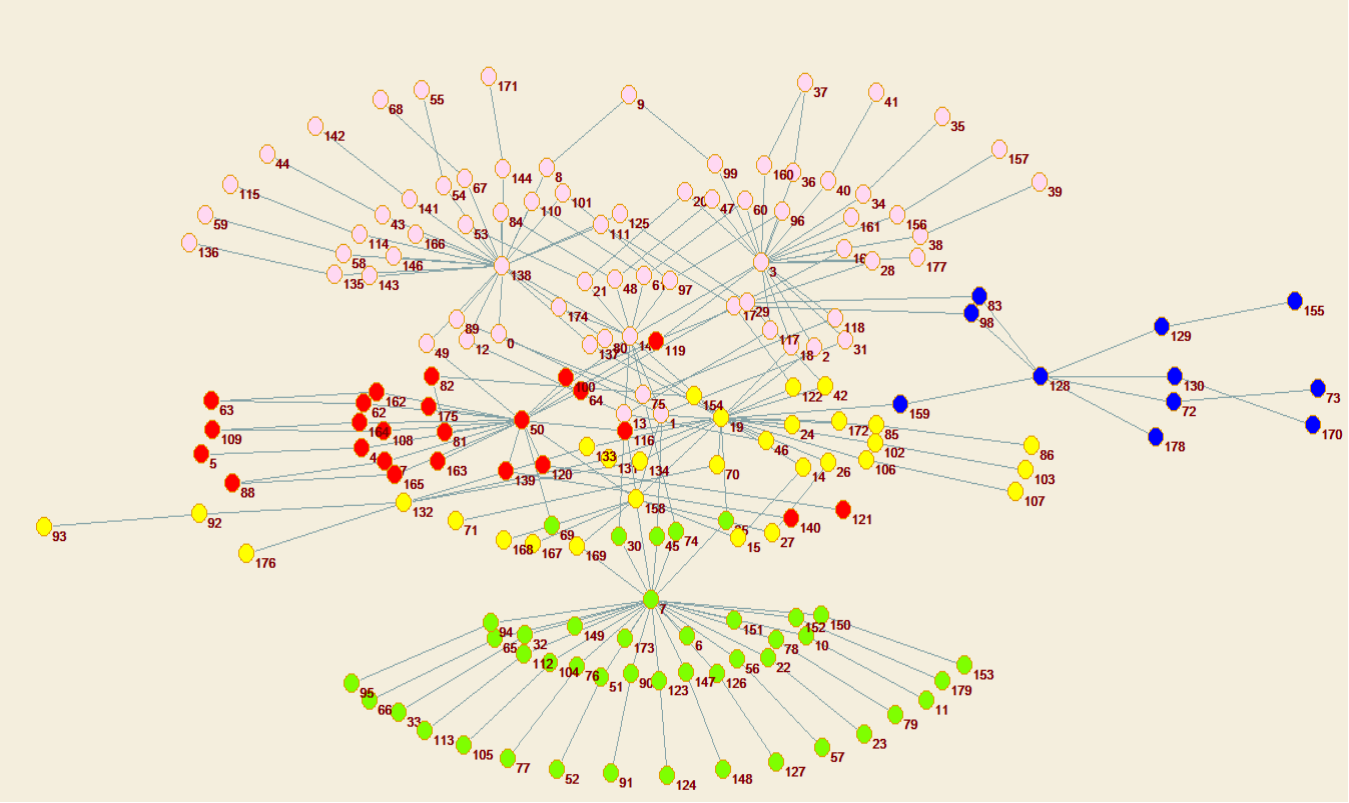

Fig 2. E = 3, M = 1.65

Result – From the 2 cluster formations we can easily depict that parameter values E= 2, M = 1.35 generates a well separated and close to ground truth value clusters.

2) Physics Collaboration Network dataset.

Expansion Parameter (E)	Inflation Parameter (R)	# Clusters Formed
2	**1.25**	**5**
2	1.5	14
2	1.75	21
2	2	24
3	1.25	1
3	**1.45**	**5**
3	1.75	10
3	2	14
4	1.65	5
4	2	9

Pajeck Visualization for the 2 sets of parameters which generates 5 clusters are shown below –

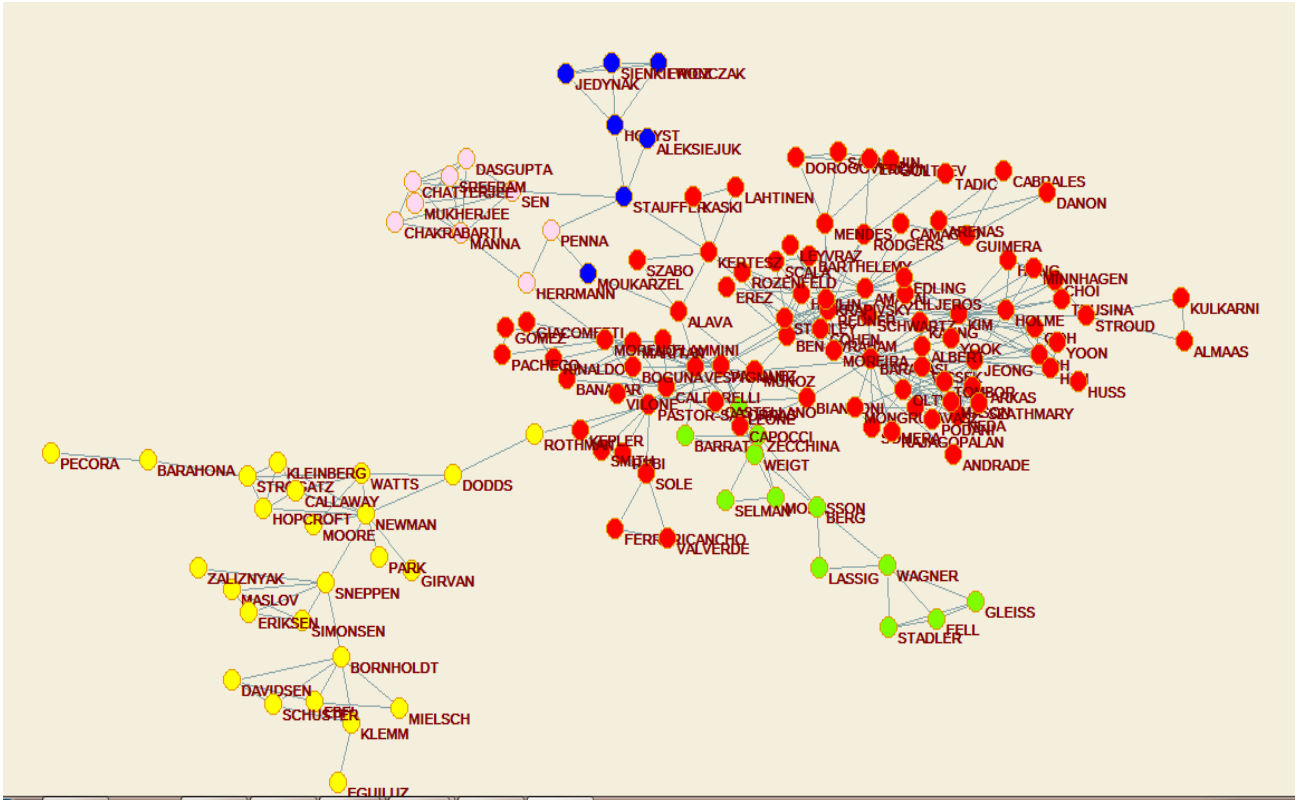

Fig 3. E = 2, M = 1.25

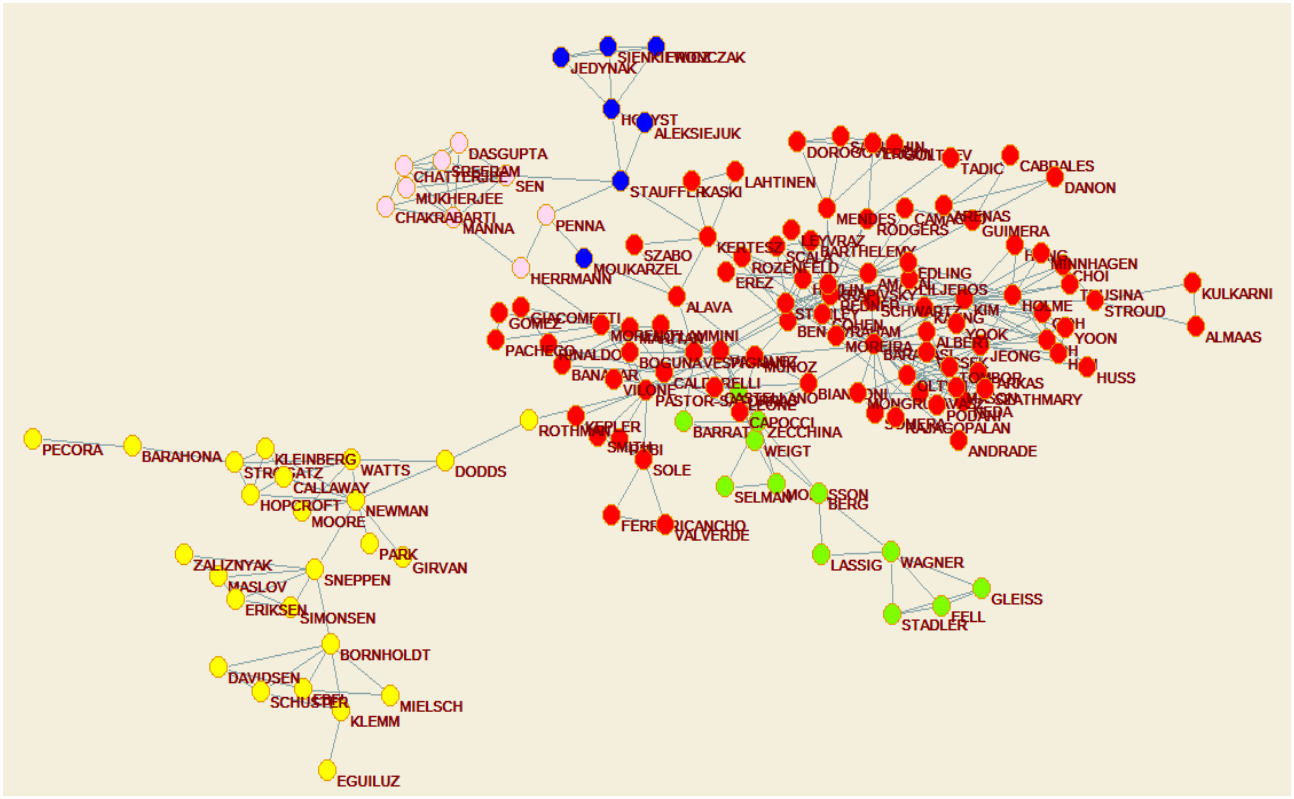

Fig 4. E = 3, M = 1.45

Result – The 2 clusters formations are more or less the same and have 5 clusters within them therefore, E = 2, M = 1.25 and E = 3, M = 1.45 are the 2 sets of optimal cluster formations.

3) Yeast Metabolic Network dataset.

Expansion Parameter	Inflation Parameter	# Clusters Formed
3	1.25	1
3	**1.34**	**7**
3	1.75	43
3	2	53
4	1.25	1
4	1.46	8
4	1.5	9
4	1.75	22
5	1.5	4
5	1.55	7
5	1.6	9

Pajeck Visualization for the 2 sets of parameters which generates 7 clusters are shown below

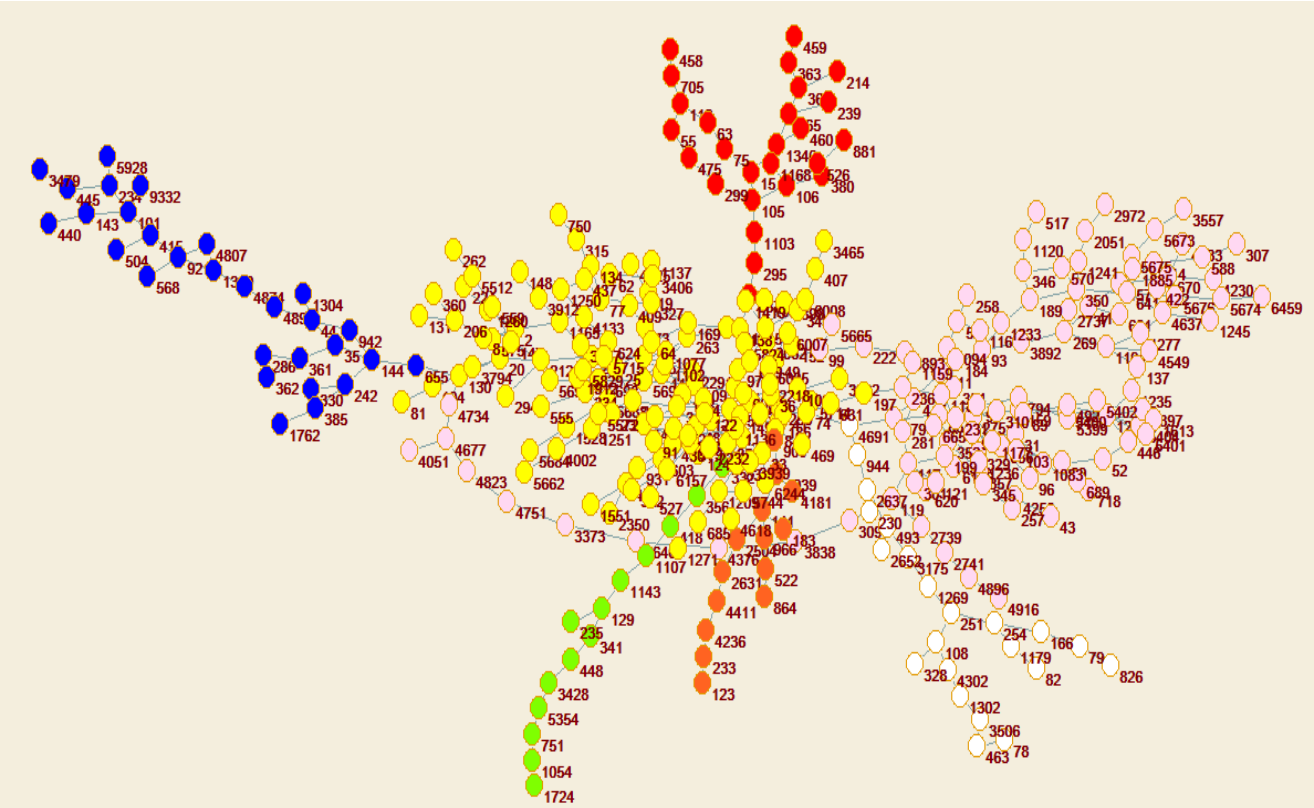

Fig. 5 E = 3, M = 1.34

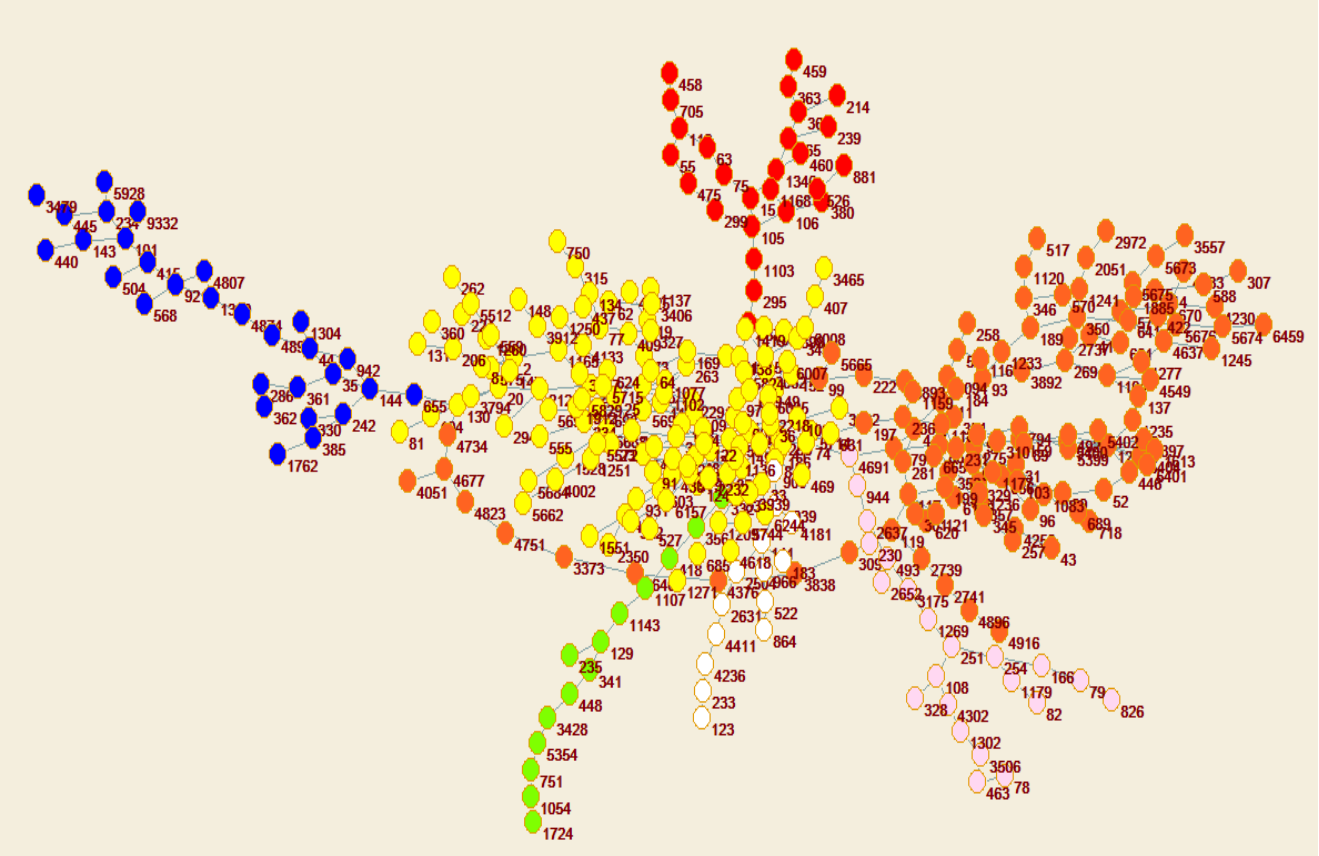

Fig 6. E = 5, M = 1.55

Result – From the 2 cluster formations we can easily depict that E = 3, M = 1.34 generates a well separated and close to ground truth value clusters.

Performance Improvement –

We used **pruning** technique to improve the performance of the system. In this technique during each iteration of the MCL algorithm we set the values of the transition matrix to zero which are very close to zero and will eventually become zero in subsequent iterations.

By applying pruning step we observed that the number of iterations required to converge were significantly reduced for a set of parameter E and M values, thus pruning step proved a performance enhancement.

Attweb_net.txt	E= 2, M = 1.35	# Iterations (with pruning)= 27 # Iterations (without pruning) = 39
Physics_collaboration_net.txt	E = 2, M = 1.25	# Iterations (with pruning)= 47 # Iterations (without pruning) = 64
Yeast_undirected_metabolic.txt	E = 3, M = 1.34	# Iterations (with pruning)= 33 # Iterations (without pruning) = 46

Analysis of experimentation results –

From the above experiment results we observed that setting expansion and inflation parameters plays an important role in generating optimal cluster formation. As we increase the expansion parameter and keep inflation parameter constant the number of clusters formed decreases while increasing the inflation parameter and keeping the expansion parameter constant increases the number of clusters formed.