A Neural Network based Community Detection Algorithm

A Thesis

submitted to

Indian Institute of Science Education and Research Pune in partial fulfilment
of the requirements for the BS-MS Dual Degree Programme
by

Sanak Mukherjee

Indian Institute of Science Education and Research Pune
Dr. Homi Bhabha Road,
Pashan, Pune 411008, INDIA.

April, 2022
Supervisor: Prof. Collins Assisi

Certificate

This is to certify that this dissertation entitled ‘A Neural Network based Community Detection Algorithm’ towards the partial fulfilment of the BS-MS dual degree programme at the Indian Institute of Science Education and Research, Pune represents work carried out by Sanak Mukherjee at Indian Institute of Science Education and Research under the supervision of Prof. Collins Assisi ? Professor, Department of Biology, during the academic year 2021-2022.

Collins Assisi

Committee: Biology

Collins Assisi

M. S. Madhusudhan

This thesis is dedicated to

My father

Dr. Keshab Mukhopadhyay

Declaration

I hereby declare that the matter embodied in the report entitled ‘A Neural Network based Community Detection Algorithm’. Here are the results of the work carried out by me at the Department of (Biology), Indian Institute of Science Education and Research, Pune, under the supervision of Collins Assisi and the same has not been submitted elsewhere for any other degree

Sanak Mukherjee

Date: 7.04.2022

Table of Contents

List of Figures

Abstract

A neural network-based community detection algorithm

Complex networks are typically not uniform random networks. Rather, they contain inhomogeneities -- densely connected groups of nodes with sparse connections to other nodes. Such densely connected sub-networks are known as communities or modules *(1)*. The presence of these modules is often an indication of distinct functions performed by the network with each module subserving a different function. Parceling functions into distinct structures can improve the evolvability of the network. Different functions may be selectively optimized without disturbing other aspects of network function.

One approach to detect modules in a network is to optimize a quality function, termed modularity, over different partitions of the network *(2)*. In this project, we propose an alternative method to detect modules by mapping the dynamics of a neural network (with connectivity based on the test network we wish to analyze) to the modular structure of a complementary network *(3)*. The neurons will be modeled as simple on-off devices that interact with each other in an antagonistic fashion. That is, if a given neuron is on, all the neurons connected to it are turned off. A slow time scale (compared to the time scale of switching) and transient inputs will be used to switch the state of the neural network. We anticipate that the identity of neurons that are synchronously activated will correspond to modules of a complementary network.

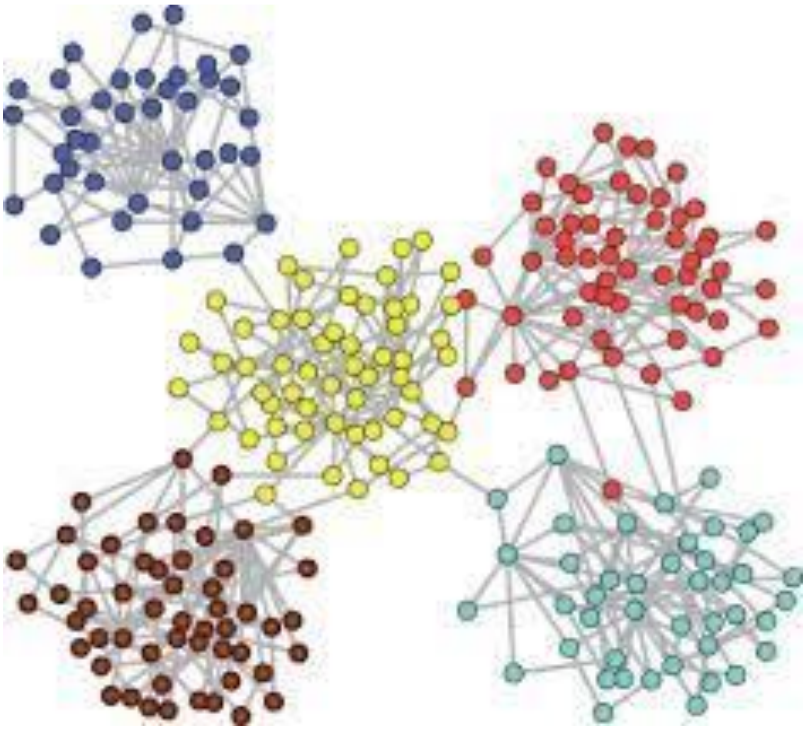
Acknowledgements

I would like to take this opportunity to thank my guide Prof. Collins Assisi , my senior in lab and a PhD pursuing student Inaiyath Shaikh . I would also like to thank the students of computational neurobiology lab in IISER Pune. I would also like to thank my friend Ankit Bhaskar and my father Keshab Mukhopadhyay.

Introduction

Community : A community is a group of vertices in a graph which are more connected among themselves than with the rest of the graph.

Fig . 1.



A standard graph is taken where there are 5 different communities . Elements of same community is denoted by the same colour.

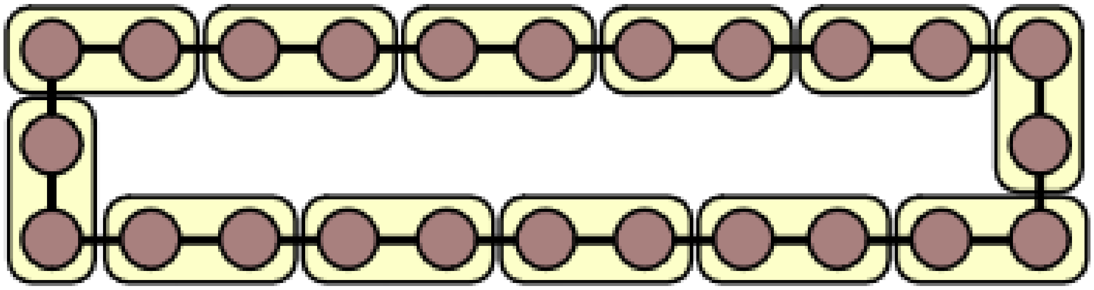
A dataset is considered. The datset consists of a set of points. The points have a set of properties defined on them. Grouping the points based on similarity of their properties into groups is called clustering . Newman's Clustering[1] makes communities based on a value standardized to the original graph. This type of clustering misses some group of nodes which should be considered as communities in themselves. A node is a vertex of a graph. At the beginning a measure of modularity[1] is introduced. A community of a graph consists of vertices which have more dense connections in between themselves compared to the rest of the graph. In Newman clustering[1] the modularity of the complete graph is zero. The partition of the graph which gives the maximum modularity[1] is taken as the group of communities. In reality when composing a graph, when the graph is generated under restricted conditions, certain structures can be introduced beforehand which already establishes the communities. Under those conditions it can be argued that the communities of the graph are already known. In such circumstances for specific cases Newman clustering does not provide with the optimal outcome. Two such cases are explained as follows :

Case 1:

A clique is a complete graph where all the vertices are connected to each other. Suppose a complete graph of 5 vertices is taken. Let it be called P in this case. Now 24 such P's are connected like pearls in a necklace. By our own creation we already know what the total communities are in it.   There should be 24 communities. Each of the clique should form a community of it's own.

Fig . 2.

-



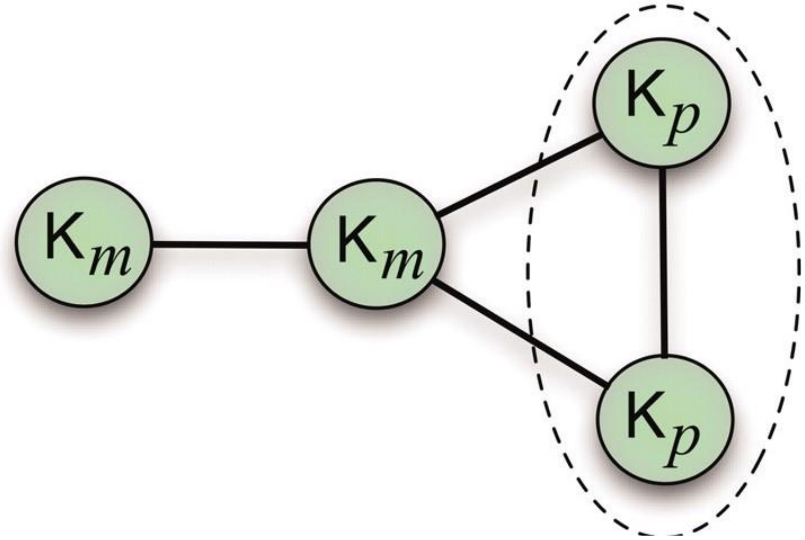
Each circle in the figure denotes a complete graph of 5 vertices. They are connected to it's neighbouring component by one edge . Each yellow rectangle clubbing 2 complete K_5 graphs denote the communities from Newman Clustering.

Now we run the Newman clustering[1] algorithm on it. The set of communities we find is different. It gives 12 communities. Two cliques are clubbed together as one community. The modularity value for both those cases are now calculated. When the above case has twelve communities the modularity[1] value is 0.8712 . When modularity[1] value of 24 such community is considered, as claimed it is less. It is 0.8674. Hence we can see in this case the algorithm fails.

Case 2 :

Let us consider another such example. Four complete graphs are taken into consideration. Two of them has 20 vertices each. Another two of them has 5 vertices each. Let the two complete graphs with 20 vertices each be called B1 and B2. The two complete graphs with 5 vertices each is called S1 and S2. B1 is connected to B2 with one edge. B2 is connected to S1 and S2 like a triangle. What I mean is that if S1 , S2 and B2 are considered like vertices, B1 and S1 are connected by one edge, S1 and S2 is connected by one edge and also so is B1 and S2. In this situation by architecting the graph in a certain way we can claim we already know the communities in them.

Fig . 3.



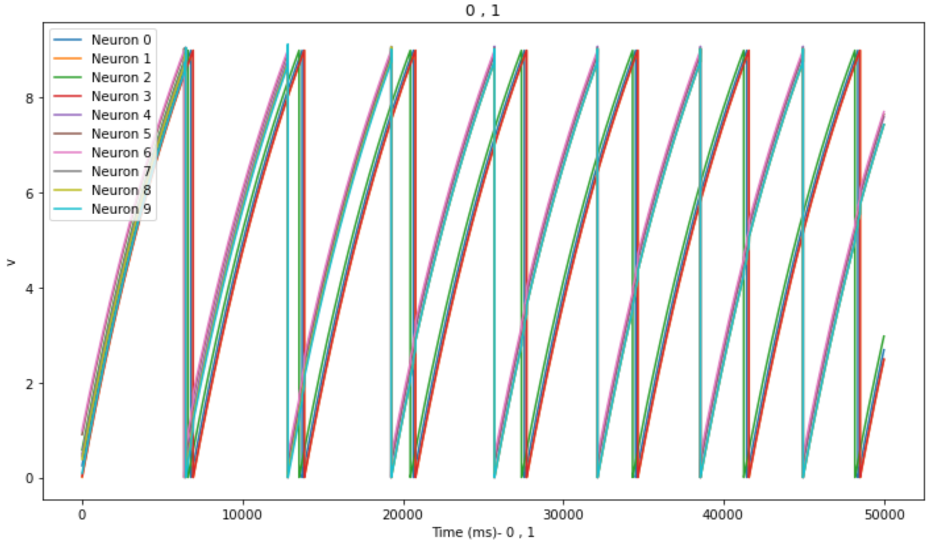
K_m and K_p denote complete graphs of m vertices and p vertices respectively. Here m=20 and p=5. They are connected to each other as shown.

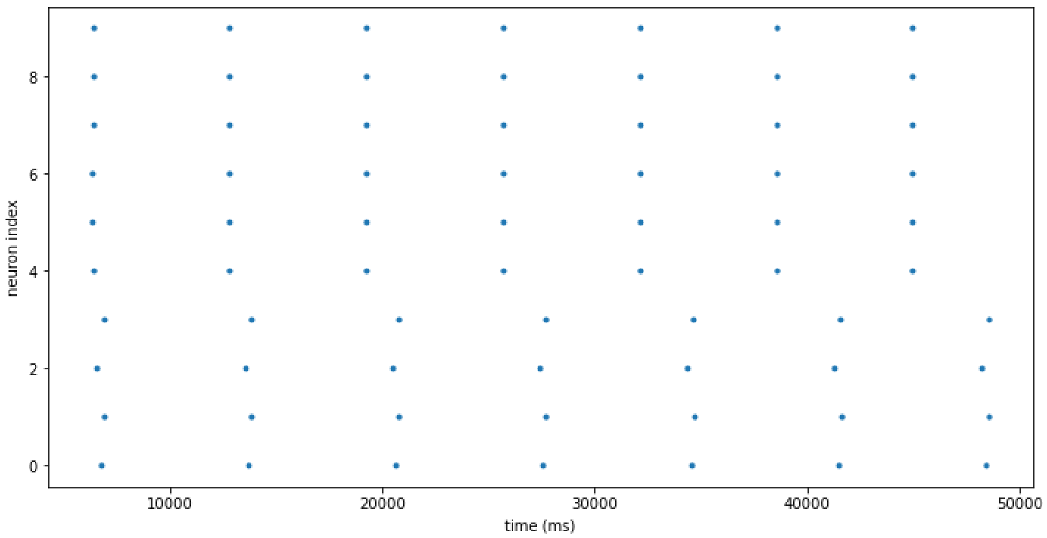
Each of the complete subgraphs in this whole graph that we have created should form a community as shown in Fig.3. Hence we should have four communities. However when we apply the Newman clustering[1] algorithm on it we get a different result. We get three communities. The two complete graphs of five vertices gets clubbed into a single community. Hence the communities are B1 , B2 and the union of S1 and S2. Once again the algorithm gives a different output than our prediction. However while running through examples like this we tend to notice a similarity in both cases.

This intuition also develops from our understanding of the process of partitioning of the graph into communities according to Newman's clustering[1] algorithm. We see in both these cases the smaller communities get clubbed together. Also those group of nodes individually sometimes form a smaller fraction of the total set of nodes. Newman modularity[1] value is based on a certain property of a graph. A group of nodes that have more interconnection than the rest of the graph will have a higher modularity index. However this process is normalized. Moreover to standardize the process, the algorithm is set such that the modularity[1] index when considering the whole graph as a community is taken as zero. Hence groups of nodes which are sparsely connected to the whole graph and less densely connected among themselves compared to the average density of the main graph, although should be considered as communities are not partitioned individually by the algorithm.

Groups of neurons consisting of no connections in them and inhibitory connections in them , tend to synchronize in spiking. Let a group of neurons be considered . They are themselves partitioned into various groups on their own under certain conditions. Under initial conditions all neurons are of similar quality. Suppose the neurons have no connections among them if they are in same partition in the total group. Now if two neurons, each from separate group is considered, they are connected to each other. Their connection is inhibitory in nature. Each of them when they spike , inhibits the spiking of the other neuron it is connected to. Under such situation we see that the individual partitions of neurons each tend to synchronize in spiking individually. Basically after an extended period of time the plot of the spiking of the neurons will be similar to individual neurons under similar conditions. If number of neurons equal to the number of partitions in previous condition is taken. After that all of them are connected to each other through inhibitory synapses. After a significant amount of time the spiking pattern of the above two conditions will be similar. The change in voltage on the receiving neuron is set to be a fraction of the threshold value of the neuron. On a micro level each spiking basically pushes the spiking of the other neuron it is connected to further down the road. The neuron which is delayed in spiking however is still closer to spiking than the other neuron. Along the way this neuron resets first and similarly delays the spiking of the neuron it is connected to. Similarly at the other end Hebb's rule[2] works. Suppose two neurons are named A and B. They are connected to each other by a synapse. The synapse is excitatory and affects both ways. Now without loss of generality A spikes first. Then it will excite B. This brings B closer to spiking than it would have. Then after B spikes it will excite A to spike before it would have for a second time. Gradually after a significant amount of time, they will end up firing together. With this concept in mind the plan is to make a neural network of the graph.

Fig. 4.





Group of 10 neurons {0,1,2,3,4,5,6,7,8,9} as denoted. Vertices {0,1,2,3,4} and {5,6,7,8,9} form complete graphs. There are a few connections in between these set of vertices. An inhibitory connection is run along the complementary network of this graph. The change of membrane potential with time and the raster plot of these neurons are shown in these figures respectively .

The new network will separate the communities in the graph . The elements of the communities will also be closer to each other. Hence with the help of a distance matrix which we can create from this we will be able to cluster the graph into communities. Now to find the distance between them we go to a different concept. For this we are introducing a concept called spike distance. In this the graph of voltage vs time for each neuron is modified a little. The graph is broken from a continuous to a discontinuous graph. The voltage at the threshold during spiking is only taken into consideration. Rest of the time the voltage is converted to zero, in order to convert the graph to a spike train. For each neuron the graph is converted similarly. Then based on each point of time, a synchronicity is calculated for two spike trains. In my work I have integrated this measure over a standard period of time to gain a distance metric I wish. The time is based on the neuronal dynamics and the total number of neurons.

In Newman clustering[1] we see that it fails to distinguish between communities under certain cases. When the density between intra-edges and inter-edges of a community comes close, faults tend to arise. This phenomenon is termed as resolution limit[3] . Now that we have created a pair-wise distance matrix, we can apply machine learning clustering algorithms on it. The main issue that is attemped to be rectified here is to have enough clarity between clusters. For resolution of this clustering, K-means clustering method and Agglomerative clustering method has been used . From the intuition and results of various observations only one of them has been chosen eventually. First in the model the initial point of each neuron has been randomly chosen with insignificant magnitude. This will cause them not to have same time of first spiking . Moreover a maximum amount of magnitude in change for spiking of a neuron to it's following neuron is calculated. This value will be in relation to other parameters of the model. The primary criteria for it would be to not destabilize the system. What is meant by this is that it should not cause any neuron to stay in continuous depolarized state of not sending any action potential , or even for a lengthy meaningful period of time. It should also not cause any neuron to send a continuous stream of action potentials for any time period. Now that the whole neural network is created it can be worked upon. However as architected the neural network is a disjoint union of excitatory connections and inhibitory connections. The magnitude of change in post-synaptic potential due to each type of connection need not be similar. Hence the maximum magnitude of change in potential in post-synaptic neuron due to generation of action potential graded and broken down to a uniform scale. Now the inhibitory post-synaptic input is gradually increased along the units of the scale. This is similarly done for the excitatory connection. Hence we get a network where the ratio of these dissimilar connections vary all over the points of a two dimensional gradient graph. The target is to find the most suitable clustering from all these observations. Newman clustering[1] was created on a concept which had a distinct mathematical structure given. Hence under deep scrutiny artificially constructed graphs could be created where our intuitive division of communities in them which did not match with the result of Newman clustering[1]. Moreover even if the communities form by connections, they segregate themselves from each other due to lack of connections. In Newman clustering[1] no account of that was taken . Hence mostly we can see that the overlap between communities were mostly complete union of them in standard examples. However we can notice that those mega communities could and should be divided into disjoint partitions. But still without any distinct universally defined concept of community , any mathematical formulation provided for it will have it's setbacks. Hence to approach this issue in a different way, a neural network is created based on the graph structure which can provide the value for us. The segregation or partitioning of the communities are provided by the inhibitory connections. The connectiveness or bonding for making the distance of points in between the compact is provided by excitatory connections. However as the network is all to all connected the impact of these uni-dimensionally oppositely directed impact of these connections created from excitatory connections along the graph and inhibitory connections along the complementary graph also never completely convert the simulation periodic. If such a situation was applicable, the approach would not have been much different from Newman clustering[1]. Yet due to the large number of datasets we can argue that the fundamental core nature of the network will have the maximum impact on the result . Now due to this model an impression of a large number of ways so that any graph can be grouped into communities is created. Each of them can be argued as a clustering based on goal. This method has also tried to address the fundamental reason Newman clustering is not applicable in certain situations based on intuition. Now along the way it needs to be figured the most similar set of observations and justify the core of it which will lead us to our result.

Materials and methods

Knowledge and understanding of algorithm is required. A machine that can go through any length to run the code is also a necessity. Knowledge of neuronal dynamics of integrate and fire model[4] is required. Exponential integrate-and-fire models[4] allows us to run neural networks easily as they are simple in terms of variables. Exponential integrate-and-fire models[4] are widely used in the field of computational neuroscience and spiking neural networks because of
(i) a solid grounding of the neuron model in the field of experimental neuroscience,
(ii) computational efficiency in simulations of the python packages which run these simulations.
(iii) mathematical clarity in the modules which run these simulations in python.

Also a depth in knowledge in graph theory is required. A graph is defined as a set of two types of objects. Let G be a graph. ‘G={V,E}’. The set of two types of objects are ‘V’ and ‘E’ . ‘V’ and ‘E’ stands for vertices and edges respectively. ‘V’ consists of a set of objects. ‘E’ consists of objects of ‘V’ combined together to form an object. Let ordering of objects of ‘V’ to form an object of ‘E’ does not matter. The objects of ‘E’ are also only pairs of objects of ‘V’ where ordering does not matter. The objects of ‘E’ do not repeat themselves. Pair of same objects from ‘V’ do not form an object of ‘E’. When a Graph qualifies the above mentioned conditions it is called a simple graph.

Neurons have three different types of potential. They are called resting potential, graded potential and action potential. At resting potential, the potential inside the neuron is less than outside the neuron . There are various ions around the cell membrane of neuron both inside and outside. However the permeability of those ions is different on either side of the cell membrane. There is also sodium-potassium pump attached to the cell membrane of the neuron. The negatively charged ions on neither side are chlorine and proteins. The positively charged ions on either side are sodium and potassium . The cell membrane has various channels which are attached to the cell membrane. The major ones with implications to this project are leaky channels and voltage gated channels. The negative ions have no means of passing through the membrane. The electric gradient tends to push the positive ions out of the cell. However the density of potassium ions is higher inside the cell than outside the cell. Similarly the density of sodium ions is higher outside the cell. However there is a sodium-potassium pump. This helps pumping three sodium ions out and two potassium ions in. This tends to keep a varying dynamic equilibrium inside the cell. A graded potential is a change in potential that can vary in size. This is caused by continuous exchange in ions across the membrane. They also vary due to the input of stimulus from connected neurons. The stimulus may be excitatory and inhibitory based on the change in voltage in causes in the affecting neuron. The resulting variation when causes a change in voltage generally, an action potential is created. This sends an electrical impulse along the length of the axon to the other end of the neuron. This in turn causes a change in potential in the neuron it was connected to.

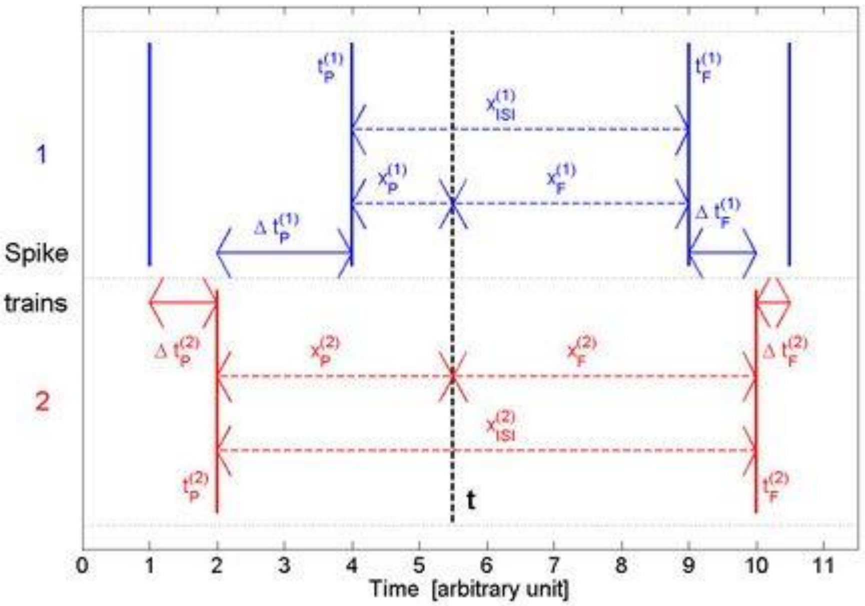
In this project a simple neuron which sends an action potential along specific intervals. The change in potential across the membrane of neuron can be simplified to '-[(dV/dt)]=(I-V)/tau' [4] . 'V' is the potential difference across the membrane . The minus sign can be ignored based on which side of the membrane you are considering. 'I' is the rise in potential difference required for the propagation of action potential. In order to balance the whole equation , 'tau' is the time constant in this.

Now while creating the system the ratio of the numbers are to be taken into consideration. The value of effect of stimuli from the preceeding neuron to the proceeding neuron should not cross 'I' or the designated value of action potential in this case. Hence if a graph has 'n' vertices, the highest value of inhibition or excitation for each stimuli should not be more than '(I/n)' in voltage. Also in previous section an idea is provide of how the inhibitory connection and excitatory connection have an effect on the spiking on the connected pair of neurons. Now the main graph is converted to a neural network. The edges of it are bidirectional excitatory synapses. Now the complement of the main graph is taken . The edges of it are bidirectional inhibitory synapses. When all the vertices of the graph is considered as neurons, all of them are connected to each other both ways . In this process without any external stimulus from other neuron , each neuron should periodically fire action potentials after a specific period of time. There are 'n' neurons . In order to give all the neurons proper time to have effect from other neurons, the total run time of the simulation is of order n.

All the coding is done in python. The packages used are
Brian 2 – For plotting the neural network
Numpy – For mathematical operations and using data structures like arrays and matrices.
Matplotlib – For plotting all graphs.
Scikit-learn – For machine learning clustering algorithms.
Scipy – For clustering algorithms and integrating functions in algorithm.
Networkx – For generating and drawing graphs.
Math – For doing mathematical operations.
Itertools – For creating partitions in sets.

Now to undergo clustering a distance matrix is required . First the graph of voltage vs time across the membrane is converted to spike train vs time for each neuron. Now for a point of time in the simulation, two different spike trains are compared. This creates a similarity or dissimilarity profile for each pairs of spike trains at specific instances of time. For ith spike train at a specific instance of time the preceeding and proceeding spike of that spike train is taken. Let them be denoted as t^i_P (t)[5] and t^i_F (t)[5] respectively. The variables notations will change accordingly. The preceeding and proceeding spikes are called previous and following spikes and are similarly denoted. For ith spike the interspike interval at time t is the time difference it's previous and following spike . It is denoted as $v_{ISI}^i(t)$[5]. Now two spike trains i and j are considered . The value of difference in time for the previous spikes at the given time is calculated. It is denoted as $\Delta t_p^{(I,j)}(t)=t_P^i(t)-t_P^j(t)$[5] . The direction of the vector is important. Similarly it is also calculated for the forward spike trains. $\Delta t_F^{(I,j)}(t)=t_F^i(t)-t_F^j(t)$ [5] . It is calculated and denoted by the equation in the previous line. For the pair of spike trains 'i' and 'j' , at any specified time , for each spike train two boundary points are established of previous and following neuron. Hence for each pair of neurons at any specific time four boundary points are defined . Now for each of these boundary points the smallest time distance with it's neighbours is calculated. Without loss of generality let us define this for the previous spike of the ith spike train at time t.

Fig. 5.



Two spike trains 1 and 2 and their spikings at arbitrary point 't' in time is shown .The variables in the figure denote their corresponding values. [5]

$\Delta t_P^i(t)=\min(|t_P^i(t) - t_a|)$[5] . Here '$t_a$' denotes all the other boundary spikes. Now the time difference in spike with the specified time in calculated to use as ratio while calculating the variables we need. For 'i'th spike at time 't', for previous spike it is calculated as '$x_P^i(t)=t-t_P^i(t)$' [5] . For ith spike at time 't', for following spike it is calculated as '$x_F^i(t)=t_F^i(t)-t$' [5]. Now for each point in time a local weight of the spike train is calculated . Now calculating it at time 't' for 'i'th spike train at time t through with respect to jth spike the following equation.

$$S_i(t)=(\Delta t_P^{i,j}(t)x_F^i(t)+ \Delta t_F^{i,j}(t)x_P^i(t))/ v_{ISI}^i(t) \qquad ....[1]$$

It is similarly calculated for the 'j'th spike too. Now enough variables are formulated to calculate the dissimilarity between two spikes . Now at a time 't' for 'i'th and 'j'th spike train the dissimilarity value 'S(t)' is

$$S(t)=(S_i(t)\, v_{ISI}^j(t)+ S_j(t)\, v_{ISI}^i(t))/(v_{ISI}^i(t)^2+ v_{ISI}^j(t)^2)...........[2]$$

Now enough variables have been defined to create the distance matrix for each pair of neurons. The above variable gives dissimilarity measure between two spike trains at a specific point. Hence by integrating it over the entire simulation time, we get a distance measure which can be used as pairwise distance for each pair of neurons.

Now modularity as explained by Newman[1] is defined. Let the main graph be 'G'. The graph is divided into 'm' communities. This value of 'm' and it's structure is provided by the algorithm itself. Let 'Q' be the modularity quotient obtained from this partition of 'G' ,then

$Q=\Sigma_{s}{}^{n}{}_{=1}[(l_s/L)-(d_s/2L)\hat{}2]$[1] . ...[3]

The terms inside the equation will be explained . The variation along the summation denotes the communities. Fundamentally, the claim of Newman clustering algorithm is that the optimum partition of communities will give the highest value of 'Q'. Now the term inside the community is calculated individually for each partition. Here 'l_s' denotes the number edges within a module. The module is the partition of nodes denoted by s. L is the total number of nodes in the whole graph . 'd_s' is the sum of all the degrees of nodes in a module. In a graph the sum of all the degrees of all the nodes is twice the value of total number of edges in a graph[6]. When the graph has no partitions by definition 'l_s=L' . Hence when the graph is not partitioned into communities the value of 'Q' becomes 0. However later a resolution factor[1] was introduced. It was denoted as 'γ' . Under it's influence the equation becomes

$Q=\Sigma_{s}{}^{n}{}_{=1}[(l_s/L)- \gamma\ (d_s/2L)\hat{}2]$[1] . ...[4]

If the value it is given is less than 1 the partition favours larger communities. Intuitively we can understand that when the previous condition is satisfied the graph has a positive modularity quotient to begin with. Within the summand of the equation, the first term increases the modularity while the second term decreases it. The first term increases with larger communities. The effect of the second term is diminished due to the value set for 'γ' . Hence we can see why the previous claim works. Similarly when the value of 'γ' is more than 1 the converse happens. Hence we can see that smaller communities get resolved.

The model is based on primarily on how neurons vary in voltage over time. It is influenced by excitatory connections. It is also influenced by inhibitory connections. The inhibitory connections will cause the neurons to send axon potentials at different time to each other. Similarly it can be seen that when two neurons excite each other, they cause each other to spike sooner than expected. Spiking of one causes a voltage rise in other which makes it spike sooner. Simultaneously the other neuron performs similarly till there is no noticeable difference in their variation of membrane potential. Hence under this condition a graph which has two vertices and is connected will be similar to a community. Also when the graph has two nodes but are not connected , its spike distance will have a value distinct and not close to zero. Now it will be seen that like this set of disconnected complete graphs will act as independent communities. The number of communities will be equal to the number of elements in the set.

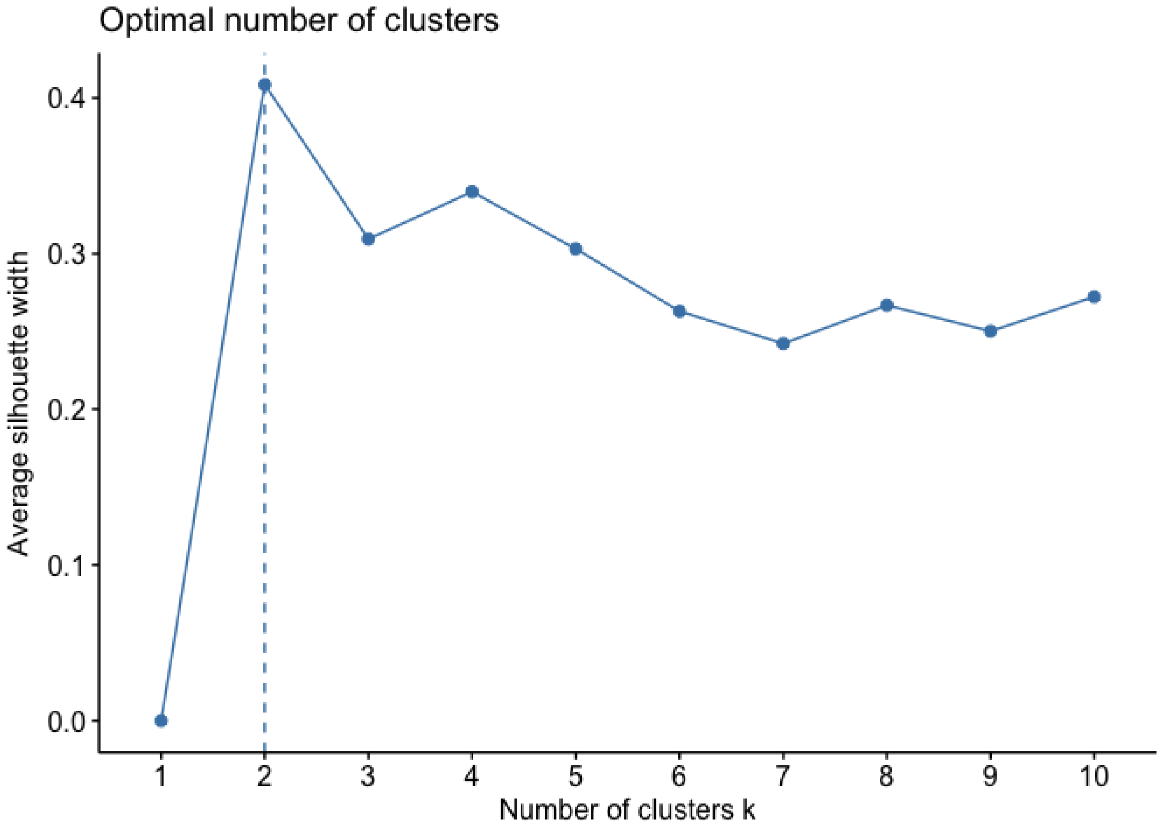
Choice of clustering

When the graph is constructed , the communities are structured and hence should be fixed. So in this situation K-means clustering is done. In K-means clustering the set of points in distance matrix is optimally clustered into k clusters based on algorithm. First randomly k points are taken in the vector space. All the points are clustered to their nearest point. In the next step the mean points of these clusters are taken as the initial k points and the algorithm is run again. It is run continually till the elements of clusters do not change. In this clustering the process of clustering do not modify the distance between points while comparing them. Hence it has been used over Agglomerative clustering. Now there are a few ways of determining the optimal number of clusters thst should be extracted from K-means clustering. Silhouette method and Elbow method are used based on the simplicity of their application and the inference of their measure.

Silhouette Method[7]

Suppose there are k clusters . The mean intra-cluster distance of all the clusters is calculated . It is designated as a. Now for each cluster the distance with the center of each cluster is measured. For each cluster in such a way the distance with it's nearest neighbour is calculated. It is denoted as b. The Silhouette Coefficient for a sample is (b - a) / max(a, b) . The value of k for which it is maximum is our desired number of clusters.

Fig. 6.

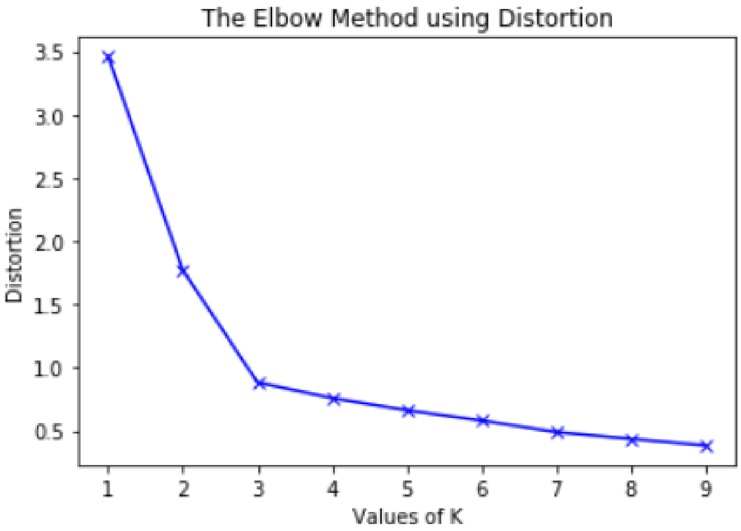


Along the y-axis silhouette coefficient is calculated . Along the x-axis the number of clusters are varied for an arbitrary example. In this case the optimal number of clusters is 2.

Elbow Method[8]

In each cluster a centroid is determined. It is the mean point of all the points in the space for a cluster . Now like variance with respect to it the square of distance of all the points in the cluster is calculated. This value is calculated for all the clusters and summed up. This measure is called Within Cluster Sum of Squares or WCSS. For one cluster it is maximum. It is zero if the total number of clusters is equal to the number of points. The value is plotted against the number of clusters . For a value of k, it's point in graph is connected to the respective points of it's neighbours as a straight line. This causes a change in slope for the graph at the integral values within it's range. The point in the graph which has the maximum impact for the change in values should be an optimum point for determining the number of clusters. The plot of this graph takes the shape of a folded hand of a human. The chosen point is named to be the elbow point for the similarity in structure. In this process only the distance between the communities in the community is calculated. This is why this reason is chosen.

Fig. 7.



On the y-axis is a measure of WCSS and on the x axis is the number of clusters . At k=3 it is the elbow point of this graph based on the requirements of this arbitrary example.

Let the method of clustering introduced in this research be called as 'Synaptic clustering'

Results and Discussion

While comparing with Newman Clustering[1] it has been noticed the existence of a resolution limit[3] which has been mentioned in previous sections. The graph has a certain number of edges. Averaging the number of edges over the number of vertices gives us the measure of the density of edges. Similarly let it be considered that the graph is already segregated into communities. Considering communities as subgraphs in themselves, a similar density measure of edges for the communities is acheived. Now as the density measure of the edges between graph and the rest of the communities tend to become closer, the value of modularity tend to decrease . The variation in modularity is also not similar to linear progression.

Now two cases are studied in the results and Synaptic clustering is applied on them.

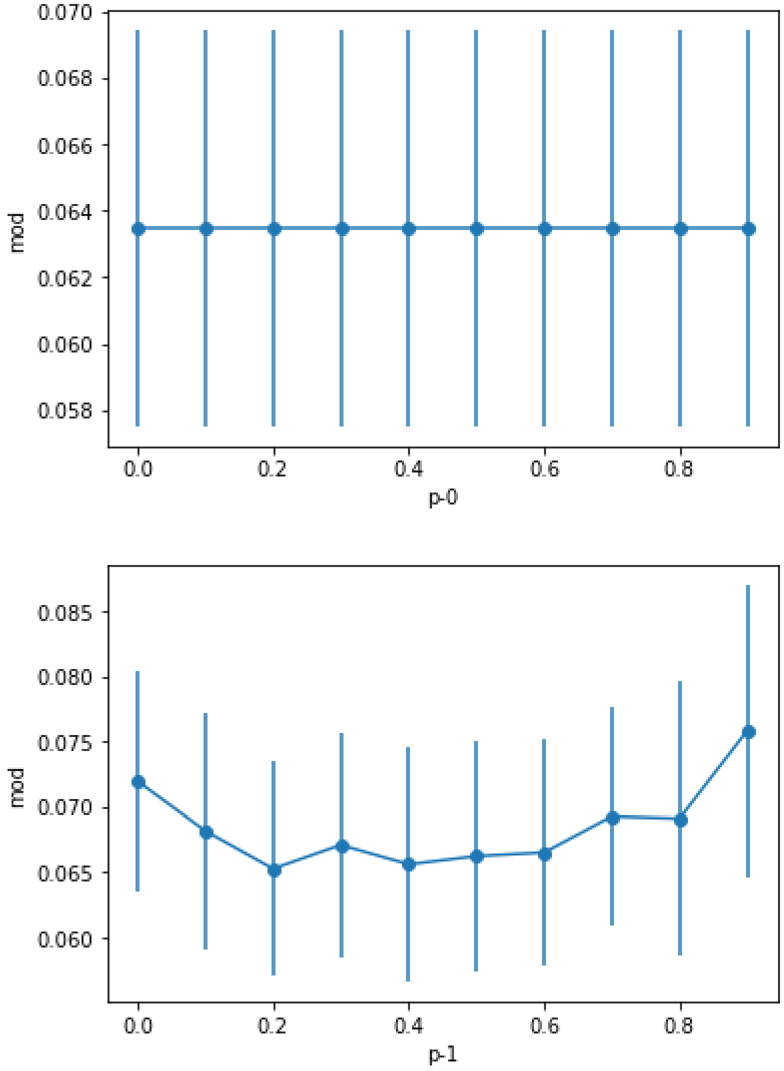
Case 1 :

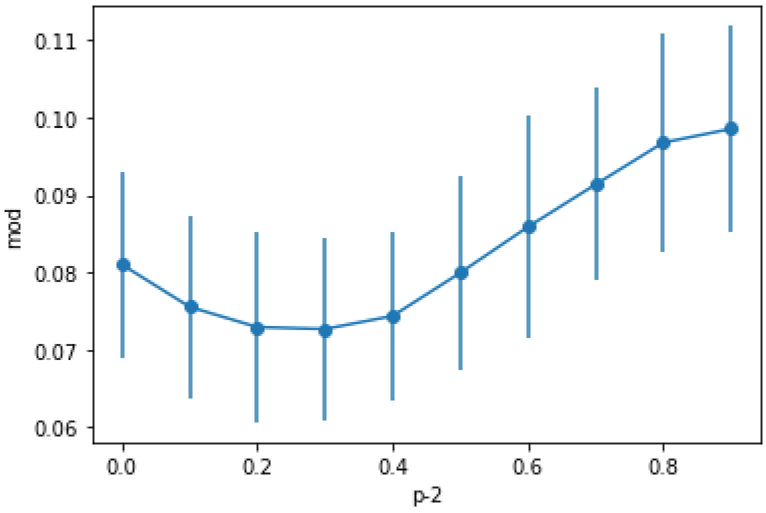
Here a graph of 10 vertices is considered . The vertices are numbered {0,1,2,3,4,5,6,7,8,9}. The set of vertices {0,1,2,3,4} form one community. The set of vertices {5,6,7,8,9} form another community. There are edges in between the two communities. It is denoted as 'InterE' . Community {0,1,2,3,4} is denoted as A. Community {5,6,7,8,9} is denoted as B. Edges inside community A is denoted as 'IntraA' . Edges inside community B is denoted as 'IntraB' . It has been used such that the density of edges in community A and density of edges in community B are similar. The density of edges in community A and B are denoted as 'dIntraC' . Now density of edges in 'InterE' is considered. It is denoted as 'dInterE' . For the graph to have the pre-specified communities , dIntraC >= dInterE condition should be satisfied. This is because communities are more densely connected than the rest of the graph.

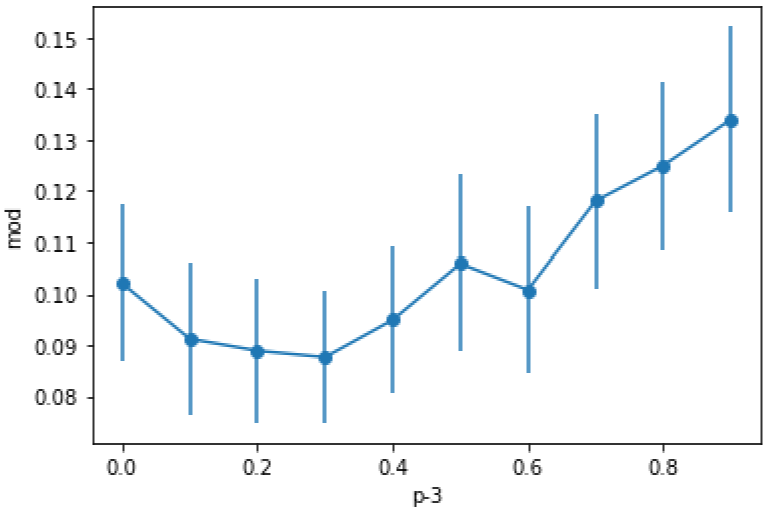
a.) dIntraC>=dInterE ....[5]
or, dIntraC-dInterE>=0 ....[6]

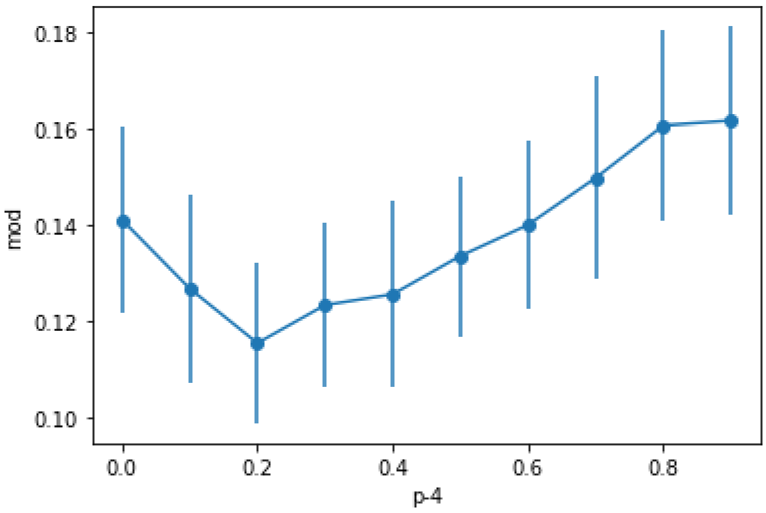
Hence the above mentioned condition can be satisfied by the accordingly .

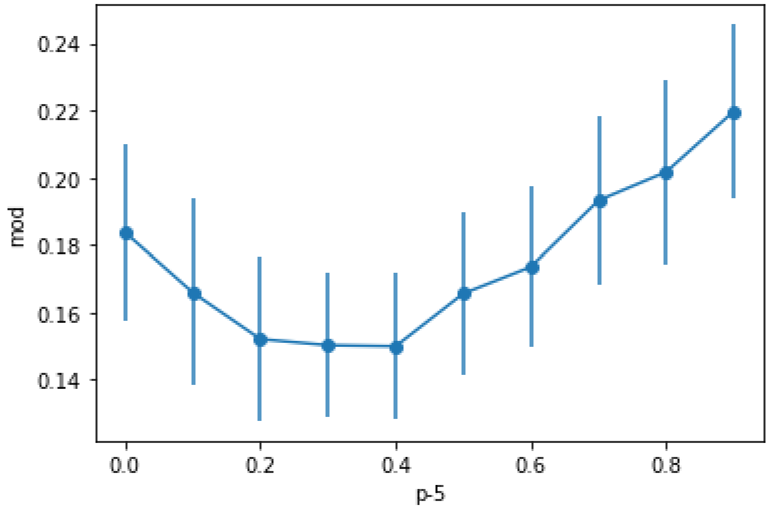
Fig. 8.

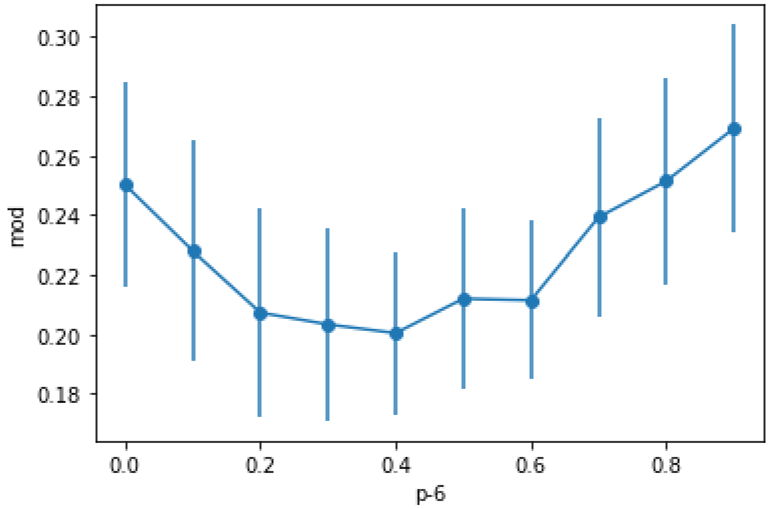

0.070
0.068
0.066
0.064
0.062
0.060
0.058
mod
0.0
0.2
0.4
0.6
0.8
p-0
0.085
0.080
0.075
0.070
0.065
0.060
mod
0.0
0.2
0.4
0.6
0.8
p-1

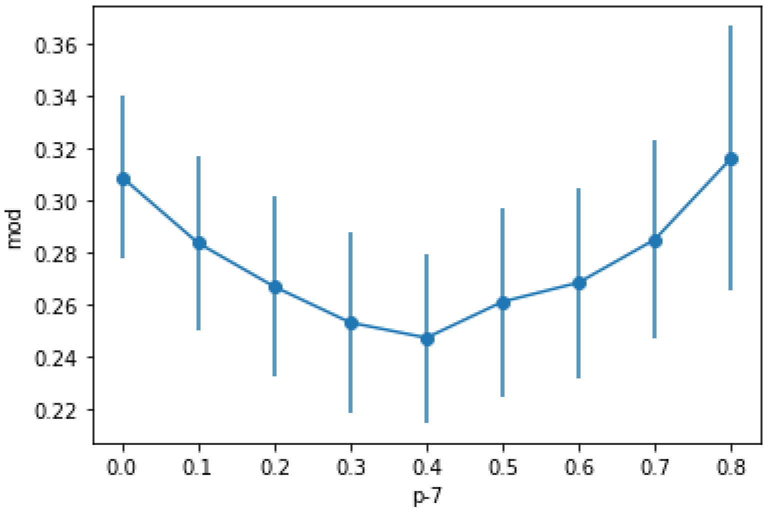

0.11
0.10
0.09
0.08
0.07
0.06
mod
0.0
0.2
0.4
0.6
0.8
p-2

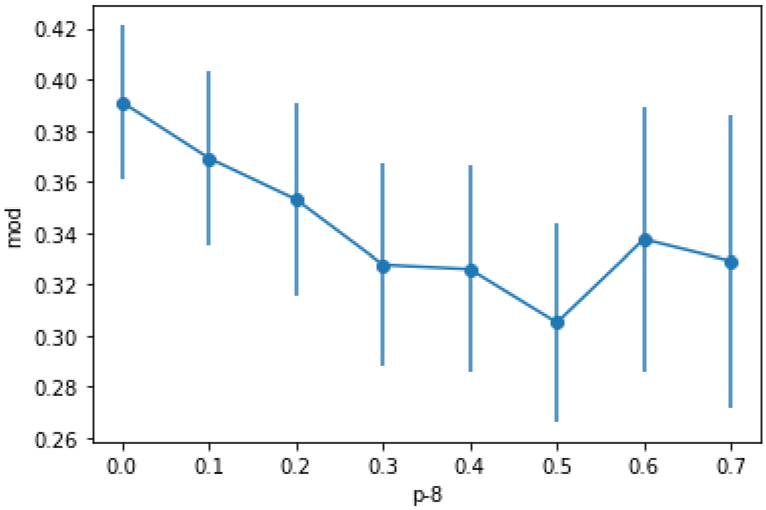

0.15
0.14
0.13
0.12
0.11
0.10
0.09
0.08
mod
0.0
0.2
0.4
0.6
0.8
p-3

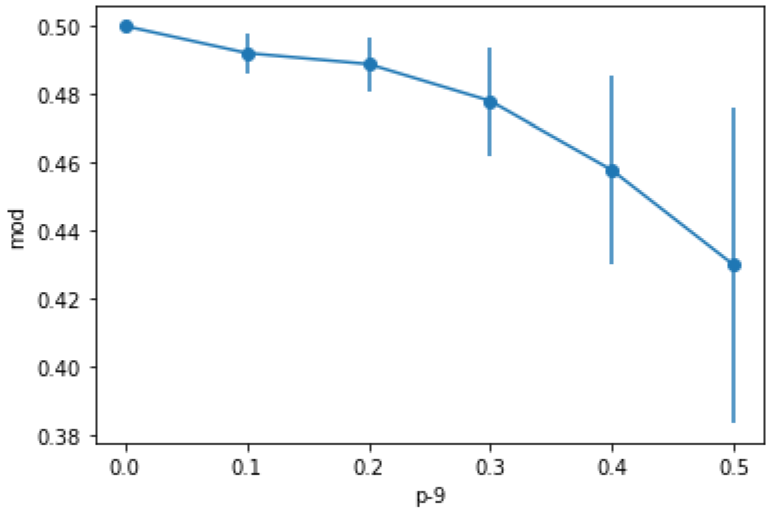

0.18
0.16
0.14
0.12
0.10
mod
0.0
0.2
0.4
0.6
0.8
p-4


mod
0.24
0.22
0.20
0.18
0.16
0.14
0.0
0.2
0.4
0.6
0.8
p-5


mod
0.30
0.28
0.26
0.24
0.22
0.20
0.18
0.0
0.2
0.4
0.6
0.8
p-6


0.36
0.34
0.32
0.30
0.28
0.26
0.24
0.22
mod
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
p-7


0.42
0.40
0.38
0.36
0.34
0.32
0.30
0.28
0.26
mod
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
p-8



X-axis denotes variation of j from 0 to required value. Y-axis denotes mean value of modularity based on partitioning the desired graph into 2 communities. The mean is calculated for 100 such iterations of the graph for each specific combination of p and j. The error bar denotes the standard deviation in such circumstances.

These are graphs showing varying probability in x axis and modularity value in y axis. The modularity values are for specific randomly generated graphs . The x label is 'p' with a number denoting one of the attributes of the randomly generated graph. The graphs are drawn in a bipartite setting, with a set of 5 vertices each, but they also have connections between them. For the number accompanying 'p' , which if considered as 'i' , the density of edges between the two sets of nodes are (1-(0.1*(i+1))). The x-axis in each such graph varies from 0 to 0.9. They denote the density of connections in each individual sets of 5 nodes each. For x-axis value denoted as 'j' , the density of edges in them is (1-(j*(0.1*(i+1)))). Hence under all circumstances these graphs have two sets of nodes more densely connected than the connections in between them. Also the density of connections in between the set of nodes decreases further along the x-axis. The maximum modularity in case of 2 communities is plotted for such graphs along y-axis. The plotted graph shows the mean for 100 such simulations, with the error bars showing their standard deviations. The curvature of the graph is also analogous to the previous demand stated .

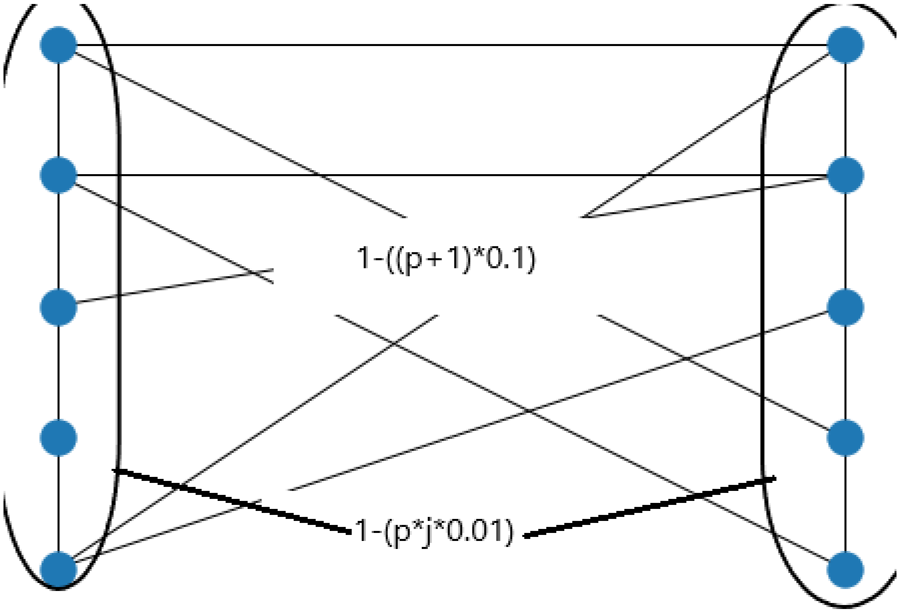
The previous stated condition is now analyzed.

1. dIntraC = 1-(p*j*0.01)
2. dInterE = 1-((p+1)*.1)

dIntraC-dInterE
= {1-(p*j*0.01)}-{1-((p+1)*.1)}
= ((p+1)*.1) –(p*j*0.01)
= p*0.01*(10-j) + 0.1 ....[j varies from 0 to 9] .....[7]

The evaluated term [p*0.01*(10-j)] is considered ‘jj’ . As ‘jj’ is always positive considering it a product of positive numbers , the above sum is always positive. Also in this we can observe that ‘jj’ increases with higher values of ‘p’ and decreases with higher values of ‘j’ . Based on equation [7] we can conclude that the requirement of equation [6] is satisfied.

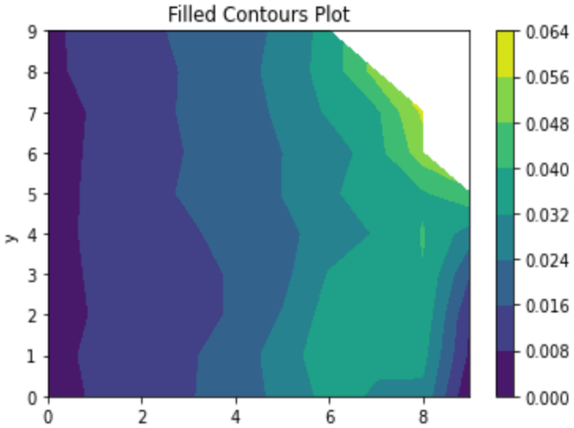
Fig. 9.



The 5 vertices encapsulated in the left are vertices {0,1,2,3,4} and the 5 vertices encapsulated in right are vertices {5,6,7,8,9} . The density of edges inside the communities and in-between them are shown accordingly.

This above graph shows a pictorial representation of how the density of the edges in various parts of the graph is shown.

Fig. 10.



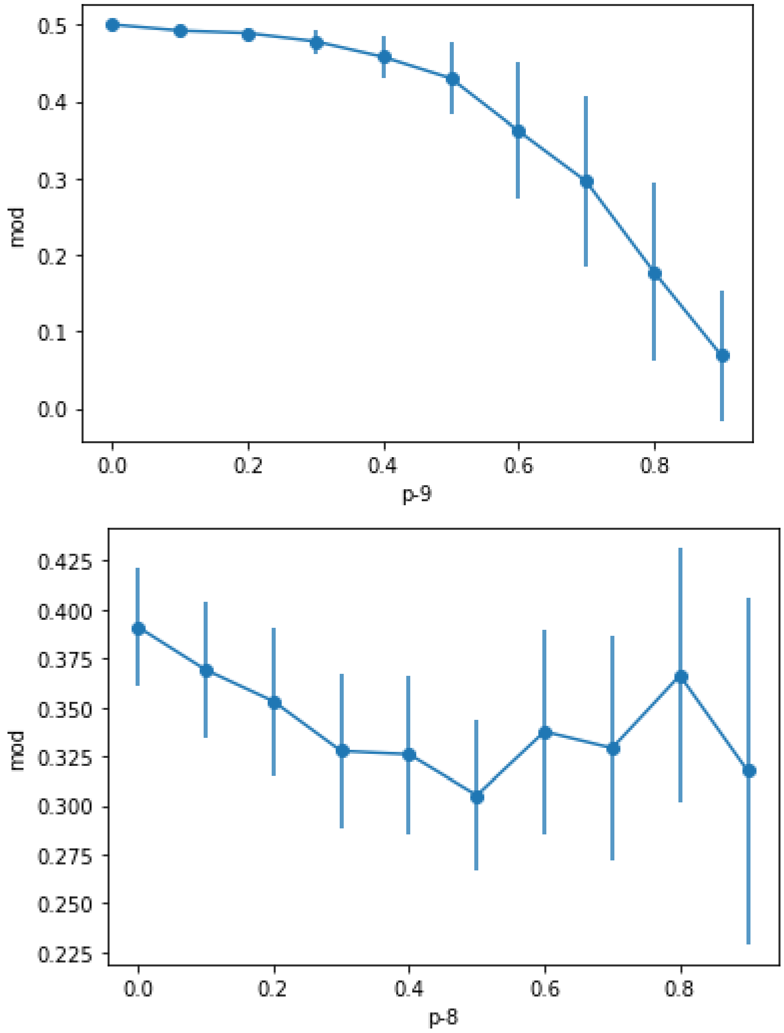
*The value along x-axis shows the value of p. The value along y-axis shows the value of j*10. The average value of modularity is plotted for such conditions. The mean is calculated for 100 such iterations of the graph for each specific combination of p and j. In case in any repetition , the graph has no edges no value is accepted for modularity.*

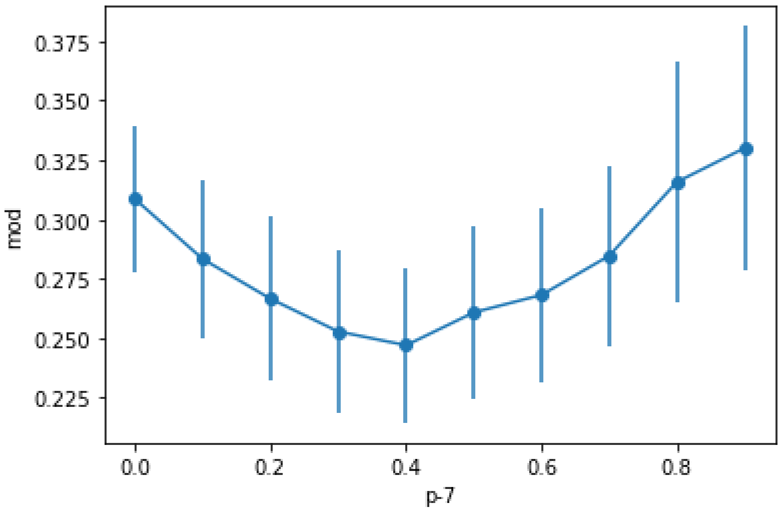
This is a contour graph of the previous example put together.

At p=9 due to the constraints put in place the graph becomes disconnected and hence is completely separate and have visually distinct two communities . Due to the heavy load of comparing all the 100 possible variations put in place while comparing Newman clustering, it has been only compared them for one such variation. In this algorithm the graph has both inhibitory connection and excitatory connection. They are given a unit in voltage with relation to the threshold value of the equation. The input due to spiking of the preceeding neuron to the proceeding neuron is varied in integral multiples of that unit from zero to ten . Without taking into consideration the variation where the input due to spiking for both excitatory connection and inhibitory connection is zero, there are ((11*11)-1)= 120 distance matrices for each graph.

In the contour graph the combination plot of the above ten graphs are made. Along x-axis is the probability of bipartite connection in the complement graph or ‘p’ as stated previously. The value of density is actually ‘(p+1)*0.1’ . In the complementary graph the density of connection in the predetermined communities is ‘p*j*0.01’ where j varies from zero to nine . The value of ‘j’ is plotted along y axis. We can see that in the contour graph . In these graphs whenever a null graph is generated by the graph generator , the modularity of that graph is set to NaN or not a number. That is why blank space in the contour graph has been observed and arbitary termination along x-axis for ‘p’=7,8,9. When the modularity for those situation is revised and set to zero, the modifications that take place are shown below. These two terms ‘p’ and ‘j’ are taken as standard for further explanation of similar situation down the line.

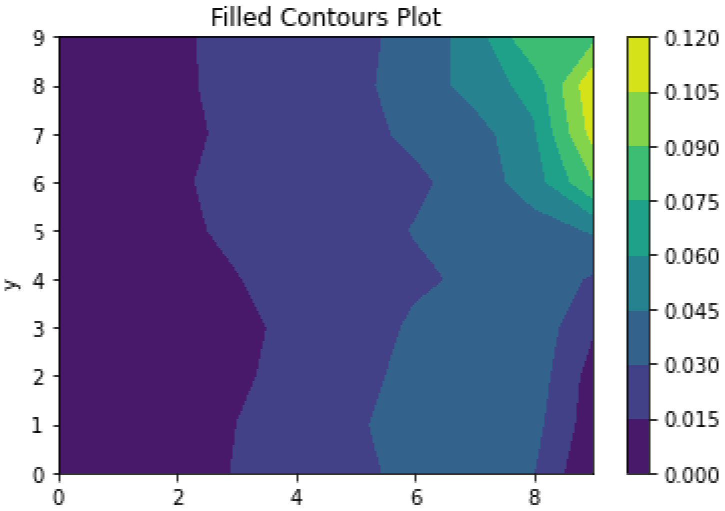
Fig. 11.


0.5
0.4
0.3
0.2
0.1
0.0
mod
0.0
0.2
0.4
0.6
0.8
p-9
0.425
0.400
0.375
0.350
0.325
0.300
0.275
0.250
0.225
mod
0.0
0.2
0.4
0.6
0.8
p-8



X-axis denotes variation of j from 0 to required value. Y-axis denotes value of modularity based on partitioning the desired graph into 2 communities. . The mean is calculated for 100 such iterations of the graph for each specific combination of p and j. In case in any repetition , the graph has no edges 0 is accepted for modularity.

Fig. 12.



*The value along x-axis shows the value of p. The value along y-axis shows the value of j*10. The average value of modularity is plotted for such conditions. The mean is calculated for 100 such iterations of the graph for each specific combination of p and j. In case in any repetition , the graph has no edges 0 is accepted for modularity.*

Due to the effect of modularity = 0 on the new revised calculations new variations in them are seen. In these settings the target has been made to already create graphs with two communities ingrained in them . Moreover of the 10 vertices the set of vertices {0,1,2,3,4} and {5,6,7,8,9} should form the two relevant communities. In order to achieve this the graphs are constructed such that the density of connection of the edges in the two communities is always higher than the connection between them.

Now the output of this is compared to the output generated by the process that has been done . Due to ease of comparison , Synaptic clustering process is applied only on one variation of this graph set. So only on the first set of graphs generated , this method of clustering is applied and compared with the results achieved from this one. However to achieve a comparison in between the both of them a new measure of similarity is considered.. For each combination of excitatory connection to inhibitory connection , a k-means clustering for two clusters is done. Suppose those two clusters are A and B. First A is matched with {0,1,2,3,4} and B is matched with {5,6,7,8,9}. The number of elements common in both A and {0,1,2,3,4} is calculated. It is similarly done for B and {5,6,7,8,9}. The sum of these two values is measured. This value is noted as S1 .Then the reverse is performed . So similarly A is matched with {5,6,7,8,9} and B is matched with {0,1,2,3,4} . The value measured from this is noted as S2. The maximum of S1 and S2 is calculated. Without loss of generality let it be S1. Now the error is denoted as 10-S1. If A is either of {0,1,2,3,4} or {5,6,7,8,9} the outcome is perfectly in line with the predicted outcome. Now the error matrix generated by this due to the first set of generated matrices under Newman Clustering[1] is denoted below.

```
array([[0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 1., 3., 4., 4., 4., 4.],
       [0., 0., 0., 5., 1., 2., 1., 2., 4., 4.],
       [0., 0., 0., 0., 0., 3., 5., 3., 4., 5.],
       [0., 0., 0., 5., 4., 4., 4., 5., 2., 5.],
       [0., 0., 0., 0., 1., 3., 5., 2., 4., 3.],
       [0., 0., 0., 0., 3., 2., 4., 5., 4., 5.],
       [0., 0., 0., 0., 1., 2., 1., 5., 5., 2.],
       [0., 0., 1., 1., 1., 5., 4., 5., 2., 4.],
       [0., 0., 0., 1., 2., 1., 1., 3., 4., 5.]])
```

Along the rows ‘p’ is varied . Along the columns ‘j’ is varied . These variables are denoted beforehand for explanation. The density of connection in between the communities decreases as p increases. Also the communities are more densely packed as j decreases. In accordance with this the error at the lower left end of the matrix is zero or negligible. Now for this the minimum error of all the 120 different distance matrices for synaptic clustering is checked and shown below.

```
array([[[4., 4., 4., 4., 4., 4., 4., 4., 4., 4.],
        [4., 4., 4., 4., 4., 3., 4., 3., 4., 2.],
        [1., 1., 1., 1., 1., 1., 1., 1., 1., 2.],
        [2., 2., 2., 2., 2., 2., 2., 2., 2., 2.],
        [1., 1., 1., 1., 1., 2., 1., 1., 2., 1.],
        [1., 1., 1., 1., 1., 1., 2., 2., 1., 1.],
        [2., 2., 2., 2., 2., 1., 2., 1., 2., 2.],
        [0., 0., 1., 1., 0., 1., 0., 0., 1., 0.],
        [1., 1., 1., 2., 2., 1., 1., 0., 0., 2.],
        [0., 0., 0., 1., 1., 1., 1., 1., 1., 1.]]])
```

Now we can compare the matrices in both cases.

- It is obserevable from these two matrices that Synaptic clustering is able to resolve the communities to a reasonable degree.
- As p value increases the overall density of the graph increases. Under such circumstances the resolution is not as clear as hoped . The segregation of communities is caused by the inhibitory network. When the network is very dense, the effect of excitatory connection dominates . This gives rise to more error.
- On the flip side for sparse networks, under circumstances it is able to outperform the Newman clustering.

In the above cases whether the graphs are connected are not taken into consideration . Even if the graph is disconnected as long as less than two components are present, the comparison is appropriate. Hence a lot of the favourable results are due to wrong interpretation. When all the graphs have less than two components , the results are taken again. The updated error values for Newman clustering is now compared.

```
array([[0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 1., 3., 4., 4., 4., 4.],
       [0., 0., 0., 5., 1., 2., 1., 2., 4., 4.],
       [0., 0., 0., 0., 0., 3., 5., 3., 4., 5.],
       [0., 0., 0., 5., 4., 4., 4., 5., 2., 5.],
       [0., 0., 0., 0., 1., 3., 5., 2., 4., 3.],
       [0., 0., 0., 0., 3., 2., 4., 5., 4., 5.],
       [0., 0., 0., 0., 1., 2., 1., 5., 5., 5.],
       [0., 0., 1., 1., 1., 5., 5., 5., 5., 5.],
       [0., 0., 0., 5., 5., 5., 5., 5., 5., 5.]])
```

The error values which have been changed to 5 from their previous value is due to an increase in communities more than 2.

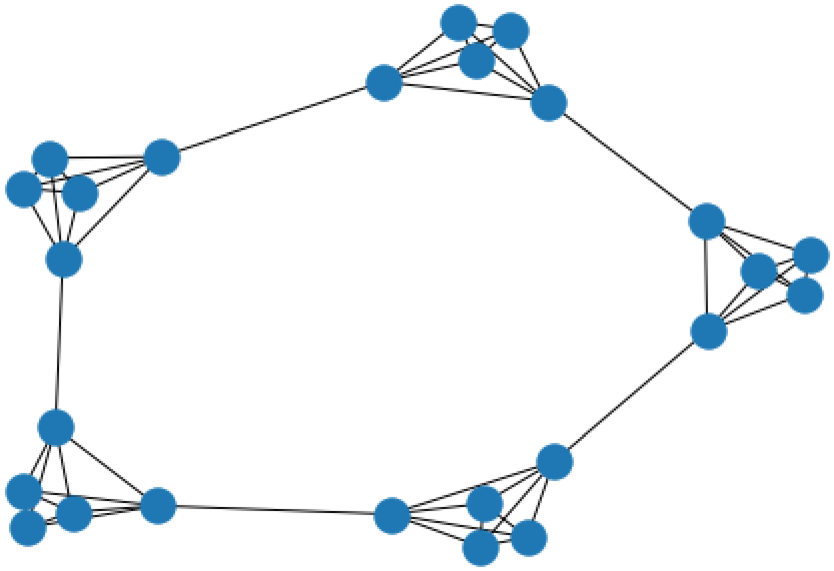
- For p=1 and j =7,9 Synaptic clustering is able to provide lesser error than Newman clustering.
- For p=2 and j>3 Synaptic clustering is able to provide lesser error than Newman clustering.
- For p=3 and j>=5 Synaptic clustering is able to provide lesser error than Newman clustering.
- For p=4 and j>4 Synaptic clustering is able to provide lesser error than Newman clustering.
- For p=5 and j>3 Synaptic clustering is able to provide lesser error than Newman clustering.

- For p=6 and j>3 Synaptic clustering is able to provide lesser error than Newman clustering.
- For p=7 and j=9 graph has more than 2 components.
- For p=8 and j>5 graph has more than 2 components.
- For p=9 and j>2 graph has more than 2 components.
- For p=7 and j=4,6,7,8 Synaptic clustering is able to provide lesser error than Newman clustering.
- For p=8 Synaptic clustering is not able to outperform Newman clustering, however the error due to synaptic clustering is not significantly higher. For j=3 their errors are the same and for other relevant values of j , the error only varies by 1.
- For p=9 due to the structure of the graph generators , the graph anyway has more than one connected component. Under such conditions only for j=0,1,2 they have exactly 2 connected components {0,1,2,3,4} and {5,6,7,8,9}. Both the clustering algorithms are able to resolve such situation equally well.
- Now it is noticed in cases for higher values of j Synaptic clustering is resolving communities more clearly. As claimed before the segregation of communities is caused by the inhibitory network. The density of edges decreases in a graph as j increases. However in such case there should be absolutely no need in even considering an excitatory network. However without external influence each neuron periodically spikes due to the equation that controls it. The distance metric is measured as an integral of spike distance at each point of time. The effect of only inhibitory stimulus will cause spikes in different communities to come closer at atleast some point of time. However if it was connected with an excitatory connection which stimulated it beforehand, it’s spiking will happen a little sooner that what would have happened only under inhibitory connection. Basically excitatory connection is provided to cause the elements of the community to be closer to each other.
- All the above comparisons are made under the assumption that the pre-determined communities are truly the actual communities in the graph. Although the graphs are generated with a certain condition, the actual communities formed might turn out to be something different.

Case 2 :

Now let’s take another example into consideration. The graph which is shown below will be used for it. It is denoted as ‘Necklace graph’ for convinience.

Fig. 13.



5 complete graphs each consisting of 5 vertices each are attached in the form of a necklace.

Here we can see that the graph has five communities. First a complete graph of five vertices are taken . Five of them are considered. Now considering each of them as a bead of a string, they are attached like a necklace. Each complete subgraph is composed of set of vertices {0,1,2,3,4},{5,6,7,8,9},{10,11,12,13,14},{15,16,17,18,19} and {20,21,22,23,24} . Each such set of vertices has only 2 edges coming out of it. They are connected to a different such set of vertices each. These communities are denoted as pre-resolved communities.

The communities should be accordingly resolved. Now by k-means clustering, setting the required clusters to five the resolved communities are checked.

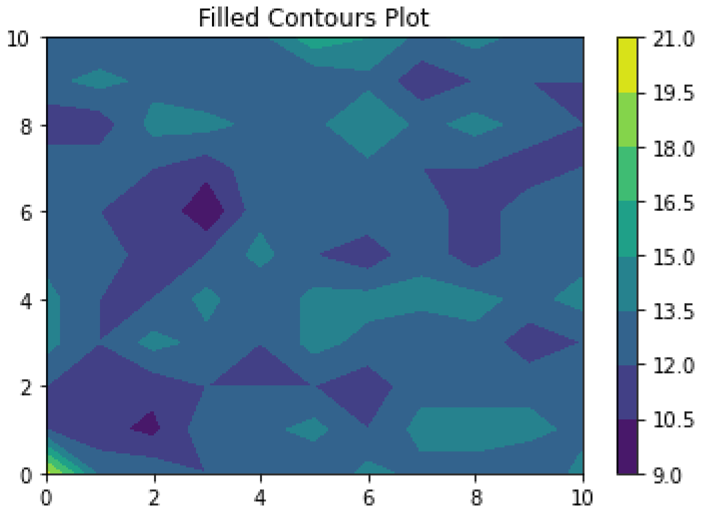
The communities are resolved for each ratio of inhibitory stimulus to excitatory stimulus. The value of excitatory stimulus is varied from 0 unit to 10 units. Similarly the value of inhibitory stimulus is varied from 0 unit to 10 units. This gives rise to {11*11}=121 different community clusters. From them the original set of nodes which should form communities is already known beforehand. So an error matrix is resolved from it. The error matrix is derived from decision trees. The communities are numbered as [0th ,1st ,2nd ,3rd ,4th]. The original communities are known beforehand. First from the set of original communities it is analyzed which of those communities has most number of elements of one community derived from Synaptic clustering. Let {20,21,22,23,24} has 4 elements of 2nd community. No other pre-resolved community has more than 3 elements from one community. Now 4 is subtracted from 25 and the algorithm is run again without the elements of 2nd community being considered . If now two or more pre-resolved communities have the same value for most number of elements , all of them are considered and the algorithm is run parallely. Finally the outcome which provides with the least value is taken as error.

The error matrix documented is shown below.
Along the rows the value of excitation is varied from 0 to 10 units.

Along the columns the value of excitation is varied from 0 to 10 units.

```
[[20.0, 12.0, 12.0, 14.0, 14.0, 13.0, 12.0, 13.0, 11.0, 13.0, 13.0],
 [13.0, 11.0, 12.0, 12.0, 12.0, 13.0, 12.0, 13.0, 11.0, 14.0, 12.0],
 [13.0, 10.0, 11.0, 14.0, 12.0, 11.0, 12.0, 12.0, 14.0, 13.0, 12.0],
 [12.0, 13.0, 12.0, 13.0, 14.0, 12.0, 9.0, 11.0, 14.0, 12.0, 13.0],
 [12.0, 13.0, 12.0, 12.0, 12.0, 14.0, 13.0, 13.0, 13.0, 12.0, 13.0],
 [12.0, 14.0, 12.0, 14.0, 14.0, 12.0, 13.0, 13.0, 13.0, 12.0, 16.0],
 [14.0, 12.0, 11.0, 13.0, 14.0, 11.0, 13.0, 13.0, 15.0, 13.0, 15.0],
 [13.0, 14.0, 13.0, 12.0, 14.0, 13.0, 13.0, 12.0, 13.0, 11.0, 13.0],
 [13.0, 14.0, 13.0, 13.0, 14.0, 11.0, 11.0, 12.0, 14.0, 12.0, 14.0],
 [12.0, 14.0, 13.0, 11.0, 13.0, 13.0, 13.0, 11.0, 13.0, 12.0, 13.0],
 [14.0, 13.0, 13.0, 12.0, 14.0, 13.0, 13.0, 12.0, 12.0, 12.0, 13.0]]
```

Fig. 12.



Along x-axis the value of excitation is varied from value 0 to 10 units and along y-axis the value of inhibition is varied from value 0 to 10. The error obtained from such combinations is plotted in the graph for the 'Necklace graph.'

First for no excitatory connection , the value of inhibitory stimulus in connection is varied from 0 unit to 10. The clustered community matrix resolved from it is now shown.

The value of inhibition varies from 0 units to 10 units along the rows and this pattern is maintained for analysis beyond this matrix too. In this matrix and all the matrices following this the pre-resolved communities {0.1.2.3.4}, {5,6,7,8,9},{10,11,12,13,14},{15,16,17,18,19} and {20,21,22,23,24} are denoted individually under first brackets separately .

```
([[[0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
    0., 0., 0., 0., 0., 0., 0., 0., 0.],
   [(4., 1., 0., 3., 2.,)( 4., 4., 2., 2., 4.,)( 3., 0., 3., 3., 0.,)( 2.,
    3., 4., 2., 4.,)( 1., 1., 1., 1., 0.)],
   [(1., 2., 3., 0., 3., )(1., 3., 3., 4., 0., )(0., 4., 1., 0., 1., )(3.,
    0., 3., 1., 0., )(2., 2., 2., 2., 0.)],
   [(1., 3., 0., 2., 2., )(1., 2., 4., 0., 2., )(1., 0., 0., 4., 4.,)( 2.,
    2., 4., 0., 0., )(3., 3., 3., 3., 1.)],
   [(1., 3., 0., 4., 2., )(2., 0., 4., 1., 0.,)( 0., 4., 4., 2., 2., )(0.,
    0., 1., 2., 2.,)( 3., 3., 3., 3., 1.)],
   [(1., 2., 4., 4., 3., )(3., 3., 4., 1., 4., )(1., 2., 1., 2., 0., )(2.,
    0., 3., 0., 3.,)( 2., 2., 2., 2., 1.)],
   [(0., 2., 1., 4., 0., )(0., 0., 1., 0., 1.,)( 4., 0., 0., 1., 0., )(4.,
    3., 3., 0., 3.,)( 2., 2., 2., 2., 4.)],
   [(0., 3., 2., 1., 1., )(1., 1., 1., 0., 4.,)( 4., 0., 1., 0., 4.,)( 0.,
    2., 2., 4., 4.,)( 3., 3., 3., 3., 2.)],
   [(1., 3., 1., 1., 0., )(1., 2., 1., 4., 0.,)( 4., 0., 2., 2., 1., )(2.,
    2., 2., 2., 4., )(3., 3., 3., 3., 1.)],
   [(3., 2., 4., 3., 3., )(1., 1., 1., 2., 3.,)( 1., 3., 2., 1., 3., )(1.,
    0., 2., 1., 0.,)( 2., 2., 2., 2., 2.)],
   [(0., 1., 4., 4., 0., )(4., 0., 4., 0., 4., )(1., 4., 3., 3., 2., )(3.,
    1., 4., 3., 2., )(1., 1., 1., 1., 3.)]]])
```

Conclusions from excitatory connection of 0 units.

- The first community separation has no connection and hence no communities.
- In all the community separations the community {20,21,22,23,24} has atleast 4 elements from the same community. For inhibition at 9 units infact it is completely of elements of 2nd community.
- For inhibition at 8 units {0,1,2,3,4} has 3 elements of 1st community, {15,16,17,18,19} has 4 elements of 2nd community and 4 elements of 3rd community is in {20,21,22,23,24} .
- Other community separations does not provide better significant results.

For for 1 unit of excitatory connection , the value of inhibitory stimulus in connection is varied from 0 unit to 10. The clustered community matrix resolved from it is now analysed.

```
([[[(3., 1., 0., 2., 0.,)( 4., 3., 0., 0., 3.,)( 2., 3., 0., 3., 4.,)( 0.,
  3., 2., 2., 2., )(1., 1., 1., 1., 0.)],
[(3., 1., 0., 0., 0., )(3., 3., 4., 3., 2.,)( 2., 2., 4., 2., 0., )(0.,
  3., 4., 4., 2., )(1., 1., 1., 1., 4.)],
[(1., 2., 0., 1., 1., )(3., 0., 0., 3., 4., )(3., 1., 3., 1., 3.,)( 0.,
  4., 1., 3., 1.,)( 2., 2., 2., 2., 0.)],
 [(3., 2., 0., 0., 3.,)( 0., 1., 4., 3., 3., )(4., 3., 1., 1., 3., )(0.,
  0., 0., 0., 3., )(2., 2., 2., 2., 1.)],
 [(3., 0., 3., 4., 3., )(3., 1., 1., 3., 2., )(1., 1., 2., 2., 2., )(3.,
  2., 4., 1., 2., )(0., 0., 0., 0., 2.)],
 [(2., 3., 0., 1., 1.,)( 2., 0., 0., 1., 2.,)( 1., 1., 4., 2., 2., )(1.,
  0., 4., 4., 2., )(3., 3., 3., 3., 4.)],
 [(3., 2., 4., 4., 1.,)( 0., 3., 3., 3., 0., )(4., 3., 1., 0., 3., )(1.,
  1., 0., 1., 0.,)( 2., 2., 2., 2., 1.)],
 [(4., 2., 3., 4., 1.,)( 0., 1., 1., 3., 3.,)( 3., 1., 4., 4., 3.,)( 1.,
  1., 3., 3., 1.,)( 2., 2., 2., 2., 0.)],
 [(3., 2., 1., 1., 1.,)( 4., 1., 3., 3., 3., )(0., 1., 0., 3., 0., )(2.,
  0., 0., 4., 1., )(2., 2., 2., 2., 4.)],
 [(4., 2., 4., 1., 0.,)( 3., 0., 4., 1., 0.,)( 1., 1., 3., 1., 3.,)( 0.,
  4., 4., 1., 1., )(2., 2., 2., 2., 0.)],
 [(4., 1., 0., 4., 3.,)( 0., 3., 3., 0., 3.,)( 3., 0., 4., 3., 2., )(0.,
  4., 2., 3., 2., )(1., 1., 1., 1., 3.)]]]])
```

Conclusions from excitatory connection of 1 units.

- Even in this case the vertices {20,21,22,23,24} seem to have 4 elements of same community . Specially {20,21,22,23} is always of the same community.
- For inhibition of 10 units {5,6,7,8,9} has 3 elements of 3rd community too.
- For inhibition of 9 units {10,11,12,13,14} has 3 elements of 1st community.
- For inhibition of 8 units {0,1,2,3,4} has 3 elements of 1st community, {5,6,7,8,9} has 3 elements of 3rd community,{15,16,17,18,19} has 3 elements of 0th community and {20,21,22,23,24} already has 4 elements of 2nd community.
- For inhibition of 6 units {5,6,7,8,9} has 3 elements of 3rd community and {15,16,17,18,19} has 3 elements of 1st community too.
- For inhibition of 3 units in {15,16,17,18,19} , {15,16,17,18} is of 0th community.
- For inhibition of 1 unit {0,1,2,3,4} has 3 elements of 0th community, {5,6,7,8,9} has 3 elements of 3rd community and {10,11,12,13,14} has 3 elements of 2nd community.
- Rest of the separations does not provide improved results.

For 2 unit of excitatory connection , the value of inhibitory stimulus in connection is varied from 0 unit to 10. The clustered community matrix resolved from it is now analysed.

```
([[[(2., 1., 0., 3., 2.,)( 0., 2., 2., 0., 2.,)( 0., 3., 0., 4., 2., )(2.,
  3., 2., 2., 2.,)( 1., 1., 1., 1., 0.)],
 [(2., 1., 4., 3., 3., )(4., 4., 2., 4., 0.,)( 3., 0., 2., 3., 2.,)( 0.,
  0., 0., 2., 0.,)( 1., 1., 1., 1., 3.)],
 [(0., 2., 1., 1., 1.,)( 0., 1., 0., 0., 0.,)( 3., 4., 4., 3., 1.,)( 1.,
  4., 3., 1., 0., )(2., 2., 2., 2., 4.)],
 [(3., 1., 2., 4., 0.,)( 0., 0., 3., 4., 4.,)( 2., 0., 3., 0., 2., )(4.,
  2., 0., 2., 3., )(1., 1., 1., 1., 4.)],
 [(0., 3., 2., 1., 0.,)( 0., 2., 2., 0., 0., )(4., 4., 4., 1., 1., )(4.,
  1., 0., 2., 2.,)( 3., 3., 3., 3., 1.)],
 [(4., 3., 2., 2., 2.,)( 1., 2., 0., 4., 4., )(0., 0., 2., 1., 0., )(4.,
  0., 1., 1., 4., )(3., 3., 3., 3., 2.)],
 [(0., 3., 4., 2., 4., )(2., 0., 4., 1., 2., )(1., 1., 0., 1., 1.,)( 1.,
  1., 1., 4., 1.,)( 3., 3., 3., 3., 4.)],
 [(4., 0., 2., 1., 4.,)( 1., 3., 2., 3., 3., )(1., 2., 2., 3., 4., )(3.,
  1., 2., 1., 3.,)( 0., 0., 0., 0., 1.)],
 [(0., 2., 4., 3., 1., )(3., 0., 4., 0., 4.,)( 4., 3., 1., 3., 1.,)( 1.,
  1., 0., 0., 4., )(2., 2., 2., 2., 1.)],
 [(2., 3., 1., 1., 0.,)( 0., 1., 2., 1., 4.,)( 4., 1., 1., 1., 2., )(4.,
  4., 2., 1., 4.,)( 3., 3., 3., 3., 4.)],
 [(3., 2., 4., 1., 0.,)( 1., 0., 4., 4., 1.,)( 3., 1., 3., 0., 3., )(1.,
  0., 0., 0., 3.,)( 2., 2., 2., 2., 3.)]]]])
```

Conclusions from excitatory connection of 2 units.

- Again we observe {20,21,22,23} is in one community in all segregations.
- For 0 unit of inhibition {5,6,7,8,9} there are 3 elements of 2nd community and {15,16,17,18,19} has 4 elements of 2nd community too.
- For 1 unit of inhibition {5,6,7,8,9} there are 3 elements of 4th community and {15,16,17,18,19} has 4 elements of 0th community too.
- For 2 units of inhibition {0,1,2,3,4} has 3 elements of 1st community and {5,6,7,8,9} has 4 elements of 0th community also.
- For 6 units of inhibition {15,16,17,18,19} and {10,11,12,13,14} has 4 elements of 1st community each also .

For 3 unit of excitatory connection , the value of inhibitory stimulus in connection is varied from 0 unit to 10. The clustered community matrix resolved from it is now analysed.

```
([[[(3., 1., 4., 3., 4., )(0., 3., 4., 0., 4., )(3., 3., 4., 3., 4., )(0.,
  0., 4., 0., 3.,)( 2., 2., 2., 2., 3.)],
 [(1., 2., 3., 4., 3.,)( 0., 0., 0., 1., 1., )(3., 3., 0., 1., 4., )(0.,
  3., 1., 0., 1., )(2., 2., 2., 2., 3.)],
 [(3., 2., 3., 3., 1., )(1., 0., 0., 0., 0.,)( 3., 0., 0., 1., 1., )(1.,
  3., 1., 1., 0.,)( 2., 2., 4., 2., 3.)],
 [(1., 0., 1., 4., 4.,)( 1., 3., 1., 1., 3.,)( 3., 1., 1., 2., 1., )(1.,
  4., 3., 4., 2., )(0., 0., 0., 0., 3.)],
 [(0., 2., 1., 4., 1.,)( 0., 1., 4., 1., 3.,)( 0., 1., 0., 1., 4., )(1.,
  3., 1., 4., 3., )(2., 2., 2., 2., 4.)],
 [(1., 3., 2., 4., 2.,)( 0., 2., 0., 0., 1.,)(2., 2., 4., 1., 4., )(0.,
  4., 1., 1., 2.,)( 3., 3., 3., 3., 4.)],
 [(3., 4., 3., 0., 3.,)( 3., 2., 3., 0., 0., )(1., 1., 1., 2., 2., )(2.,
  2., 2., 2., 0.,)( 4., 4., 4., 4., 1.)],
 [(2., 3., 0., 2., 4.,)( 2., 2., 2., 2., 0.,)( 1., 2., 0., 1., 4., )(1.,
  4., 0., 4., 4., )(3., 3., 3., 3., 2.)],
 [(3., 0., 2., 2., 1.,)( 4., 2., 3., 2., 1., )(1., 2., 2., 3., 1., )(3.,
  1., 3., 4., 2., )(0., 0., 0., 0., 4.)],
 [(1., 0., 2., 1., 2., )(2., 4., 4., 2., 2.,)( 1., 1., 4., 4., 4., )(2.,
  1., 2., 1., 3., )(0., 0., 0., 0., 1.)],
 [(2., 0., 4., 1., 3.,)( 2., 3., 2., 3., 3., )(3., 2., 4., 2., 4.,)( 1.,
  4., 1., 1., 2., )(0., 0., 0., 0., 4.)]]]])
```

Conclusions from excitatory connection of 3 units.

- For inhibition of unit 2 {20,21,22,23,24} has 3 elements of 2nd community. Sor all other clusterings {2,21,22,23,24} has 4 elements of one community.
- For inhibition of unit 0 {15,16,17,18,19} has 3 elements of 0th community , {10,11,12,13,14} has 3 elements of 3rd community and {2,21,22,23,24} has 4 elements of 2nd community.
- For inhibition of unit 2 {5,6,7,8,9} has 4 elements of 0th community and {11,12,13,14,10} has 3 elements of first community.
- For inhibition of unit 3 {5,6,7,8,9} and {10,11,12,13,14} has 3 elements of first community each also .
- For inhibition of unit 6{15,16,17,18,19} has 4 elements of 2nd community{10,11,12,13,14 } has 3 elements of 1st community.
- For inhibition of unit 7 {5,6,7,8,9} has 4 elements of 2nd community , {15,16,17,18,19} has 3 elements of 4th community.
- For inhibition of unit 9 {5,6,7,8,9} has 3 elements of 2nd community , {10,11,12,13,14} has 3 elements of 4th community also.

For 4 unit of excitatory connection , the value of inhibitory stimulus in connection is varied from 0 unit to 10. The clustered community matrix resolved from it is now analysed.

```
([[[(3., 1., 0., 0., 3., )(3., 2., 2., 2., 0., )(0., 2., 4., 3., 4., )(3.,
  4., 3., 0., 0., )(1., 1., 1., 1., 4.)],
 [(0., 1., 2., 4., 3.,)( 0., 4., 2., 3., 2.,)( 2., 2., 0., 4., 2.,)( 3.,
  0., 2., 3., 3., )(1., 1., 1., 1., 3.)],
 [(1., 2., 0., 0., 0.,)( 4., 3., 1., 0., 1.,)( 0., 0., 0., 4., 4.,)( 1.,
  1., 0., 1., 3.,)( 2., 2., 2., 2., 3.)],
 [(1., 3., 0., 4., 0., )(1., 0., 1., 1., 0.,)( 2., 2., 2., 1., 1.,)( 0.,
  2., 4., 2., 1.,)( 3., 3., 3., 3., 0.)],
 [(3., 4., 2., 3., 0., )(1., 3., 1., 3., 0., )(3., 2., 2., 1., 2.,)( 1.,
  0., 3., 2., 0.,)( 4., 4., 4., 4., 0.)],
 [(0., 3., 1., 3., 1., )(0., 4., 2., 2., 1.,)( 0., 2., 4., 0., 2.,)( 0.,
  1., 4., 2., 0., )(3., 3., 3., 3., 2.)],
 [(3., 2., 0., 0., 0.,)( 3., 0., 1., 1., 1., )(0., 1., 0., 3., 0.,)( 3.,
  3., 1., 4., 0., )(2., 2., 2., 2., 3.)],
 [(1., 2., 3., 3., 1., )(0., 1., 3., 3., 4., )(1., 3., 0., 4., 0., )(4.,
  3., 4., 3., 0.,)( 2., 2., 2., 2., 4.)],
 [(3., 4., 2., 1., 0.,)( 3., 1., 3., 2., 2., )(3., 3., 1., 3., 2., )(1.,
  1., 2., 3., 2., )(4., 4., 4., 4., 2.)],
 [(0., 3., 2., 4., 2.,)( 0., 3., 1., 1., 3., )(2., 1., 2., 1., 2., )(3.,
  2., 0., 0., 0., )(3., 3., 3., 3., 1.)],
 [(3., 0., 2., 4., 3., )(4., 1., 1., 3., 3.,)( 3., 3., 1., 2., 1., )(3.,
  2., 4., 2., 2., )(0., 0., 0., 0., 2.)]]]])
```

Conclusions from excitatory connection of 4 units.

- Once again we observe that {20,21,22,23} is clustered in one community in all of them.
- For inhibition of unit 1 {10,11,12,13,14} has 3 elements of 2nd community and {15,16,17,18,19} has 3 elements of 3rd community too.
- For inhibition of unit 2 we see a good match. {0,1,2,3,4} and {10,11,12,13,14} has 3 elements of 0th community each, {15,16,17,18,19} has 3 elements of 1st community.
- For inhibition of unit 3 {5,6,7,8,9} has 3 elements of 1st community , {10,11,1,13,14} has 3 elements of 2nd community too.
- For inhibition of unit 6 {0,1,2,3,4} and{10,11,12,13,14} has 3 elements of 0th community each and {5,6,7,8,9} has 3 elements of 1st community too.
- For inhibition of unit 9 {10,11,12,13,14} has 3 elements of 2nd community and {15,16,17,18,19} has 3 elements of 0th community also.

For 5 unit of excitatory connection , the value of inhibitory stimulus in connection is varied from 0 unit to 10. The clustered community matrix resolved from it is now analysed.

```
([[[(0., 1., 2., 0., 4., )(0., 2., 2., 0., 3.,)( 0., 0., 0., 3., 0., )(2.,
  4., 4., 4., 3.,)( 1., 1., 1., 1., 2.)],
 [(4., 1., 2., 2., 0.,)( 3., 4., 0., 3., 0.,)( 2., 3., 4., 2., 0., )(2.,
  3., 4., 2., 4., )(1., 1., 1., 1., 3.)],
 [(0., 2., 4., 4., 4., )(1., 0., 3., 1., 1., )(1., 3., 4., 3., 0., )(3.,
  3., 4., 1., 0., )(2., 2., 2., 2., 1.)],
 [(2., 3., 0., 1., 0., )(0., 2., 0., 4., 0., )(2., 1., 1., 0., 2., )(0.,
  0., 1., 1., 0., )(3., 3., 3., 3., 0.)],
 [(0., 2., 1., 3., 1.,)( 4., 4., 0., 0., 1., )(1., 3., 4., 0., 1.,)( 0.,
  4., 3., 4., 1., )(2., 2., 2., 2., 4.)],
 [(4., 3., 0., 1., 1., )(4., 0., 1., 2., 4.,)( 2., 2., 2., 4., 2., )(1.,
  1., 4., 4., 2., )(3., 3., 3., 3., 4.)],
 [(0., 2., 1., 1., 1.,)( 3., 0., 0., 0., 3.,)( 1., 1., 0., 0., 1., )(4.,
  3., 1., 3., 0.,)( 2., 2., 2., 2., 1.)],
 [(1., 0., 4., 2., 1.,)( 3., 2., 4., 3., 3., )(4., 3., 2., 1., 2.,)( 4.,
  2., 2., 1., 1., )(0., 0., 0., 0., 2.)],
 [(1., 4., 1., 3., 1., )(3., 2., 2., 0., 3., )(2., 3., 0., 3., 1.,)( 2.,
  1., 2., 0., 3.,)( 4., 4., 4., 4., 2.)],
 [(1., 0., 3., 3., 2.,)( 3., 4., 4., 2., 1.,)( 1., 1., 2., 1., 1., )(4.,
  3., 2., 1., 3., )(0., 0., 0., 0., 2.)],
 [(4., 1., 3., 2., 0.,)( 2., 3., 0., 3., 1., )(2., 1., 0., 4., 3., )(0.,
  3., 2., 4., 3., )(1., 1., 1., 1., 3.)]]]])
```

Conclusions from excitatory connection of 5 units.

- Once again we observe that {20,21,22,23} is of same group in all clusterings.
- {10,11,12,13,14} has 4 elements of 0th community and {15,16,17,18,19} has 3 elements of 4th community.
- For inhibition of unit 6 {0,1,2,3,4} has 3 elements of 1st community, {5,6,7,8,9} has 3 elements of 0th community amd {10,11,12,13,14} has 3 elements of 1st community too.

For 6 unit of excitatory connection , the value of inhibitory stimulus in connection is varied from 0 unit to 10. The clustered community matrix resolved from it is now analysed.

```
([[[(3., 0., 2., 2., 4., )(1., 2., 3., 4., 1., )(1., 1., 4., 3., 3., )(1.
  3., 2., 2., 3.,)( 0., 0., 0., 0., 3.)],
[(4., 3., 2., 2., 0.,)( 1., 0., 0., 1., 0.,)( 1., 1., 2., 4., 1.,)( 4.,
 1., 0., 2., 2., )(3., 3., 3., 3., 1.)],
[(2., 0., 4., 2., 2.,)( 1., 1., 2., 4., 4., )(1., 3., 1., 1., 1.,)( 2.,
 4., 1., 1., 3., )(0., 0., 0., 0., 2.)],
[(2., 1., 0., 4., 3.,)( 3., 0., 3., 2., 2.,)( 2., 3., 0., 0., 3., )(3.,
 0., 3., 0., 0.,)( 1., 1., 1., 1., 2.)],
[(1., 2., 0., 1., 1.,)( 4., 3., 0., 4., 0.,)( 0., 1., 1., 0., 1., )(0.,
 1., 1., 4., 1., )(2., 2., 2., 2., 4.)],
[(0., 2., 1., 0., 0.,)( 3., 1., 0., 1., 1.,)( 0., 1., 0., 0., 1., )(3.,
 3., 3., 1., 3.,)( 2., 2., 2., 2., 4.)],
[(1., 3., 4., 2., 2.,)( 0., 0., 1., 1., 2., )(1., 0., 2., 1., 4., )(4.,
 1., 0., 0., 0.,)( 3., 3., 3., 3., 1.)],
[(2., 0., 1., 4., 2., )(1., 3., 1., 2., 4.,)( 3., 1., 3., 4., 4.,)( 2.,
 4., 1., 1., 1., )(0., 0., 0., 0., 3.)],
[(1., 2., 1., 3., 0., )(0., 0., 3., 4., 1.,)( 2., 0., 1., 1., 4.,)( 0.,
 0., 4., 3., 1., )(2., 2., 2., 2., 3.)],
[2., 0., 2., 2., 1., 1., 1., 0., 1., 3., 4., 1., 2., 2., 3., 1.,
 2., 1., 3., 0., 0., 0., 0., 0., 3.],
[(1., 4., 3., 0., 1., )(3., 1., 1., 3., 2., )(3., 3., 2., 1., 0., )(2.,
 3., 1., 1., 0., )(4., 4., 4., 4., 2.)]]]])
```

Conclusions from excitatory connection of 6 units.

- We again observe that {20,21,22,23,24} is always in one community.
- For inhibition in unit 1 {5,6,7,8,9} has 3 elements in 0th community,{10,11,12,13,14} has 3 elements of 1st community too.
- For inhibition in unit 2 {0,1,2,3,4} has 3 elements of 2nd community and {10,11,12,13,14} has 4 elements of 1st community also.
- For inhibition of unit 4 {0,1,2,3,4} and {15,16,17,18,19} has 3 elements of 1st community and also {10,11,12,13,14} has 3 elements of 1st community.
- For unit 5 of inhibition {0,1,2,3,4} and {10,11,12,13,14} has 3 elements of 0th community each. {5,6,7,8,9} has 3 elements of 1st community and {15,16,17,18,19} has 4 elements of 3rd community . Moreover {20,21,22,23,24} has 4 elements of 2nd community. This is by far the best result observed till now.
- For inhibition of unit 9 {0,1,2,3,4} has 3 elements of 2nd community , {5,6,7,8,9} has 3 elements of 1st community.

For 7 unit of excitatory connection , the value of inhibitory stimulus in connection is varied from 0 unit to 10. The clustered community matrix resolved from it is now analysed.

```
([[[(2., 1., 3., 2., 4., )(4., 0., 0., 2., 3.,)( 0., 3., 4., 2., 4.,)( 0.,
  2., 4., 2., 2., )(1., 1., 1., 1., 3.)],
 [(1., 2., 4., 0., 3.,)( 4., 1., 0., 4., 1.,)( 3., 1., 3., 0., 3.,)( 3.,
  4., 0., 3., 1., )(2., 2., 2., 2., 4.)],
 [(1., 0., 3., 1., 3.,)( 2., 4., 3., 3., 4., )(4., 1., 2., 1., 2., )(3.,
  2., 1., 4., 3., )(0., 0., 0., 0., 2.)],
 [(0., 2., 3., 0., 3.,)( 1., 0., 0., 4., 4., )(0., 3., 3., 0., 1., )(4.,
  4., 4., 4., 3., )(2., 2., 2., 2., 0.)],
 [(1., 0., 4., 2., 3.,)( 3., 1., 4., 2., 3., )(4., 4., 1., 3., 3.,)( 4.,
  2., 1., 2., 3.,)( 0., 0., 0., 0., 1.)],
 [(1., 4., 0., 0., 2.,)( 3., 3., 1., 0., 1., )(3., 2., 2., 1., 1., )(0.,
  2., 0., 2., 0.,)( 4., 4., 4., 4., 0.)],
 [(0., 3., 2., 2., 2., )(0., 1., 1., 4., 2., )(1., 1., 0., 0., 2., )(1.,
  0., 4., 1., 1., )(3., 3., 3., 3., 0.)],
 [(3., 2., 3., 3., 0., )(1., 0., 1., 0., 0.,)( 1., 1., 4., 0., 0., )(0.,
 3., 4., 1., 1., )(2., 2., 2., 2., 1.)],
[(3., 4., 0., 0., 0., )(1., 0., 0., 2., 1., )(1., 0., 3., 1., 2., )(1.,
 2., 3., 2., 1., )(4., 4., 4., 4., 2.)],
[(0., 3., 1., 1., 1.,)( 0., 1., 0., 0., 1.,)( 4., 1., 2., 2., 0., )(0.,
 1., 4., 4., 0.,)( 3., 3., 3., 3., 1.)],
[(0., 4., 2., 3., 0., )(1., 0., 1., 2., 2.,)( 1., 0., 1., 0., 0., )(1.,
 1., 0., 2., 0., )(4., 4., 4., 4., 1.)]]])
```

Conclusions from excitatory connection of 7 units.

- We again observe that {20,21,22,23,24} is always in one community.
- For inhibition of 6 units {0,1,2,3,4} has 3 elements of 2nd community, {15,16,17,18,19} has 3 elements of 1st community too.
- For inhibition of 7 units {0,1,2,3,4} has 3 elements of 3rd community, {5,6,7,8,9} has 3 elements of 0th community.
- For inhibition of 9 units {0,1,2,3,4} has 3 elements of 1st community, {5,6,7,8,9} has 3 elements of 0th community.
- For more effects of inhibition and excitation it is being observed that the communities are not being clustered more clearly. Although the ratio of them is required to get these results, the magnitude of excitation or inhibition due to action potential should have been of lesser magnitude.

For 8 unit of excitatory connection , the value of inhibitory stimulus in connection is varied from 0 unit to 10. The clustered community matrix resolved from it is now analysed

```
([[[(3., 0., 2., 2., 1.,)( 3., 4., 4., 1., 3., )(3., 4., 1., 4., 2., )(1.,
   1., 4., 4., 3., )(0., 0., 0., 0., 1.)],
 [(3., 2., 4., 1., 1., )(0., 4., 3., 0., 1., )(1., 1., 3., 3., 4., )(3.,
  1., 0., 1., 0.,)( 2., 2., 2., 2., 0.)],
 [(1., 2., 4., 0., 0., )(1., 3., 0., 1., 4.,)(3., 1., 3., 0., 3., )(3.,
  1., 4., 3., 1.,)( 2., 2., 2., 2., 1.)],
 [(3., 1., 4., 4., 4.,)( 2., 3., 0., 0., 0., )(4., 0., 0., 0., 3., )(4.,
  4., 2., 0., 4., )(1., 1., 1., 1., 4.)],
 [(3., 1., 4., 0., 4.,)( 4., 4., 3., 3., 2., )(3., 4., 2., 0., 2., )(3.,
  2., 2., 0., 4., )(1., 1., 1., 1., 0.)],
 [(0., 2., 1., 0., 3., )(0., 0., 0., 1., 4.,)( 1., 3., 0., 3., 3., )(4.,
  0., 4., 0., 4., )(2., 2., 2., 2., 1.)],
 [(1., 2., 1., 0., 1., )(3., 3., 1., 3., 4.,)(0., 0., 0., 3., 3.,)( 4.,
  3., 0., 1., 0., )(2., 2., 2., 2., 0.)],
 [(4., 1., 4., 1., 4.,)( 2., 2., 3., 4., 3.,)( 0., 2., 2., 0., 4., )(2.,
  1., 3., 4., 2., )(1., 1., 1., 1., 0.)],
 [(2., 3., 1., 4., 2., )(2., 1., 4., 1., 1., )(2., 1., 2., 0., 2., )(2.,
  3., 1., 4., 1., )(3., 3., 3., 3., 2.)],
 [(1., 2., 1., 1., 1.,)( 0., 4., 0., 0., 3.,)( 0., 1., 0., 0., 3., )(0.,
 3., 1., 4., 0.,)( 2., 2., 2., 2., 1.)],
[(0., 3., 1., 2., 1., )(2., 4., 0., 2., 0., )(2., 1., 1., 0., 0.,)( 1.,
 2., 1., 2., 1., )(3., 3., 3., 3., 0.)]]])
```

Conclusions from excitatory connection of 8 units.

- We again observe that {20,21,22,23} is always in one community.
- For inhibition of unit 3 {0,1,2,3,4} has 3 elements of 4th community,{5,6,7,8,9} and {10,11,12,13,14} has 3 elements of 0th community each.
- For inhibition of unit 5 {5,6,7,8,9} has 3 elements of 0th community, {10,11,12,13,14} has 3 elements of 3rd community, {15,16,17,18,19} has 3 elements of 4th community and as stated {20,21,22,23,24} has 4 elements of 2nd community.
- For inhibition of unit 6 {0,1,2,3,4} has 3 elements of 1st community, {5,6,7,8,9} has 3 elements of 3rd community and {10,11,12,13,14} has 3 elements of 0th community and {20,21,22,23,24} has 4 elements of 2nd community.
- For inhibition of unit 8 {5,6,7,8,9} has 3 elements of 1st community, {10,11,12,13,14} has 3 elements of 2nd community too.
- For inhibition of unit 9 {0,1,2,3,4} has 4 elements of 1st community ,{5,6,7,8,9} has 3 elements of 0th community , {10,11,12,13,14} also has 3 elements of 0th community.

Conclusions from excitatory connection of 8 units.

For 9 unit of excitatory connection , the value of inhibitory stimulus in connection is varied from 0 unit to 10. The clustered community matrix resolved from it is now analysed

```
([[[(3., 1., 3., 4., 2.,)( 4., 2., 3., 0., 0.,)( 0., 0., 2., 2., 2., )(2.,
  2., 3., 2., 2., )(1., 1., 1., 1., 3.)],
 [(4., 2., 3., 0., 1.,)( 4., 1., 3., 3., 1.,)( 0., 3., 0., 3., 3., )(0.,
  3., 0., 1., 1.,)( 2., 2., 2., 2., 4.)],
 [(1., 2., 1., 0., 3., )(1., 0., 0., 0., 4.,)( 1., 1., 4., 0., 0., )(3.,
  3., 4., 1., 0., )(2., 2., 2., 2., 3.)],
 [(0., 1., 2., 2., 3.,)( 2., 4., 4., 2., 4., )(4., 4., 2., 4., 0.,)( 2.,
  0., 0., 0., 0., )(1., 1., 1., 1., 2.)],
 [(1., 3., 0., 1., 1., )(4., 0., 1., 0., 1.,)( 4., 4., 0., 0., 0.,)( 2.,
  0., 1., 1., 0.,)( 3., 3., 3., 3., 4.)],
 [(3., 0., 3., 3., 2.,)( 2., 1., 1., 2., 4.,)( 2., 1., 4., 1., 3.,)( 2.,
  2., 1., 3., 4., )(0., 0., 0., 0., 1.)],
 [(0., 3., 1., 2., 2.,)( 4., 0., 1., 0., 4.,)( 0., 1., 2., 1., 1., )(2.,
  2., 0., 1., 1., )(3., 3., 3., 3., 4.)],
 [(4., 1., 2., 2., 2.,)( 3., 3., 0., 2., 4.,)( 4., 2., 2., 4., 0.,)( 4.,
  0., 2., 0., 0., )(1., 1., 1., 1., 0.)],
[(3., 2., 0., 1., 1., )(0., 0., 0., 1., 1.,)( 0., 1., 3., 0., 3., )(1.,
 0., 0., 4., 1.,)( 2., 2., 2., 2., 1.)],
[(0., 3., 2., 1., 2., )(2., 0., 1., 0., 4., )(2., 2., 2., 2., 1.,)( 1.,
 0., 0., 2., 0., )(3., 3., 3., 3., 0.)],
[(3., 0., 3., 1., 4., )(3., 4., 1., 4., 1., )(1., 1., 2., 4., 2.,)( 3.,
 1., 4., 3., 3.,)( 0., 0., 0., 0., 1.)]]]])
```

Conclusions from excitatory connection of 9 units.

- We again observe that {20,21,22,23} is always in one community.
- For inhibition of unit 0{10,11,12,13,14} and {15,16,17,18,19} has 3 elements of 2^{nd} community each.
- For inhibition of unit 3 {5,6,7,8,9} and {10,11,12,13,14} has 3 elements of 4^{th} community each and {15,16,17,18,19} has 4 elements of 0^{th} community .
- For inhibition of unit 4 {0,1,2,3,4} has 3 elements of 1^{st} community , {10,11,12,13,14} has 3 elements of 0^{th} community.
- For inhibition of unit 7 {0,1,2,3,4} 3 elements of 2^{nd} community, {15,16,17,18,19} has 3 elements of 0^{th} community.
- For inhibition of unit 9 {10,11,12,13,14} has 4 elements of 2^{nd} community, {15,16,17,18,19} has 3 elements of 0^{th} community.

For 10 unit of excitatory connection , the value of inhibitory stimulus in connection is varied from 0 unit to 10. The clustered community matrix resolved from it is now analysed

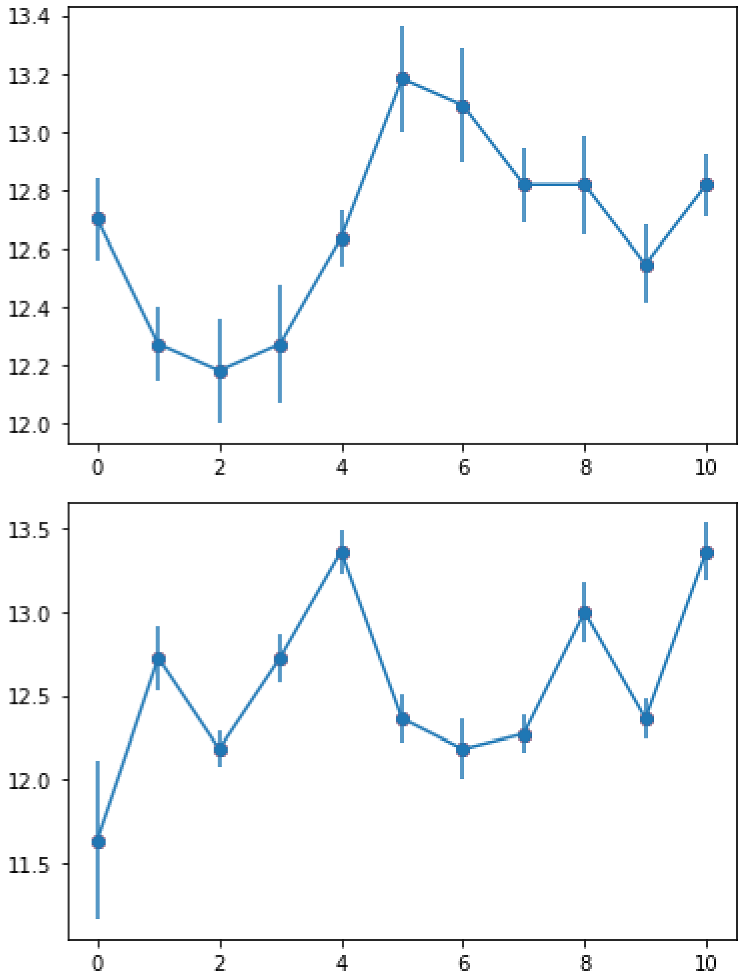
```
([[[(0., 1., 3., 4., 2.,)( 2., 3., 2., 4., 3., )(3., 2., 4., 0., 2., )(3.,
  0., 4., 2., 0., )(1., 1., 1., 1., 3.)],
 [(4., 2., 3., 1., 0.,)( 1., 4., 1., 0., 1., )(4., 4., 3., 0., 0., )(3.,
  0., 3., 0., 3., )(2., 2., 2., 2., 3.)],
 [(0., 2., 1., 1., 1.,)( 0., 1., 1., 1., 0.,)( 4., 0., 3., 3., 1., )(3.,
  1., 3., 1., 0., )(2., 2., 2., 2., 0.)],
 [(1., 3., 4., 0., 4.,)( 4., 2., 1., 2., 4., )(1., 0., 1., 0., 0., )(0.,
  2., 0., 2., 0., )(3., 3., 3., 3., 2.)],
 [(2., 4., 2., 0., 1., )(1., 2., 2., 0., 0., )(1., 0., 2., 1., 0., )(0.,
  1., 0., 2., 3.,)( 4., 4., 4., 4., 0.)],
 [(0., 4., 1., 1., 0., )(1., 1., 3., 3., 0., )(1., 1., 2., 0., 3., )(1.,
  2., 1., 0., 2., )(4., 4., 4., 4., 3.)],
 [(0., 2., 3., 4., 1., )(1., 0., 0., 1., 0.,)(0., 1., 3., 0., 3.,)( 4.,
  4., 0., 1., 3.,)( 2., 2., 2., 2., 1.)],
 [(1., 0., 4., 1., 2.,)( 1., 1., 1., 3., 2., )(1., 3., 1., 4., 4., )(3.,
  3., 2., 3., 3., )(0., 0., 0., 0., 4.)],
 [(2., 3., 2., 0., 2.,)( 0., 4., 0., 0., 4.,)( 1., 1., 1., 2., 0., )(2.,
  4., 1., 0., 2., )(3., 3., 3., 3., 2.)],
 [(1., 0., 3., 2., 2., )(1., 2., 4., 3., 4.,)( 4., 3., 4., 3., 4.,)( 1.,
 4., 3., 3., 3.,)( 0., 0., 0., 0., 2.)],
[(4., 0., 1., 2., 4., )(2., 3., 1., 3., 2., )(3., 4., 1., 2., 3.,)( 2.,
 3., 1., 1., 2., )(0., 0., 0., 0., 3.)]]])
```

Conclusions from excitatory connection of 10 units.

- We again observe that {20,21,22,23} is always in one community.
- For inhibition of 1 unit {5,6,7,8,9} has 3 elements of 1st community and {15,16,17,18,19} has 3 elements of 3rd community.
- For inhibition of 2 unit {0,1,2,3,4} and {5,6,7,8,9} has 3 elements of 1st community each.
- For inhibition of 3 unit {10,11,12,13,14} and {15,16,17,18,19} has 3 elements of 0th community each.
- For inhibition of 7 units {5,6,7,8,9} has 3 elements of 1st community and {15,16,17,18,19} has 3 elements of 3rd community.
- For inhibition of 8 units {0,1,2,3,4} has 3 elements of 2nd community, {5,6,7,8,9} has 3 elements of 0th community {10,11,12,13,14} has 3 elements of 1st community .

- From the above it has been analyzed all the community clusters . They are compared with what they are supposed to be.
- Despite the values being not indicative of lack of resolution, elements of community{20,21,22,23,24} always tends to fall under same community. Considering the symmetric nature of the graph , it can be argued that with introduction of better sophistication in the model, the communities can be more clearly resolved.

Fig. 14.



In both the graphs the x-axis denote the unit and y-axis denote the error for it's defined case. In the first graph the value of excitatory stimuli is varied from 0 to 10 units for synaptic clustering of the 'Necklace Graph' . The y-axis denote the mean error for the each value of excitatory stimuli for range of inhibitory stimuli varying from 0 to 10 units. The error bar is also plotted for it denoting the standard deviation over it. For the second graph the value of inhibitory stimuli is varied from 0 to 10 units for synaptic clustering of the 'Necklace Graph' . The y-axis denote the mean error for the each value of inhibitory stimuli for range of excitatory stimuli varying from 0 to 10 units. The error bar is also plotted for it denoting the standard deviation over it.

Conclusions

- For graphs with high degree of connections, Synaptic clustering has been unable to provide results which are satisfactory as observed from Case 1 when comparing it with Newman clustering. To tackle this values if inhibitory and excitatory stimuli should be manipulated to finer values to reduce the chaotic nature of the system.

- While Synaptic clustering was applied on necklace graph, one community was consistently resolved over all ratios of inhibitory to excitatory stimuli. Considering the symmetric nature of the graph , with further upgrade on the system , it can be definately used to resolve graphs to finer communities.

References

1. Newman, M. E. J. “Modularity and Community Structure in Networks.” *Proceedings of the National Academy of Sciences*, vol. 103, no. 23, June 2006, pp. 8577–82. *DOI.org (Crossref)*, https://doi.org/10.1073/pnas.0601602103.

2. “Donald Hebb.” *The Decision Lab*, https://thedecisionlab.com/thinkers/neuroscience/donald-hebb. Accessed 5 Apr. 2022.

3. Fortunato, Santo, and Marc Barthelemy. “Resolution Limit in Community Detection.” *Proceedings of the National Academy of Sciences*, vol. 104, no. 1, Jan. 2007, pp. 36–41. *arXiv.org*, https://doi.org/10.1073/pnas.0605965104.

4. Gerstner, Wulfram, et al. *Neuronal Dynamics: From Single Neurons to Networks and Models of Cognition*. Cambridge University Press, 2014.

5. Kreuz, Thomas. “SPIKE-Distance.” *Scholarpedia*, vol. 7, no. 12, Dec. 2012, p. 30652. *www.scholarpedia.org*, https://doi.org/10.4249/scholarpedia.30652.

6. *Sum of Degrees of Vertices Theorem | Gate Vidyalay*. https://www.gatevidyalay.com/tag/sum-of-degrees-of-vertices-theorem/#:~:text=Handshaking%20Theorem%2D,of%20edges%20contained%20in%20it. Accessed 5 Apr. 2022.

7. Kumar, Satyam. “Silhouette Method — Better than Elbow Method to Find Optimal Clusters.” *Medium*, 21 Sept. 2021, https://towardsdatascience.com/silhouette-method-better-than-elbow-method-to-find-optimal-clusters-378d62ff6891.

8. “Elbow Method (Clustering).” *Wikipedia*, 25 Mar. 2022. *Wikipedia*, https://en.wikipedia.org/w/index.php?title=Elbow_method_(clustering)&oldid=1079206422