Spring 2021 Meeting of the APS New England Section

Study on the Time-scale Separation in Communities Networks with Consensus Dynamics

Richard Kyung
Hyungsoon Henry Kim
CCRG-NJ

Abstract

In this paper, dynamics, time-scales and communities were studied using graph theory. Simulations were performed by writing a code to simulate consensus dynamics on a network, and verify that the dynamics asymptotically converges towards a constant state.
Adjacency matrix (unweighted) of a structured network with random groups was discussed in this research to study the consensus dynamics on this network which displays a time-scale separation.
The presented code showed a plot of the vector set and x(t) and they converged to the average value after sufficient time steps.
In contrast to the normal patterns, a greater time-scale separation was observed. This was because there were many less edges connecting the different communities: intuitively meaning the communities have less of an effect on each other, or that to have an effect it will take much more time. Increasing the number of random edges between communities— i.e. the magnitude of the perturbation to the adjacency matrix of the three separate communities—will reduce the time-scale separation, making it so that the communities reach the same consensus value at a certain time.
Results show that until around t = 0:05, approximate consensus is reached within each group, then a consensus is reached between the groups.

Introduction

A network is a system made of nodes connected by links. Links can be undirected or directed, and unweighted or weighted. In the mathematical literature, a network is called a graph. It is defined as

$$\mathcal{G} = (V, E)$$

where V is a set of nodes (also called vertices) and E is a set of links (also called edges).

A network can be represented by the corresponding N × N adjacency matrix. Being adjacent means that two nodes are directly connected by a link. In the case of unweighted networks, the entries of the adjacency matrix are given by

$$A_{ij} = \begin{cases} 1 & \text{if node } v_i \text{ is adjacent to node } v_j, \\ 0 & \text{otherwise.} \end{cases}$$

Methods

Graph theory and networks have a close relation.After Euler's development and the famous story of the seven bridges of Konigsberg, networks the dyadic relationships between these units. Once this abstraction has been operated, standardized tools can then be applied to otherwise very different systems.have become a central toolbox in a range of disciplines. Social systems, Web and neuroscience are all the relevant study fields of graph theory and networks. Within the framework of Network Science, a system is modeled as a set of nodes, representing the individual units of the system, and a set of links, representing

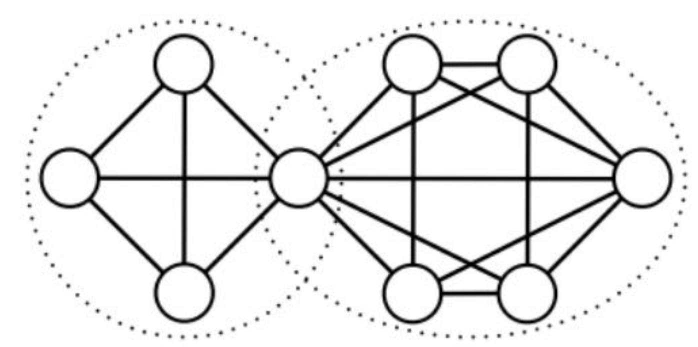
A broad range of dynamical and structural properties of networks is characterised by spectral properties of a matrix describing the network. Depending on the problem at hand, we often use the adjacency matrix (denoted by A) and the Laplacian matrix (denoted by L).

Spectral properties of networks have been studied in detail, and various bounds are available. In this research, we present the basic spectral properties of undirected networks. The Laplacian and normalised Laplacian are defined by

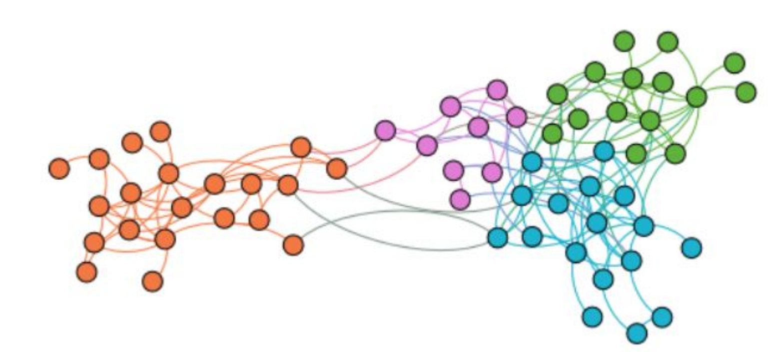
$$L_{ij} = k_i\delta_{ij} - A_{ij}, \qquad \tilde{L}_{ij} = \delta_{ij} - \frac{A_{ij}}{\sqrt{k_i k_j}}.$$

$$L = D - A, \qquad \tilde{L} = D^{-1/2} L D^{-1/2} = I - D^{-1/2} A D^{-1/2},$$

For the adjacency matrix, it is customary to order the eigenvectors from the largest λ1 to the smallest λN , whereas the eigenvalues are usually ordered from the smallest to the largest for the Laplacian matrices.



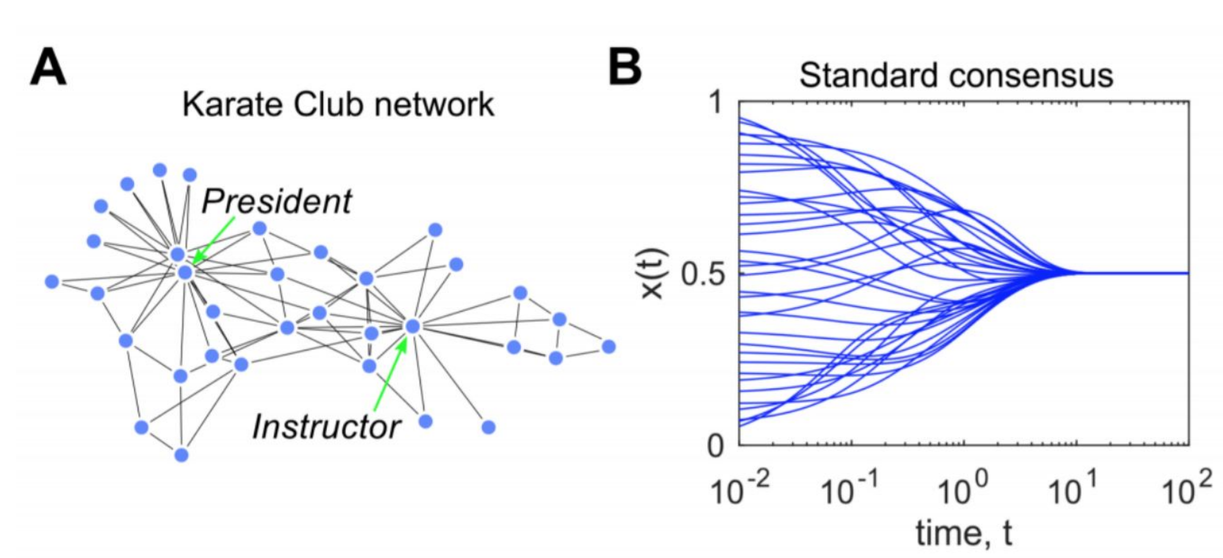
Two communities are shown by dotted circles. One node belongs to both communities.



A social network of bottlenose dolphins. Four communities detected by the Louvain algorithm implemented on gephi (http://www.gephi.org) are shown by different colours. Schematic of overlapping communities.

Dynamics, time-scales, and communities

Illustration of a consensus dynamics on the Karate Club network. A Karate Club network orginally analysed by Zachary. B Consensus dynamics on the Karate club network starting from a random initial condition. As time progresses the states of the individual nodes become more and more aligned, and eventually reach a consensus value, equal to the arithmetic average of the initial condition.



Results

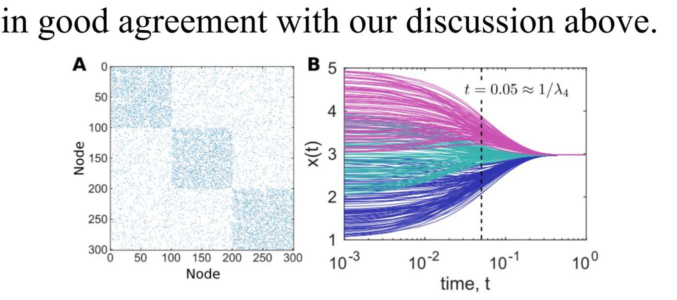
Consensus has been one of the most popular and well studied dynamics on networks. This is due to both its analytic tractability as well as its simplicity in approximating several fundamental behaviors. For instance, in socio-economic domains consensus provides a model for opinion formation in a society of individuals.
For engineering systems, it has been considered as a basic building block for an efficient distributed computation of global functions in networks of sensors, robots, or other agents.

To define a standard consensus dynamics, consider a given connected network of n nodes and adjacency matrix A. Let us endow each node with a scalar state variable xi ∈ R. The (average) consensus dynamics on such a network is then defined as:

$$\dot{x} = -Lx, \text{ (consensus dynamics)}$$

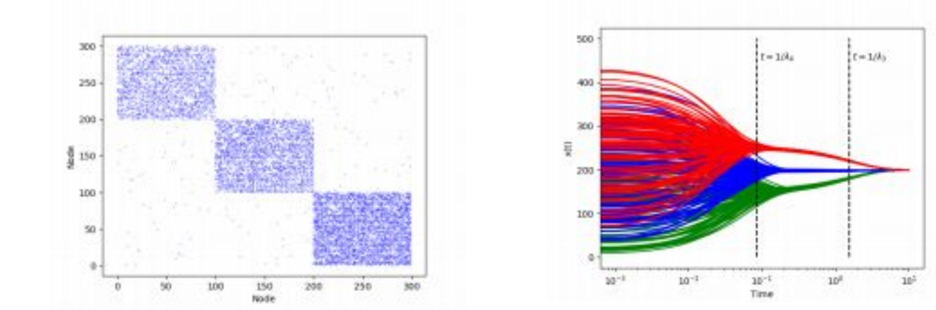
where L is the graph Laplacian.

Adjacency matrix (unweighted) of a structured network with 5 groups, as discussed in the text. B A consensus dynamics on this network displays a time-scale separation: until around t = 0.05, approximate consensus is reached within each group (groups indicated by color); then a consensus is reached between the groups. Note that for the shown network λ4 = 18, in good agreement with our discussion above.



a. The following code outputs a plot of the vector x_t. On all graphs tested, x_t converged to the average value $1^T \cdot x_0/n$ after sufficient time steps:

b. The following code will replicate similar results to Figure 23. Figures shown on next page after code.



Conclusion

In contrast to the sample figures, here we see a greater time-scale separation. This is because there are many less edges connecting the different communities here, intuitively meaning the communities have less of an effect on each other, or that to have an effect it will take much more time. Increasing the number of random edges between communities—i.e. the magnitude of the perturbation to the adjacency matrix of the three separate communities—will reduce the time-scale separation, making it so that the communities reach the same consensus value at time $1/\lambda_4$. This parameter can be controlled by line 22 in the code*, although note that random edges are also added into the communities. This doesn't change the conclusion of the simulation since the communities are arbitrary connected graphs anyways.