Random Walk on a Network

The Impact of Random Diffusion Processes in a Non-Spatial Domain

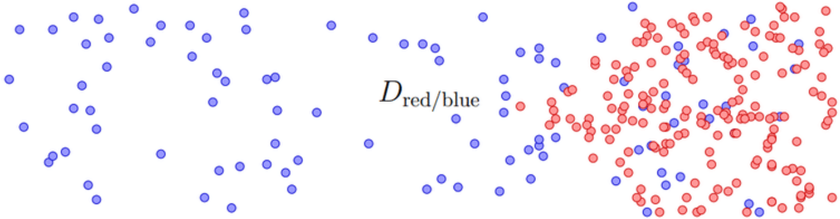
Nathan Pacey

18njp@queensu.ca

April 9, 2024

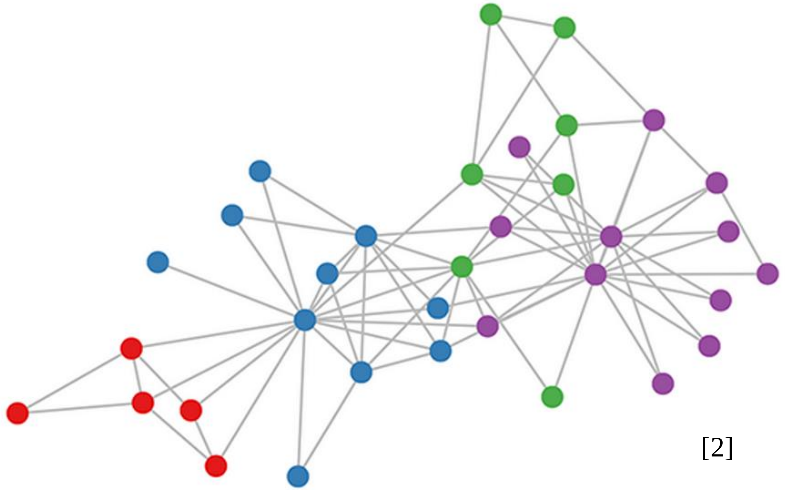
Final Presentation for ENPH 317 Mathematical Methods in Physics II

Introduction



[1]

Traditionally **randomness and diffusion are examined within spatial realms** such as particles moving from high to low concentration areas.



[2]

In network theory, space is conceptualized through connections rather than physical proximity. **Making random walks across networks—paths from node to node driven by probabilities**.

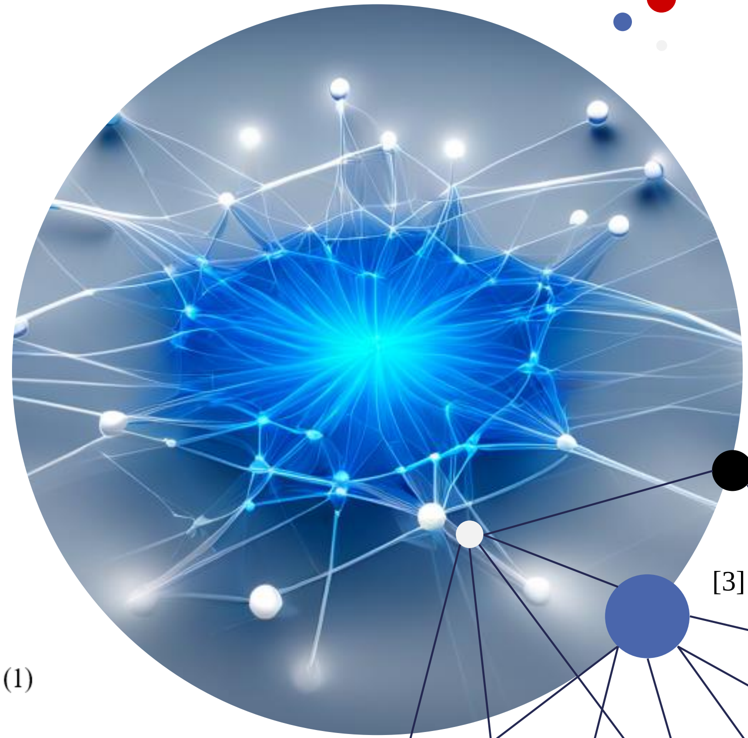
This project delves into such random walks, probing into the steady state behaviors that reveal the underlying dynamics and structural influences of networks.

Problem Definition

This study examines how a random walk unfolds across a network, with nodes as individuals and edges as their links.

Leveraging an adjacency matrix A, that captures community relationships, the **aim is to model the random walk mathematically** and pinpoint the **steady-state probability of an individual holding a box** [4].

$$A_{ij} = \begin{cases} 1 & if\ person\ i\ and\ j\ are\ friends \\ 0 & if\ person\ i\ and\ j\ are\ not\ friends \end{cases} \quad (1)$$



[3]

Problem Methodology

The simulation will determine the network's steady state and its probability distribution by **calculating the likelihood of each individual holding the 'box' over time**, considering their number of connections k_j [4].

$$p_i(t) = \sum_j \frac{A_{ij}}{k_j} p_j(t-1) \tag{2}$$

Equilibrium is achieved when the probability $p_i(t)$ **stabilizes with time approaching infinity**, influenced by the network's configuration and the strength of connections, which ultimately determine the probability distribution. This is analyzed using the graph Laplacian [4].

$$L = D - A \tag{3}$$

With the degree matrix D, described by the degree of connections k_j along the diagonal elements of node j.

$$D = diag(k_1, k_2, k_3, \ldots, k_N) \tag{4}$$

Problem Methodology

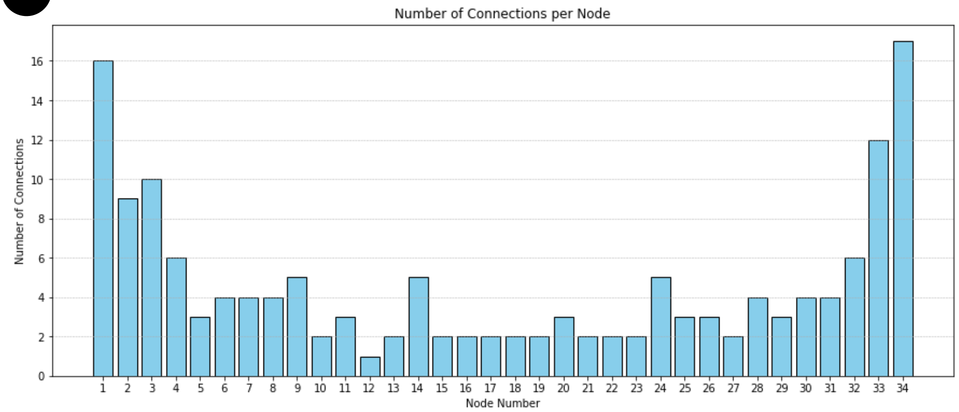
The probability equation can be simplified to the discrete-time Markov chain.

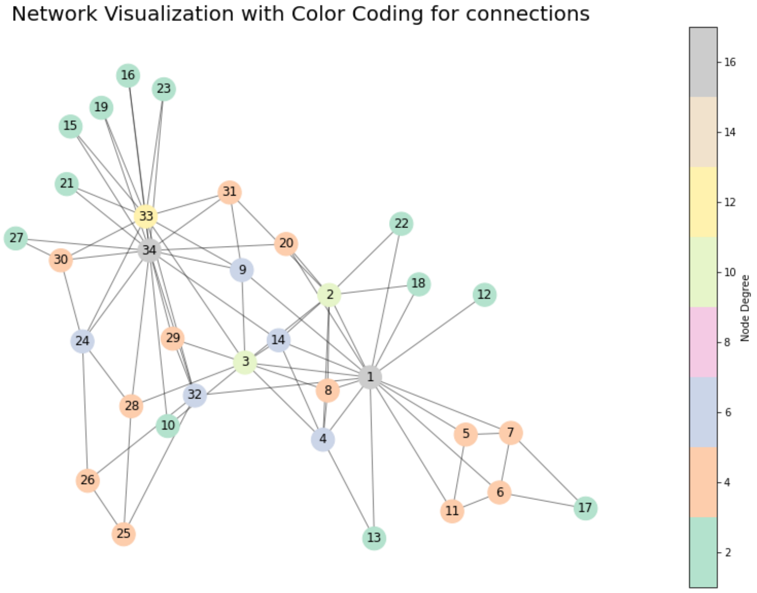
$$p_i(t) = \sum_j D_{jj}^{-1} A_{ij}\, p_j(t-1) \tag{5}$$

The Laplacian's eigenvectors, particularly the Fiedler vector, highlight steady-state traits. In spectral graph theory, the Fiedler vector—associated with the second smallest eigenvalue of the Laplacian—indicates the network's connectivity; a higher Fiedler value suggests greater overall connectivity.

[5]

Graph Structure



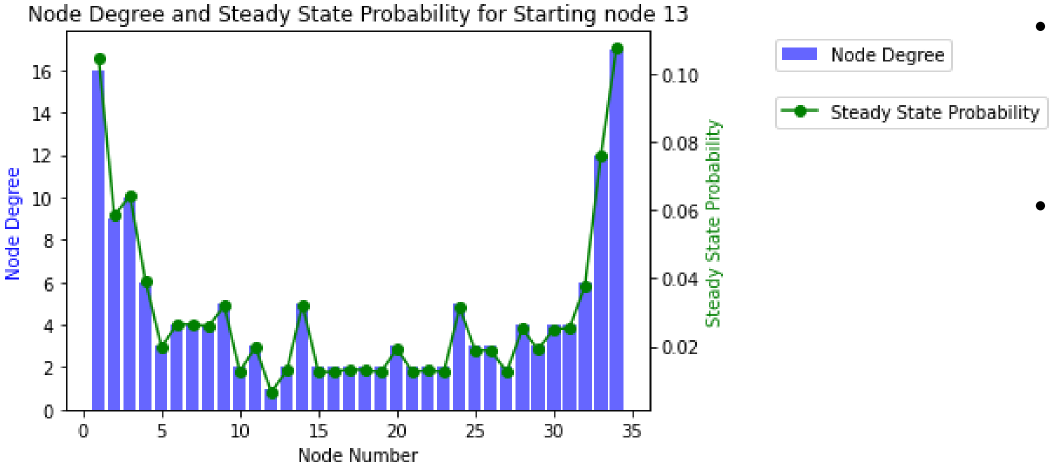


First step was to understand network connections:

- Utilized matplotlib and networkx to visualize the network and interpret the structure.
- A histogram quantifies connection frequency, pinpointing influential nodes.
- Connection count directly informs the probability distribution in steady-state analysis.

Steady State Analysis

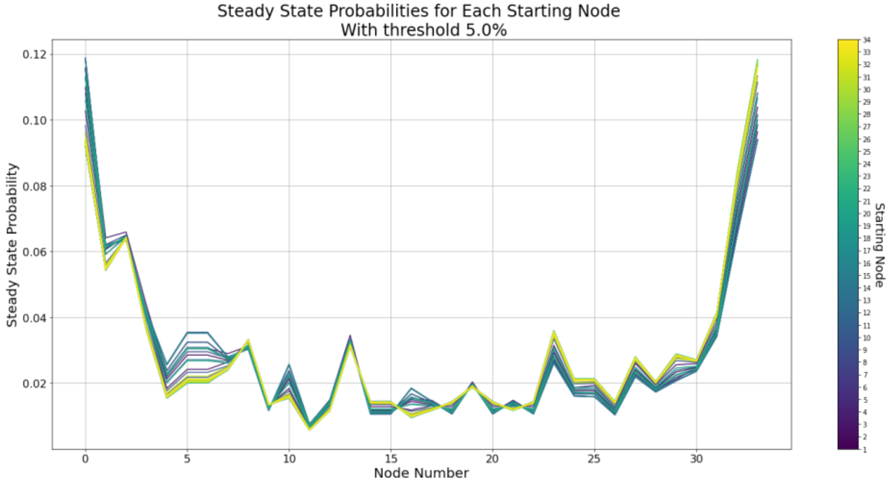
- Implemented a discrete-time Markov chain in Python to calculate the network's steady-state probability, using adjacency and degree matrices to represent node connections.
- The simulation iterated until probability changes were minimal, establishing an equilibrium distribution that reflects the long-term likelihood of each node holding the 'box'.

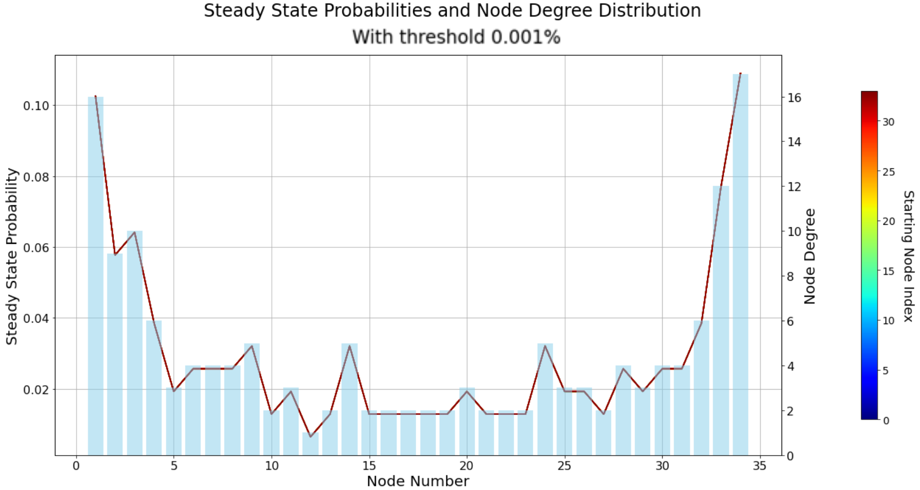


- Produced a visualization correlating steady-state probabilities with node degrees to assess how closely the distribution matches the network's connectivity.
- Can see slight variations due to the iteration accepted error margins (threshold)

Steady State Solution

As the threshold approaches zero, the differences in probabilities originating from different starting nodes decrease, and the **probabilities converge to a single steady-state solution**

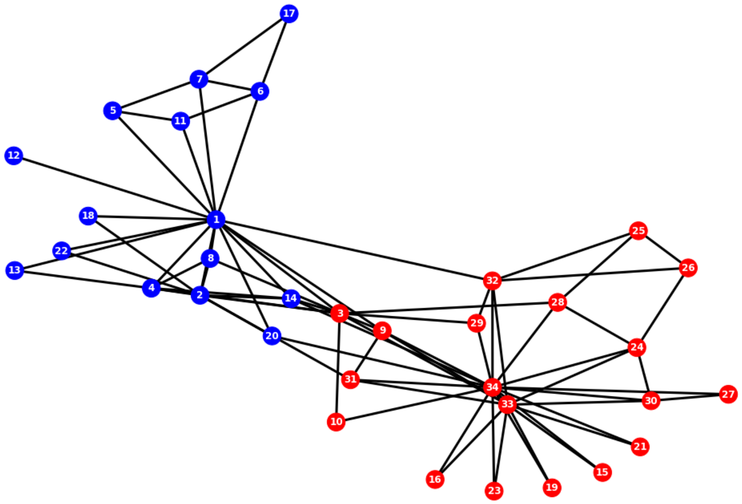




As expected, **the steady state solution is directly proportional to the node degrees**.

Laplacian Relationship to Probability Distribution

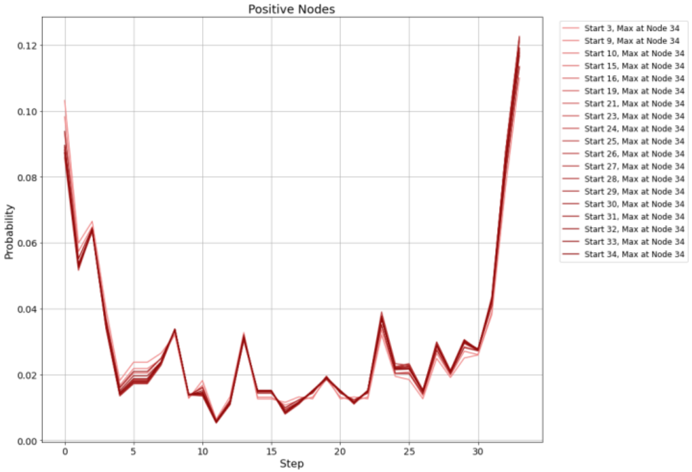
- The **Fiedler vector**, related to the network's Laplacian matrix, **reflects algebraic connectivity**; its magnitude signals the network's overall connectedness [6].
- **This vector aids in discerning network clusters**, with its distinct component values indicating groups of internally dense connections among nodes [7].
- The Fiedler vector for each node was determined where the order is ascending from node 1 to 34.

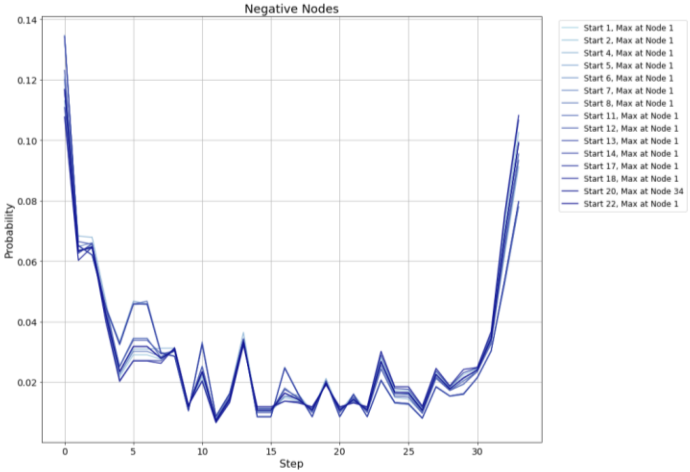



```
Fiedler vector:
 [-0.11213743 -0.04128789  0.02321896 -0.05549978 -0.28460453 -0.32372722
 -0.32372722 -0.052586    0.05160128  0.09280089 -0.28460453 -0.21099295
 -0.1094613  -0.01474197  0.16275078  0.16275078 -0.42276533 -0.10018142
  0.16275078 -0.01363713  0.16275078 -0.10018142  0.16275078  0.15569457
  0.15302558  0.16096292  0.18710956  0.12766405  0.0951523   0.16765031
  0.07349964  0.09875343  0.13034546  0.11890326]

Nodes with positive Fiedler vector components:
 [ 3  9 10 15 16 19 21 23 24 25 26 27 28 29 30 31 32 33 34]
Nodes with negative Fiedler vector components:
 [ 1  2  4  5  6  7  8 11 12 13 14 17 18 20 22]
```

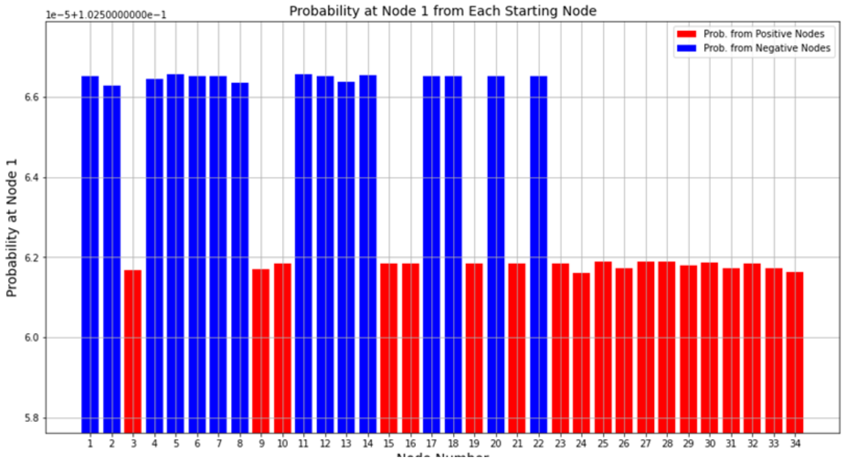
Probability Impacts of the Fiedler Vector

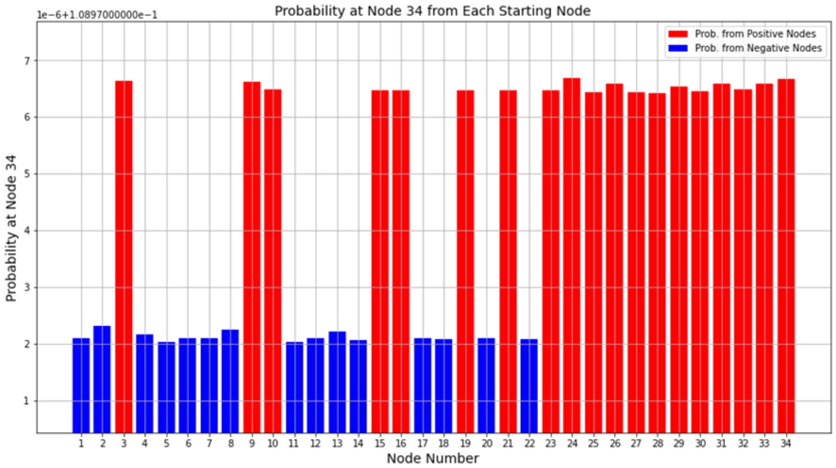




- Threshold adjustments highlight how Fiedler vectors distinguish node groups, revealing preferences for specific nodes in the box's path.
- Visualization of steady-state probabilities at a 10% threshold accentuates the division between node groups, correlating higher probabilities with either node 1 or node 34 depending on their Fiedler vector sign.

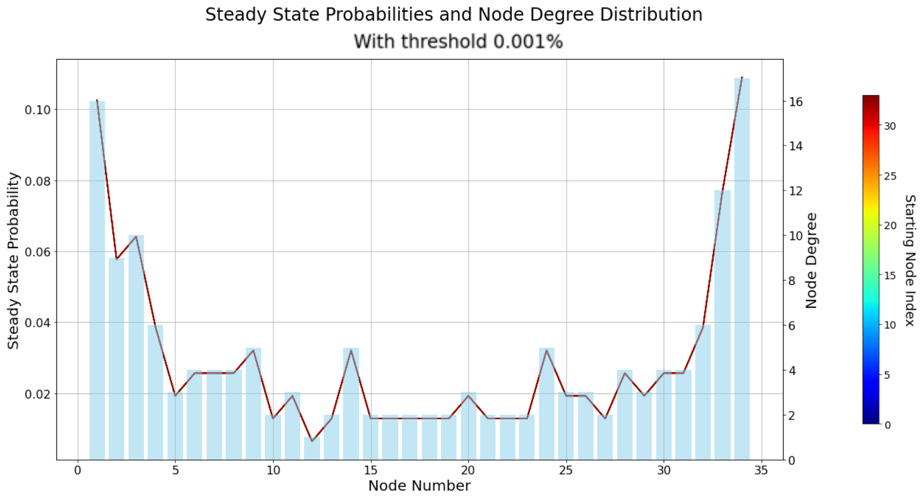
Probability Impacts of the Fiedler Vector

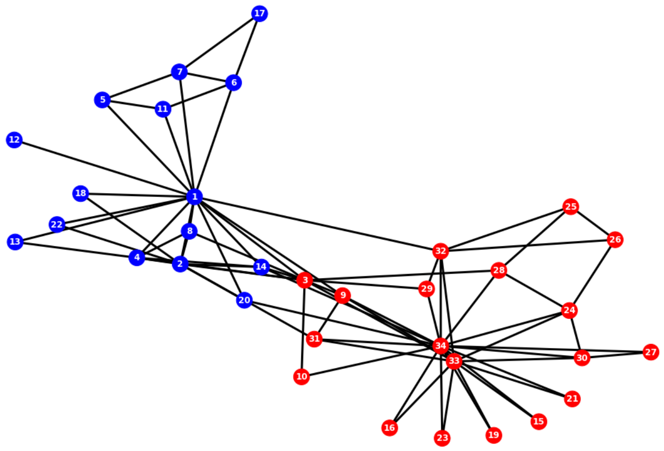




- Distinct groupings are still prevalent for very small thresholds (0.001%).
- This outcome is anticipated because raising the threshold reduces the number of iterations in the random walk, making the results more sensitive to the immediate connections of each node, essentially emphasizing the local network structure around each starting point.

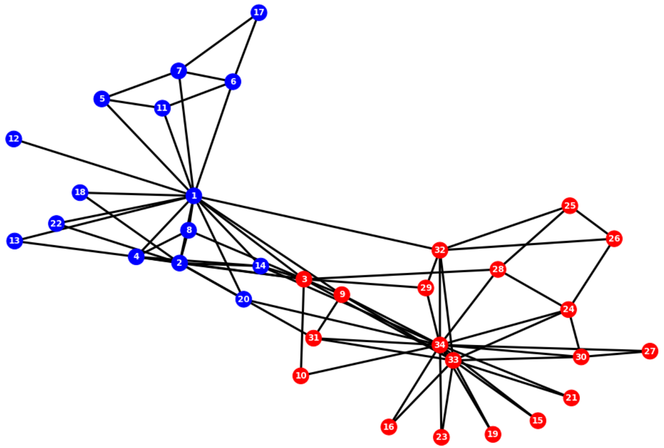
Results Summary





- As thresholds decrease towards zero, steady-state probabilities across varying starting points converge to one distribution reflecting node connectivity. Node 34 consistently shows the highest probability, aligning with its high connection count.
- Analysis of Fiedler vectors reveals a clear dichotomy in the network's structure, dividing nodes into two clusters with distinctive probability outcomes influenced by their links within the network.

Suggestions



- Nodes with negative Fiedler vectors tend to direct the 'box' towards node 1, while those with positive vectors favor node 34, highlighting distinct network clusters and influencing node interaction strategies.

- For added nodes **desiring the 'box,' forming a connection with node 34 is advisable** due to its central role; however, those aligned with negative Fiedler vector nodes might find better prospects with node 1.

- Conversely, for new nodes **avoiding** the **'box' suggests connecting with nodes like node 12** and **avoiding nodes 34 and 1**.

Extensions?

More clusters and added friendship dynamics?

References

[1] N. D. Ro, "Local Equilibrium Approach to Quantum Transport in Normal Fluids," ResearchGate, September 2022. [Online]. Available: https://www.researchgate.net/figure/Illustration-of-particle-diffusion-Higher-concentration-of-red-particles-on-the-right_fig2_363645602. [Accessed 5 April 2024].

[2] D. Karunarathna, "Introduction to Graph Convolutional Networks (GCN)," Medium, 11 November 2020. [Online]. Available: https://dilinikarunarathna.medium.com/introduction-to-graph-convolutional-networks-gcn-2235ed69875d. [Accessed 22 Mach 2024].

[3] Daisie Team, "Guide to Diffusion Model Hash in Networking," DAISIE, 7 August 2023. [Online]. Available: https://blog.daisie.com/guide-to-diffusion-model-hash-in-networking/. [Accessed 5 April 2024].

[4] G. V. Anders, "Resource: Networks," Queen's University, Kingston, 2024.

[5] C. Brune, "Geometric Deep Learning," Applied Analysis, SACS, Applied Mathematics, 30 July 2021. [Online]. Available: https://bathicmsworkshop.github.io/ChristophBrune.pdf. [Accessed 3 April 2024].

[6] E. Weisstein, "Fiedler Vector," Wolfram, [Online]. Available: https://mathworld.wolfram.com/FiedlerVector.html. [Accessed 4 April 2024].

[7] J. Demmel, "CS 267: Notes for Lecture 23, April 9, 1999. Graph Partitioning, Part 2," UC Berkeley, [Online]. Available: https://people.eecs.berkeley.edu/~demmel/cs267/lecture20/lecture20.html. [Accessed 4 April 2024].