Examination of network dynamics using random walks

Peter Larsen and Nayo Ogilvie

Modeling Complex Systems

December 4th, 2020

Agenda for Presentation

- What we are modelling
- How we modeled
- Results
- Further steps

What are we modelling?

- Part 1: Comparisons Using Random Walks Between A Standard 3D Space and a Non-Uniform Dimension Network
- Part 2: Random Walk With Specific Origins Based on Centralities
- Part3: Navigating a Network with Random Walks While Including Bias from a Paired Network

What are we modelling?

- Part 1: Comparisons Using Random Walks Between A Standard 3D Space and a Non-Uniform Dimension Network
- Part 2: Random Walk With Specific Origins Based on Centralities
- Part3: Navigating a Network with Random Walks While Including Bias from a Paired Network

What are we modelling?

- Part 1: Comparisons Using Random Walks Between A Standard 3D Space and a Non-Uniform Dimension Network
- Part 2: Random Walk With Specific Origins Based on Centralities
- Part3: Navigating a Network with Random Walks While Including Bias from a Paired Network

Part 1: How is it being modeled?

- We created a network of Marvel Comics characters using an adjacency matrix and linear algebra manipulations
- The following are two graphical representations of the Person to Person Network (P2P)

P2P Representation 1

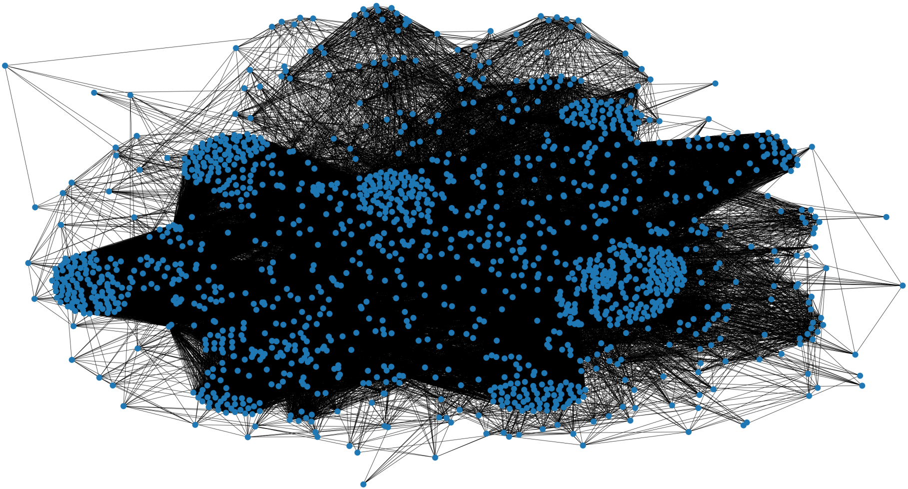


Figure: Python package "networkx" representation of P2P

P2P Representation 2

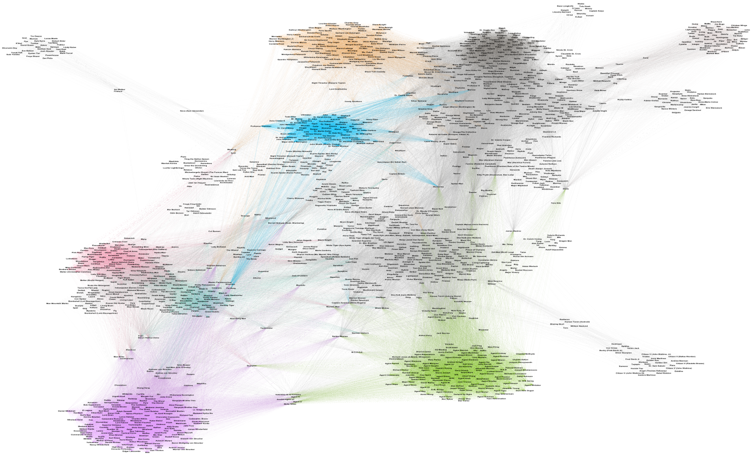


Figure: GePhi representation of P2P

Part 1: How is it being Modeled?

- Start from a random node and randomly move to a neighboring node
- Perform this movement action 1000 times
- To smooth over irregularities, iterate the above 100 times and average all results.
- This is denoted as NonUni for Non-Uniform Dimension/Degree

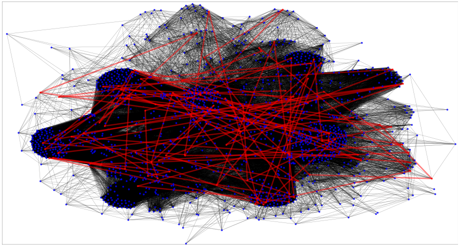


Figure: Random Walk for 100 Steps

Part 1: How is it being Modeled?

- To compare and contrast our results, we created a simple 3D random walk and ran it for 1000 timesteps
- To smooth over irregularities, iterate the above 100 times and average all results
- This is denoted as Uni for Uniform Dimension/Degree

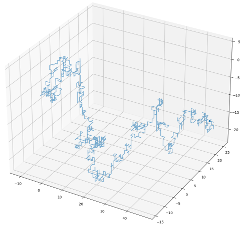


Figure: Random Walk for 1000 Steps

Part 1: Results

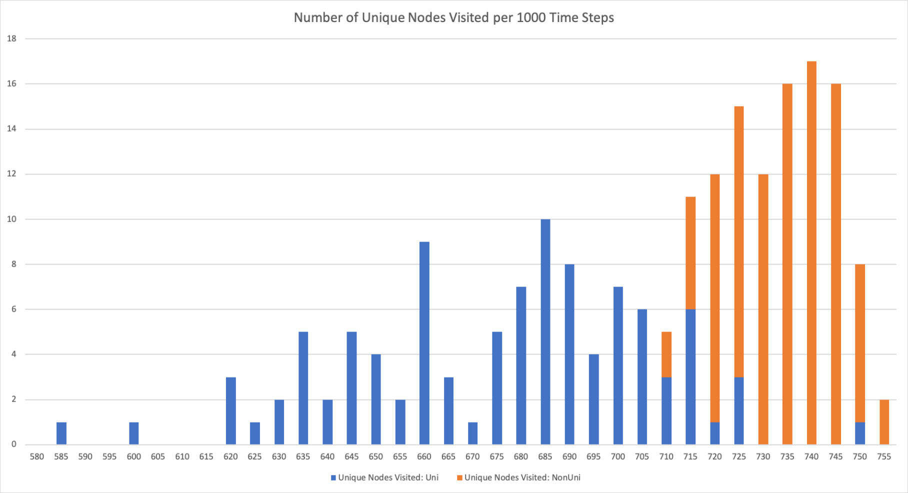
- In order to compare the Uni and NonUni random walks, we implemented two metrics: Coverage Fraction (CF), which measures the amount of unique nodes visited by each run, and Rate of Return (RoR), which measures the overall success of the random walk returning to its origin at some point within its run
- Based on the work by Polya (1921), Watson (1939), McCrea and Whipple (1940), Domb (1954), and Glasser and Zucker (1977), it can be mathematically shown that the RoR for a standard 3D random walk is 0.340537...

$$
\begin{aligned}
P(3) &= 1 - \frac{1}{u(3)} = 0.340537\ldots \\
u(3) &= \frac{3}{(2\pi)^3} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \frac{dxdydz}{3 - \cos(x) - \cos(y) - \cos(z)} \\
&\approx 1.516386
\end{aligned}
$$

Part 1: Results

- The CF for the Uni runs was 0.6741, or 67.41%. The CF for the NonUni runs was 0.7313, or 73.13%. This means that on average, the random walk in the P2P network was reaching approximately 57 more unique nodes per run than the 3D random walk.
- The RoR for the Uni runs was 34%. The RoR for the NonUni runs was 44%. This means that, aside from our results for the 3D run matching near perfectly with the mathematics, a NonUni run was 10% more likely to return to the origin at some point within a 1000 timesteps.

Part 1: Results



Part 2: How is it being Modeled?

- Start from a pre-selected node and randomly move to a neighboring node
- Perform this movement action 1000 times

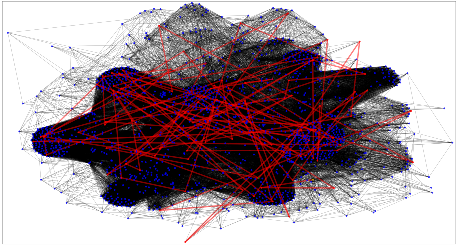


Figure: Random Walk pre-selected start for 100 Steps

Part 2: How is it being Modeled?

- The nodes in the P2P were selected based upon their centrality scores in 3 different measures: betweenness centrality, closeness centrality, and eigenvector centrality.
- Betweenness Centrality is calculated by the following: $C_B(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{s,t}(v)}{\sigma_{s,t}}$, where $\sigma_{s,t}$ represents the total number of shortest paths from node s to node t and $\sigma_{s,t}(v)$ is the number of those paths that pass through v.
- Closeness Centrality is calculated by the following: $C(x) = \frac{1}{\sum_y d(y,x)}$, where $d(y,x)$ is the length of the shortest path between nodes x and y.
- Eigenvector Centrality is calculated by the following: $x_v = \frac{1}{\lambda} \sum_{t \in G} a_{v,t} x_t$, where $a_{v,t} = 1$ if nodes x and t are neighbors and 0 otherwise,x_t represents the eigenvector centrality score of t, and λ is a constant.

Part 2: Results

- Average CF for the Top 5 Betweenness: 72.907%
- Average CF for the Middle 5 Betweenness: 73.286%
- Average CF for the Bottom 5 Betweenness: 72.487%

Part 2: Results

- Average CF for the Top 5 Eigenvector: 73.566%
- Average CF for the Middle 5 Eigenvector: 73.306%
- Average CF for the Bottom 5 Eigenvector: 73.366%

Part 2: Results

- Average CF for the Top 5 Closeness: 73.566%
- Average CF for the Middle 5 Closeness: 72.807%
- Average CF for the Bottom 5 Closeness: 72.567%

Part 2: Follow up Steps

- Determine CF of the above results with 100 iterations for each specific node in those categories to smooth out results

Part 3: How is it being Modeled?

- Using weights from the other network created by the adjacency matrix (multiplying the matrix by its transpose one direction gave P2P, while multiplying in the other direction gave the Group to Group network (G2G)), we want to guide a random walker by influencing the adjacency lists in P2P with those weights.
- Designate one random groups as the origin and one random group that is not the origin as the target. We select a random member of the origin group that is not a member of the target group. Then, for each of that nodes neighbors, we determine the group they are a part of with the highest connectivity to the target group, and multiply their number by the weight of connectivity.
- Ideally, this will change an adjacency list of {A: B, C, D, E, Z} with target group R such that weights are as follows: BR = 5, CR = 11, DR = 1, ER = 0, ZR = 8 into an adjacency list like so {A: B,B,B,B,B,C,C,C,C,C,C,C,C,C,C,C,D,Z,Z,Z,Z,Z,Z,Z,Z}. Then you randomly choose one of those nodes, and repeat.

Part 3: Results

- To be Determined over the next few days

Conclusion

- Created a network of Marvel Comics characters using an adjacency matrix and linear algebra manipulations
- Part 1: Implement and Test Random Walks for Uni and NonUni networks.
- Part 2: Implement and Test Random Walks for pre-selected nodes in our NonUni network there are further tests to run.
- Part 3: Started Implementing the model and will have results shortly.