(Mis)information Spread in Hierarchical Social Media Networks

Dav King & Mitch Harrison

Introduction

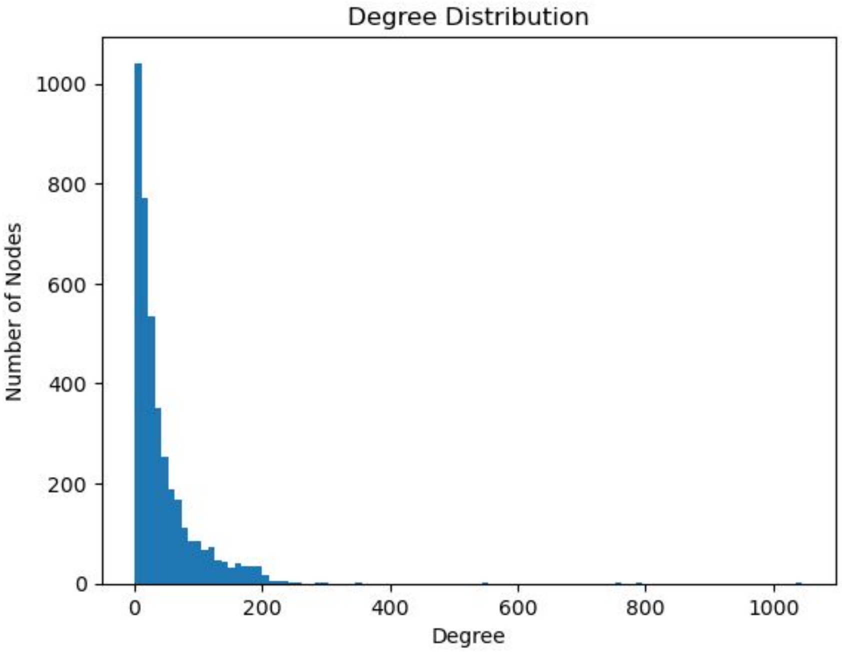
- In recent years, social media has taken off as a huge source of connection between individuals in society.
  - Facebook reports upwards of 3 billion monthly users, a staggering amount of connection that would have been unfathomable even twenty years ago.
- At the same time, media literacy has been declining, and misinformation has run rampant, impacting elections, political unrest, and more.
- Modeling the spread of information in a social media network is thus key to understanding how to combat misinformation (and spread useful information, if needed).

WhatsApp Study (Nobre et al., 2021)

- Analyzed the spread of misinformation on WhatsApp during the 2018 Brazilian elections.
- Explored misinformation spread at multiple levels: individuals, WhatsApp groups, and latent user communities.
- A small fraction of individuals are responsible for sharing a huge proportion of the total information, often introducing it.
- While the individuals sharing the most misinformation change, the communities that they belong to are very stable.
- There is some suggestion that misinformation campaigns may be coordinated, and individuals sharing more misinformation tend to position themselves so that they are central to the network.
    - They propose two potential avenues of misinformation spread: The network “backbone”, which is very stable and may be coordinated, and the network “periphery”, which is likely uncoordinated and reactionary.

The Dataset

- Stanford University gathered “circles” (friends lists) from Facebook users via survey.
- Dataset is an undirected graph. This is necessarily the case from Facebook.
- V = 4,039 | E = 88,234
- Added 20 nodes to account for hierarchical structure (more on that later).



Background

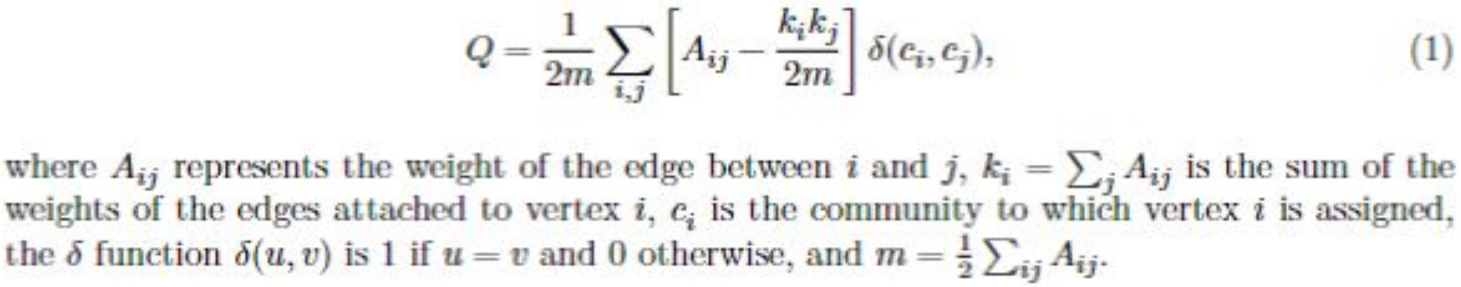
- In a social media network, there is often a latent hierarchical structure.
- Users are a part of communities, which themselves compose higher-level communities, until (generally) several levels of hierarchy are defined.
- Unless there is a predefined structure (i.e., user groups), identifying these communities can be difficult. However, some algorithms have been developed.

Evaluating Community Detection

- Though many techniques to evaluate community detection have been developed, the most widely used is **modularity**.
- Modularity is a scalar in [-1, 1] that measures the density of links inside communities compared to links between communities.
- In other words, it is the density of links within a community relative to the density of links that would be expected in a random graph.

Modularity

Modularity is defined



$$Q = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j), \tag{1}$$

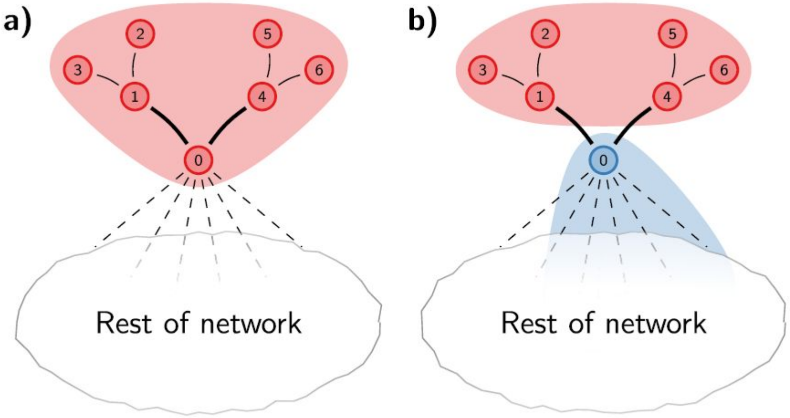
where A_{ij} represents the weight of the edge between i and j, $k_i = \sum_j A_{ij}$ is the sum of the weights of the edges attached to vertex i, c_i is the community to which vertex i is assigned, the δ function $\delta(u, v)$ is 1 if $u = v$ and 0 otherwise, and $m = \frac{1}{2} \sum_{ij} A_{ij}$.

Louvain Algorithm (Blondel et al., 2008)

- This was the first truly scalable algorithm for community detection
- Repeat two steps until convergence:
  - Place every node in its own community. For each node *i*, evaluate the modularity gain by placing *i* into the community of each neighbor *j*. Place node *i* into the community that maximizes modularity gain (provided the gain is positive). Repeat sequentially for all nodes until no further improvement can be achieved.
  - Build a new network, whose nodes are the communities from step 1. The weights of the edges are the sum of the weights of the edges between nodes in the corresponding two communities. Internal edges become self-loops. Repeat step 1 on the new network.

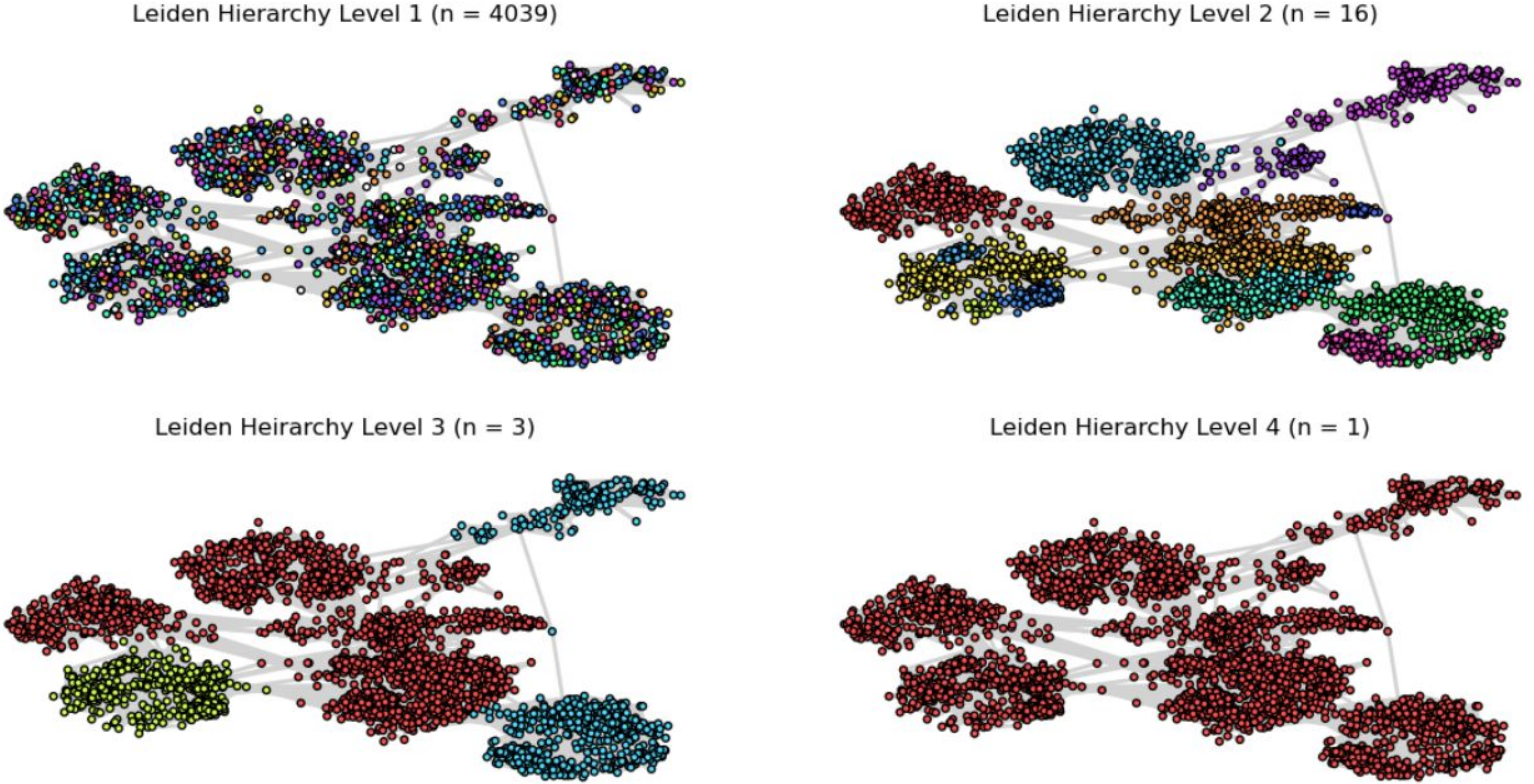
Leiden Algorithm (Traag, Waltman, & van Eck, 2019)

- The Louvain algorithm has some unaddressed problems: Moving nodes may internally disconnect communities, but there is no procedure for de-aggregating communities



Leiden Algorithm

- This algorithm is very similar to the Louvain algorithm, with a couple of key differences.
    - In phase 1: Only consider moving nodes whose neighborhoods have changed. The rest are irrelevant, and only slow down the algorithm.
    - After developing a partition *P*, perform the same local moving of nodes internally within each developed community.
    - This can allow the communities to divide into smaller communities, though that will not always occur.



Leiden Hierarchy Level 1 (n = 4039)

Leiden Hierarchy Level 2 (n = 16)

Leiden Heirarchy Level 3 (n = 3)

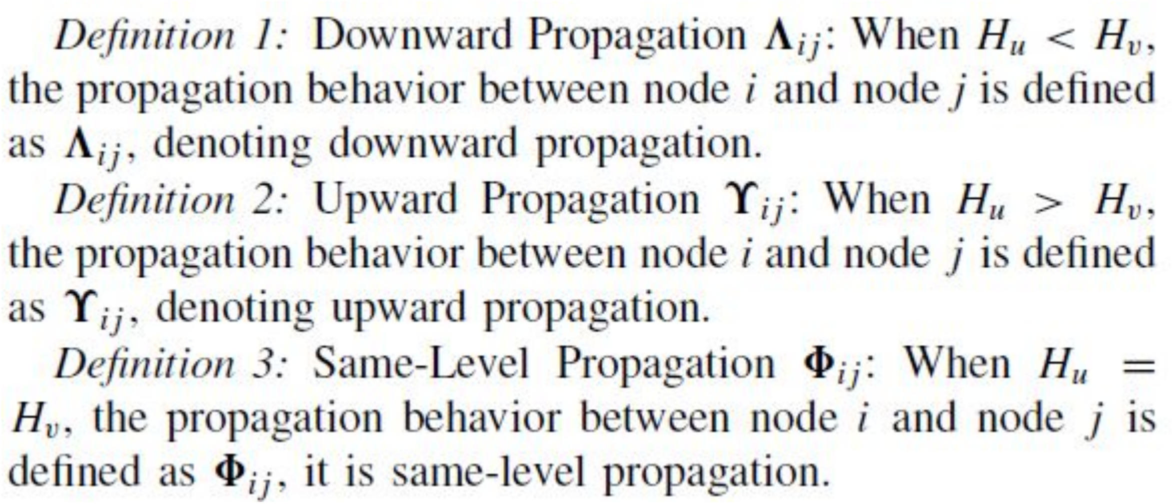
Leiden Heirarchy Level 4 (n = 1)

H-SIR: Information Dissemination in a Hierarchical Network (Nian et al., 2024)

- Network propagation has long been studied in the context of disease spread: the SIR model was first developed in the 1960s, and remains popular today.
- We can utilize this algorithm to model the spread of information in a network, adding a twist to account for the hierarchy.
- There are three node states:
    - **Susceptible (S) State:** The individual is not yet informed of the information, but has the potential to acquire information.
    - **Infected (I) State:** The individual has acquired the information and has the potential to spread it to others.
    - **Recovery (R) State:** The individual is no longer willing to get the information nor spread the information to others.
- This model also requires the definition of **node interest**, which is the interest of each node in participating in information dissemination (i.e., a propagation probability).

H-SIR: Information Dissemination in a Hierarchical Network (Nian et al., 2024)

Assume that the hierarchy level of susceptible node *i* is H_i and the hierarchy level of infected node *j* is H_j (higher numbers = higher level). Then, define the following:



Definition 1: Downward Propagation $\mathbf{\Lambda}_{ij}$: When $H_u < H_v$, the propagation behavior between node i and node j is defined as $\mathbf{\Lambda}_{ij}$, denoting downward propagation.

Definition 2: Upward Propagation $\mathbf{\Upsilon}_{ij}$: When $H_u > H_v$, the propagation behavior between node i and node j is defined as $\mathbf{\Upsilon}_{ij}$, denoting upward propagation.

Definition 3: Same-Level Propagation $\mathbf{\Phi}_{ij}$: When $H_u = H_v$, the propagation behavior between node i and node j is defined as $\mathbf{\Phi}_{ij}$, it is same-level propagation.

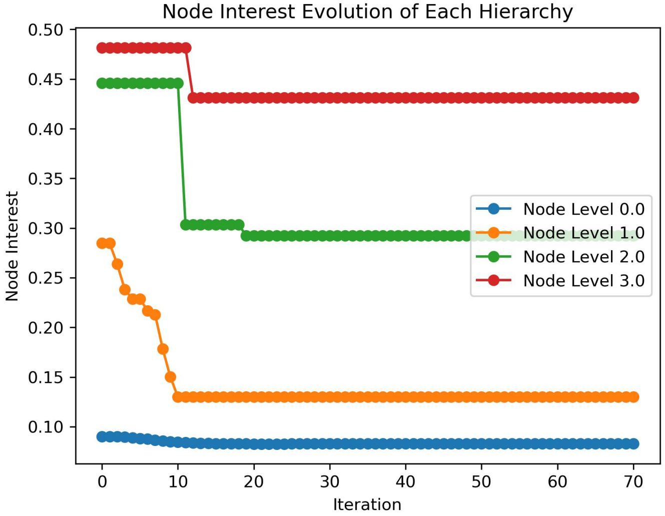
H-SIR: Information Dissemination in a Hierarchical Network (Nian et al., 2024)

At time t + 1, an infected node I propagates information to infected node S with a probability given by the node interest. This value is affected by relative positioning in the hierarchy: Upward propagation is much more difficult than downward propagation.

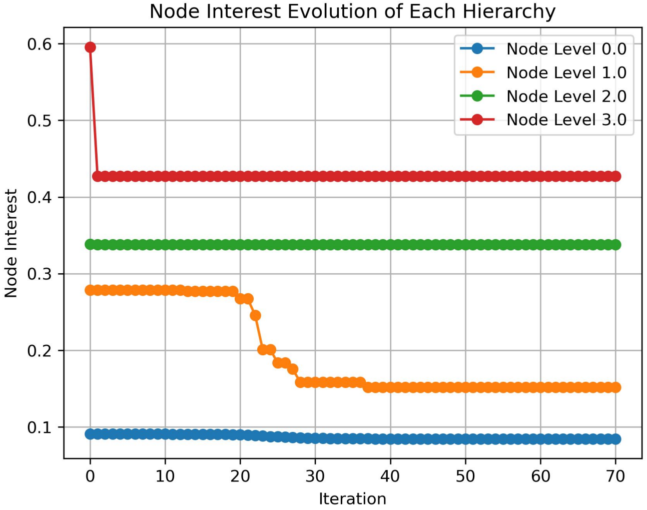
Following this, the I-state node reverts to an R-state node with recovery probability γ.

H-SIR Interest Convergence

Random Starting Nodes

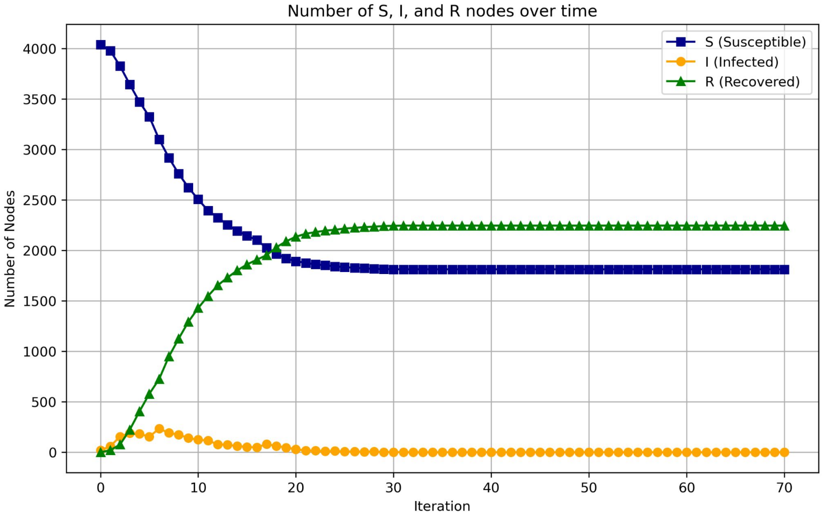


High-Level Starting Nodes

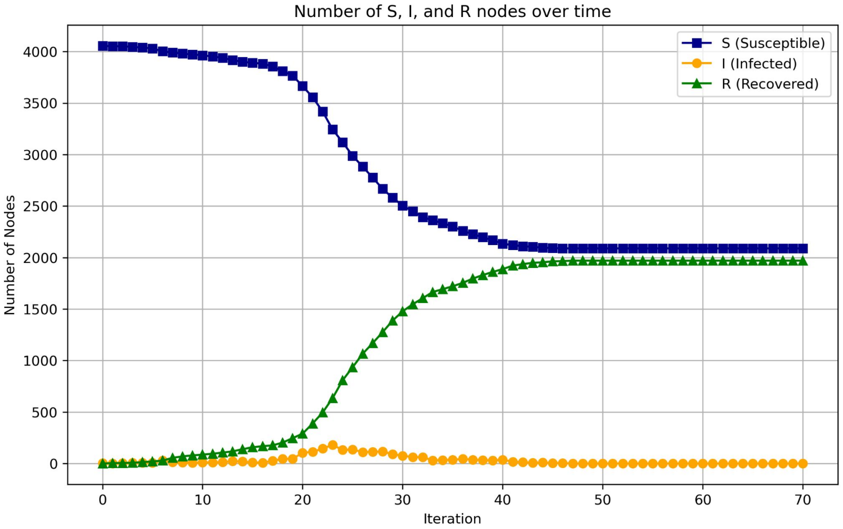


H-SIR Infection Convergence

Random Starting Nodes



High-Level Starting Nodes



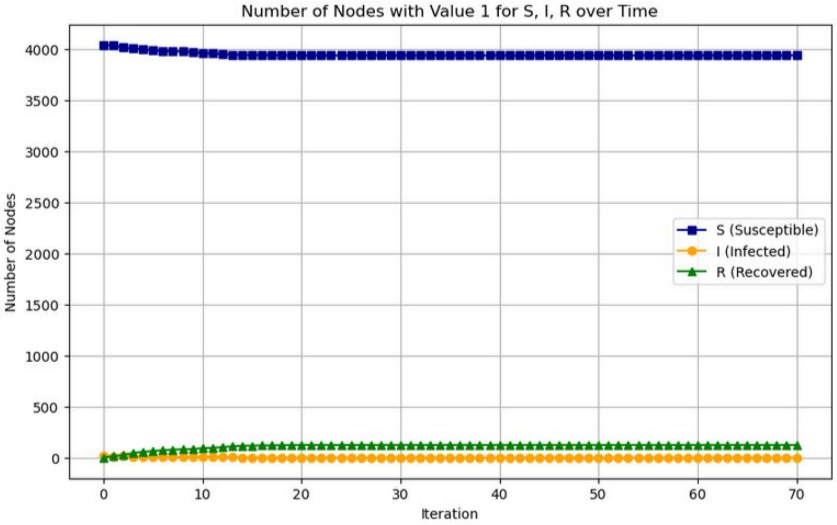
Idea

What happens if we initialize the information spread in the network's most central nodes?

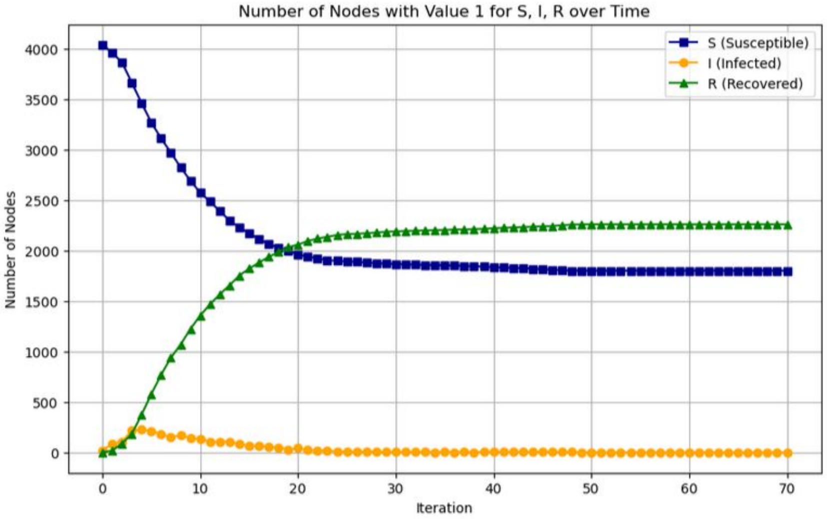
- Calculate the leading eigenvector of the graph adjacency matrix.
- Values ≈ a measurement of node centrality.
- Instead of initializing *k* nodes at random to be infected, initialize the *k* most central nodes.
- Hypothesis: This initialization will lead to much greater dissemination of information through the network.

H-SIR Infection Convergence

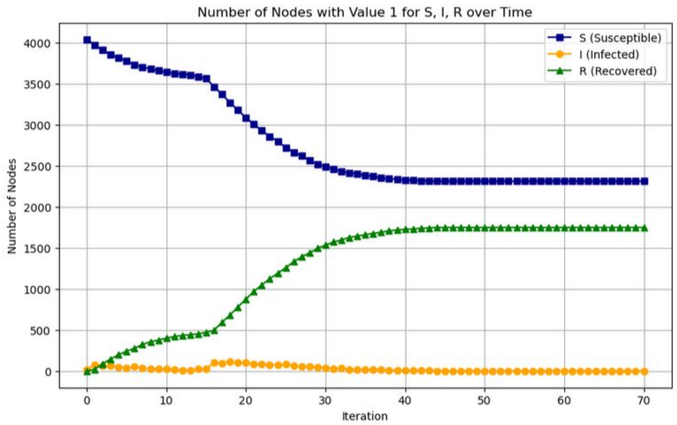
Centrality-Based Starting Nodes

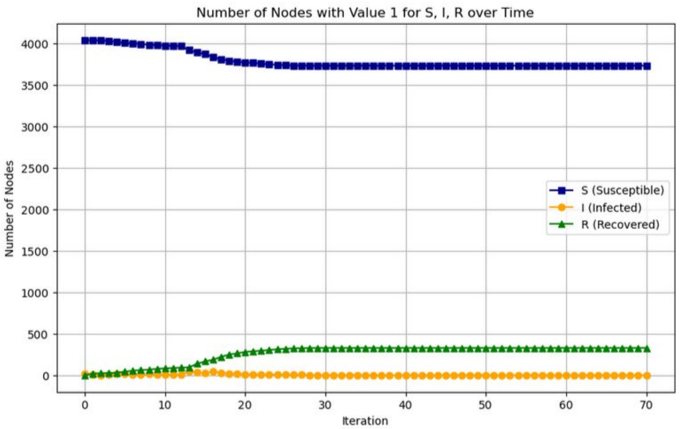


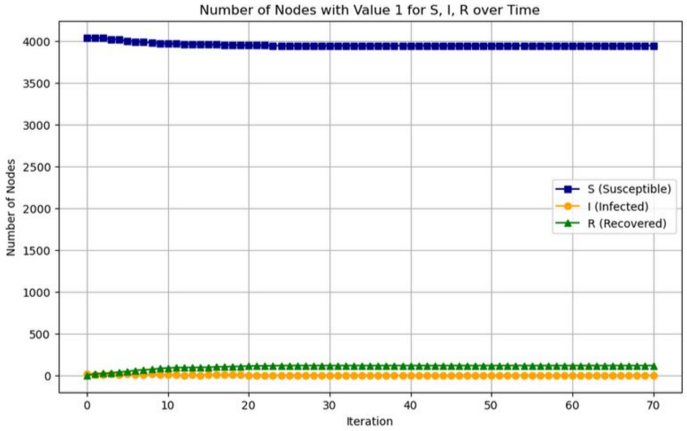
Centrality + Random Combination

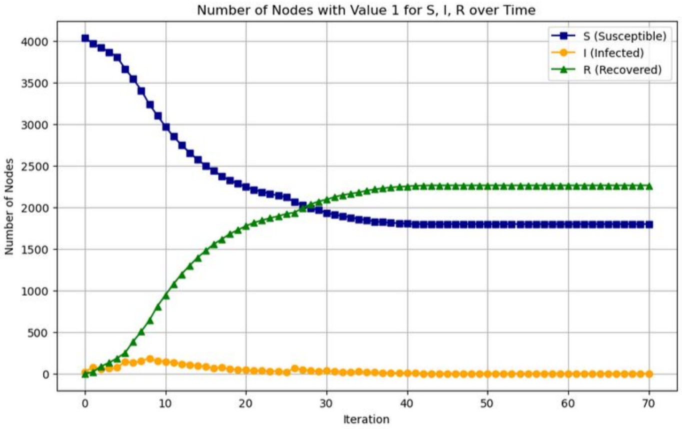


Output of Four Straight Runs with Centrality-Based Initialization









Conclusions

- Grassroots support matters: Messages being broadcast (indirectly) to the whole network will not necessarily propagate to individuals.
- Even with a massive number of nodes initially infected, infections decline very quickly, and reaching the entire network is highly unlikely.
- Initialization matters: Selecting only the most central nodes can actually impair the performance of the system, though this is not a guaranteed outcome.
- The H-SIR development study (on synthetic data) suggested much more favorable conditions for information propagation than these empirical results support.
  - This is only a subgraph of the Facebook network, and our communities are also algorithmically detected - we are open to debate around which one is “correct”.

Limitations

- While edges between high-level nodes should be weighted by the number of edges in the original graph, our implementation did not do this.
    - This does not affect the algorithm, but it does affect the spectral convergence starting point.
- Clustering is not one-to-one in the real world: A user in H_0 may belong to multiple communities in H_1, which may not map one-to-one in H_2, etc.
- This was a very limited network that sampled from Facebook's massive network: Dynamics could change in a different sample, and would likely change across the full network.

Questions?