Refining the Framework for Closing Gaps in Information Access in Networks

by

Dennis Robert Windham

B.A., University of Colorado Boulder

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Master of Science

Department of Computer Science

2024

Committee Members:

Aaron Clauset, Chair

Daniel Larremore

Zachary Kilpatrick

Windham, Dennis Robert (M.S., Computer Science)

Refining the Framework for Closing Gaps in Information Access in Networks

Thesis directed by Prof. Aaron Clauset

In contemporary networks, information spread is inherently unequal. Disparities stem from the diverse structural makeups of these networks. Such inequalities manifest in various contexts, including healthcare and employment. While traditional influence maximization algorithms strive to maximize overall information spread, they tend to overlook the notion of fairness in information access, and leave the most vulnerable individuals behind. Addressing the gap in information access is crucial for designing effective public health campaigns and outreach strategies. This paper aims to refine the framework for closing gaps in information access in networks by introducing the concept of spreadability, a new metric for evaluating algorithm performance, and ten new algorithms that aim to minimize the gap between the most and least advantaged nodes in a network. We evaluate the performance of old and new algorithms on a large, comprehensive cross-domain network corpus under several spreadability regimes, and find that none of the existing heuristics are superior to all others. We also discover that under certain conditions, conventional algorithm evaluation accuracy may prove insufficient to gauge the performance of algorithms on large, structurally complex networks. Our findings highlight the importance of network structure in shaping the efficacy of approaches to closing information access gaps, and we believe that observations and tools introduced in this work will be instrumental in designing effective intervention strategies in the future.

Acknowledgements

I would like to thank my advisor, Aaron Clauset, for his guidance and support throughout this project. I would also like to thank Alex Crane and Blair D. Sullivan of the University of Utah, and Sorelle A. Friedler of the Haverford College, for their feedback and suggestions. Finally, I would like to thank my family for their unwavering support and encouragement. This work was also supported in part by National Science Foundation Award IIS 1956183 (DRW, AC).

Contents

Tables

Table

Figures

Figure

Chapter 1

Introduction

In social networks, access to information is unequally distributed. This inequality is a natural consequence of heterogeneous structures observed in networks – it is rather intuitive that highly-connected individuals have more opportunities both to receive and to spread knowledge, and conversely, persons with few connections are less likely to partake in exchanges of information.

For example, in a pandemic, access to crucial resources, such as money, food and healthcare, is more difficult for socially disadvantaged groups, in particular due to their limited connectedness in social networks [6]. Likewise, in professional social networks such as LinkedIn, employment opportunities are dictated by connectedness: job seekers who are well-networked are likely to fill a lucrative opening sooner than others [18]. In fact, even in the context of organizations, it has been shown that connectedness of employees of a company to those of other companies in the same industry can be a predictor of the company's success [13].

Two popular approaches to spreading information in networks are broadcasting and word-of-mouth [2][11][19]. Broadcasting is a one-to-many approach, where a single source sends information to all nodes in the network. Word-of-mouth, on the other hand, is a many-to-many approach, where information is spread through the network by individuals sharing it with their neighbors [2]. The latter approach leverages the structure of the network to spread information, and real-world data suggests that word-of-mouth can be more effective than broadcasting in certain settings [2].

As discussed by Banerjee at al. in the study of information dissemination [2], under the word-of-mouth approach one can improve information access in a real-world network through careful,

intentional seeding, wherein we choose an individual or a set of individuals to be the first to receive the information. This is the idea behind the field of influence maximization, which aims to identify the best set of individuals to seed with information in order to maximize the spread of that information through the network [11].

The influence maximization problem has been studied extensively in the literature, and a variety of algorithms have been proposed to solve it. Traditionally, the goal of these algorithms has been to maximize the spread of information in a network, often by selecting the set of individuals that will maximize the expected number of activated nodes [11]. However, this approach focuses on the overall spread of information, and does not take into account the fairness of the spread between individuals in the network.

Improving information access of marginalized nodes improves fairness in networks. Indeed, taking the notion of "fairness" to be equitable treatment and opportunities for all people in a community, it is only fair that one should not be systemically punished for being imperfectly positioned with respect to other members of a given social network. The quality of one's information access lends itself to mathematical expression. Given a graph G, we take $S \subset G$ to be the seed set, where $k \in S$ represents an individual with absolute knowledge who propagates it to the rest of the network. Then, for a node $i \in G$, we say that π_i represents the information access of the individual i under S [4].

There exist numerous dynamical systems to model information propagation through networks [11][9][8]. A popular choice in prior work, also adopted by us, is the independent cascade model [4]. The independent cascade model is a stochastic model of information propagation in networks, where each edge has a one-time attempt of transmitting information from an active node to its inactive neighbors with a set probability of transmission [11]. It is a simple model that captures the essential dynamics of information propagation in networks, and has been used extensively in the literature to study the spread of information in networks [11] [4].

The independent cascade model offers a convenient framework for studying the spread of information on a network, and enables us to devise intervention strategies for improving the esti-

mated information access of the least advantaged nodes therein. Of particular interest to us is the following problem: given a network G and a budget constraint k, how can we select k seeds that minimize the gap between the most advantaged and the least advantaged nodes in the network, displaying information access of $\pi_{\max}$ and $\pi_{\min}$ respectively. That is, how do we minimize the gap π_{gap}, defined as:

$$\pi_{\text{gap}} = |\pi_{\max} - \pi_{\min}| \tag{1.1}$$

Under the independent cascade model, computing exact π_i is NP-hard [4]. Therefore, researchers have resorted to the Monte Carlo simulation-based approach, `ProbEst`, seen in Section 2.3, to estimate π_i for each node in the network [4]. However, the accuracy of the estimated π_i is usually limited to $\frac{1}{1000}$ due to computational cost [4]. It is unknown whether this accuracy cap could become a limiting factor when studying large, structurally complex networks.

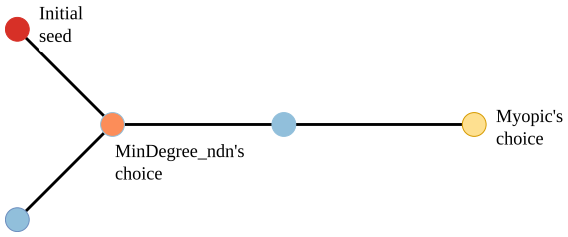


Figure 1.1: Example of how two algorithms could make different choices on a network.

Prior work suggests that there exists an ideal, combinatoric algorithm that maximizes $\pi_{\min}$ on a budget by considering all possible seed sets of size k. However, while this algorithm is technically a general solution to the problem, it is computationally infeasible to run on even the smallest networks. Instead, researchers have proposed heuristics that aim to approximate the ideal algorithm [4]. Such heuristics may produce different seed sets, as seen in Figure 1.1, and therefore different π_{gap}, on the same network. However, the heuristics' performance has been evaluated on a structurally limited range of networks [4]. It remains unclear how the structure of a network shapes the efficacy of algorithms implementing said heuristics. Furthermore, it is unknown whether there is one algorithm which always generates the best seed set, and if not, how to choose the best algorithm given only

characteristics of a network. Additionally, there is presently no convenient metric for comparing the performance of algorithms on a network.

By design, the independent cascade model takes in the parameter α, which represents the probability of successful transmission of information from one node to another [11]. The choice of α is not trivial, as the structure of the network may influence how well information spreads under a given α. Prior work focuses on several pre-selected α values [4], not accounting for network structures. A more robust approach to choosing α would help us design and evaluate intervention strategies that aim to improve information access of the disadvantaged individuals in a network.

Progress on these questions would take us closer to understanding how we could, given a budget constraint, devise an optimal intervention technique that serves to improve information access of the disadvantaged individuals. This knowledge would be instrumental in designing effective public health campaigns, outreach strategies, et cetera, that aim to improve fairness in social networks by reaching even the most socially isolated, and therefore vulnerable, individuals [1]. This is particularly important because conventional influence maximization techniques have been shown to be ineffective in improving information access of the disadvantaged nodes [4].

In our work, we introduce the concept of spreadability, which we define as the ability of a network to spread information under a given α, and subsequently pick α values individually for each network based on a target spreadability. To compare performance of algorithms, we propose a new metric, β, which corresponds to the slope of the line of best fit for `ProbEst`-based evaluations after selecting k seeds. We also design ten new algorithms that aim to minimize π_{gap} on a budget, evaluate and compare their performance alongside algorithms from prior work. We evaluate the performance across several target spreadabilities and budgets k using the metric β on a large, comprehensive cross-domain network corpus presented in Section 2.1.

We find that while there is a computationally expensive algorithm that performs best on average, it is oftentimes outperformed or matched by other, cheaper algorithms under certain spreadability conditions, suggesting that none of the existing heuristics are a clear superior choice for all situations. We also make findings regarding the efficacy of `ProbEst` given the accuracy cap

of $\frac{1}{1000}$ in the context of large, structurally complex networks, suggesting that in low spreadability conditions, this accuracy limit is insufficient to soundly evaluate algorithms on many large networks. Finally, we introduce machine learning to study whether we can predict the best algorithm for a given network, and how well the best algorithm will perform on a given network, given only the network's features, and find that both of these tasks are quite challenging.

Chapter 2

Methods & Materials

2.1 Network Corpus

In order to investigate the effects of network structure on the performance of the algorithms, we constructed a corpus of 174 networks from six domains: biological, social, economic, technological, transportation, and informational. The networks in the corpus have at least 500 nodes, and are unipartite, that is, they only have one type of node [14]. We simplified each network by ignoring edge weights, edge direction and self-loops, and took the largest connected component of each network, which itself must contain at least 500 nodes. We also ensured that no domain accounts for more than 25% of all networks in the corpus. The corpus statistics are presented in Table 2.1. Data for the corpus was obtained from [7], [3] and [17]. An overview of the corpus is presented in Figure 2.1.

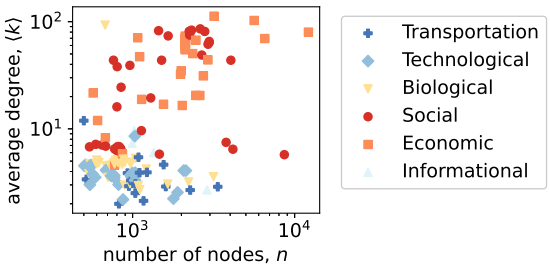


Figure 2.1: Overview of the network corpus compiled for this study.

	Num. Net-works	$\langle \mathbf{n} \rangle$, Avg. Num. Nodes	$\langle \mathbf{m} \rangle$, Avg. Num. Edges	$\langle \mathbf{k} \rangle$, Avg. Degree	$\langle \mathbf{C} \rangle$, Avg. Clust. Coeff.	$\langle \ell \rangle$, Avg. Diame-ter	$\langle \sigma^{\mathbf{2}} \rangle$, Avg. Deg. Variance
Corpus	174	1495.42	26288.68	22.60	0.24	13.79	1997.12
Biological	34	927.50	2830.24	7.00	0.08	11.59	285.87
Social	44	1647.75	29055.57	28.11	0.42	11.02	1099.06
Economic	43	2367.40	71946.23	51.97	0.38	7.19	6647.82
Technological	32	843.13	1651.50	4.07	0.07	17.84	69.55
Transportation	17	1243.65	2148.53	3.80	0.10	35.76	37.79
Informational	4	1561.75	4124.25	6.41	0.11	8.00	174.06

Table 2.1: Network corpus statistics. $\langle n \rangle$ is the average number of nodes, $\langle m \rangle$ is the average number of edges, $\langle k \rangle$ is the average degree, $\langle C \rangle$ is the average clustering coefficient, $\langle l \rangle$ is the average path length, and $\langle \sigma^2 \rangle$ is the average variance of the degree distribution.

2.2 Independent Cascade Model

In our work, we leverage the independent cascade model to study the spread of information in networks and produce π_i. Under the independent cascade model, given a network G and a set of activated nodes $Q \subseteq G$, we grow a forest by flipping a coin for each edge $e_{i,j}$, where $i \in G \backslash Q$, $j \in Q$, exactly once, so that i is added to Q on a successful flip. Here, $e_{i,j}$ is never considered as a transmission path after the initial flip. We define the probability of successful transmission to be α. As noted earlier, producing the exact π_i is NP-hard, but there is a Monte Carlo simulation-based approach, `ProbEst`, which we adopt in our work [4].

Under `ProbEst`, given transmission probability α, number of simulation rounds R, a seed set S, we estimate π_i for every $i \in G$ using R independent cascades originating from S [4]. Notice that since some nodes in the network are known to be seeded by the design of the study, we have that $\pi_{\max} = 1$, therefore π_{gap} is minimized when $\pi_{\min}$ is maximized [4]. The time complexity of `ProbEst` is $\mathcal{O}(R(|S| + 2m))$ [4], and in practice, we have seen that `ProbEst` slows down as more nodes are added to the seed set, since we end up producing longer cascades. Past work settled on $R = 1000$ as a balance between accuracy and computational cost, which we incorporate in our work as well [4]. However, we saw that at $R = 1000$, `ProbEst` is already quite slow. `ProbEst` is used to evaluate the performance of algorithms on a network, and is the basis for the β metric, which we introduce

in Section 2.6.1.

2.3 Algorithms from Prior Work

The key heuristics for maximizing $\pi_{\min}$ introduced in prior work are as follows: `Random`, `Greedy`, `Myopic`, `Naive Myopic` and `Gonzalez` [4]. The goal of these heuristics is to select a seed set that maximizes $\pi_{\min}$ on a budget k. `Random` selects k seeds by randomly sampling nodes from the network. `Gonzalez` is a heuristic that, given a seed set, selects the node that is, on average, the furthest node away from the seed set as the next seed [4]. `Greedy` is a brute-force algorithm that selects a new seed by evaluating the marginal gain of adding each node to the seed set using `ProbEst`, and choosing the node with the highest marginal gain as the next seed. Due to immensely high computational cost, `Greedy` is not practical for most networks [4], and is not being used in our work.

In contrast, `Myopic` uses `ProbEst` with the current seed set, and selects the node with the lowest π_i as the next seed. `Naive Myopic` is similar to `Myopic`, but only runs `ProbEst` once, and proceeds to select k lowest π_i nodes as the seed set. In the past, on a small set of networks, `Myopic` has been shown to perform best. However, since `Myopic` incorporates `ProbEst`, it is computationally expensive, and is, in fact, always the second slowest algorithm, after `Greedy` [4].

2.4 New Algorithms

One of our goals was to design new algorithms that aim to minimize π_{gap} on a budget, while offering alternative, faster approaches to selecting the seed set when compared to `Myopic`, as well as matching or exceeding the performance of `Myopic` on the corpus. We designed a variety of algorithms, each with a different approach to selecting the seed set. The new algorithms can be grouped into three families: BFS-based, PPR-based and Topology-based.

The first family of algorithms includes `Myopic BFS` and `Naive Myopic BFS`. Here, we swap out the `ProbEst` component of `Myopic` and `Naive Myopic` with a breadth-first search to estimate π_i. The breadth-first search component is initialized with a random initial seed s_0, transmission

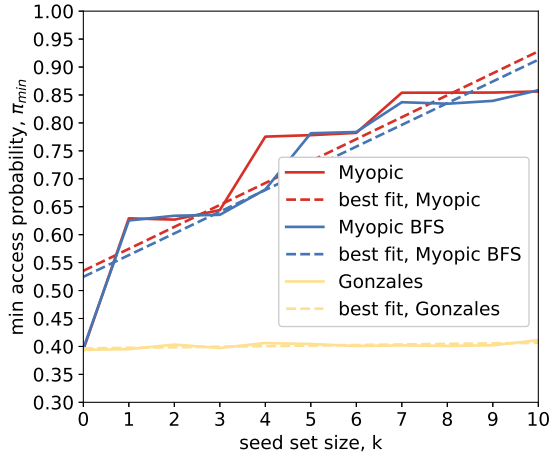


Figure 2.2: Performance of `Myopic`, `Myopic BFS` and `Gonzalez` on a network, with the lines of best fit, averaged over 20 runs. Evaluated on a large economic network with 2113 nodes and 57927 edges; $\alpha = 0.4$. Budget of $k = 10$ algorithm-computed interventions, in addition to one random initial seed. `Myopic BFS` matches `Myopic`'s performance, while `Gonzalez` lags behind.

probability α, and proceeds to "peel" the network starting s_0 in breadth-first fashion, layer by layer, estimating π_i for each node. The π_i estimate is based on the probability of i receiving a transmission from s_0 through any nodes it connects to in the previous layer, as well as through nodes it is connected to in its own layer. All subsequent iterations update existing π_i estimates during the breadth-first traversal from new candidate seeds.

While this approach is imperfect, it is much faster than `ProbEst`, taking $\mathcal{O}(n \cdot \langle \mathbf{k} \rangle)$ per iteration, where for our networks $\langle \mathbf{k} \rangle << 1000$ and $n < m$, as opposed to `ProbEst`'s $\mathcal{O}(1000(|S| + 2m))$ per iteration. The key design principle behind this algorithm was to capture complex network structures that `Gonzalez` could not account for. Unlike `Gonzalez`, which only considers the overall distance from the seed set, our `BFS`-based algorithms incorporate more local information when estimating π_i. In Figure 2.2, observe that `Myopic BFS` can almost match `Myopic`'s performance in situations where `Gonzalez` lags behind.

The second family, made up of `Myopic PPR` and `Naive Myopic PPR`, attempts to leverage the power of Personalized PageRank (PPR) to estimate π_i. We use personalized PageRank, which is known for its efficiency, to perform a random walk with biased restarts from the seed set [15].

While the ranking produced by PPR does not directly translate to π_i, we wanted to investigate the possibility of a correlation between lower ranking and lower π_i values. For the news algorithms, the PPR component then replaces `ProbEst` from `Myopic` and `Naive Myopic` described in Section 2.3. We did not implement PPR by hand, and instead relied on NetworkX's built-in implementation [10].

The third family of algorithms uses network topology to estimate π_i. These algorithms are based on the intuition that the structure of the network should be meaningful in estimating π_i. One such algorithm is `LeastCentral`, which selects the non-seed node with the lowest closeness centrality as the next seed. Another variation of this algorithm, `LeastCentral_n`, similarly locates the non-seed node with the lowest closeness centrality, but proceeds to select its neighbor with the highest degree as the next seed. Here, closeness centrality is the inverse of the average shortest path length from a node to all other reachable nodes in the network [4]. Hence, lower closeness centrality implies the node is less reachable by the rest of the network, and therefore is expected to have lower π_i.

The four remaining topology-based algorithms exploit the degree structure of a network to make decisions. The inspiration for these algorithms comes from an early observation that `Myopic` tends to select nodes with low degrees as seeds, and we wanted to investigate whether this was a meaningful strategy. The first two algorithms look for a non-seed node with the lowest degree and the lowest harmonic centrality among same-degree nodes, and proceed to choose the node itself as the new seed in the case of `MinDegree_hc`, and the highest-degree neighbor of the node in the case of `MinDegree_hcn`. The two remaining variations of the Degree-based algorithms replace harmonic centrality in the aforementioned logic with neighbor degree, preferring the node with the highest neighbor degree, and are called `MinDegree_nd` and `MinDegree_ndn` respectively.

2.5 Algorithm Initialization

Notice that `Myopic`, `Naive Myopic` and `Gonzalez` inherently require an initial seed set to run. Past work considered initializing the seed set with the highest degree node, and left other initial-

ization methods for future studies. We pick up the baton and consider initializing these algorithms with random nodes. One advantage of this approach is that it eliminates bias in performance of the algorithms, since performance is contingent on the initial seed set, as seen in Figure 2.3.

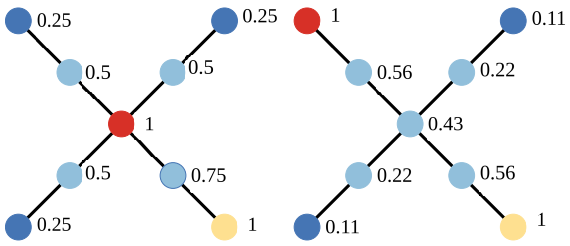


Figure 2.3: Two small networks with different initial seeds, in red, and seeds selected by `Myopic`, in yellow. The remaining nodes have had their access probabilities estimated by `ProbEst` under $\alpha = 0.5$ to two decimal places. The first network has $\pi_{\min} = 0.25$, the second displays $\pi_{\min} = 0.11$, showing the importance of the initial seed set.

To further mitigate the impact of the initial seed set on our comparison, we initialize every algorithm with the same random seed within an evaluation round for a given network, even for algorithms that do not require initialization. This way, we can ensure that the performance of the algorithms is not biased by the initial seed set. The initial seed does not count against the budget k, as seen in Figure 2.2.

2.6 Spreadability

Prior work used $\alpha = 0.3, 0.4, 0.5$ to evaluate the performance of the algorithms [4]. Picking α isn't simple, as the network's configuration could affect the efficiency of information propagation. Intuitively, denser, more connected networks should spread information more easily than sparse networks, hence the same α may produce larger cascades on the former networks, and much smaller cascades on the latter. We introduce the concept of spreadability, which we define as the ability of a network to spread information under a given α.

For a given α, we compute spreadability on a network by averaging over R_s trials, where for each trial we randomly sample an initial seed from the network, run the independent cascade

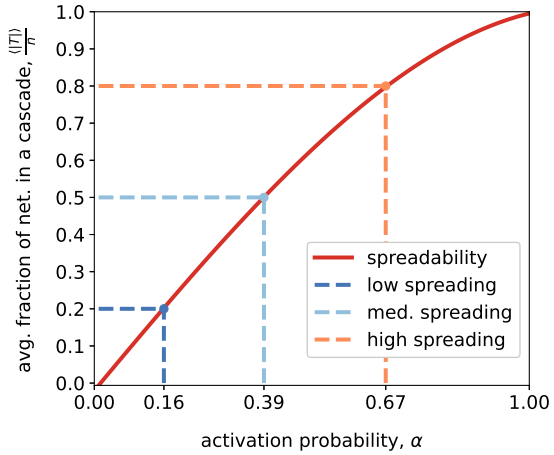


Figure 2.4: Spreadability computation on a network. The vertical axis corresponds to the average fraction of network nodes found in a tree T grown through an independent cascade from a random initial seed for a given α.

model only once with the given α, and then record the fraction of nodes in the network that were activated on average. This approach produces a monotonic increasing curve, as seen in Figure 2.4.

We have found that computing spreadability for each $\alpha \in \{0.01, 0.02, ...0.99\}$ provides generous granularity in choosing α values close to our target spreadabilities of $0.2, 0.5, 0.8$, denoted to be low, medium and high spreadabilities respectively. We set $R_s = 1000$, as the spreadability curve tends to stabilize in this area and we get diminishing returns for larger R_s.

2.6.1 New Metric for Algorithm Evaluation

We propose a metric for evaluating the relative performance of algorithms, β, which corresponds to the average slope of the lines of best fit for `ProbEst`-based evaluations after selecting the first 10 seeds, as seen in Figure 2.2. This new metric is useful because it shows the marginal improvement in $\pi_{\min}$ that we can expect when asking an algorithm to draw a new seed, which allows for a direct comparison of the performance of the algorithms. We use this metric to evaluate the performance of our algorithms on the network corpus.

Chapter 3

Results

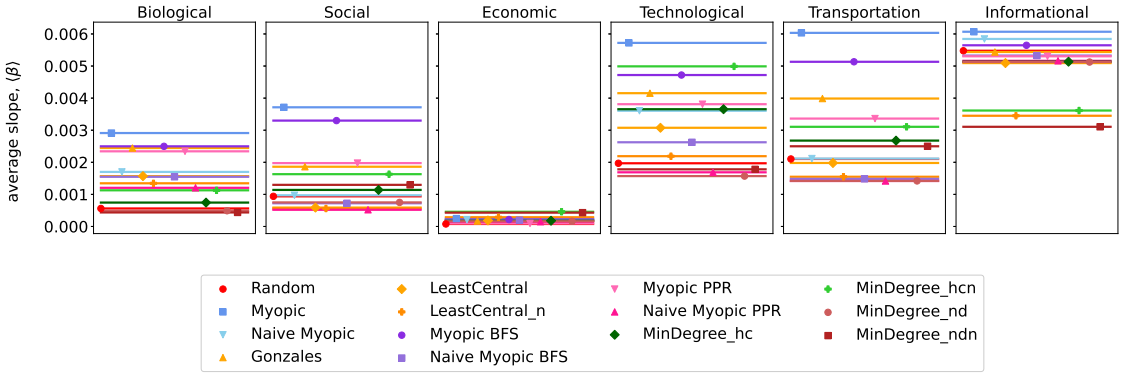
3.1 Performance of Intervention Algorithms on the Corpus

Our new algorithms, together with the algorithms from prior work, add up to a total of 14 algorithms that we evaluate on 174 networks from the corpus. Since we are interested in algorithms that operate well on a tight budget, we set the target number of seeds to $k = 10$. For each of the three target spreadabilities, we obtained a $10 \times 14 \times 174$ matrix of results, where each entry corresponds to the performance of an algorithm after adding $k_n \in \{1, \ldots, 10\}$ seeds in a network averaged over 20 runs. The results for medium and high spreadabilities are very similar, so for brevity, we omit high spreadability from the remainder of our analysis.

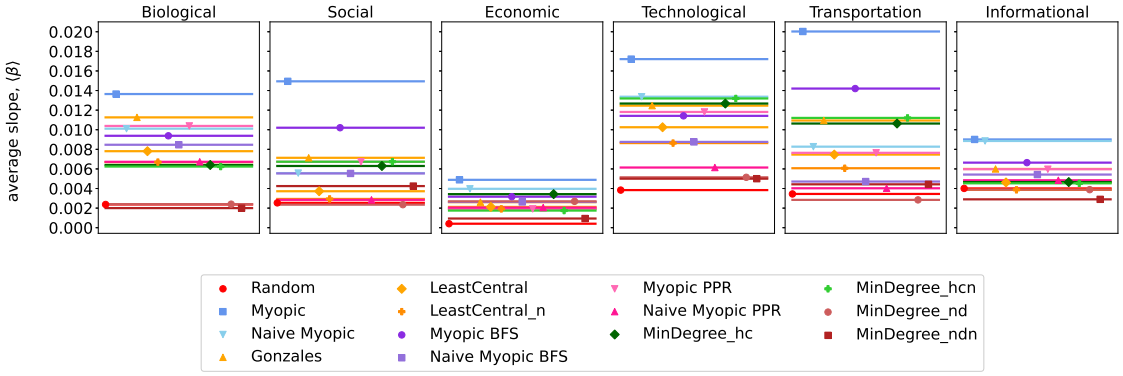
In Figure 3.1, observe that under medium spreadability, on average, `Myopic` performs best. In a low spreadability regime, the performance difference between `Myopic` and other algorithms is less pronounced. In fact, in the economic domain, `Myopic` is outperformed by `MinDegree_ndn` and `MinDegree_hcn` on average. Recall that these two algorithms select highest-degree neighbors of apparent disadvantaged nodes, which are indeed slightly better at reaching other disadvantaged nodes in the network. This suggests that the structure of the network plays a role in the performance of the algorithms.

In Figure 3.2, we present performance of the algorithms on the corpus in a different way. Here, we sort by network size within domain in ascending order and normalize relative to `Myopic`. Thus, we are able to gauge how well the algorithms perform relative to `Myopic` on each network.

From Figure 3.2, observe that `Myopic` is not universally the best choice. In fact, for medium



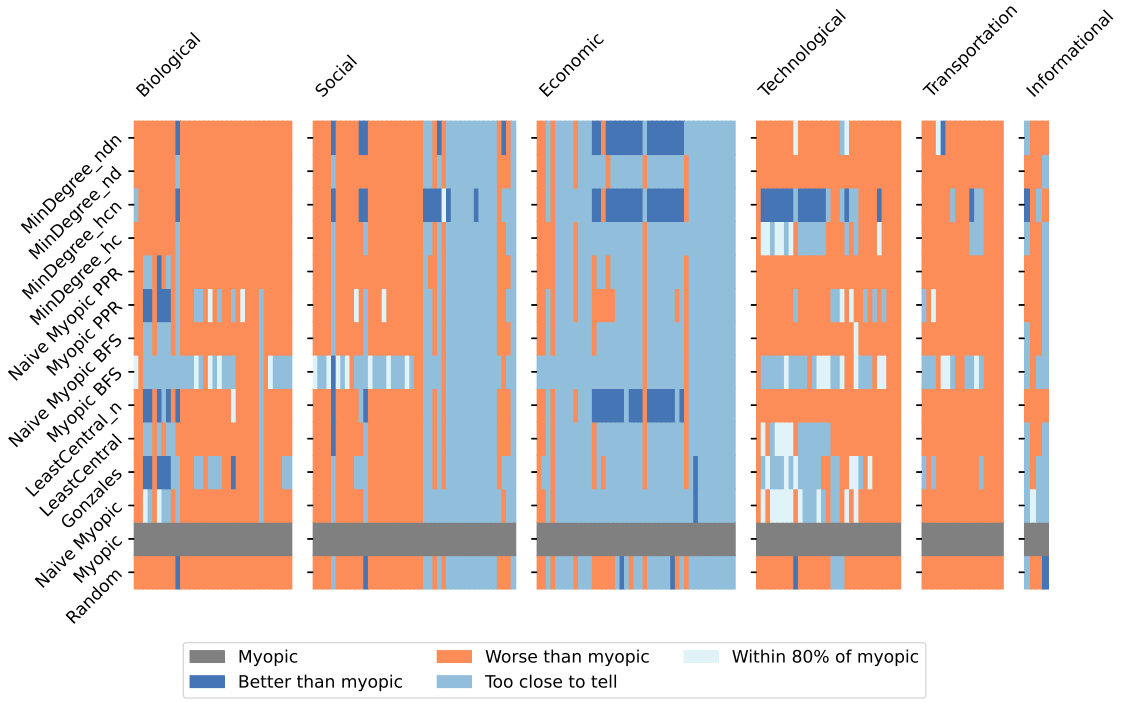
(a)



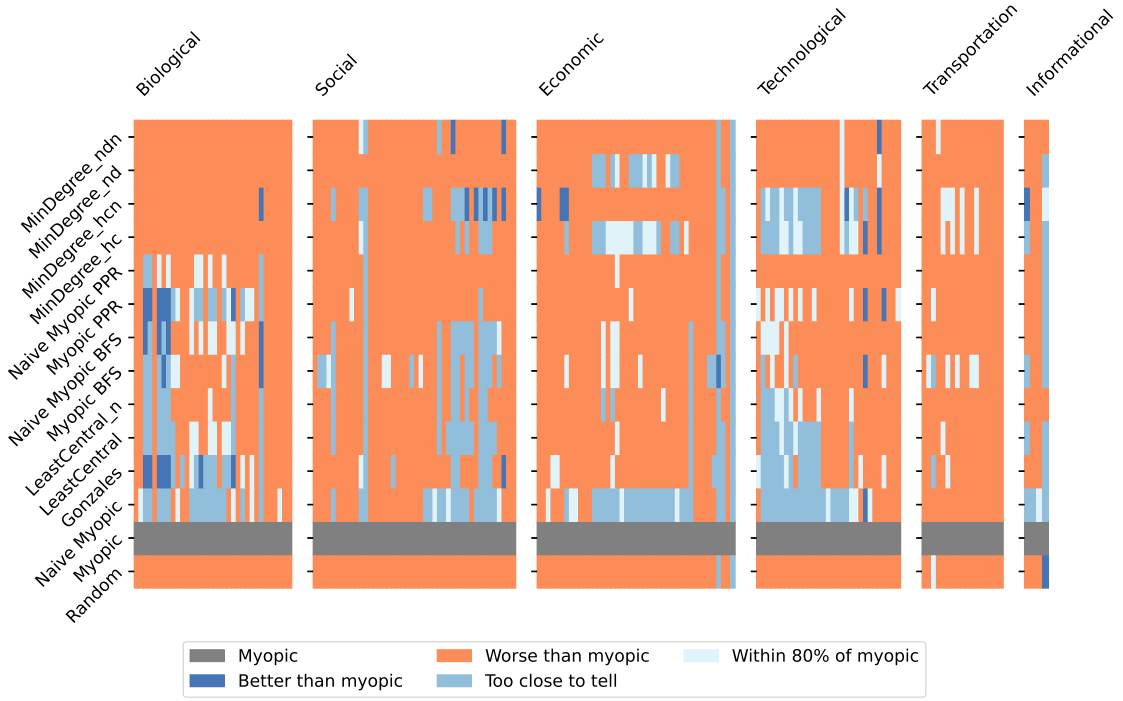
(b)

Figure 3.1: Mean performance of the intervention algorithms on each domain in the corpus. (a): low spreadability. (b): medium spreadability.

spreadability, only 24% of the corpus networks do not have another algorithm which performs at least 80% as well as `Myopic`, as seen in Figure 3.2b. Further, from Figure 3.2a, in low spreadability conditions, only 12% of the corpus sees no algorithm perform at least 80% as well as `Myopic`. Note that in the case of low spreadability, the results are muddied: given that low-spreading α are generally quite small, on some networks, the precision of `ProbEst` set to 1000 independent cascades may not be enough to gauge the performance of an algorithm. In a situation where both `Myopic` and another algorithm could not be evaluated to the desired precision, we assign "Too close to tell" (Figure 3.2a).



(a)



(b)

Figure 3.2: Performance of the intervention algorithms on the corpus networks, normalized relative to `Myopic`, sorted by network size within domain in ascending order. (a): low spreadability, 12% of the corpus has no algorithm better or at least within 80% of `Myopic`'s performance; (b): medium spreadability, 24% of the corpus has no algorithm better or at least within 80% of `Myopic`'s performance.

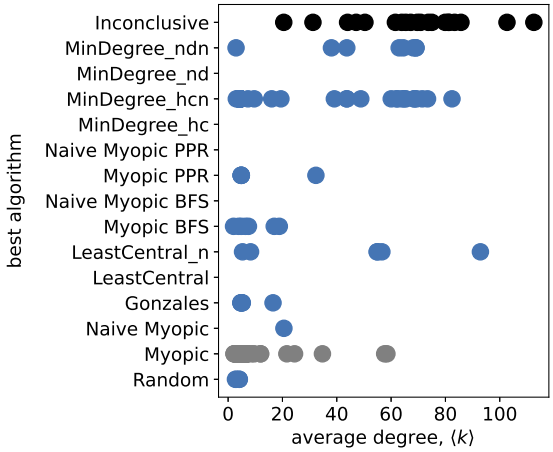


Figure 3.3: Best-performing algorithm on a network vs average degree of the network under low spreadability. Each network's corresponding single most-performant algorithm is plotted with a circle.

We further investigate the effects of network characteristics on algorithm performance in Figure 3.3. Here, a pattern emerges: `Myopic` performs best on networks with low average degree in a low spreadability regime, but as the average degree of the network increases, `Myopic` less frequently performs best. This suggests that network structures produced from various average degrees have a profound influence on performance of algorithms.

We suspect that `Myopic` underperforms as a result of targeting the nodes with the lowest π_i. The final seed set produced by `Myopic` is generally made up of low-degree nodes located on the periphery of the network and far removed from one another, and likewise they may be too far removed from the rest of disadvantaged nodes to meaningfully improve their π_i values.

Observe that in Figure 3.3, we also have that `MinDegree_hcn` becomes more performant with increasing average degree. Furthermore, in Figure 3.1a, `MinDegree_hcn` outperforms `Myopic` in the economic domain on average, and from Table 2.1, we see that indeed, the economic domain has the highest average node degree.

Recall that `MinDegree_hcn` picks the highest degree neighbor of a low-degree node. By design, this algorithm strategically selects seeds that are likely to be very close to disadvantaged nodes, but have better-than-worst engagement in the network. Due to this increased level of engagement,

the information spread from these seeds is more likely to reach more disadvantaged nodes in the network, and therefore improve their π_i values.

From Figures 3.1, 3.2, a natural research tangent arises: we see that Myopic does best on average, but for most networks there is another algorithm that may get close or outperform it. Moreover, we have seen that network structures play a role in performance of algorithms. Then, it would be valuable to come up with a solution that can either select the best algorithm for a given network using the network's features, or at least predict what is the highest $\pi_{\min}$ that any algorithm can produce.

3.2 Machine Learning

The problems of predicting how well one can do on a network, and what the best algorithm would be, lend themselves to machine learning. To train the models, we extracted the following features for all networks: domain, number of nodes, average degree, maximum degree, degree variance, clustering coefficient, average shortest path, diameter, assortativity. We then trained random forest classifiers to predict the best algorithm for a given network, and random forest regressors to predict the performance of the best algorithm on a given network, all using the 70-30 train-test split. In both cases, we employed 100 estimators after hyperparameter tuning. Our experimental data of choice for these tasks was the performance of the algorithms on the corpus under medium spreadability.

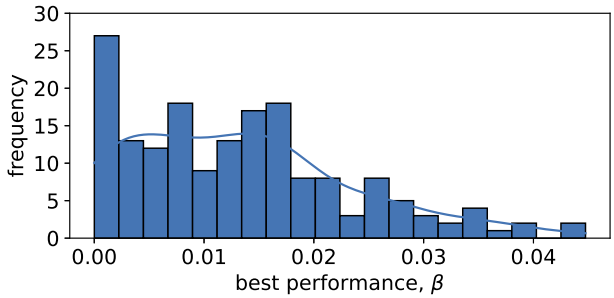


Figure 3.4: Distribution of the best β values for the regression task.

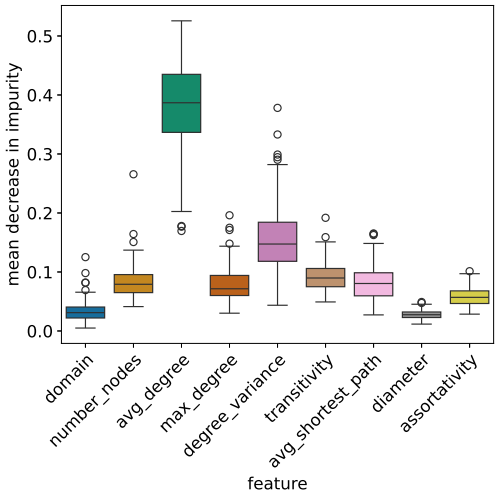
For the regression task, where we attempt to predict the best possible $\pi_{\min}$ for a given network, we found that the mean absolute error over 100 trained models is ≈ 0.004, whereas the mean squared error is ≈ 0.000046. Albeit seemingly low, the errors are not negligible if we compare them to the performance of the algorithms found in Figure 3.2b and the distribution of best β values seen in Figure 3.4.

This result suggests that while it is possible to get a sense of the best attainable performance for a given network using only the network's features, it is not a trivial task, and could require a more sophisticated model and more training data.

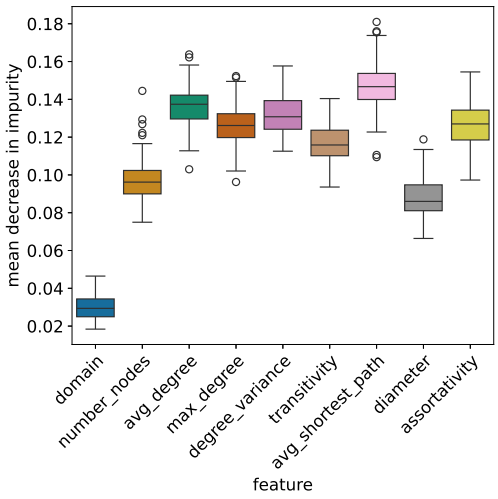
To study feature importance, we used `SciPy`'s built-in feature importance calculation, which is based on the Gini impurity [16]. The most important feature for regression, according to the random forest models, is the average degree, which lines up with our earlier observations (Figure 3.5).

The classification task is even more challenging. If we take our target to be the best algorithm for a given network, the models come out with an accuracy of ≈ 0.66. While this may seem like a decent result, in actuality it is not much better than guessing `Myopic` every time. In fact, a typical run of the model yields a confusion matrix where `Myopic` is the most common prediction, although the models also resolve `Gonzalez` and `MinDegree_hcn` most of the time (Figure 3.6). However, it is important to note that in every instance where the model guesses `Myopic` wrong, `Myopic` actually turns out to be the second-best algorithm.

From the classification task, we saw that the domain of a network contributes the least to the prediction, whereas the other features are roughly equally important, as seen in Figure 3.5b. This is a surprising result, as we expected the domain to be a significant factor in predicting the best algorithm for a given network. It can be inferred that the rest of our feature set already captures the domain information in a more meaningful, albeit indirect, way.



(a)



(b)

Figure 3.5: Feature importances for the machine learning tasks. (a): regression. (b): classification.

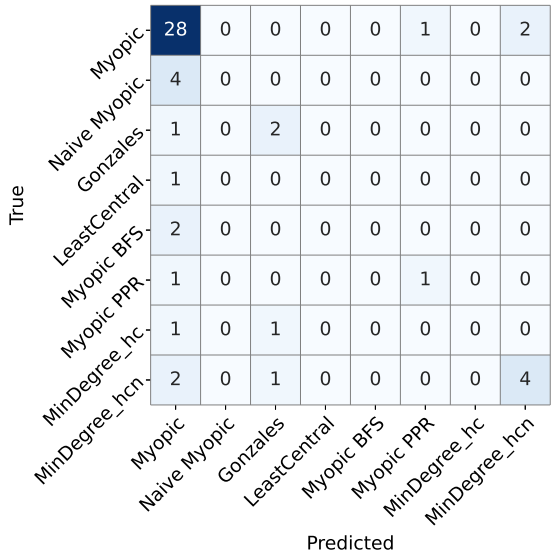


Figure 3.6: Confusion matrix of the best algorithm prediction model.

Chapter 4

Discussion

In our study, we saw that the performance of intervention algorithms is contingent on the structure of the network, and that generally, no considered algorithm is universally the best choice. With the introduction of a large, diverse cross-domain network corpus, a new performance metric, ten new algorithms and the concept of spreadability on a network, we found that `Myopic` consistently performs well, but there is usually an algorithm that performs close to it or better. However, we also found that predicting the performance of algorithms given only the features of a network is not an easy task. Moreover, we discovered that under certain conditions, `ProbEst` is not accurate enough to gauge the performance of an algorithm, and that α need to be chosen carefully, as structural features of a network may impact how well information spreads under a given α.

These findings have implications for the design of intervention strategies that aim to improve information access of the disadvantaged individuals in a network. Our work suggests that the structure of the network plays a significant role in the performance of the algorithms, with average degree being a particularly important factor. Importantly, we saw that the hitherto standard usage of `ProbEst` to evaluate algorithm performance, when taken at the conventional precision of $\frac{1}{1000}$, is not enough under low spreadability conditions to either compare algorithm performance of guide `Myopic`. Given the computational costs of `ProbEst`, it may not be feasible to increase the precision of the performance metric computation, which is a significant limitation of the current approach.

Our work provokes exciting new questions about how to improve fairness in social networks. Namely, one may wonder whether the use of `ProbEst` to evaluate the performance of the algorithms

is the best approach, and whether there is a better way that is both accurate and computationally feasible. Moreover, applications of machine learning to this problem are still an open question, and it would be interesting to see if more sophisticated models could predict the best algorithm for a given network, or the performance of the best algorithm on a given network, with higher accuracy.

We believe that the introduction of spreadability and the new performance metric β are important steps towards understanding how to design effective intervention strategies that aim to improve information access of the disadvantaged individuals in a network, and that our own new lightweight algorithms are a right step in the direction of producing more scalable and performant algorithms.

Bibliography

[1] Marcus Alexander, Laura Forastiere, Swati Gupta, and Nicholas A. Christakis. Algorithms for seeding social networks can enhance the adoption of a public health intervention in urban india. Proceedings of the National Academy of Sciences, 119(30):e2120742119, 2022.

[2] Abhijit Banerjee, Emily Breza, Arun G Chandrasekhar, and Benjamin Golub. When less is more: Experimental evidence on information delivery during india's demonetization. Working Paper 24679, National Bureau of Economic Research, June 2018.

[3] Aaron Clauset, Ellen Tucker, and Matthias Sainz. The colorado index of complex networks. `https://icon.colorado.edu/`, 2016.

[4] Benjamin Fish, Ashkan Bashardoust, Danah Boyd, Sorelle Friedler, Carlos Scheidegger, and Suresh Venkatasubramanian. Gaps in information access in social networks? In The World Wide Web Conference, WWW '19. ACM, May 2019.

[5] Linton C. Freeman. Centrality in social networks conceptual clarification. Social Networks, 1(3):215–239, 1978.

[6] Joshua P. Garoon and Patrick S. Duggan. Discourses of disease, discourses of disadvantage: A critical analysis of national pandemic influenza preparedness plans. Social Science & Medicine, 67(7):1133–1142, 2008.

[7] Amir Ghasemian, Homa Hosseinmardi, and Aaron Clauset. Evaluating overfit and underfit in models of network community structure. IEEE Transactions on Knowledge and Data Engineering, page 1–1, 2019.

[8] Jacob Goldenberg, Barak Libai, and Eitan Muller. Talk of the network: A complex systems look at the underlying process of word-of-mouth. Marketing Letters, 2001.

[9] M. Granovetter. Threshold models of collective behavior. The American Journal of Sociology, 83(6):1420–1443, 1978.

[10] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics, and function using networkx. In Gaël Varoquaux, Travis Vaught, and Jarrod Millman, editors, Proceedings of the 7th Python in Science Conference, pages 11 – 15, Pasadena, CA USA, 2008.

[11] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '03, page 137–146, New York, NY, USA, 2003. Association for Computing Machinery.

[12] W. O. Kermack and A. G. McKendrick. A contribution to the mathematical theory of epidemics. Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character, 115(772):700–721, 1927.

[13] Xin Li, Frank Nagle, and Aner Zhou. Mapping organizational-level networks using individual-level connections: Evidence from online professional networks. Harvard Business School Strategy Unit Working Paper, (24-010), August 2023.

[14] M. E. J. Newman. Networks. Oxford University Press, Oxford, UK, 2nd edition, 2018.

[15] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. In The Web Conference, 1999.

[16] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12:2825–2830, 2011.

[17] Tiago P. Peixoto. The netzschleuder network catalogue and repository. `https://doi.org/10.5281/zenodo.7839981`, 2020.

[18] Dashun Wang and Brian Uzzi. Weak ties, failed tries, and success. Science, 377(6612):1256–1258, 2022.

[19] Duncan J. Watts, Peter Sheridan Dodds, John Deighton served as editor, and Tulin Erdem served as associate editor for this article. Influentials, networks, and public opinion formation. Journal of Consumer Research, 34(4):441–458, 2007.